

SEC290

Fundamentals of Infrastructure Security

Louie Venegas

Introduction

Throughout this presentation, we will explore several fundamental concepts of network security through exercises conducted in labs over the past eight weeks. These concepts include threat intelligence and reconnaissance, vulnerability management and scanning, cloud security, infrastructure security, Identity and Access Management (IAM), software development security, security monitoring, incident response programs, indicators of compromise, forensic analysis, and risk management.

Manual Vulnerability Analysis

In this activity I worked with three virtual machines Ubuntu Web (main testing platform), Easy (Windows Server 2003) and Malware (Windows 7). On these machines I performed the following activities.

- | | |
|---|---|
| <p>1. Initial Reconnaissance:</p> <ul style="list-style-type: none">Using Nmap to identify hosts and open portsNetwork enumeration using nbtscan <p>2. Vulnerability Testing:</p> <ul style="list-style-type: none">Testing for MS08-067 on older Windows systems (Easy VM)Testing for MS17-010 on Windows 7 (Malware VM)Using Nmap scripting engine for vulnerability detection | <p>3. Exploitation Process:</p> <ul style="list-style-type: none">Using Metasploit Framework (MSF) to exploit discovered vulnerabilitiesGaining shell access to vulnerable systemsUpgrading basic shells to Meterpreter shells for advanced functionality <p>4. Post-Exploitation:</p> <ul style="list-style-type: none">Exploring various post-exploitation modulesManaging multiple sessionsUsing advanced Meterpreter commands |
|---|---|

The lab demonstrates the practical application of vulnerability scanning, exploitation, and post-exploitation techniques

Intrusion Analysis using Wireshark

The objective of this lab was to learn fundamental protocol analysis using Wireshark, while focusing on packet capture, traffic analysis, and identifying common attacks. In this activity I used two virtual machines setting up captures between Ubuntu Web VM and OWASP-BWA VM.

The following components were presented in this activity:



Basic Protocol

Analysing ICMP (ping) traffic

Examining TCP three-way handshake

Analyzing HTTP traffic and following HTTP streams

Observing SSH encrypted traffic



Display Filters:

Using filters to extract specific traffic types

Understanding TCP flags (Using mnemonic:

Analysis of packet structure and encapsulation

•Common Attacks Analysis:

- Using Nmap for network scanning
- Analyzing port scan artifacts
- Understanding stealth scan techniques (-sS flag)

•Attack Analysis Practice:

- Examining a pre-captured pcap file (basicattacks2.pcap)
- Identifying different operating systems based on ICMP characteristics
- Analyzing suspicious flag settings

The lab provides practical experience in network protocol analysis, traffic capture and filtering, attack pattern recognition, security tool usage (Wireshark, Nmap), and network attack indicator identification. I gained practical experience examining protocols, capturing network traffic, using security tools, and identifying threat indicators in network communications.

Basic attack analysis

- Captures 20 and 22 are both ICMP traffic, but the time-to-live and data field sizes differ. *Capture 20 data size is 98 bytes, ttl is 64 and Capture 22 data size is 74 bytes and ttl is 128.*
- Do a little Internet research to discover which operating systems use the specific values in their ping commands. What operating system generated the echo request in capture 20? *Echo request in capture 20 is generated by a Linux OS*
- Review packet no. 37 and beyond, what do you think is taking place here? *There is a malicious attack going on (port scanning)*
- Look at capture 22846. What is suspicious about the flag settings in this packet? *It's suspicious that the SYN flag for 0x0000 isn't set but the fin flag for 0x001 is set. This is unusual because SYN is used to initiate a connection and FIN is used to terminate a connection. Having both flags set simultaneously is an indication of a scan or attempt to bypass firewall rules.*
- What is the IP address of the host being targeted? Src 192.168.25.200 Dst. 192.168.25.1

Open SSL

The objective of this activity was to create, test, and analyze SSL/TLS connections using OpenSSL and Wireshark. Using a virtual machine running Ubuntu, called the Ubuntu Web VM, I went over the following concepts:

- SSL/TLS Setup:
 - Creating public and private keys
 - Generating self-signed certificates
 - Setting up an SSL server
- Testing Process:
 - Testing the SSL connection using Firefox
 - Using OpenSSL client commands
 - Managing TLS version compatibility
- Wireshark Analysis:
 - Capturing SSL/TLS traffic
 - Configuring Wireshark for SSL decryption
 - Setting up RSA keys for traffic decryption
 - Following SSL streams
- Troubleshooting:
 - Handling TLS version mismatches
 - Resolving certificate errors
 - Verifying encrypted connections

During this lab I demonstrated practical encryption implementation and analysis, setting up secure connections and decrypting SSL/TLS traffic with proper key access.

Notice the GET request in the Info column.

Creating and testing an SSL/TLS file cont'd

The screenshot displays the Wireshark network protocol analyzer interface. The main pane shows a packet capture of an SSL/TLS session. The left pane lists the captured packets, with packet 10 selected. The right pane shows the details of the selected packet, which is an HTTP GET request. The details pane is expanded to show the 'Hypertext Transfer Protocol' section, which contains the following information:

```
GET / HTTP/1.0
HTTP/1.0 200 OK
Content-type: text/html

<HTML><BODY BGCOLOR="#ffffff">
<pre>

s_server -www -cipher AES256-SHA -key server.pem -cert server.crt
Secure Renegotiation IS supported
Ciphers supported in s_server binary
TLSv1.3 : TLS_AES_256_GCM_SHA384 TLSv1.3 : TLS_CHACHA20_POLY1305_SHA256
TLSv1.3 : TLS_AES_128_GCM_SHA256 SSLv3 : AES256-SHA
---
Ciphers common between both SSL end points:
AES256-SHA
Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:Ed25519:Ed448:RSA-PSS+SHA256:RSA-
PSS+SHA384:RSA-PSS+SHA512:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA+SHA256:RSA-
+SHA384:RSA+SHA512:ECDSA+SHA224:ECDSA+SHA1:RSA+SHA224:RSA+SHA1:DSA+SHA224:DSA+SHA1:DSA-
+SHA256:DSA+SHA384:DSA+SHA512
Shared Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:Ed25519:Ed448:RSA-PSS-
+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA-
+SHA256:RSA+SHA384:RSA+SHA512:ECDSA+SHA224:ECDSA+SHA1:RSA+SHA224:RSA+SHA1:DSA+SHA224:DSA-
+SHA1:DSA+SHA256:DSA+SHA384:DSA+SHA512
Supported Elliptic Groups: X25519:P-256:X448:P-521:P-384
Shared Elliptic groups: X25519:P-256:X448:P-521:P-384
---
No server certificate CA names sent
---
New, SSLv3, Cipher is AES256-SHA
SSL-Session:
  Protocol : TLSv1.2
  Cipher : AES256-SHA
  Session-ID:
  Session-ID-ctx: 01000000
  Master-Key:
0974EA235BF0005AB80FE5C226932ACF9CB1B445472A10F8E5B8995C3F9051789FEB0C62F2BC4D11A96746EBB30D4E
DE
  PSK Identity: None
  PSK Identity hint: None
  SRP username: None
  Start Time: 1727049722
  Timeout : 7200 (sec)
  Verify return code: 0 (ok)
  Extended master secret: yes
---
  0 items in the session cache
  0 client connects (SSL_connect())

1 client pkt, 1 server pkt, 1 turn.
```

The bottom of the details pane shows the 'Find' field and the 'Find Next' button. The status bar at the bottom indicates 'Entire conversation (2,114 bytes)' and 'Show and save data as ASCII'.

Snort

Working with three virtual machines: Security Onion IDS, Ubuntu Web, and OWASP-BWA during this activity I was introduced to an intrusion prevention system called Snort. I used Snort to practice the following:

Testing Snort Rules:

- Use Sguil (a web frontend for IDS alerts) to monitor network activity
- Generate an XMAS scan attack using Nmap and observe the alerts
- Analyze the attack using both Sguil and Wireshark to examine TCP flag values

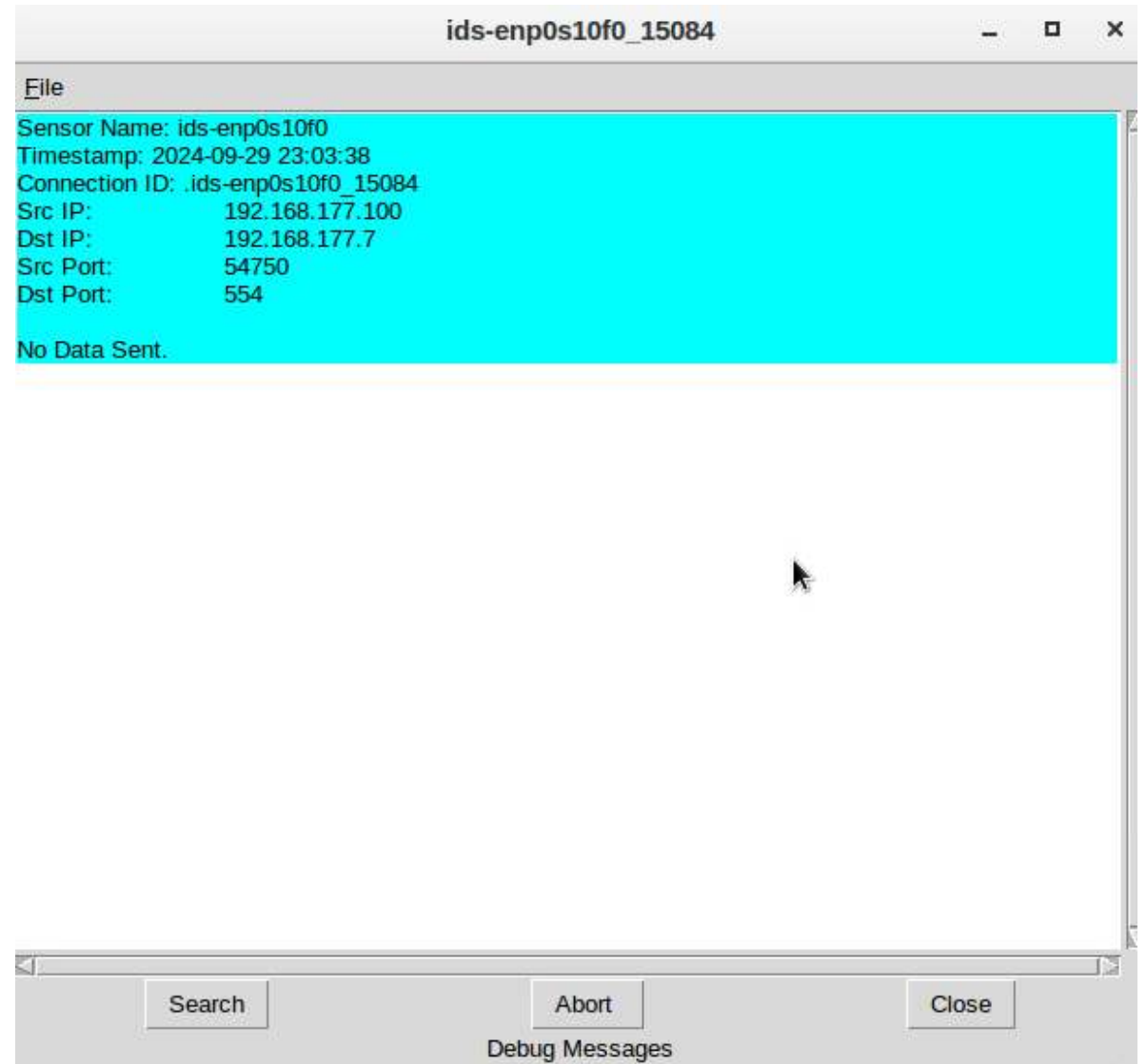
Creating Custom Snort Rules:

- Write a custom rule to detect ICMP traffic
- Test by sending ping requests between machines
- Monitor the resulting alerts in Sguil
- Analyze captured packets in Wireshark
- Verify rule effectiveness and document results

The lab provided hands-on experience with network security tools, intrusion detection, and traffic analysis.

Testing Snort rules

XMAS scan alert.



Testing Snort rules cont'd

TCP packets generated by the
XMAS scan.

*enp0s10f0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp Expression...

No.	Time	Source	Destination	Protocol	Length	Info
1069	289.844432800	192.168.177.100	192.168.177.7	TCP	60	54750 → 554 [FIN, PSH, URG] Seq=1 Win=1...
1070	289.845359800	192.168.177.100	192.168.177.7	TCP	60	54750 → 80 [FIN, PSH, URG] Seq=1 Win=10...
1071	289.846040900	192.168.177.100	192.168.177.7	TCP	60	54750 → 113 [FIN, PSH, URG] Seq=1 Win=1...
1072	289.847027400	192.168.177.100	192.168.177.7	TCP	60	54750 → 111 [FIN, PSH, URG] Seq=1 Win=1...
1073	289.847028700	192.168.177.100	192.168.177.7	TCP	60	54750 → 443 [FIN, PSH, URG] Seq=1 Win=1...
1074	289.848081500	192.168.177.100	192.168.177.7	TCP	60	54750 → 5900 [FIN, PSH, URG] Seq=1 Win=...
1075	289.848603500	192.168.177.100	192.168.177.7	TCP	60	54750 → 53 [FIN, PSH, URG] Seq=1 Win=10...
1076	289.848604600	192.168.177.100	192.168.177.7	TCP	60	54750 → 1723 [FIN, PSH, URG] Seq=1 Win=...
1077	289.849355700	192.168.177.100	192.168.177.7	TCP	60	54750 → 21 [FIN, PSH, URG] Seq=1 Win=10...
1078	289.849884200	192.168.177.100	192.168.177.7	TCP	60	54750 → 199 [FIN, PSH, URG] Seq=1 Win=1...
1079	290.945757800	192.168.177.100	192.168.177.7	TCP	60	54751 → 199 [FIN, PSH, URG] Seq=1 Win=1...

Acknowledgment number: 0
0101 = Header Length: 20 bytes (5)

Flags: 0x029 (FIN, PSH, URG)

Window size value: 1024
[Calculated window size: 1024]
[Window size scaling factor: -1 (unknown)]
Checksum: 0xe7fc [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0

[Timestamps]
[Time since first frame in this TCP stream: 0.000000000 seconds]
[Time since previous frame in this TCP stream: 0.000000000 seconds]

```
0000 00 15 5d 00 ba 06 00 15 5d 00 ba 09 08 00 45 00  ..]...[...E
0010 00 28 9c e2 00 00 2f 06 0b 3a c0 a8 b1 64 c0 a8  (..../.1..d
0020 b1 07 d5 de 02 2a 3c a9 cb 4f 00 00 00 00 50 29  ....*<.0...P)
0030 04 00 e7 fc 00 00 00 00 00 00 00 00 00 00 00 00  ....
```

Creating Snort rules

Shows a ping activity alert.

SGUIL-0.9.0 - Connected To localhost

Query Reports Sound: Off ServerName: localhost UserName: infosec UserID: 2 2024-09-29 23:33:07

RealTime Events Escalated Events

T	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
T	2000	ids-enp0s...	4.15084	2024-09-29 23:03:38	192.168.177.100	54750	192.168.177.7	554	6	Nmap XMAS Tree Scan
T	60	ids-enp0s...	4.17084	2024-09-29 23:33:07	192.168.177.100		192.168.177.47		1	GPL ICMP_INFO PING

P Resolution Agent Status Snort Statistics System Msg

Reverse DNS ☒ Enable External DNS

IP:
Name:

IP:
Name:

Query: ☒ None ☐ Src IP ☐ Dst IP

☒ Show Packet Data ☒ Show Rule

alert icmp \$EXTERNAL_NET any -> \$HOME_NET any (msg:"GPL ICMP_INFO PING *NIX";
itype:8; content:"[10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F]"; depth:32;
class:icmp; sid:3100265; rev:0; meta-date:created_at 2010-09-23; updated_at 2010-09-23;)

IP	Source IP	Dest IP	Ver	HL	TOS	len	ID	Flags	Offset	T
IP	192.168.177.100	192.168.177.47	4	5	0	84	57015	2	0	64

ICMP	Type	Code	ChkSum	ID	S
ICMP	8	0	46415	3171	1

DATA	
B3 E3 F9 66 00 00 00 00 BD 2E 0D 00 00 00 00 00	...f.....
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F	
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F	!"#\$%&'()*
30 31 32 33 34 35 36 37	01234567

OS: ~ SGUIL-0.9.0 SGUIL-0.9.0 - Connected Capturing from enp0s10f0 infosec@IDS: ~

ICMP packets generated by the ping activity.

Creating Snort rules cont'd

The image shows a Wireshark packet capture window titled '*enp0s10f0'. The filter bar is set to 'icmp'. The packet list shows 16 ICMP packets (No. 217 to 228) alternating between requests and replies. The packet details pane shows the structure of a selected ICMP Echo (ping) reply packet, including Ethernet II, Internet Protocol Version 4, and Internet Control Message Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
217	64.109785300	192.168.177.100	192.168.177.47	ICMP	98	Echo (ping) request id=0xc63, seq=...
218	64.111931500	192.168.177.47	192.168.177.100	ICMP	98	Echo (ping) reply id=0xc63, seq=...
221	65.111292100	192.168.177.100	192.168.177.47	ICMP	98	Echo (ping) request id=0xc63, seq=...
222	65.113007300	192.168.177.47	192.168.177.100	ICMP	98	Echo (ping) reply id=0xc63, seq=...
223	66.113468600	192.168.177.100	192.168.177.47	ICMP	98	Echo (ping) request id=0xc63, seq=...
224	66.114963300	192.168.177.47	192.168.177.100	ICMP	98	Echo (ping) reply id=0xc63, seq=...
225	67.115245600	192.168.177.100	192.168.177.47	ICMP	98	Echo (ping) request id=0xc63, seq=...
226	67.116967200	192.168.177.47	192.168.177.100	ICMP	98	Echo (ping) reply id=0xc63, seq=...
227	68.116412700	192.168.177.100	192.168.177.47	ICMP	98	Echo (ping) request id=0xc63, seq=...
228	68.118284500	192.168.177.47	192.168.177.100	ICMP	98	Echo (ping) reply id=0xc63, seq=...

Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
Ethernet II, Src: Microsof_00:ba:09 (00:15:5d:00:ba:09), Dst: Microsof_00:ba:0c (00:15:5d:00:ba:0c)
Internet Protocol Version 4, Src: 192.168.177.100, Dst: 192.168.177.47
Internet Control Message Protocol

```
0000 00 15 5d 00 ba 0c 00 15 5d 00 ba 09 08 00 45 00  --]---]---E-
0010 00 54 70 41 40 00 40 01 06 82 c0 a8 b1 64 c0 a8  -TpA@-@---d-
0020 b1 2f 08 00 d2 08 0c 63 01 2a dd e4 f9 66 00 00  -/---c-*---f-
0030 00 00 7d 4b 05 00 00 00 00 00 10 11 12 13 14 15  --}K-----
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25  -...!""#$%
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35  &'()*+,-./012345
0060 36 37 67
```

wireshark_enp0s10f0_20240929233805_GwhhWO.pcapng Packets: 228 · Displayed: 168 (73.7%) Profile: Default

Live Memory Analysis

I practiced various live memory analysis techniques, working with two virtual machines Ubuntu Web for Linux tools and a Malware VM for Windows tools.

Linux Process Analysis:

Using command line tools (ps, lsof, netstat) I examined running processes, open files, and network connections

Checked system logs and boot information

Created and monitored test network ports

Windows Tools Analysis:

Using Process Hacker I examined process relationships and properties

Using Process Monitor I tracked registry changes and application behavior

Using Process Monitor

- Tracked registry changes in real-time
- Set filters to focus on specific activities
- Used Notepad as a test case to see how applications store settings
- Learned to track and analyze registry modifications

Optional Volatility Framework:

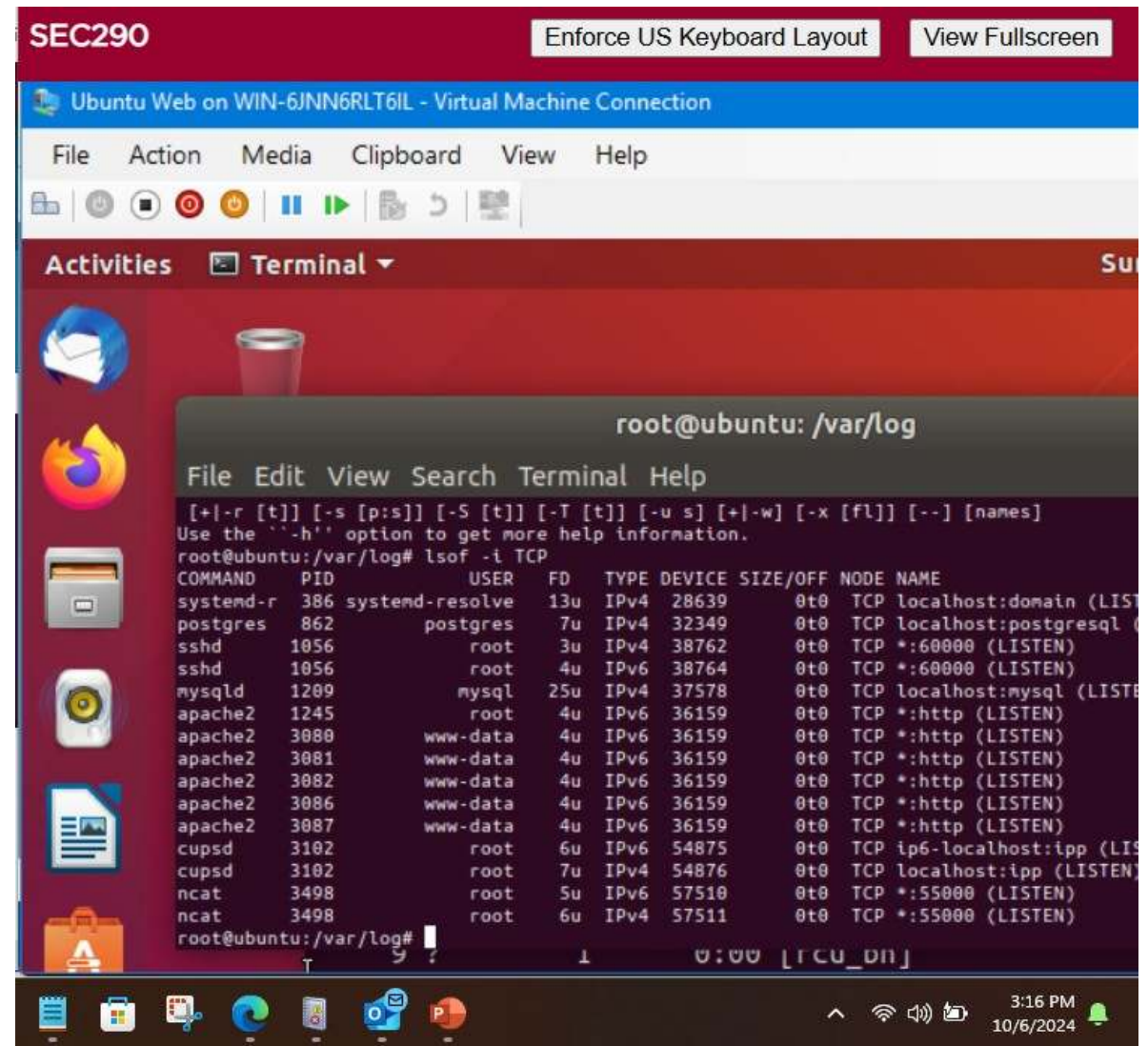
Analyzing memory dumps

Using various commands to examine processes and connections

Identifying malware through memory analysis

Linux Processes

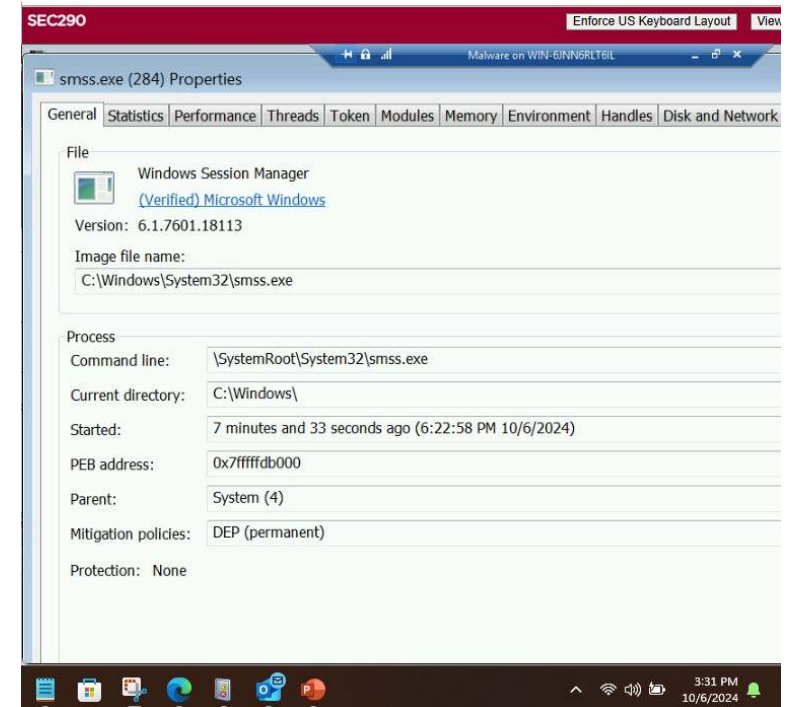
port 55000 open for both IPv4 and IPv6.



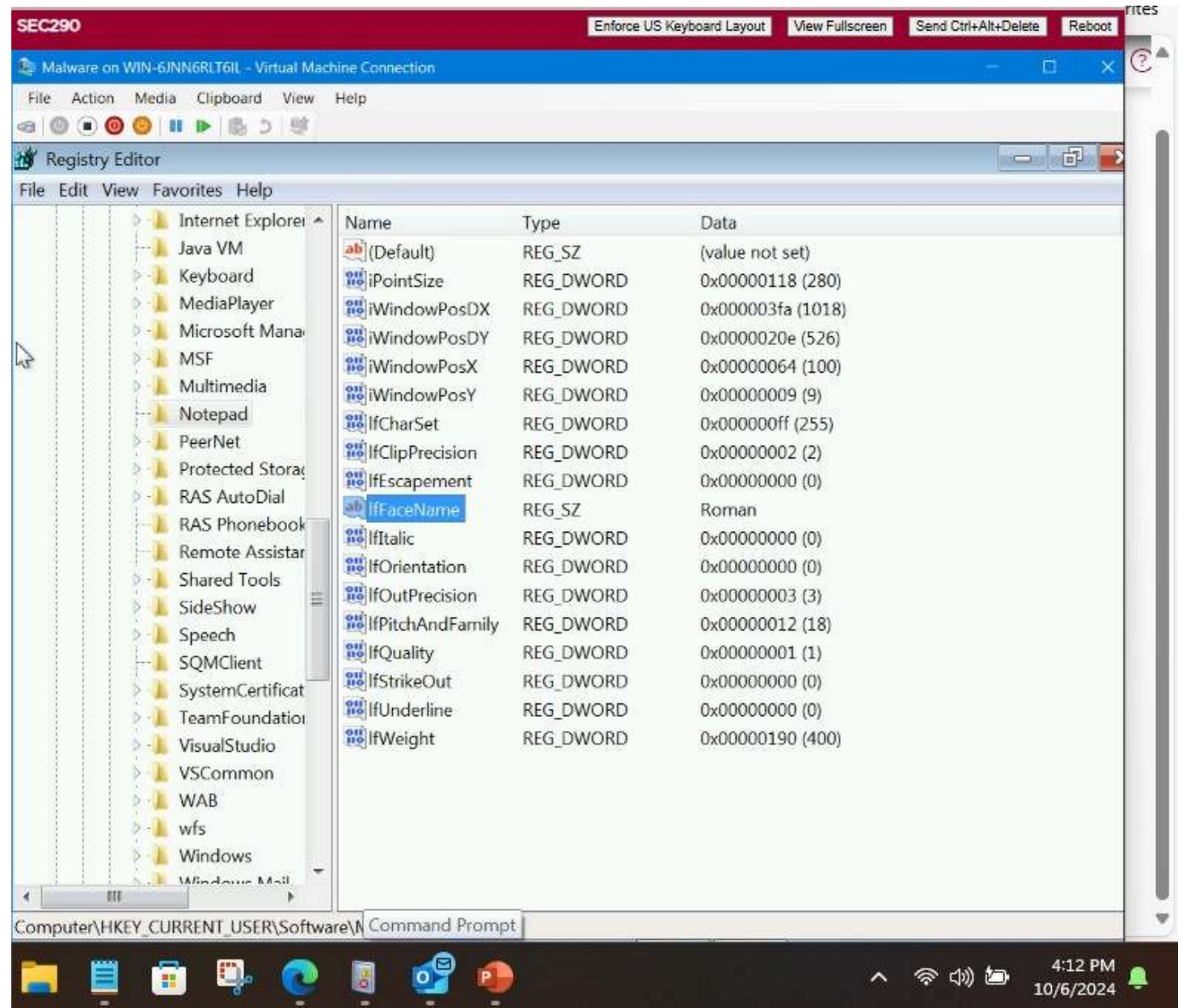
```
SEC290 Enforce US Keyboard Layout View Fullscreen
Ubuntu Web on WIN-6JNN6RLT6IL - Virtual Machine Connection
File Action Media Clipboard View Help
Activities Terminal
root@ubuntu: /var/log
File Edit View Search Terminal Help
[+|-r [t]] [-s [p:s]] [-S [t]] [-T [t]] [-u s] [+|-w] [-x [fl]] [--] [names]
Use the '-h' option to get more help information.
root@ubuntu:/var/log# lsof -i TCP
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
systemd-r 386 systemd-resolve 13u IPv4 28639 0t0 TCP localhost:domain (LISTEN)
postgres 862 postgres 7u IPv4 32349 0t0 TCP localhost:postgresql (LISTEN)
sshd 1056 root 3u IPv4 38762 0t0 TCP *:60000 (LISTEN)
sshd 1056 root 4u IPv6 38764 0t0 TCP *:60000 (LISTEN)
mysqld 1209 mysql 25u IPv4 37578 0t0 TCP localhost:mysql (LISTEN)
apache2 1245 root 4u IPv6 36159 0t0 TCP *:http (LISTEN)
apache2 3080 www-data 4u IPv6 36159 0t0 TCP *:http (LISTEN)
apache2 3081 www-data 4u IPv6 36159 0t0 TCP *:http (LISTEN)
apache2 3082 www-data 4u IPv6 36159 0t0 TCP *:http (LISTEN)
apache2 3086 www-data 4u IPv6 36159 0t0 TCP *:http (LISTEN)
apache2 3087 www-data 4u IPv6 36159 0t0 TCP *:http (LISTEN)
cupsd 3102 root 6u IPv6 54875 0t0 TCP ip6-localhost:ipp (LISTEN)
cupsd 3102 root 7u IPv4 54876 0t0 TCP localhost:ipp (LISTEN)
ncat 3498 root 5u IPv6 57510 0t0 TCP *:55000 (LISTEN)
ncat 3498 root 6u IPv4 57511 0t0 TCP *:55000 (LISTEN)
root@ubuntu:/var/log#
```

Process Hacker

-
- Shows properties of a chosen process.



Process Monitor



Firewall

Using three virtual machines Ubuntu Web, Firewall Machine, and DMZ Machine I created a realistic network environment. I practiced configuring security rules, managing network access, and filtering traffic between different network segments. The following principles were covered in this activity:

Network Setup and Basic Routing:

- Configuring IP addresses and interfaces
- Setting up basic routing between networks
- Testing connectivity with ping
- Using nmap for port scanning

Time-Based Access Control:

- Configuring iptables rules with time restrictions
- Setting up SSH access only during business hours
- Testing access during and outside allowed times

Packet Filtering Rules:

- Setting default policies
- Creating specific allow/deny rules
- Working with TCP/UDP ports

Stateful Firewall Configuration:

- Configuring connection tracking
- Setting up stateful filtering rules
- Managing established connections

Network Address Translation (NAT):

- Implementing Dynamic NAT (DNAT)
- Setting up PREROUTING and POSTROUTING rules

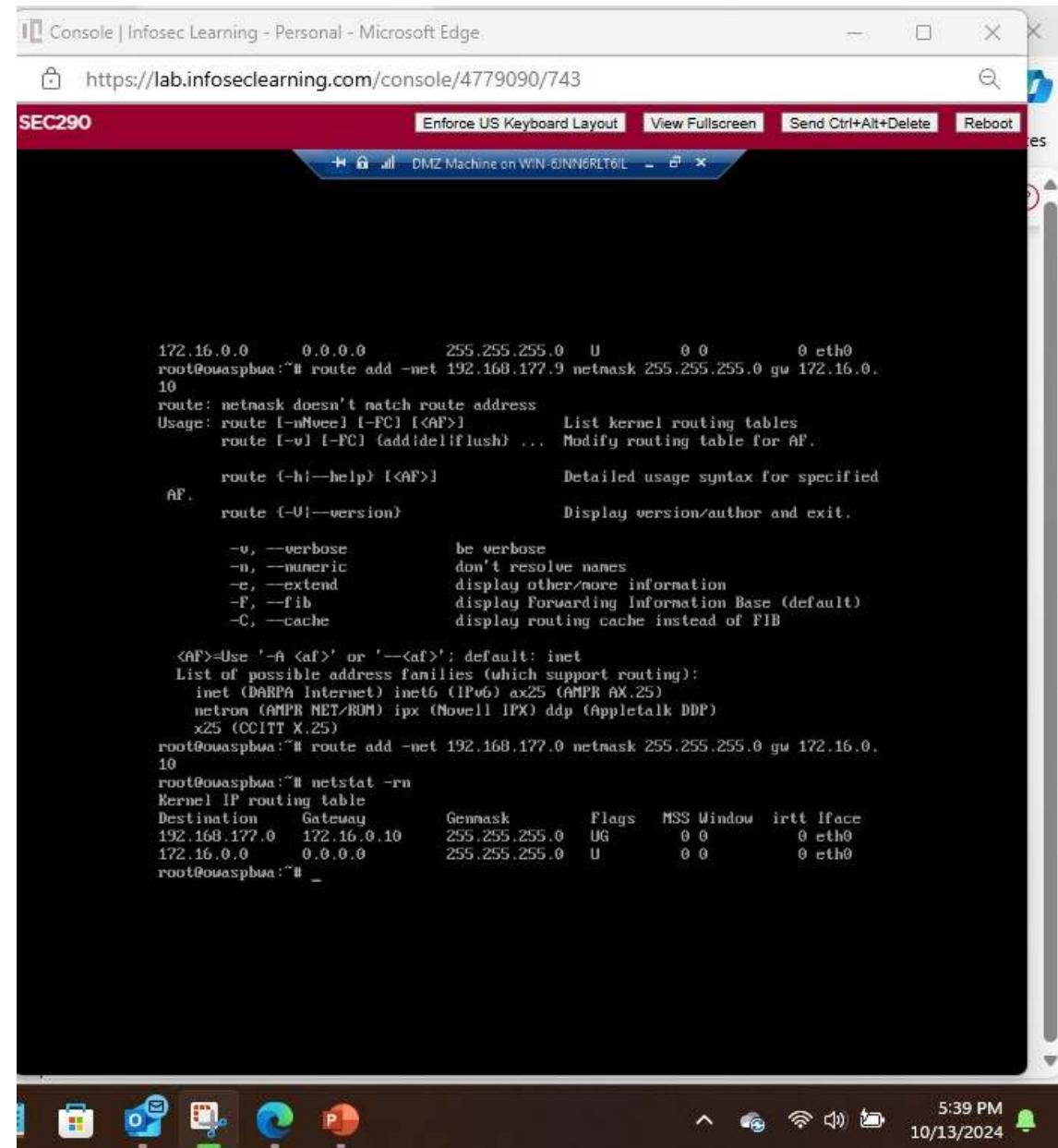
Managing port forwarding

Security Hardening:

- Implementing brute force protection for SSH
- Setting up black hole routing
- Configuring bogon filtering
- Creating custom chains for specific security purposes

Output of the DMZ Route Table.

Time-based Access



The screenshot shows a web browser window titled "Console | Infosec Learning - Personal - Microsoft Edge" with the URL "https://lab.infoseclearning.com/console/4779090/743". The terminal interface has a red header bar with "SEC290" and buttons for "Enforce US Keyboard Layout", "View Fullscreen", "Send Ctrl+Alt+Delete", and "Reboot". Below the header, a blue bar indicates the environment: "DMZ Machine on WIN-6JNN6RLT6IL".

The terminal output shows the following commands and results:

```
172.16.0.0    0.0.0.0      255.255.255.0  U        0 0        0 eth0
root@owaspbua:~# route add -net 192.168.177.9 netmask 255.255.255.0 gw 172.16.0.10
route: netmask doesn't match route address
Usage: route [-mMvee] [-FC] [<AF>]          List kernel routing tables
       route [-v] [-FC] {add|del|flush} ...  Modify routing table for AF.

       route {-h|--help} [<AF>]             Detailed usage syntax for specified AF.
       route {-U|--version}                 Display version/author and exit.

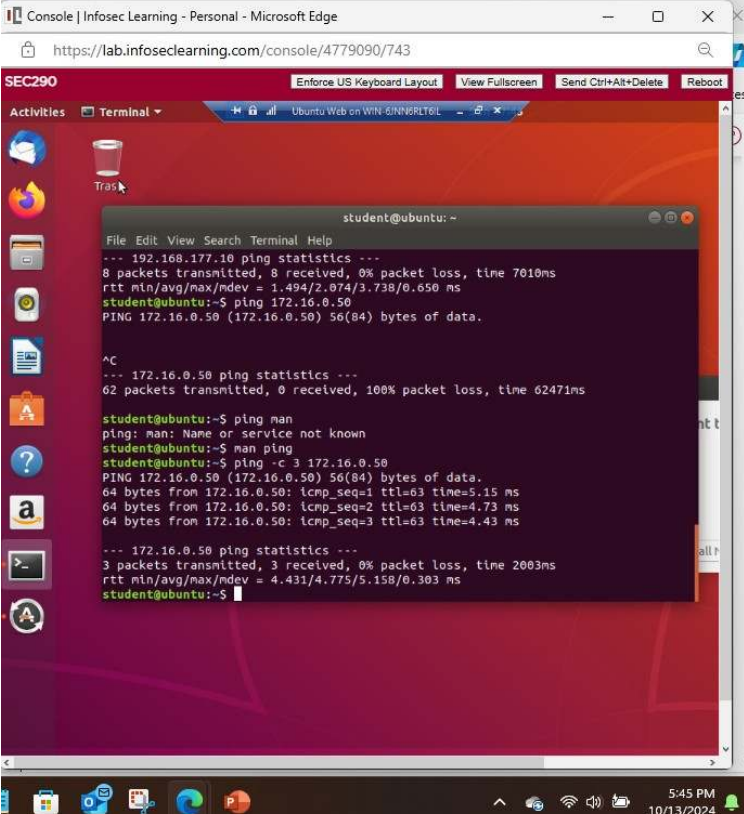
       -v, --verbose                        be verbose
       -n, --numeric                        don't resolve names
       -e, --extend                        display other/more information
       -F, --fib                           display Forwarding Information Base (default)
       -C, --cache                         display routing cache instead of FIB

<AF>=Use '-A <af>' or '--<af>': default: inet
List of possible address families (which support routing):
inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
netron (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
x25 (CCITT X.25)
root@owaspbua:~# route add -net 192.168.177.0 netmask 255.255.255.0 gw 172.16.0.10
root@owaspbua:~# netstat -rn
Kernel IP routing table
Destination        Gateway           Genmask          Flags   MSS Window  irtt Iface
192.168.177.0      172.16.0.10      255.255.255.0    UG        0 0          0 eth0
172.16.0.0         0.0.0.0          255.255.255.0    U        0 0          0 eth0
root@owaspbua:~#
```

The Windows taskbar at the bottom shows the time as 5:39 PM on 10/13/2024, along with various system icons and open application windows.

Time-based Access

- Ping from the Ubuntu Web VM and the DMZ VM.

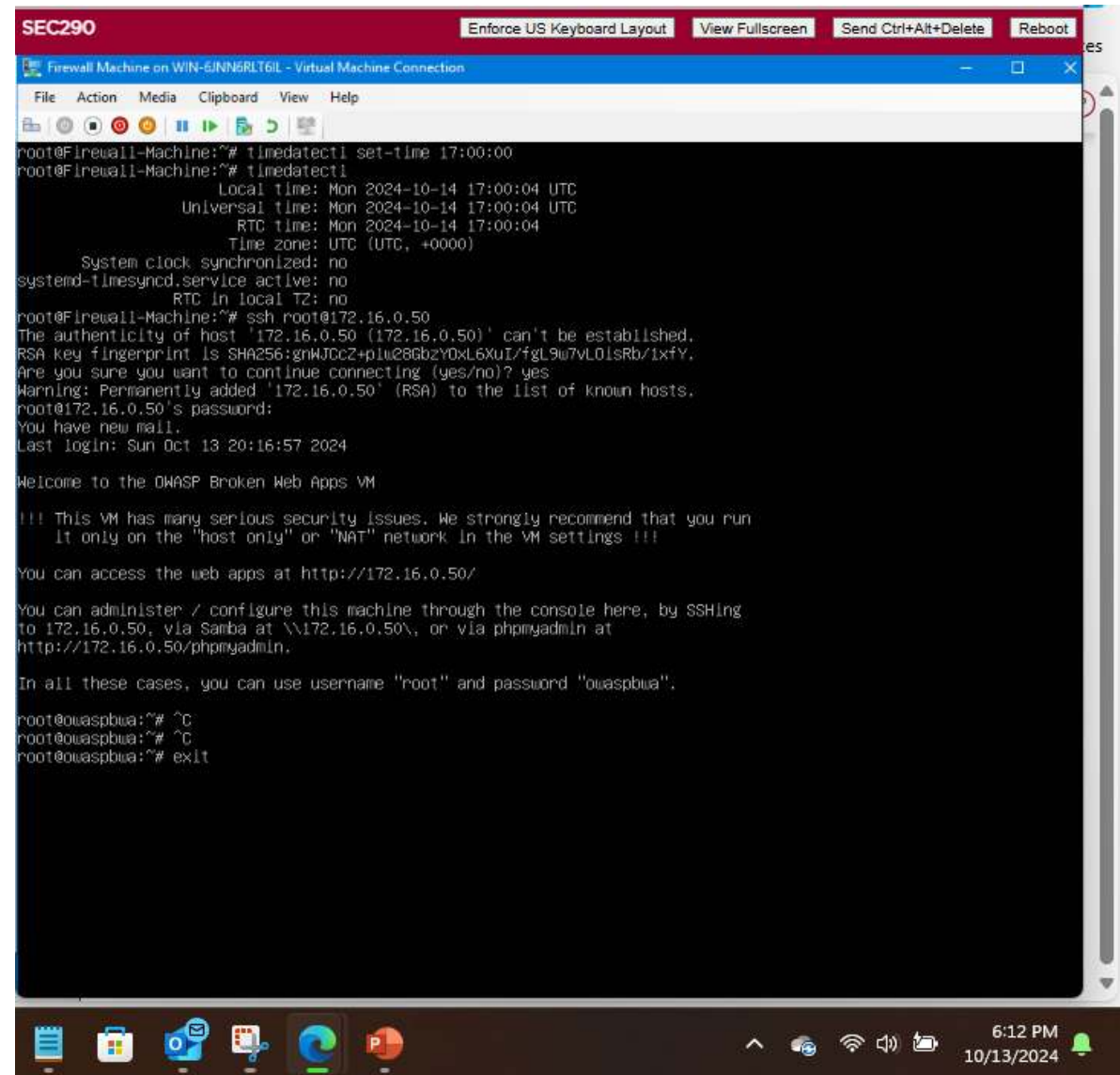


The screenshot shows a web browser window titled "Console | Infosec Learning - Personal - Microsoft Edge" with the URL "https://lab.infoseclearning.com/console/4779090/743". The browser displays a virtual machine interface for "SEC290" with a terminal window open. The terminal shows the following output:

```
student@ubuntu: ~  
File Edit View Search Terminal Help  
--- 192.168.177.10 ping statistics ---  
8 packets transmitted, 8 received, 0% packet loss, time 7010ms  
rtt min/avg/max/mdev = 1.494/2.074/3.738/0.650 ms  
student@ubuntu:~$ ping 172.16.0.50  
PING 172.16.0.50 (172.16.0.50) 56(84) bytes of data.  
  
^C  
--- 172.16.0.50 ping statistics ---  
62 packets transmitted, 0 received, 100% packet loss, time 62471ms  
  
student@ubuntu:~$ ping man  
ping: man: Name or service not known  
student@ubuntu:~$ man ping  
student@ubuntu:~$ ping -c 3 172.16.0.50  
PING 172.16.0.50 (172.16.0.50) 56(84) bytes of data.  
64 bytes from 172.16.0.50: icmp_seq=1 ttl=63 time=5.15 ms  
64 bytes from 172.16.0.50: icmp_seq=2 ttl=63 time=4.73 ms  
64 bytes from 172.16.0.50: icmp_seq=3 ttl=63 time=4.43 ms  
  
--- 172.16.0.50 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2003ms  
rtt min/avg/max/mdev = 4.431/4.775/5.158/0.303 ms  
student@ubuntu:~$
```

Time-based Access

Two time-based access rules in the FORWARD chain



```
SEC290 Enforce US Keyboard Layout View Fullscreen Send Ctrl+Alt+Delete Reboot
Firewall Machine on WIN-6JNN6RLT6IL - Virtual Machine Connection
File Action Media Clipboard View Help
root@Firewall-Machine:~# timedatectl set-time 17:00:00
root@Firewall-Machine:~# timedatectl
          Local time: Mon 2024-10-14 17:00:04 UTC
          Universal time: Mon 2024-10-14 17:00:04 UTC
              RTC time: Mon 2024-10-14 17:00:04
              Time zone: UTC (UTC, +0000)
System clock synchronized: no
systemd-timesyncd.service active: no
          RTC in local TZ: no
root@Firewall-Machine:~# ssh root@172.16.0.50
The authenticity of host '172.16.0.50 (172.16.0.50)' can't be established.
RSA key fingerprint is SHA256:gnWJcc2+piw28GbzyDxL6XuI/fgL9w7vL0IsRb/1x+y.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.0.50' (RSA) to the list of known hosts.
root@172.16.0.50's password:
You have new mail.
Last login: Sun Oct 13 20:16:57 2024

Welcome to the OWASP Broken Web Apps VM

!!! This VM has many serious security issues. We strongly recommend that you run
    it only on the "host only" or "NAT" network in the VM settings !!!

You can access the web apps at http://172.16.0.50/

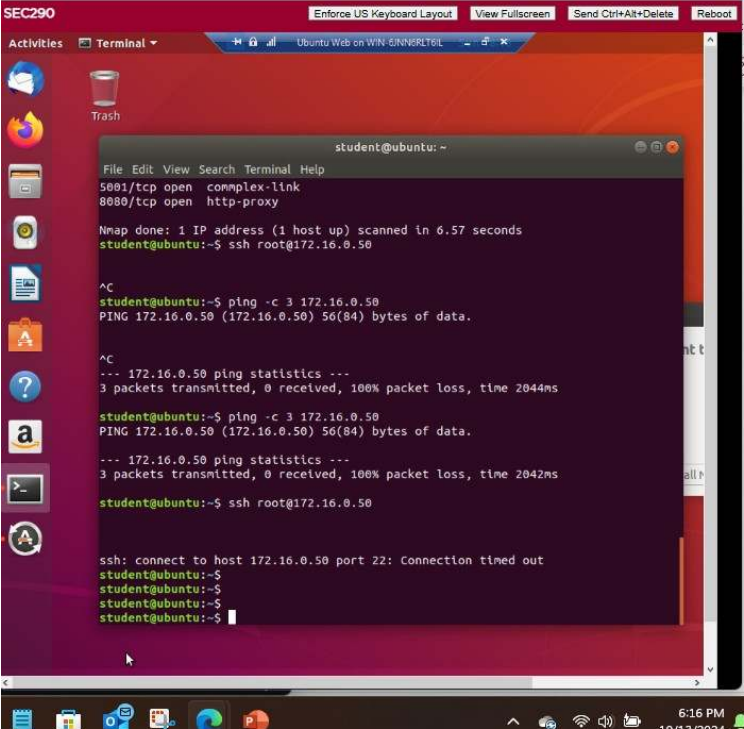
You can administer / configure this machine through the console here, by SSHing
to 172.16.0.50, via Samba at \\172.16.0.50\, or via phpmyadmin at
http://172.16.0.50/phpmyadmin.

In all these cases, you can use username "root" and password "owaspbwa".

root@owaspbwa:~# ^C
root@owaspbwa:~# ^C
root@owaspbwa:~# exit
```

I was able to log in via SSH from the firewall to the DMZ, as you can see in the previous picture.

In the picture here it shows the successful ping between Ubuntu and the DMZ. But when I try to SSH It times out.



The screenshot shows a terminal window titled 'student@ubuntu: ~' with the following content:

```
File Edit View Search Terminal Help
5001/tcp open  complex-link
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 6.57 seconds
student@ubuntu:~$ ssh root@172.16.0.50

^C
student@ubuntu:~$ ping -c 3 172.16.0.50
PING 172.16.0.50 (172.16.0.50) 56(84) bytes of data.

^C
--- 172.16.0.50 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2044ms

student@ubuntu:~$ ping -c 3 172.16.0.50
PING 172.16.0.50 (172.16.0.50) 56(84) bytes of data.

--- 172.16.0.50 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2042ms

student@ubuntu:~$ ssh root@172.16.0.50

ssh: connect to host 172.16.0.50 port 22: Connection timed out
student@ubuntu:~$
student@ubuntu:~$
student@ubuntu:~$
student@ubuntu:~$
```

The terminal window is part of a desktop environment with a sidebar on the left containing icons for Trash, Firefox, and other applications. The top of the window shows a menu bar with 'SEC290' and 'Enforce US Keyboard Layout'. The bottom of the window shows a taskbar with icons for various applications and a system tray with the time '6:16 PM' and date '10/13/2024'.

Challenges in the Project

- Managing multiple VMs running simultaneously (up to 3 at once)
- Troubleshooting failed ping tests between
- Machines
- Understanding and configuring multiple network interfaces
- Managing elevated privileges correctly (sudo commands)
- Having to carefully type long commands accurately
- Understanding the relationships between processes in Process Hacker
- Following registry changes in Process Monitor

Career Skills

1. Security Tool Proficiency:

Experience with industry-standard tools like Snort, iptables, and Process Monitor

Understanding of both Linux and Windows security tools

Ability to use monitoring and analysis tools in real-time

2. Network Security Configuration:

Hands-on firewall configuration experience

Implementation of access controls

Experience with NAT and routing

3. Incident Response & Analysis:

Memory analysis techniques

Recognition of suspicious activity

Understanding system logs and alerts

4. System Administration:

Working across multiple operating systems (Linux/Windows)

Understanding of system processes and services

5. Security Policy Implementation:

Time-based access management

Understanding of security best practices

References

Chapple, M., & Seidl, D. (2020). CompTIA CySA+ study guide: Exam CS0-002 (2nd ed.). Wiley.