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.

1. Initial Reconnaissance:
   - Using Nmap to identify hosts and open ports
   - Network enumeration using nbtscan
2. Vulnerability Testing:
   - 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
3. Exploitation Process:
   - Using Metasploit Framework (MSF) to exploit discovered vulnerabilities
   - Gaining shell access to vulnerable systems
   - Upgrading basic shells to Meterpreter shells for advanced functionality
4. Post-Exploitation:
   - Exploring various post-exploitation modules
   - Managing multiple sessions
   - Using 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 Analysis:

Analyzing 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

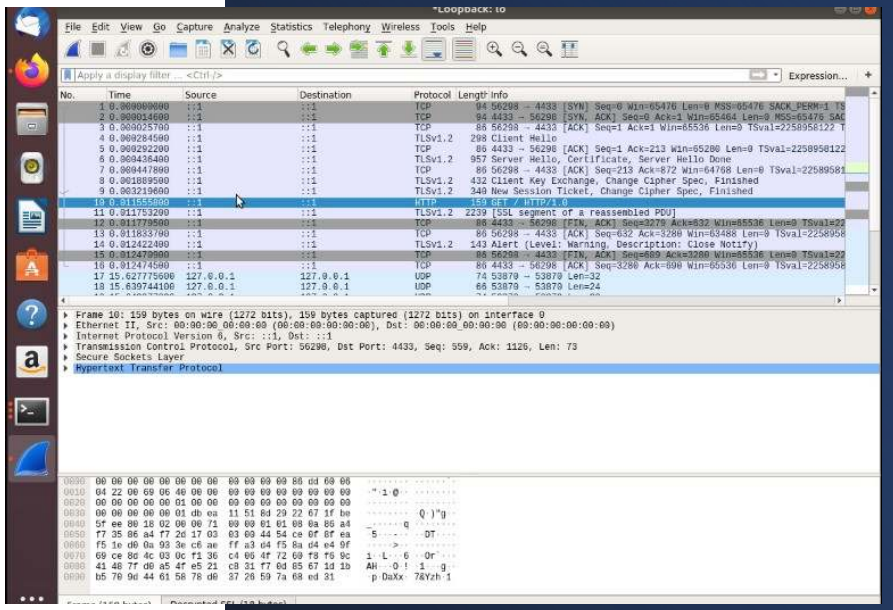
The objective of this activity was to create, test, and analyze SSL/TLS connections using OpenSSL and Wireshark. Using a virtual machine running Ubuntu, called the Ubuntu Web VM, I went over the following concepts:

- 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

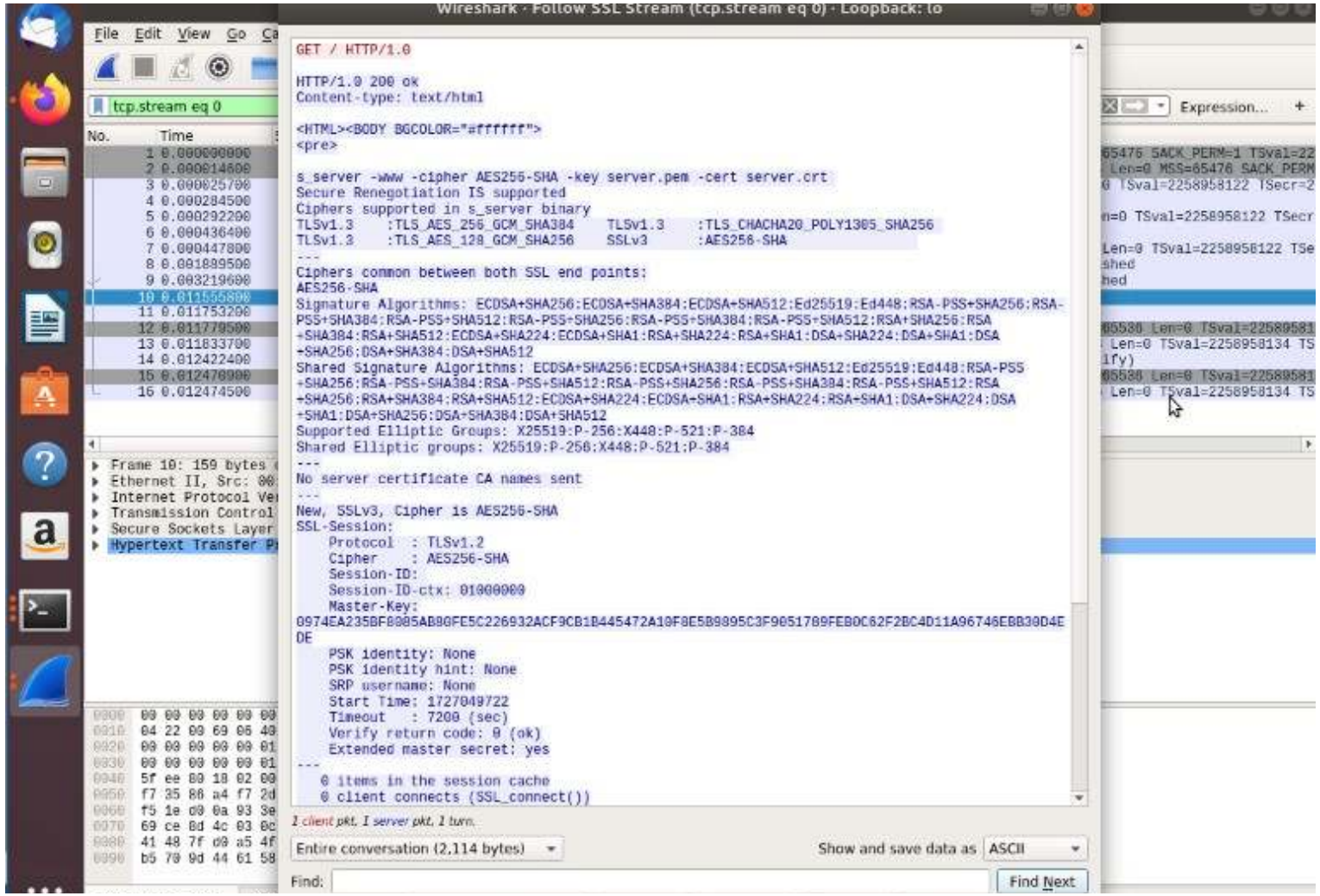
During this lab I demonstrated practical encryption implementation and analysis, setting up secure connections and decrypting SSL/TLS traffic with proper key access.

Creating and testing an SSL/TLS file

Notice the GET request in the Info column.



Creating and testing an SSL/TLS file cont’d



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

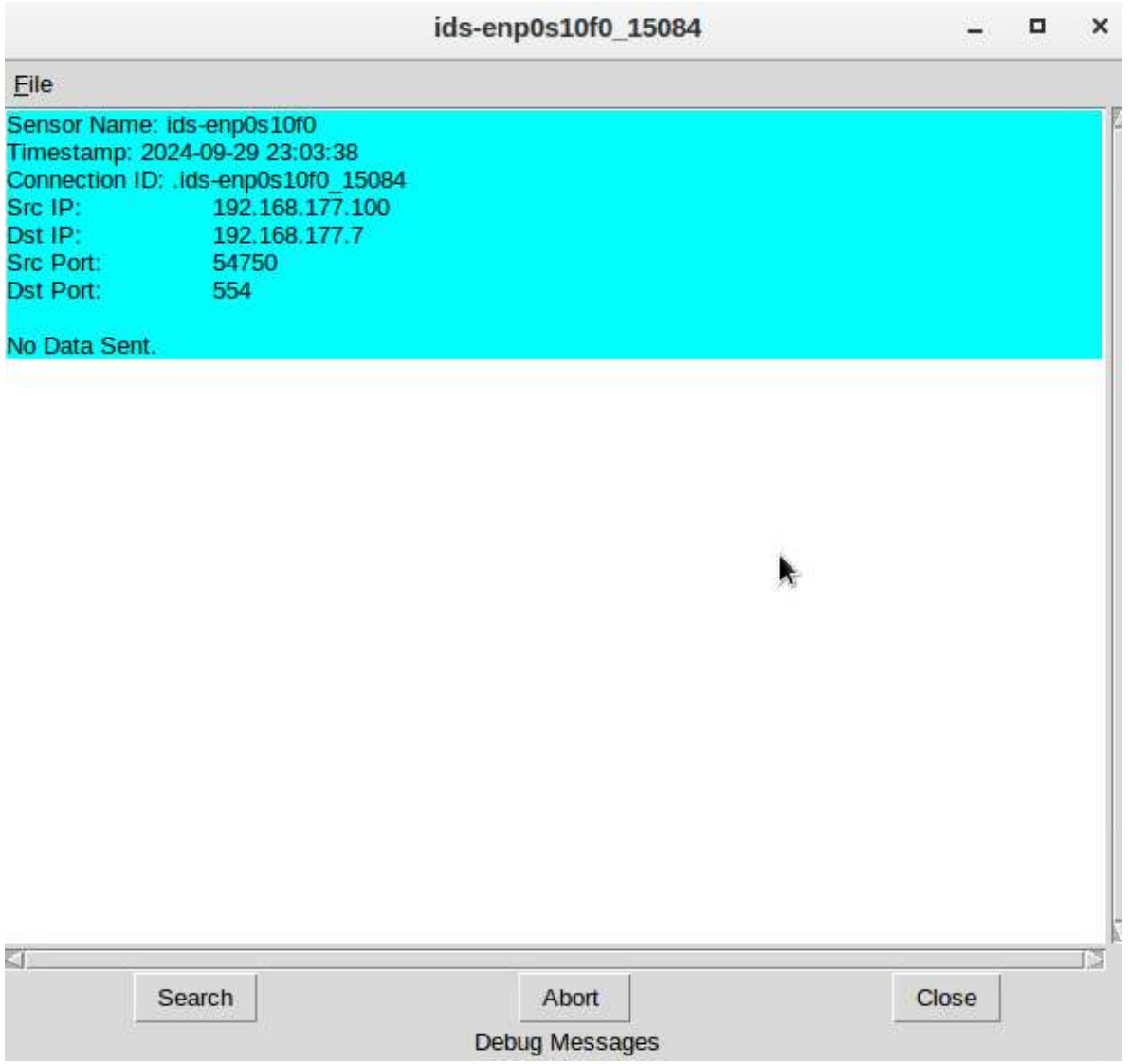
Creating Custom Snort Rules:

- 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.

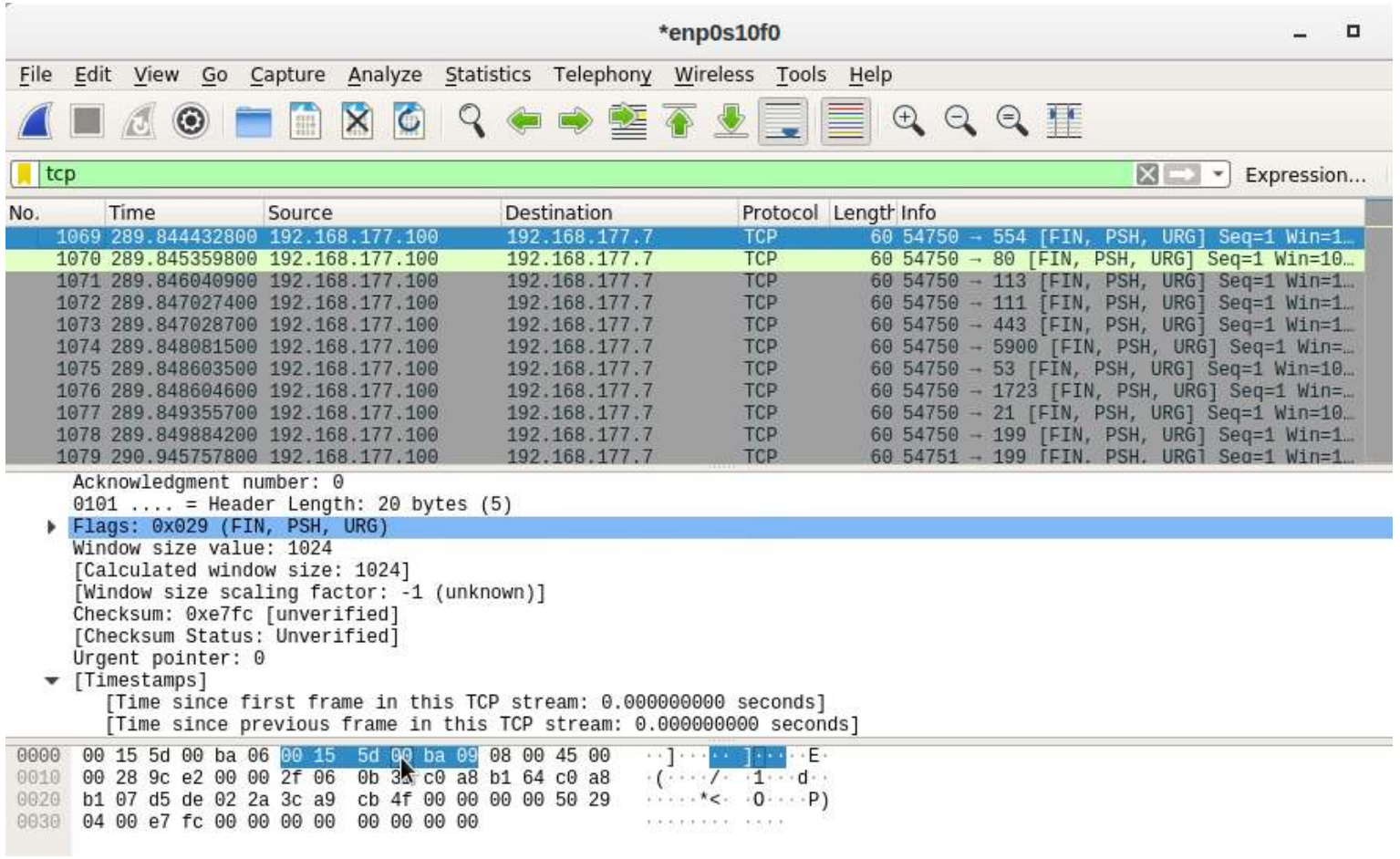
Testing Snort rules

XMAS scan alert.



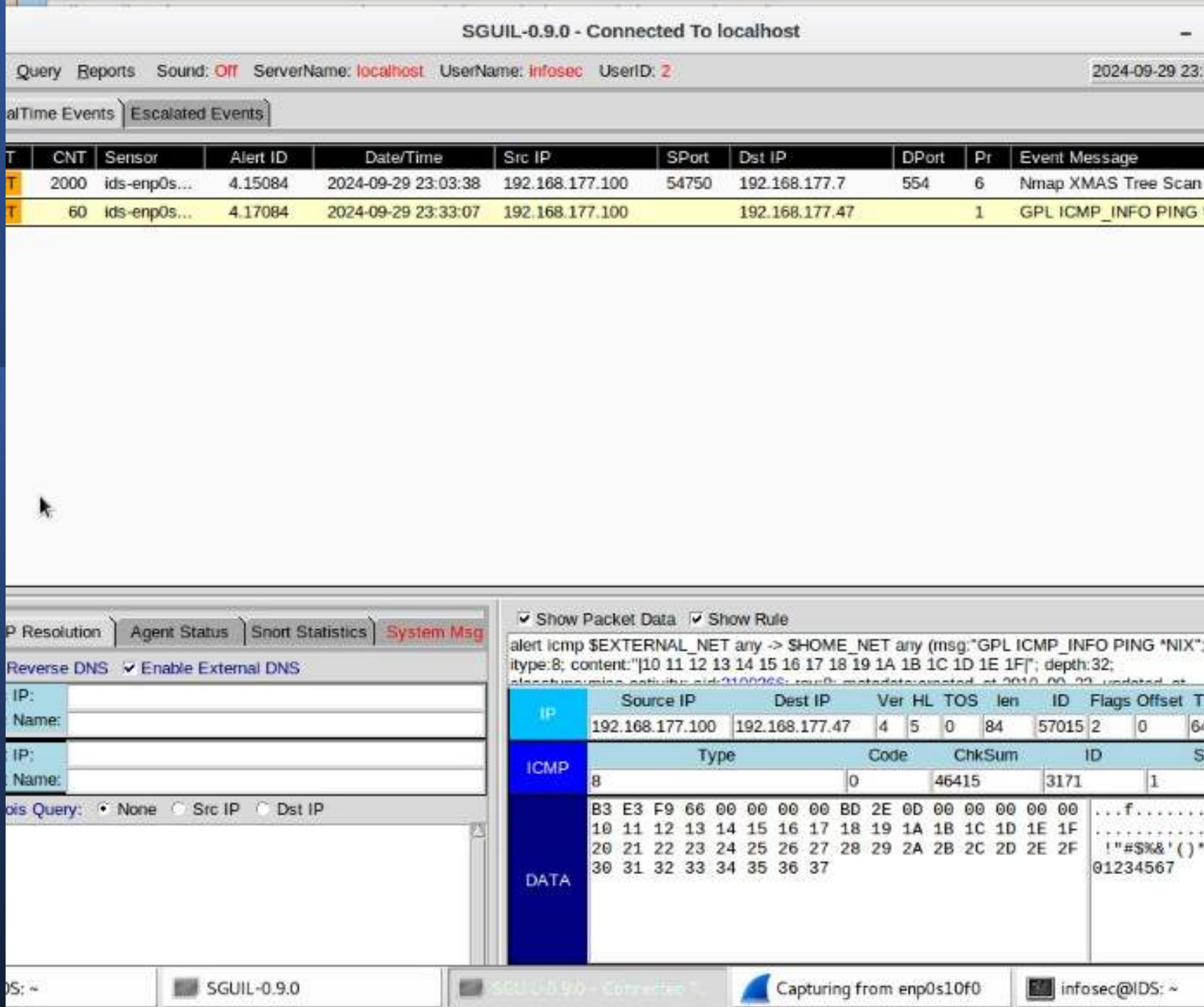
Testing Snort rules cont'd

TCP packets generated by the XMAS scan.



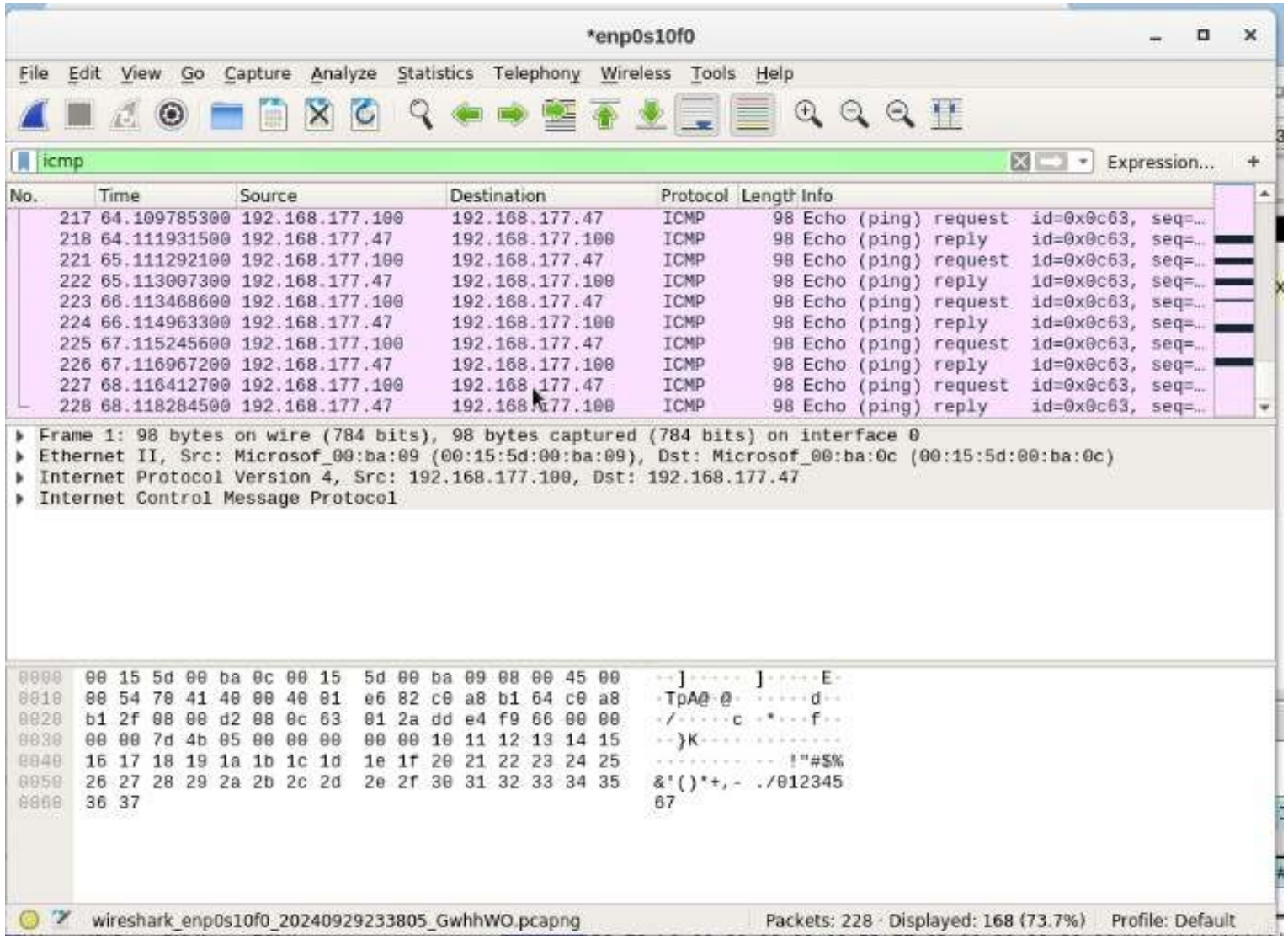
Shows a ping activity alert.

Creating Snort rules



ICMP packets generated by the ping activity.

Creating Snort rules cont'd



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:

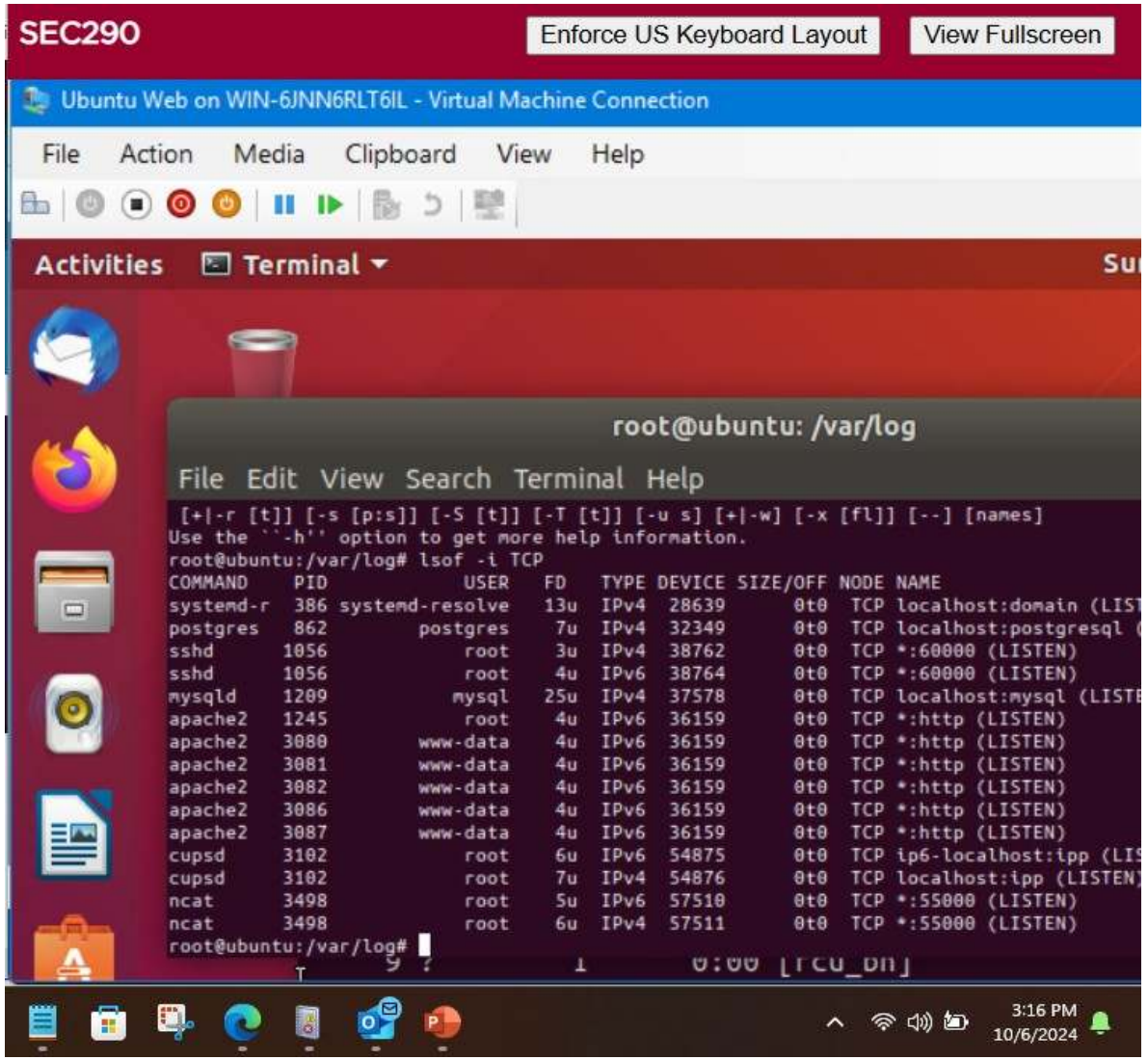
Analyzing memory dumps

Using various commands to examine processes and connections

Identifying malware through memory analysis

Linux Processes

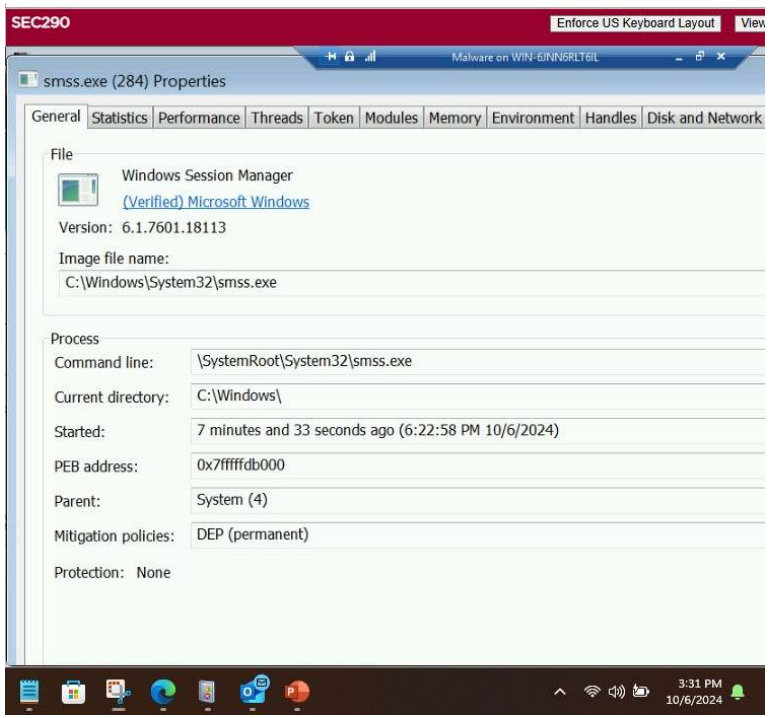
port 55000 open for both IPv4 and IPv6.



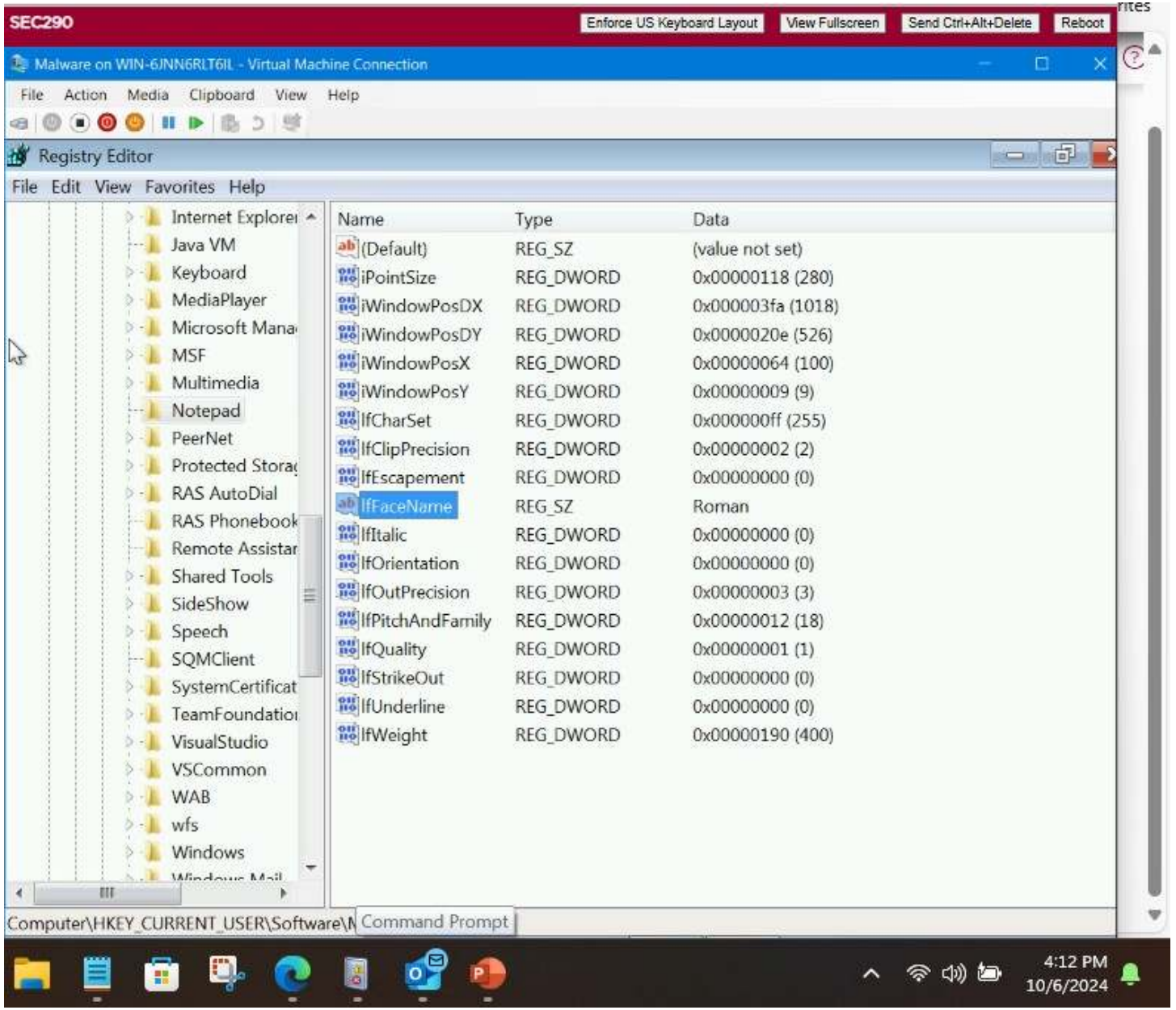
Process Hacker

•.

•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):

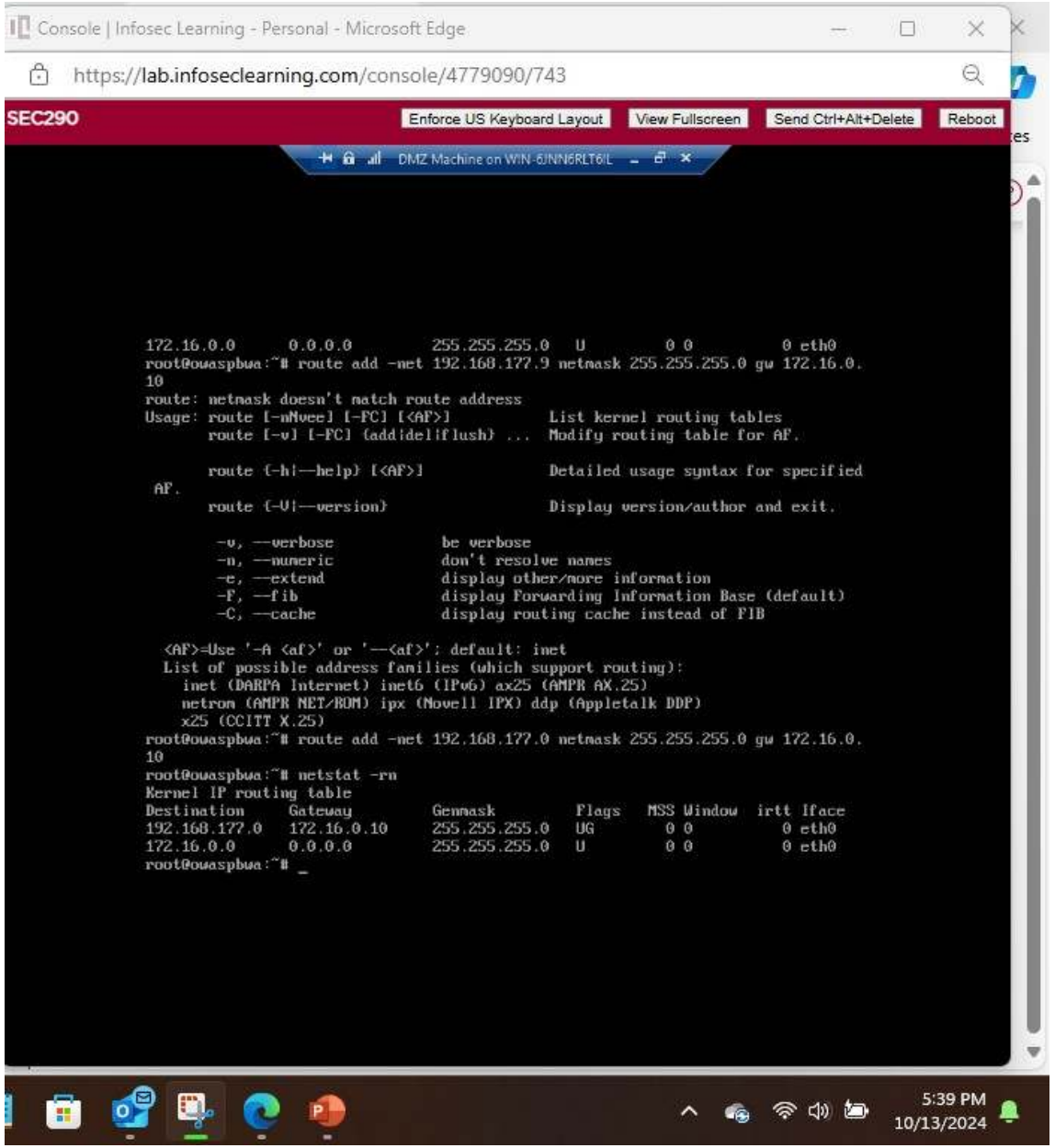
- Implementing Dynamic NAT (DNAT)
- Setting up PREROUTING and POSTROUTING rules

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

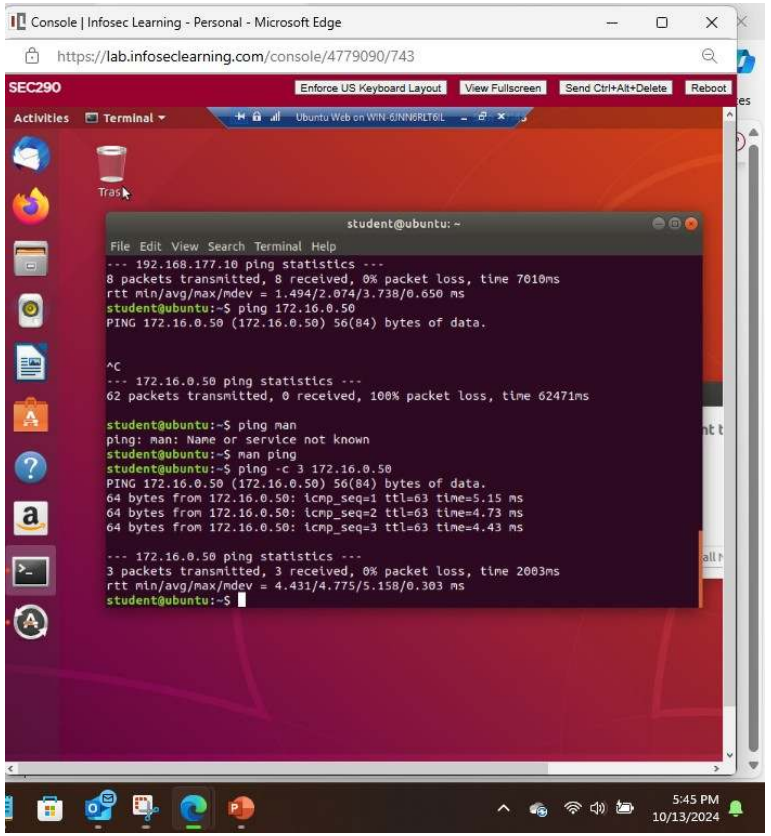
Output of the DMZ Route Table.

Time-based Access



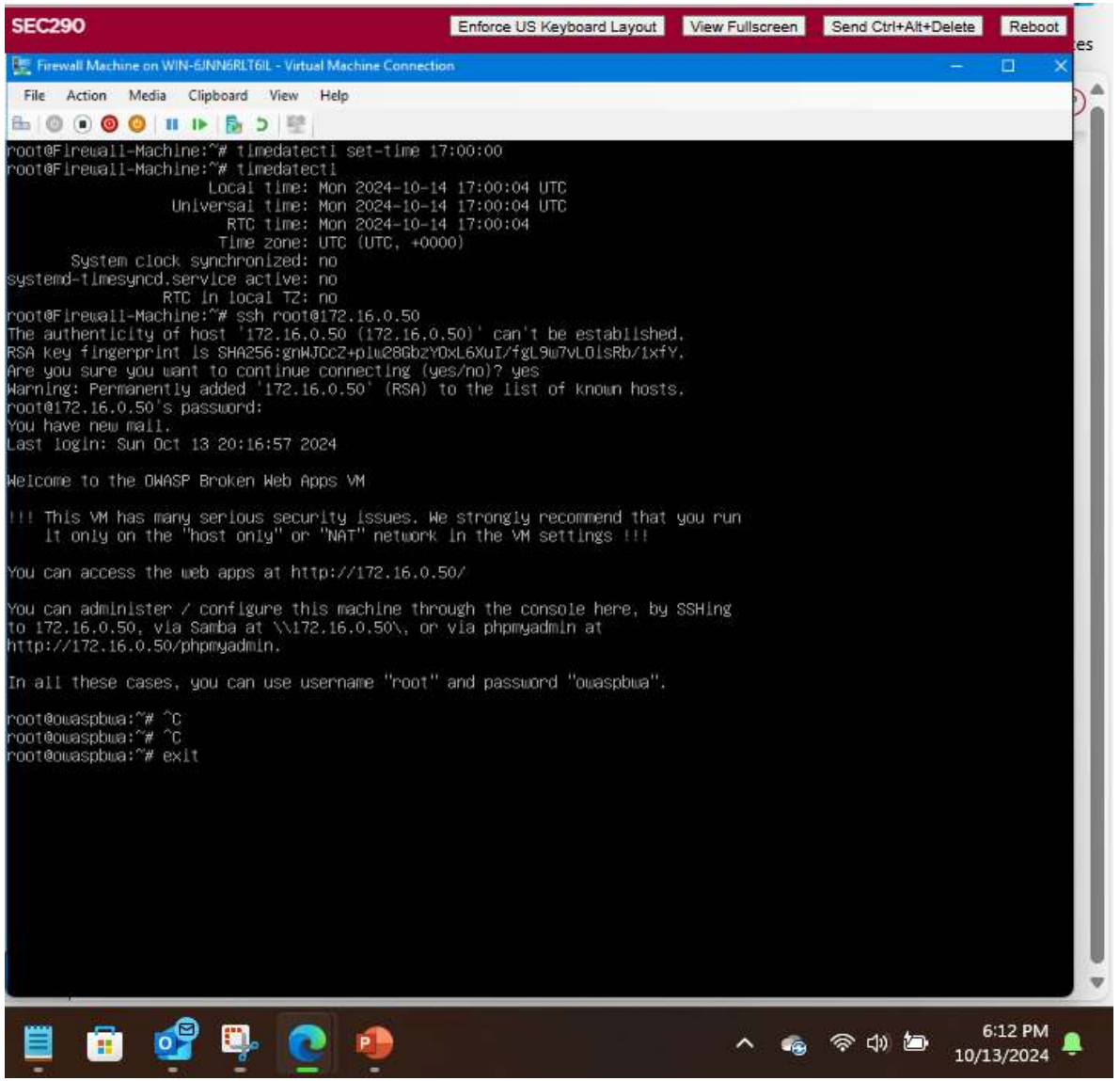
Time-based Access

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



Time-based Access

Two time-based access rules in the FORWARD chain



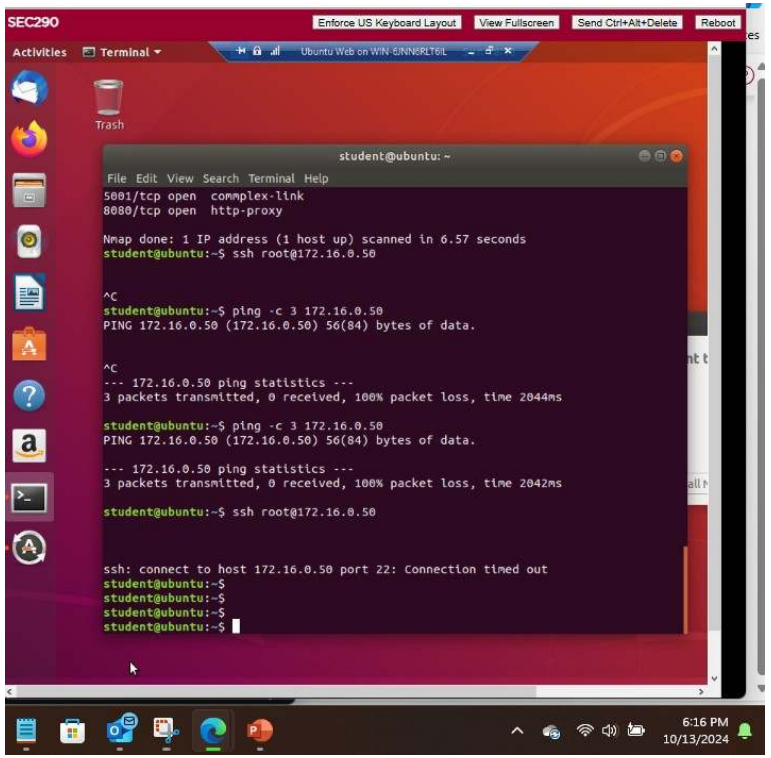
I was able to log in via SSH from the firewall to

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.



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.