

Red vs Blue

Design and Implementation of a Virtual Enterprise IT Infrastructure

Final Project Documentation

By

Team C

Pranay Garapati

Swarupa Jeedimetla

Dheeraj Kumar Ammineni

Manisharan Chakali

Brahmendra Chowdary Ponduri

A Master's Research Project Submitted to
The School for Professional Studies
in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Cybersecurity

Instructor

Syam Sai Siddabhattula

Saint Louis University
December 2025

Powered by **Ears Up Cyber Security LLC**

Table of Contents

Introduction	3
Virtualization Setup	3
Proxmox Virtual Environment	4
• VyOS Router Configuration	5
• pfSense Firewall Configuration	6
• Windows VM Configuration	7
• Ubuntu VM Configuration	9
Firewall & Router Setup	11
• VyOS and pfSense Connectivity Check	11
• Connectivity Testing Checklist	13
Firewall Security & Controlled Exposure with pfSense & Windows 10	13
• Creating Non-Admin User and Enabling Services	13
• pfSense Firewall Rules	13
• Connectivity and Ports Testing Checklist	14
Wazuh & Sysmon Configuration	15
• Ubuntu-Wazuh Manager Deployment	15
• pfSense Firewall Configuration	16
• Windows 10 Wazuh Agent & Sysmon Setup	17
Windows 10 Home & Ubuntu Hardening	19
Threat Intelligence & MITRE ATT&CK Mapping	21
VPN Configuration & Connectivity	24
Red Team Offensive Simulation	25
• Finding 1: SMB Access Achieved	25
• Finding 2: IIS Running with Default Configuration & TRACE Enable	27
• Finding 3: Network Reconnaissance Identified Open Services	27
• Finding 4: SMB Login Brute-Force Attempts Blocked	29
• Finding 5: SMB Signing Disabled	30
• Finding 6: Remote Service Information Disclosure via RPC	31
• Finding 7: FTP Access on Port 123	32
Blue Team Defensive Simulation	34
Learning Outcomes	39
Reflection	40
References	

Final Project Documentation

Introduction

This was a project that created and secured a complete virtualized enterprise network setting based on Proxmox VE, VyOS routing, pfSense firewalling, inner-network segmentation, and Wazuh monitoring. Both the Red Team's offensive testing and the Blue Team's defensive monitoring were supported by the environment. Within a few weeks, the team has configured the base infrastructure, segmented the internal network, installed security tooling, and conducted realistic attack-and-defense scenarios.

Virtualization Setup

The first task was to prepare the groundwork of our virtual laboratory environment with the help of Proxmox Virtual Environment (VE). We accomplished it through the installation of a complete virtual network, which comprised WAN and LAN bridges (Ford, 2019). This setup included:

- VyOS Router in the public WAN IP (203.0.115.2/25), and private LAN IP (178.18.3.1/24).
- Ubuntu on the LAN bridge that had the IP of 178.18.3.2/24.
- PfSense firewall with a WAN facing IP (178.18.3.3/24) and internal LAN IP (192.168.102.1/24).
- Windows 10 VM, with the internal IP 192.168.102.2/24, which we configured to work behind and get the PfSense firewall protection.

Figure 1

Network Diagram



Proxmox Virtual Environment

Proxmox VE is an open-source and full-fledged server management software to virtualize businesses.

It is a program that develops and operates virtual machines (VMs) (Ford, 2019).

Figure 2

Proxmox VE Login

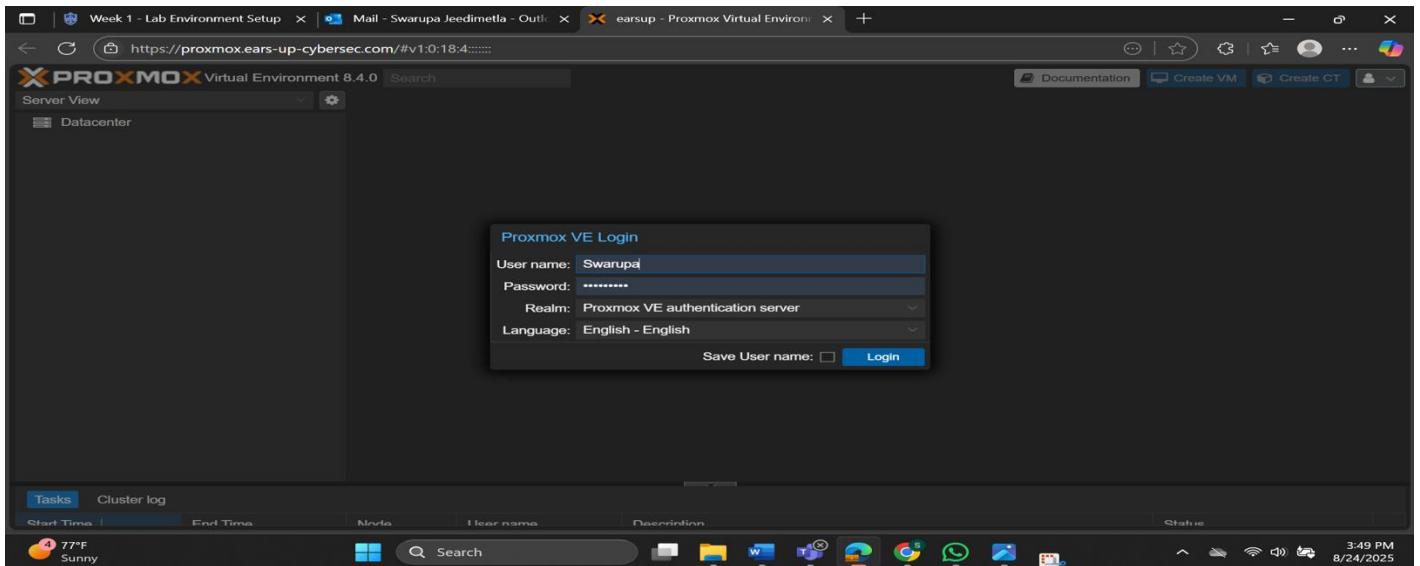
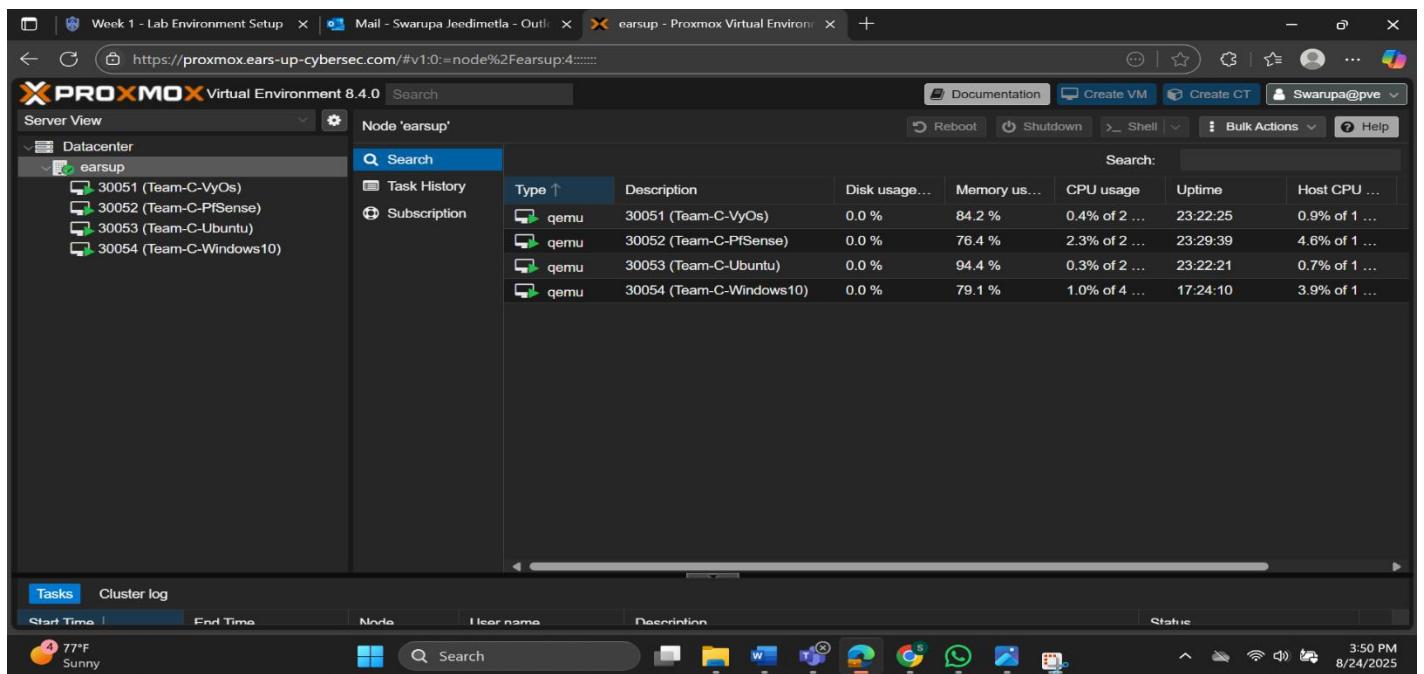


Figure 3

Proxmox Dashboard

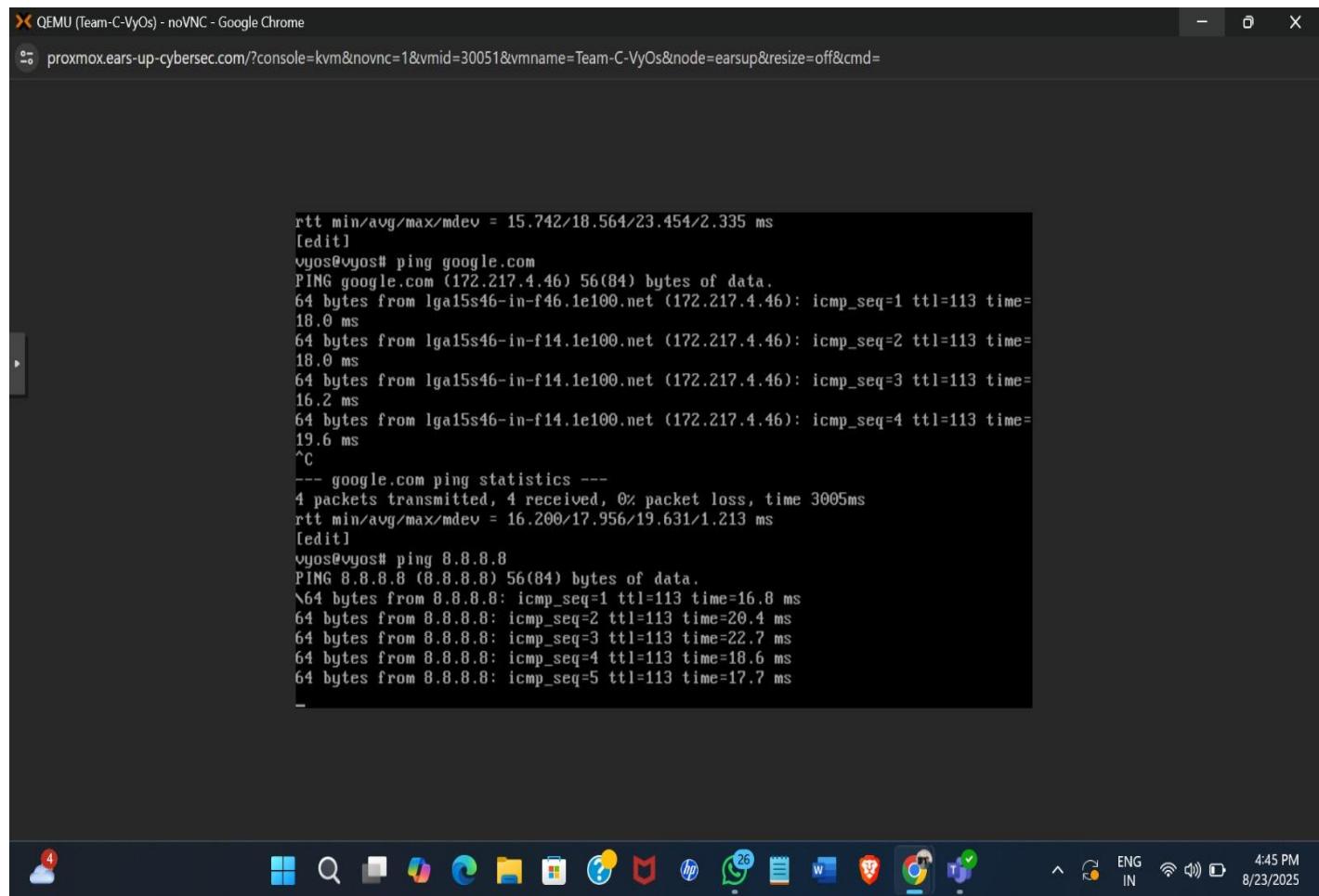


VyOS Router Configuration

VyOS router was configured as the central routing and NAT device with the WAN interface configured as 203.0.115.2/25 and a default route configured with 203.0.115.1 and the LAN interface as 178.18.3.1/24. The 8.8.8.8 and 1.1.1.1 DNS servers were configured to allow name resolution. To offer access to the internet to internal hosts, a Source NAT rule was established to mask all the traffic originating from the WAN interface eth0 and the 178.18.3.0/24 LAN out. Ping 8.8.8.8 tests confirmed that the internet was reachable, and ping google.com tests confirmed that DNS resolution was operating properly (VyOS, 2024).

Figure 4

Connection Verification



The screenshot shows a terminal window titled "QEMU (Team-C-VyOs) - noVNC - Google Chrome". The URL in the address bar is "proxmox.ears-up-cybersec.com/?console=kvm&novnc=1&vmid=30051&vmname=Team-C-VyOs&node=earsup&resize=off&cmd=". The terminal output displays three ping commands:

```
rtt min/avg/max/mdev = 15.742/18.564/23.454/2.335 ms
[edit]
vyos@vyos# ping google.com
PING google.com (172.217.4.46) 56(84) bytes of data.
64 bytes from lga15s46-in-f46.1e100.net (172.217.4.46): icmp_seq=1 ttl=113 time=18.0 ms
64 bytes from lga15s46-in-f14.1e100.net (172.217.4.46): icmp_seq=2 ttl=113 time=18.0 ms
64 bytes from lga15s46-in-f14.1e100.net (172.217.4.46): icmp_seq=3 ttl=113 time=16.2 ms
64 bytes from lga15s46-in-f14.1e100.net (172.217.4.46): icmp_seq=4 ttl=113 time=19.6 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 16.200/17.956/19.631/1.213 ms
[edit]
vyos@vyos# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=16.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=20.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=113 time=22.7 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=113 time=18.6 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=113 time=17.7 ms
```

pfSense Firewall Configuration

To separate internet and internal traffic, we connected vtnet0 to the WAN (vmbr1) and vtnet1 to the LAN (vmbr2). In the console, the LAN IP was configured as 192.168.102.1/24 with the gateway being blank, and IPv6 was ignored. The range of the LAN DHCP server was set to 192.168.102.100 - 192.168.102.200. Using pfSense (WAN 178.18.3.3/24, LAN 192.168.102.1/24), we were able to ping the Windows VM (192.168.102.2) and 8.8.8.8 and could confirm being connected.

Figure 5

pfSense IP address

```
The IPv4 LAN address has been set to 192.168.102.1/24
You can now access the webConfigurator by opening the following URL in your web
browser:
http://192.168.102.1/

Press <ENTER> to continue.
QEMU Guest - Netgate Device ID: 658b7a3f465d42ee92cd

*** Welcome to pfSense 2.8.0-RELEASE (amd64) on pfSense ***

WAN (wan) -> vtnet0 -> v4: 172.18.3.3/24
LAN (lan) -> vtnet1 -> v4: 192.168.102.1/24

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

Enter an option: ■
```

Figure 6

Connection Verification

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

Enter an option: 7

Enter a host name or IP address: 192.168.102.2

PING 192.168.102.2 (192.168.102.2): 56 data bytes
64 bytes from 192.168.102.2: icmp_seq=0 ttl=128 time=0.594 ms
64 bytes from 192.168.102.2: icmp_seq=1 ttl=128 time=0.600 ms
64 bytes from 192.168.102.2: icmp_seq=2 ttl=128 time=0.710 ms

--- 192.168.102.2 ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.594/0.635/0.710/0.053 ms

Press ENTER to continue.
```

Figure 7*Connection Verification*

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

Enter an option: 7

Enter a host name or IP address: 8.8.8.8

PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=112 time=19.317 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=112 time=16.400 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=112 time=16.088 ms

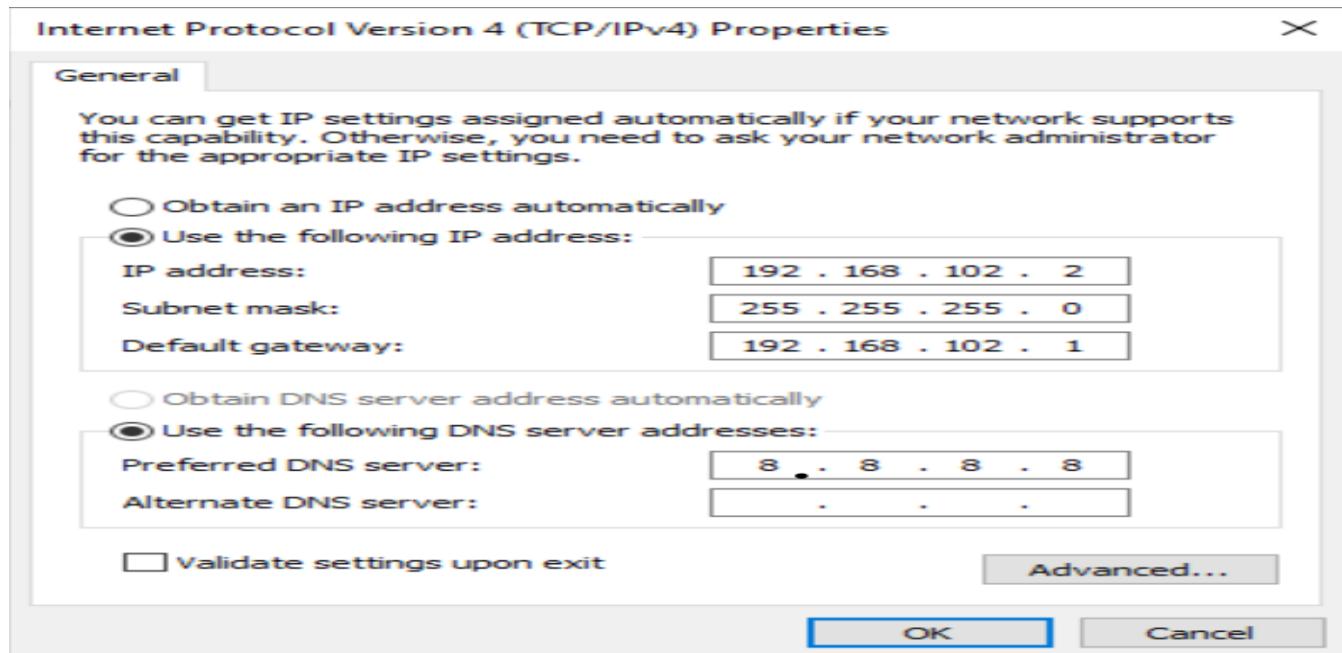
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 16.088/17.269/19.317/1.454 ms

Press ENTER to continue.
```

Windows VM Configuration

The Windows VM was configured with a static IP (192.168.102.2), subnet mask (255.255.255.0), default gateway (192.168.102.1), and DNS of Google public DNS (8.8.8.8) to allow the system to resolve the name. This manual setup also enabled Windows VM to communicate with the pfSense gateway and the rest of the systems on the network. We confirmed that we had internet connectivity by being able to ping the pfSense LAN gateway (192.168.102.1) and 8.8.8.8 to make sure that we can use the internet and DNS resolution by going to Google.com, which confirmed that the Windows VM can communicate with the local environment and the internet.

Figure 8*IPv4 Properties*

**Figure 9***IP Settings*

← Settings — □ ×

Network 3

If you set a data limit, Windows will set the metered connection setting for you to help you stay under your limit.

[Set a data limit to help control data usage on this network](#)

IP settings

IP assignment:	Manual
IPv4 address:	192.168.102.2
IPv4 subnet prefix length:	24
IPv4 gateway:	192.168.102.1
IPv4 DNS servers:	8.8.8.8

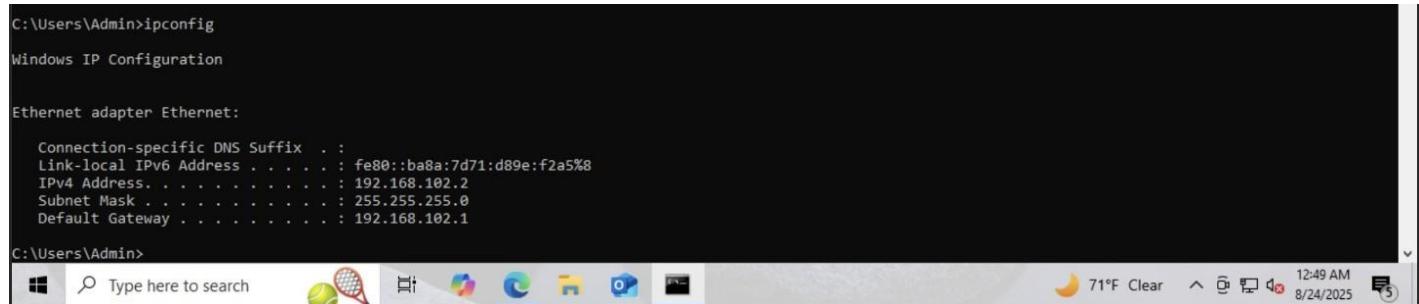
[Edit](#)

**Properties**

Link speed (Receive/Transmit):	10/10 (Gbps)
Link-local IPv6 address:	fe80::ba8a:7d71:d89e:f2a5%8
IPv4 address:	192.168.102.2
IPv4 DNS servers:	8.8.8.8
Manufacturer:	Red Hat, Inc.

Figure 10

Windows IP Address



```
C:\Users\Admin>ipconfig
Windows IP Configuration

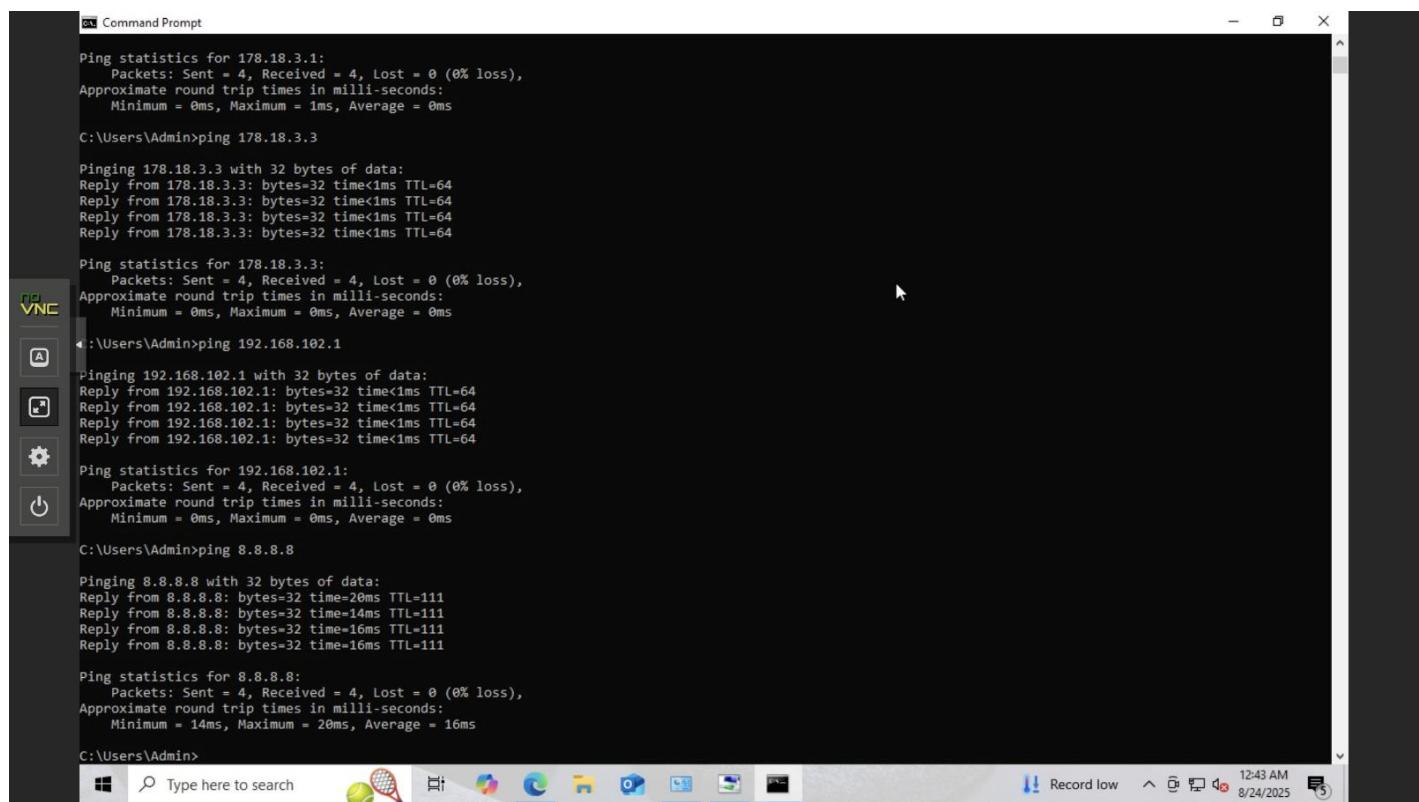
Ethernet adapter Ethernet:

  Connection-specific DNS Suffix  . :
  Link-local IPv6 Address . . . . . : fe80::ba8a:7d71:d89e:f2a5%8
  IPv4 Address. . . . . : 192.168.102.2
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 192.168.102.1

C:\Users\Admin>
```

Figure 11

Connection Verification



```
Command Prompt

Ping statistics for 178.18.3.1:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\Admin>ping 178.18.3.3

Pinging 178.18.3.3 with 32 bytes of data:
Reply from 178.18.3.3: bytes=32 time<1ms TTL=64

Ping statistics for 178.18.3.3:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Admin>ping 192.168.102.1

Pinging 192.168.102.1 with 32 bytes of data:
Reply from 192.168.102.1: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.102.1:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Admin>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=20ms TTL=111
Reply from 8.8.8.8: bytes=32 time=14ms TTL=111
Reply from 8.8.8.8: bytes=32 time=16ms TTL=111
Reply from 8.8.8.8: bytes=32 time=16ms TTL=111

Ping statistics for 8.8.8.8:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 14ms, Maximum = 20ms, Average = 16ms

C:\Users\Admin>
```

Ubuntu VM Configuration

We then cleared the old addresses on the Ubuntu VM with sudo ip addr flush dev ens18 (substituting eth0 with ens18, which is our network device of the interface) to put the interface in a state where it was

prepared to start with a new IP. At that point, we gave the VM 178.18.3.2/24 sudo ip addr add 178.18.3.2/24 dev ens18 and brought the interface up sudo ip link set ens18 up. VyOS router configured as the default gateway by sudo ip route add default via 178.18.3.1, which makes it available to the other networks and the internet. To set DNS with Google Public DNS, we used sudo bash -c echo nameserver 8.8.8.8 /etc/resolv.conf. Ping of the router (178.18.3.1), 8.8.8.8, and Google.com was successfully done, and it confirmed that there was proper access to the local and internet with domain name resolution.

Figure 12

Ubuntu IP address

```
anonymous@anonymous-Standard-PC-i440FX-PIIX-1996:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:90:08:41 brd ff:ff:ff:ff:ff:ff
        altnet enp0s18
        inet 178.18.3.2/24 scope global ens18
            valid_lft forever preferred_lft forever
anonymous@anonymous-Standard-PC-i440FX-PIIX-1996:~$ sudo ip addr flush dev eth0
Device "eth0" does not exist.
anonymous@anonymous-Standard-PC-i440FX-PIIX-1996:~$
```

Figure 13

Connection Verification

```
QEMU (Team-C-Ubuntu) - noVNC - Google Chrome
proxmox.ears-up-cybersec.com/?console=kvm&novnc=1&vmid=30053&vmmname=Team-C-Ubuntu&node=earsup&resize=off
anonymous@anonymous-Standard-PC-i440FX-PIIX-1996:~$ sudo ip link set ens18 up
anonymous@anonymous-Standard-PC-i440FX-PIIX-1996:~$ sudo ip route add default via 178.18.3.1
anonymous@anonymous-Standard-PC-i440FX-PIIX-1996:~$ sudo bash -c 'echo "nameserver 8.8.8.8" > /etc/resolv.conf'
PING 178.18.3.1 (178.18.3.1) 56(84) bytes of data.
64 bytes from 178.18.3.1: icmp_seq=1 ttl=64 time=0.664 ms
64 bytes from 178.18.3.1: icmp_seq=2 ttl=64 time=0.306 ms
64 bytes from 178.18.3.1: icmp_seq=3 ttl=64 time=0.301 ms
--- 178.18.3.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2083ms
rtt min/avg/max/mdev = 0.301/0.423/0.664/0.169 ms
anonymous@anonymous-Standard-PC-i440FX-PIIX-1996:~$ ping -c 3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=112 time=19.6 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=112 time=20.2 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=112 time=18.8 ms
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 18.781/19.538/20.219/0.589 ms
anonymous@anonymous-Standard-PC-i440FX-PIIX-1996:~$ ping -c 3 google.com
PING google.com (172.217.4.46) 56(84) bytes of data.
64 bytes from ord38s18-in-f14.1e100.net (172.217.4.46): icmp_seq=1 ttl=112 time=18.5 ms
64 bytes from lga15s46-in-f46.1e100.net (172.217.4.46): icmp_seq=2 ttl=112 time=16.9 ms
64 bytes from lga15s46-in-f46.1e100.net (172.217.4.46): icmp_seq=3 ttl=112 time=17.8 ms
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 16.898/17.736/18.465/0.644 ms
anonymous@anonymous-Standard-PC-i440FX-PIIX-1996:~$
```

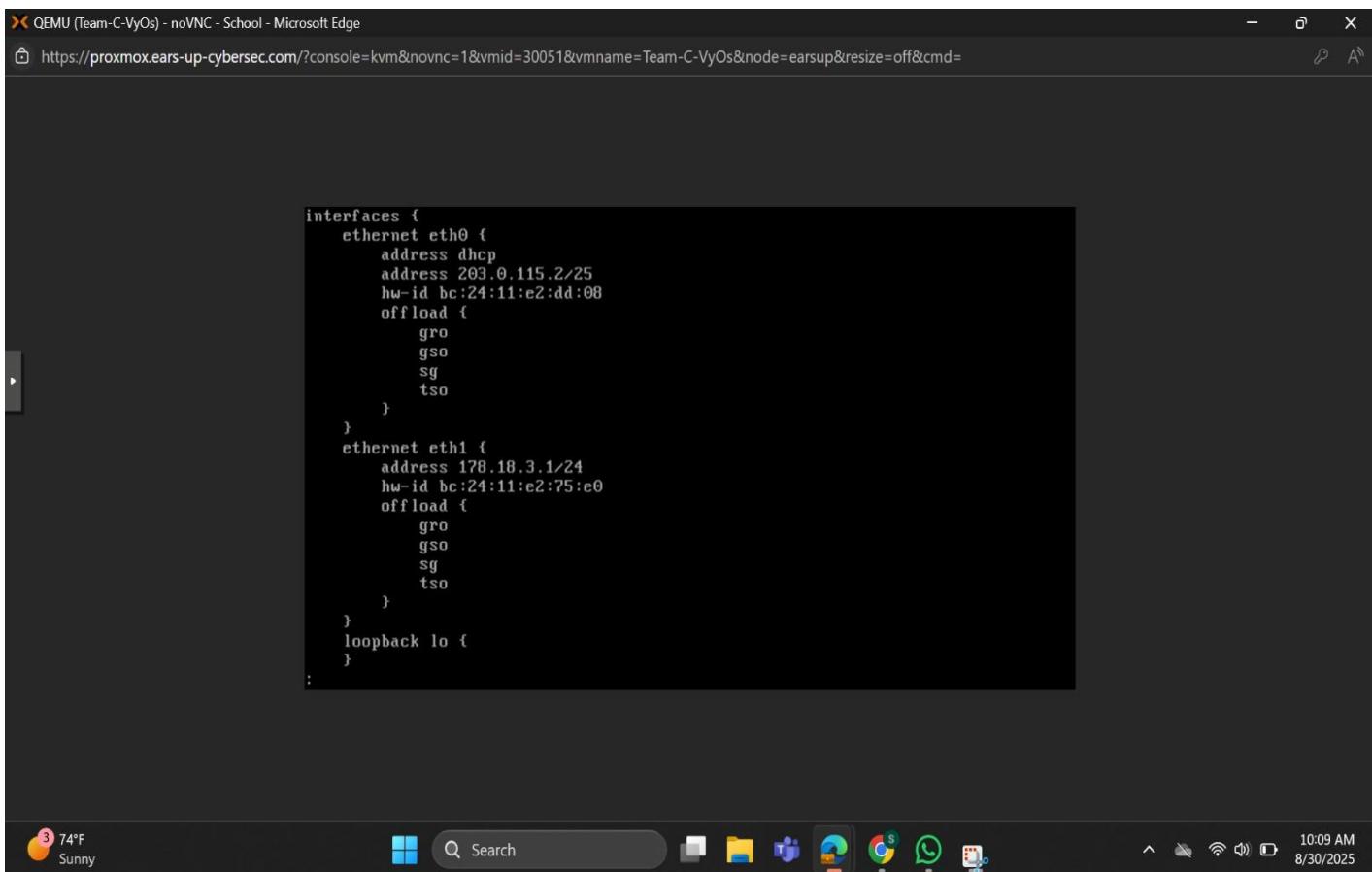
Firewall & Router Setup

VyOS and pfSense Connectivity Check

We configured VyOS router by setting the WAN interface (eth0) with the IP 203.0.115.2/25 and default gateway 203.0.115.1 and the LAN interface (eth1) with the IP 178.18.3.1/24. To configure a NAT and add DNS servers (8.8.8.8 and 1.1.1.1) we added a masquerade rule for the subnet 178.18.3.0/24 and redirected traffic flowing over eth0. Once the configuration has been saved and the configuration was verified, we ensured that we could ping the destination successfully to verify that we had internet connectivity and that we could resolve DNS addresses (Samuel, 2018).

Figure 14

Configuration of VyOS Router



```
interfaces {
    ethernet eth0 {
        address dhcp
        address 203.0.115.2/25
        hw-id bc:24:11:e2:dd:08
        offload {
            gro
            gso
            sg
            tso
        }
    }
    ethernet eth1 {
        address 178.18.3.1/24
        hw-id bc:24:11:e2:75:e0
        offload {
            gro
            gso
            sg
            tso
        }
    }
    loopback lo {
    }
};
```

Figure 15

Interfaces of VyOS Router

```

eth1      178.18.3.1/24      bc:24:11:e2:75:e0  default  1500  u/u
lo       127.0.0.1/8        00:00:00:00:00:00  default  65536  u/u
          ::1/128

vyos@vyos:~$ sudo ip addr del 192.168.102.101/24 dev eth0
vyos@vyos:~$ ip addr show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
p default qlen 1000
    link/ether bc:24:11:e2:dd:08 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    altname ens18
    inet 203.0.115.2/25 brd 203.0.115.127 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::be24:11ff:fe2:dd08/64 scope link
        valid_lft forever preferred_lft forever
vyos@vyos:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface  IP Address      MAC           URF      MTU   S/L   Description
on

-----
eth0      203.0.115.2/25  bc:24:11:e2:dd:08  default  1500  u/u
eth1      178.18.3.1/24   bc:24:11:e2:75:e0  default  1500  u/u
lo       127.0.0.1/8     00:00:00:00:00:00  default  65536  u/u
          ::1/128

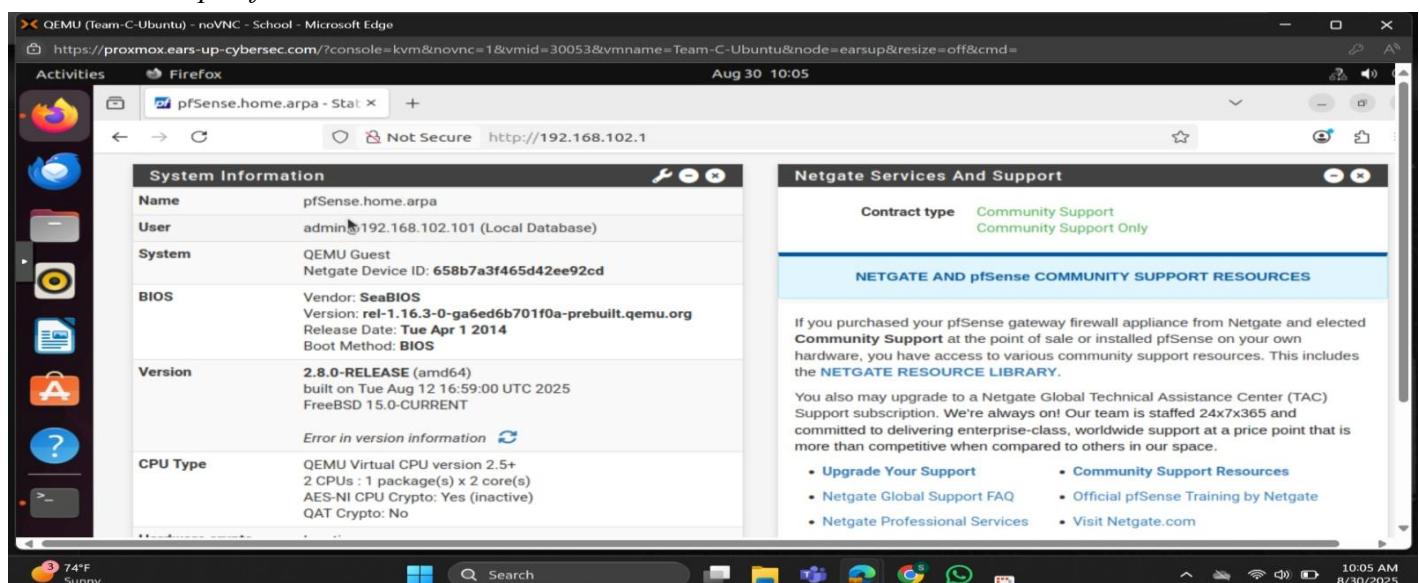
vyos@vyos:~$ 

```

We adjusted the pfSense VM by configuring the WAN (pfSense) into vtnet0 (vmbr1) and configuring the LAN (pfSense) into vtnet1 (vmbr2) and setting LAN IP to 192.168.102.1/24 and DHCP server on to 192.168.102.100-192.168.102.200. Once the WAN was set to 178.18.3.3/24, we verified connectivity by being able to ping the windows VM (192.168.102.2) and the 8.8.8.8. Then, we were able to connect to the pfSense Web GUI using the Ubuntu VM, which ensured that the network was correctly integrated, and it was possible to manage it.

Figure 16

pfSense Web GUI open from the Ubuntu VM



Connectivity Testing Check List

Table 1

Connectivity Checklist

Source to Destination	Expected Result	Status (UP/Down)
VyOS to Internet	Ping 8.8.8.8 succeeds	Up
VyOS to Ubuntu	Ping 178.18.3.2 succeeds	Up
VyOS to pfSense	Ping 178.18.3.3 succeeds	Up
Ubuntu to Internet	Ping 8.8.8.8 Succeeds	Up
Ubuntu to VyOS	Ping 178.18.3.1 succeeds	Up
Ubuntu to pfSense	Ping 178.18.3.3 succeeds	Up
pfSense to Internet	Ping 8.8.8.8 succeeds	Up
PfSense to VyOS	Ping 178.18.3.1 succeeds	Up
pfSense to Ubuntu	Ping 178.18.3.2 succeeds	Up

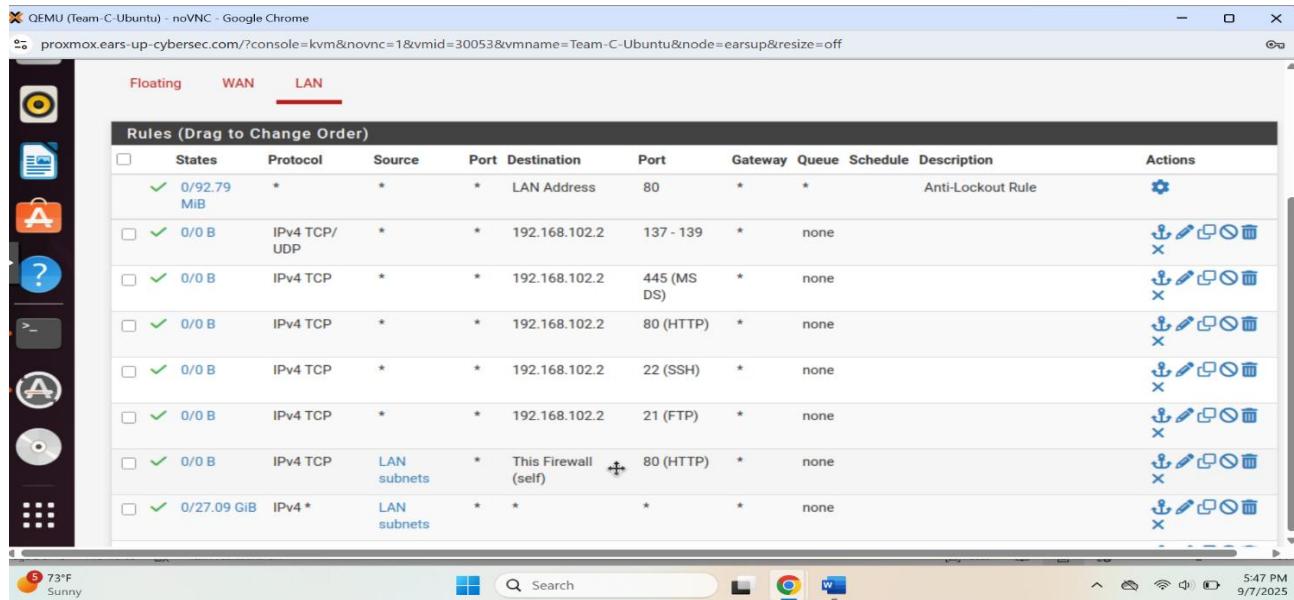
Firewall Security & Controlled Exposure with pfSense & Windows 10

Creating Non-Admin User and Enabling Services

Windows 10 VM has been configured with a static IP 192.168.102.2, and the connection has been verified using ipconfig and ping 8.8.8.8. The default access was made by one of the non-admin users named Jimmy. Major services were enabled and tested, FTP through IIS Server (port 21), SSH through OpenSSH Server, SMB with shared folder, NetBIOS through TCP/IP, and an HTTP server through Python on ports 80 and all the ports that were evaluated to be correctly functioning.

pfSense Firewall Rules

The Windows 10 VM was permitted to access the LAN firewall rules and WAN NAT on pfSense (192.168.102.2) through FTP, SSH, HTTP, SMB, and NetBIOS. Ubuntu tests were performed.

Figure 17*pfSense Rules***Connectivity and Ports Testing Checklist**

Connection tests showed that pfSense could connect with Windows (192.168.102.2), windows could connect with pfSense via the LAN/WAN and the internet and that all services were active including FTP, SSH, HTTP, SMB, and NetBIOS.

Table 2*Connectivity Checklist*

Source to Destination	Expected Result	Status (UP/Down)
PfSense to Windows	Ping 192.168.102.2 succeeds	Up
Windows to PfSense (WAN)	Ping 178.18.3.3 succeeds	Up
Windows to PfSense (LAN)	Ping 192.168.102.1 succeeds	Up
Windows to Internet	Ping 8.8.8.8 succeeds	Up

Table 3*Ports Testing*

Ports	Status (UP/Down)
FTP (21)	Up
SSH (22)	Up
HTTP (80)	Up
SMB (445)	Up
NetBIOS (137,138,139)	Up

Wazuh & Sysmon Configuration*Ubuntu-Wazuh Manager Deployment*

Ubuntu VM was then updated and the Wazuh Manager installed, and its service monitored as active by systemctl status wazuh-manager. Wazuh web dashboard was opened, default credentials were used to successfully log in, and everything was deployed, the dashboard was available.

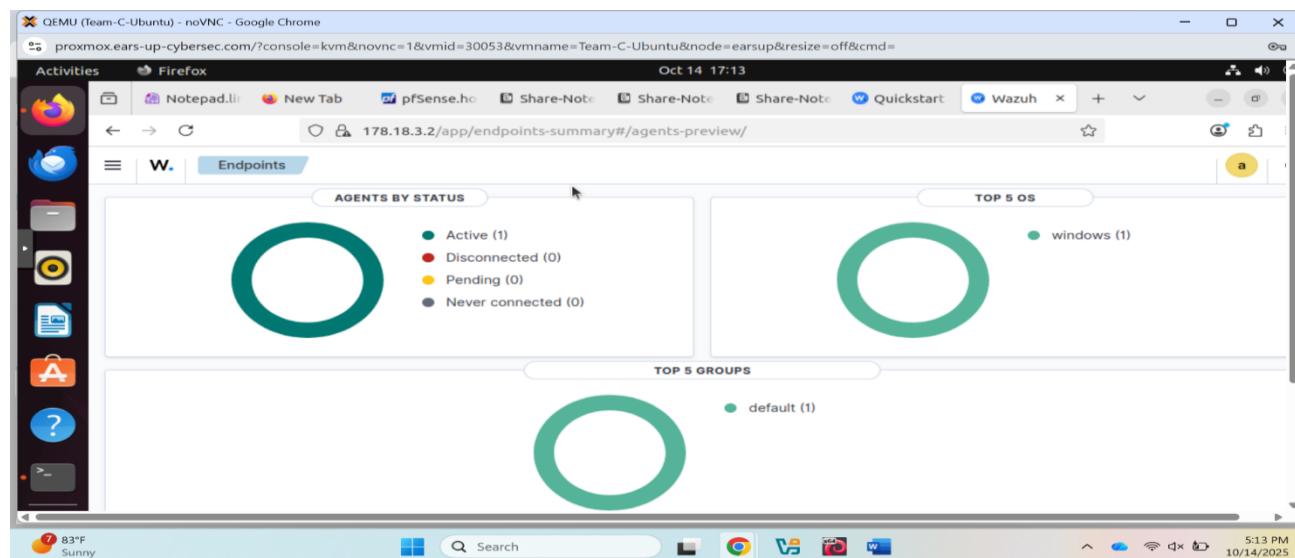
Figure 18*Wazuh Dashboard in Ubuntu*

Figure 19*Wazuh Dashboard in Ubuntu*

The screenshot shows the Wazuh dashboard interface. At the top, there's a header bar with the URL `proxmox.ears-up-cybersec.com/?console=kvm&novnc=1&vmid=30053&vmname=Team-C-Ubuntu&node=earsup&resize=off&cmd=`. Below the header, the main content area has a title "Endpoints" and a specific host entry "DESKTOP-23M8HBN". Under this, there are sections for "Threat Hunting", "File Integrity Monitoring", and "More...". A detailed table provides information about the agent:

ID	Status	IP address	Version	Group	Operating system	Cluster node	Registration date
001	active	192.168.102.2	Wazuh v4.13.0	default	Microsoft Windows 10 Home 10.0.19045.6216	node01	Oct 14, 2025 @ 12:05:59.000

Below the table, it says "Last keep alive" was at "Oct 14, 2025 @ 17:27:23.000". A "System inventory" section follows, showing:

Cores	Memory	CPU	Host name	Serial number
4	4GB	QEMU Virtual CPU version 2.5+	DESKTOP-23M...	unknown

The bottom of the dashboard includes a search bar, a toolbar with various icons, and a status bar showing the date and time.

pfSense Firewall Configuration

The Web GUI was used to set the LAN firewall rules in pfSense that it was permitted to pass Wazuh logs to the Ubuntu VM that it accepts TCP/UDP traffic on port 1514 and 1515 to the Wazuh Manager. The rules were saved, accessed and verified as live in the firewall logs.

Figure 20*pfSense Rules*

The screenshot shows the pfSense Firewall Rules configuration. The table lists the following rules:

Protocol	Source	Destination	Service	Action	Notes	
IPv4 TCP	*	192.168.102.2	22 (SSH)	*	none	
IPv4 TCP	*	192.168.102.2	21 (FTP)	*	none	
IPv4 TCP	LAN subnets	*	This Firewall (self)	80 (HTTP)	*	none
IPv4 *	LAN subnets	*	*	*	none	Default allow LAN to any rule
IPv4 *	LAN subnets	*	*	*	none	Default allow LAN IPv6 to any rule
IPv4 *	LAN subnets	*	*	*	none	Wazuh Log Forwarding
IPv4 TCP/ UDP	*	178.18.3.2	1514 - 1515	*	none	

The bottom of the screen shows the pfSense footer: "pfSense is developed and maintained by Netgate. © ESF 2004 - 2025 View license." and the system status bar: "MIA - TEX Video highlight", "11:42 PM 9/21/2025".

Windows 10 Wazuh Agent & Sysmon setup

Wazuh Agent was installed and configured to communicate with the Wazuh Manager on the Windows 10 Ubuntu, and the service was observed to be running. Sysmon was deployed using a custom XML configuration and the ossec.conf was modified to scan the System and Security logs. When reconnecting agent to manager once again the connection was verified in the Wazuh Dashboard and Sysmon events were generated and recorded in real time to be checked.

We configured Sysmon and used a custom XML configuration file to allow detailed logging of relevant security events, such as process creation, network connections, registry changes, and file activity, and tuned include/exclude rules in the XML and then ensured that the configured events were being logged properly in the Sysmon Operational log and being sent to our monitoring platform (Rodriguez, 2017).

Figure 21

Sysmon

```
PS C:\Windows\system32> Get-Service sysmon64
Status      Name            DisplayName
----      ----            -----------
Running    Sysmon64        sysmon64

PS C:\Windows\system32>
```

Figure 22

Event Viewer

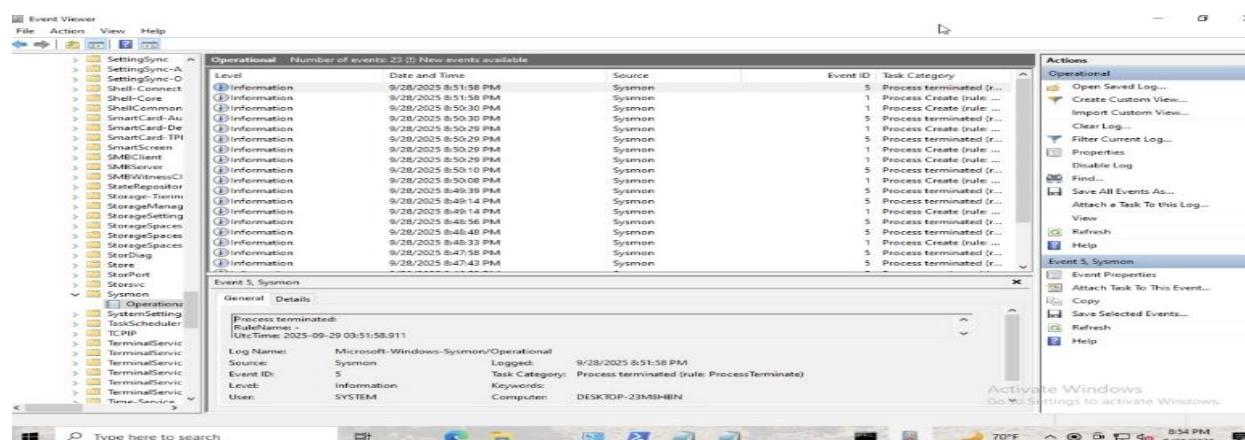


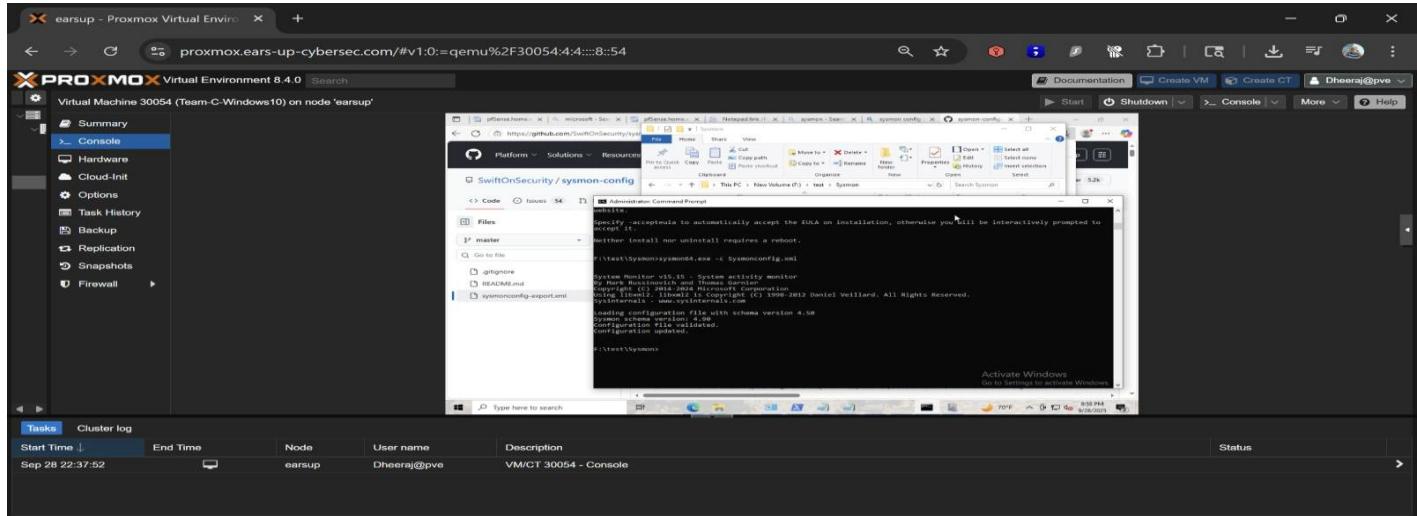
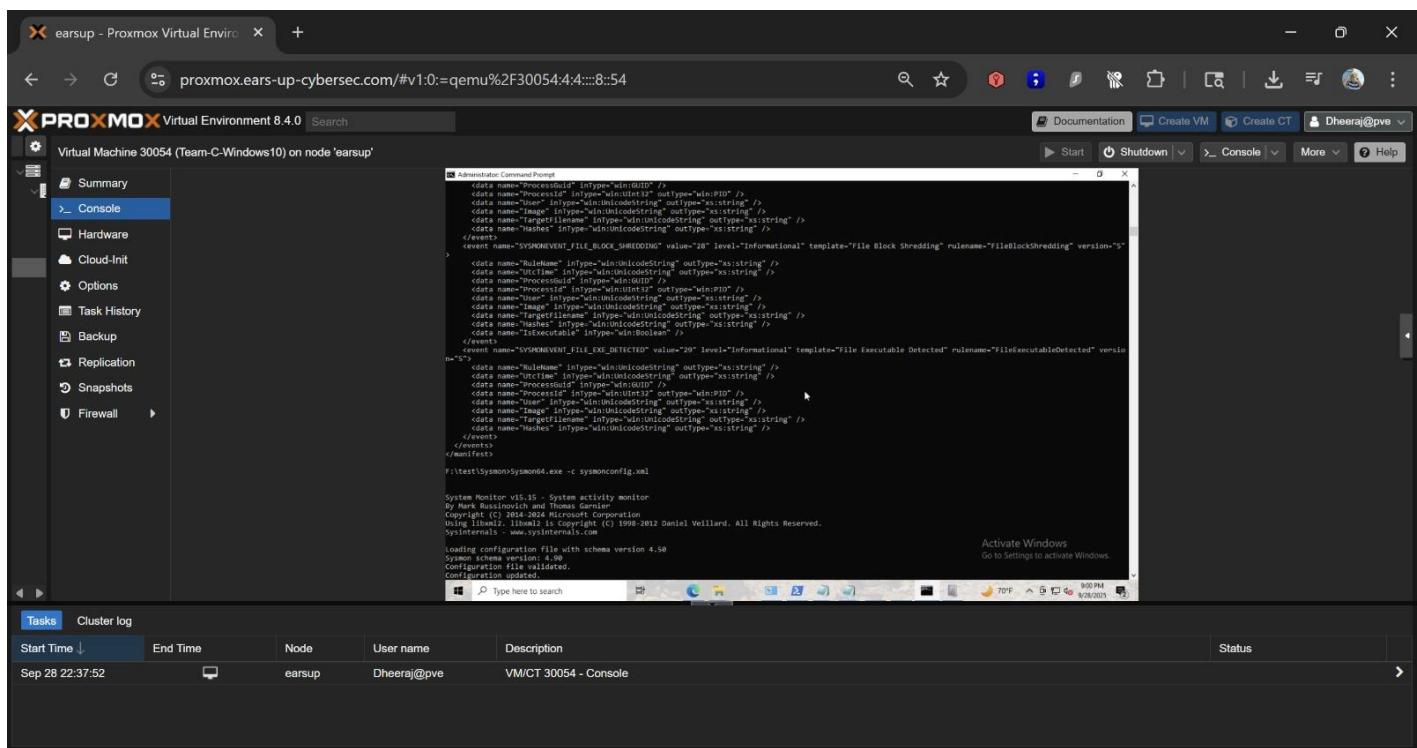
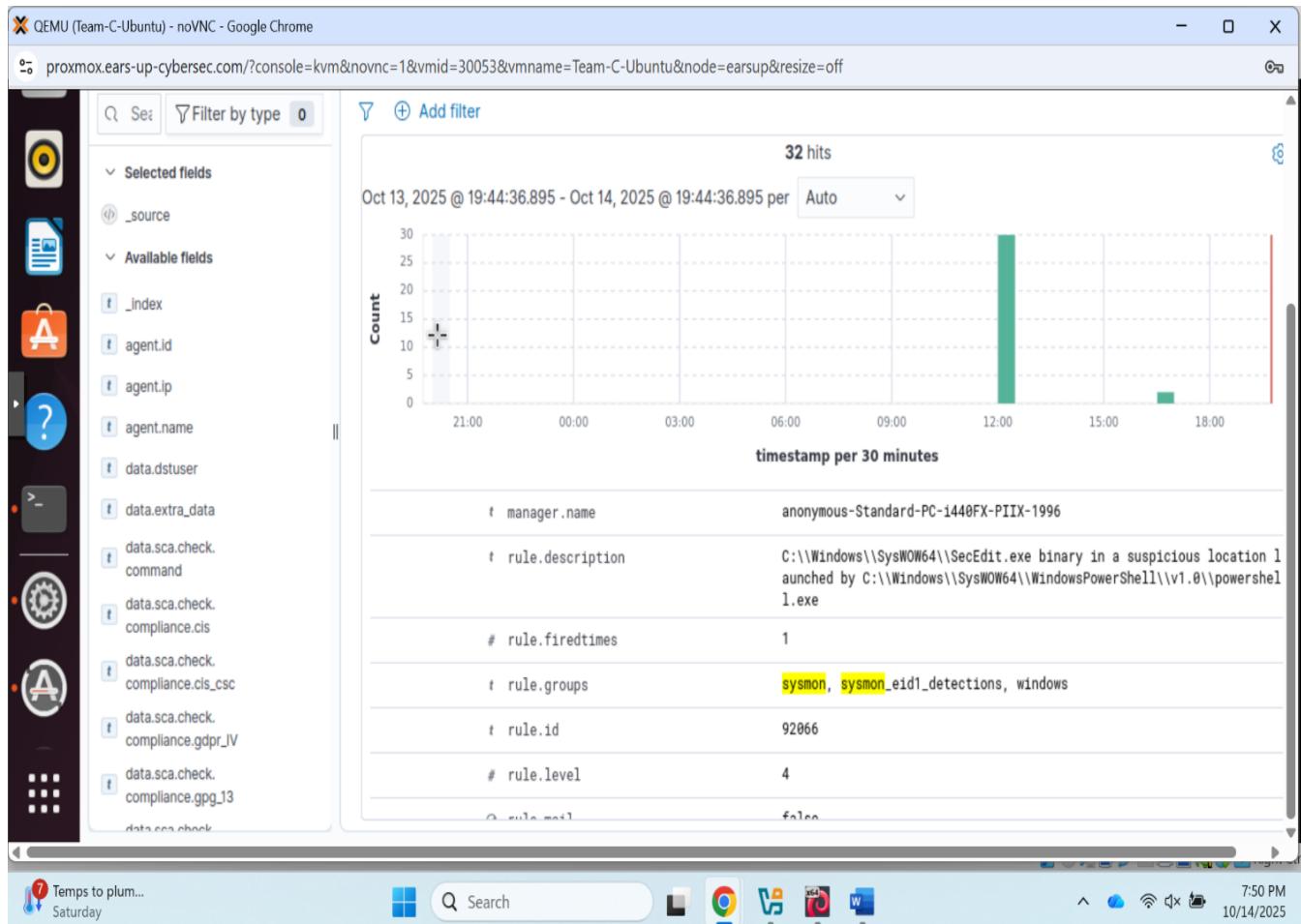
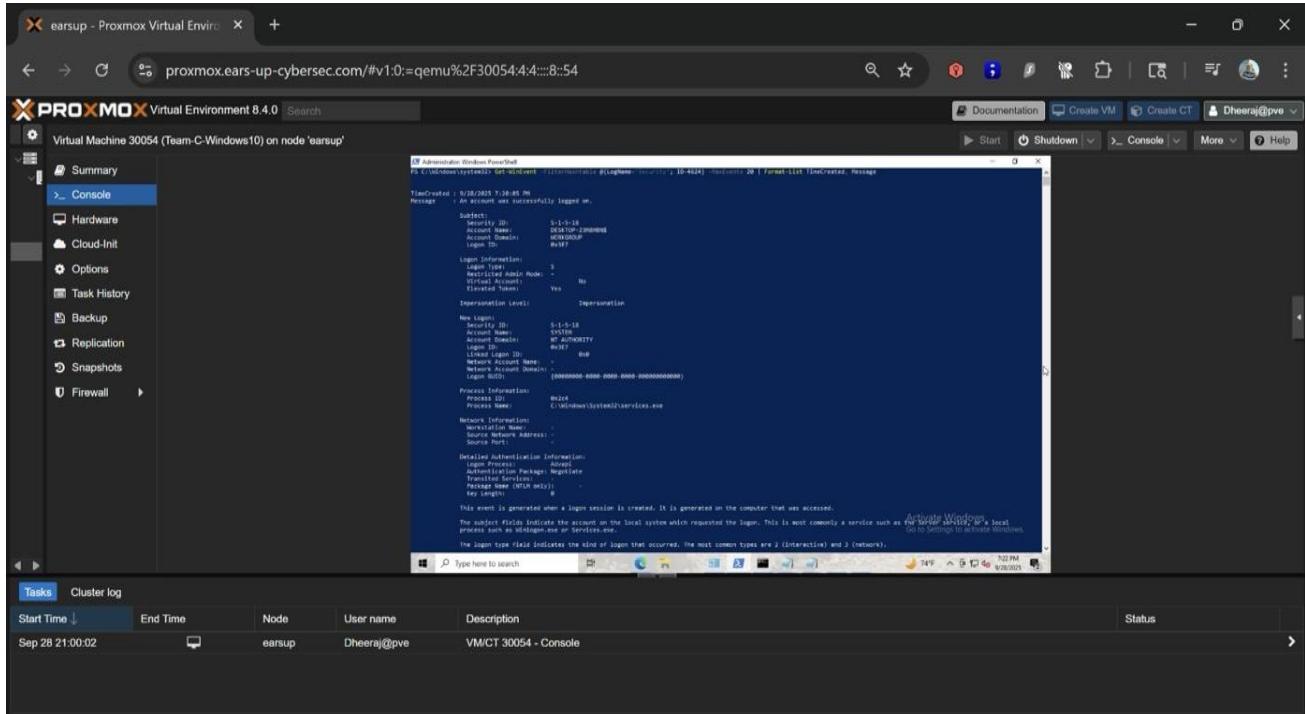
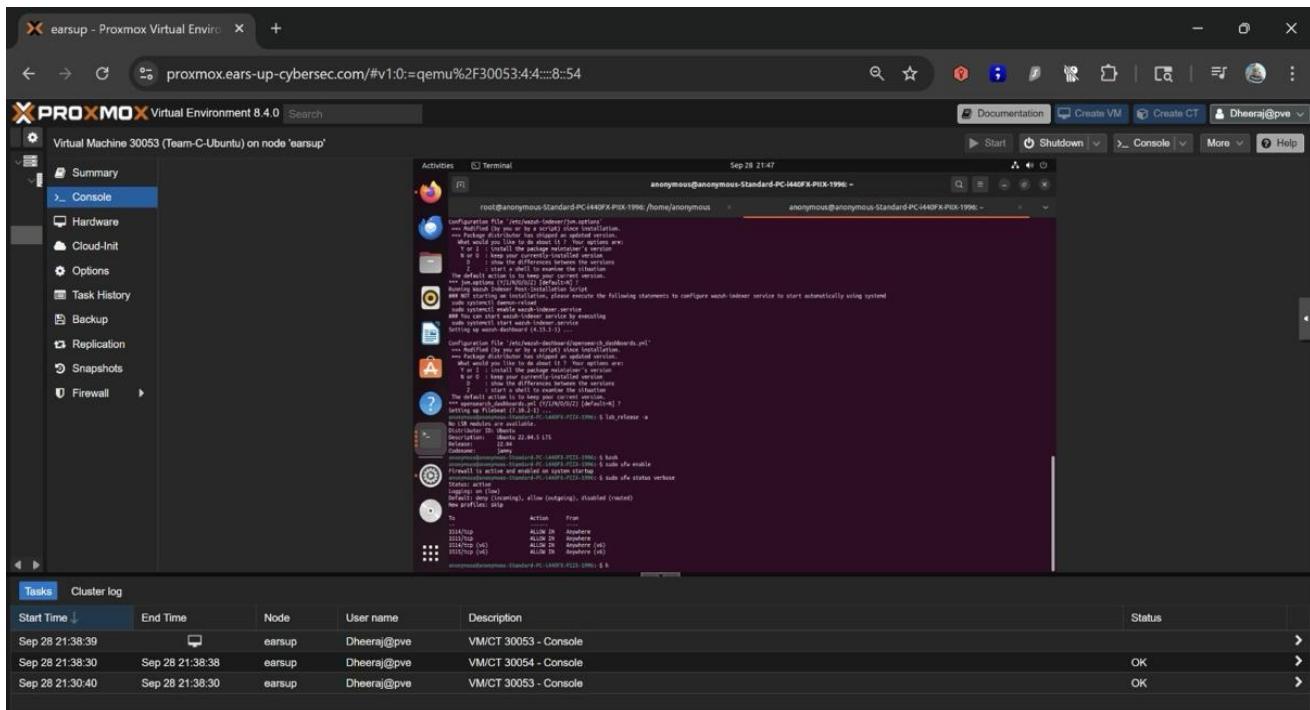
Figure 23*XML file***Figure 24***XML File*

Figure 25*Wazuh Alerts***Windows 10 Home and Ubuntu Hardening**

Windows 10 Home was safeguarded by having windows defender on, disabling optional features like Remote Registry, enabling Controlled Folder Access, and reviewing of the Event Viewer logs to ensure that auditing is taking place. Ubuntu was hardened through update, enabling UFW with default-deny rule, permitting SSH to a subnet only within the internal subnet, turned off unnecessary services, giving correct user permissions, and checking the system logs of the system. To ensure that both systems had the appropriate level of security, Wazuh compliance was verified to enable CIS Benchmark modules, hardening policies, and generate reports.

Figure 26*Windows Hardening***Figure 27***Ubuntu Hardening*

Threat Intelligence & MITRE ATT&CK Mapping

We also made the mapping of MITRE ATT+CK in Wazuh possible and verified it using the dashboard. On DESKTOP-23M8HBN, technique A (T1105: Ingress Tool Transfer, rule ID 92213) was observed on October 29 as powershell.exe was used to run a script in a Temp file, which implies tool transfer to Command-and-Control. On the same day, Technique B (T1078: Valid Accounts, rule ID 5501) has been observed on the anonymous-Standard-PC-i440FX-PIIX-1996, and a valid account (root) was used in a PAM session, which could indicate the possibility of privilege escalation (Jiang et al., 2025).

Figure 28

MITRE ATT&CK Dashboard

Figure 29

T1105 Details

The screenshot shows a Firefox browser window with the address bar at `127.0.0.1/app/mitre-attack#/overview?tab=mitre&tabView=events&tabRedirect=techniques&_a=(q)`. The title bar says "MITRE ATT&CK". The main content area displays a table of log entries under "Document Details".

Table	JSON
<code>_index</code>	wazuh-alerts-4.x-2025.10.29
<code>agent.id</code>	001
<code>agent.ip</code>	192.168.102.2
<code>agent.name</code>	DESKTOP-23M8HBN
<code>data.win.eventdata.creationUtcTime</code>	2025-10-29 19:34:27.481
<code>data.win.eventdata.image</code>	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
<code>data.win.eventdata.processGUID</code>	{863ed747-6c3c-6982-4705-000000000000}
<code>data.win.eventdata.processId</code>	13808
<code>data.win.eventdata.targetFileName</code>	C:\Users\Admin\AppData\Local\Temp__PSScriptPolicyTest_0jh1l3za.151.ps1
<code>data.win.eventdata.user</code>	DESKTOP-23M8HBN\Admin
<code>data.win.eventdata.utcTime</code>	2025-10-29 19:34:27.481
<code>data.win.system.channel</code>	Microsoft-Windows-Sysmon/Operational
<code>data.win.system.computer</code>	DESKTOP-23M8HBN
<code>data.win.system.eventID</code>	11

Figure 30*T1105 Details*

The screenshot shows a Firefox browser window with the URL [http://127.0.0.1/app/mitre-attack#/overview?tab=mitre&tabView=events&tabRedirect=techniques&_a=\(q](http://127.0.0.1/app/mitre-attack#/overview?tab=mitre&tabView=events&tabRedirect=techniques&_a=(q). The title bar says "Activities Firefox". The main content area has a header "Document Details" with tabs "View surrounding documents" and "View single document". The table lists log entries with columns for timestamp, agent.name, rule ID, and raw data. A detailed view of a log entry for T1105 is shown on the right.

	data.win.system.processID	10964
#	data.win.system.providerGuid	{5770385f-c22a-43e0-bf4c-06f5698ffbd9}
#	data.win.system.providerName	Microsoft-Windows-Sysmon
#	data.win.system.severityValue	INFORMATION
#	data.win.system.systemTime	2025-10-29T19:34:27.494996Z
#	data.win.system.task	11
#	data.win.system.threadID	10352
#	data.win.system.version	
#	decoder.name	windows_eventchannel
#	id	1761766468.3298737
#	input.type	log
#	location	EventChannel
#	manager.name	anonymous-Standard-PC-1440FX-PIIX-1996
#	rule.description	Executable file dropped in folder commonly used by malware
#	rule.firetimes	1
#	rule.groups	sysmon, sysmon_eid11_detections, windows
#	rule.id	92213

Figure 31*T1105 Details*

The screenshot shows a Firefox browser window with the URL [http://127.0.0.1/app/mitre-attack#/overview?tab=mitre&tabView=events&tabRedirect=techniques&_a=\(q](http://127.0.0.1/app/mitre-attack#/overview?tab=mitre&tabView=events&tabRedirect=techniques&_a=(q). The title bar says "Activities Firefox". The main content area has a header "Document Details" with tabs "View surrounding documents" and "View single document". The table lists log entries with columns for timestamp, agent.name, rule ID, and raw data. A detailed view of a log entry for T1105 is shown on the right.

	data.win.system.threadID	10352
#	data.win.system.version	2
#	decoder.name	windows_eventchannel
#	id	1761766468.3298737
#	input.type	log
#	location	EventChannel
#	manager.name	anonymous-Standard-PC-1440FX-PIIX-1996
#	rule.description	Executable file dropped in folder commonly used by malware
#	rule.firetimes	1
#	rule.groups	sysmon, sysmon_eid11_detections, windows
#	rule.id	92213
#	rule.level	15
#	rule.mail	true
#	rule.mitre.id	T1105
#	rule.mitre.tactic	Command and Control
#	rule.mitre.technique	Ingress Tool Transfer
#	timestamp	Oct 29, 2025 @ 14:34:28.816

Figure 32

T1105 Details

The screenshot shows a Firefox browser window with the URL <https://127.0.0.1/app/mitre-attack#/overview?tab=mitre&tabView=intelligence&tabRedirect=techniques&id=T1105>. The page displays the 'Details' section for T1105, which is named 'Ingress Tool Transfer'. It includes the creation date (May 31, 2017), modified date (Apr 14, 2023), and version (2.2). The 'Description' section states: 'Adversaries may transfer tools or other files from an external system into a compromised environment. Tools or files may be copied from an external adversary-controlled system to the victim network through the command and control channel or through alternate protocols such as `ftp`. Once present, adversaries may also transfer/spread tools between victim devices within a compromised environment (i.e. `Lateral Tool Transfer`).'. Below this, it notes that files can be transferred via various web services and native tools. A 'Groups' section lists 'menuPass' as a threat group active since at least 2006.

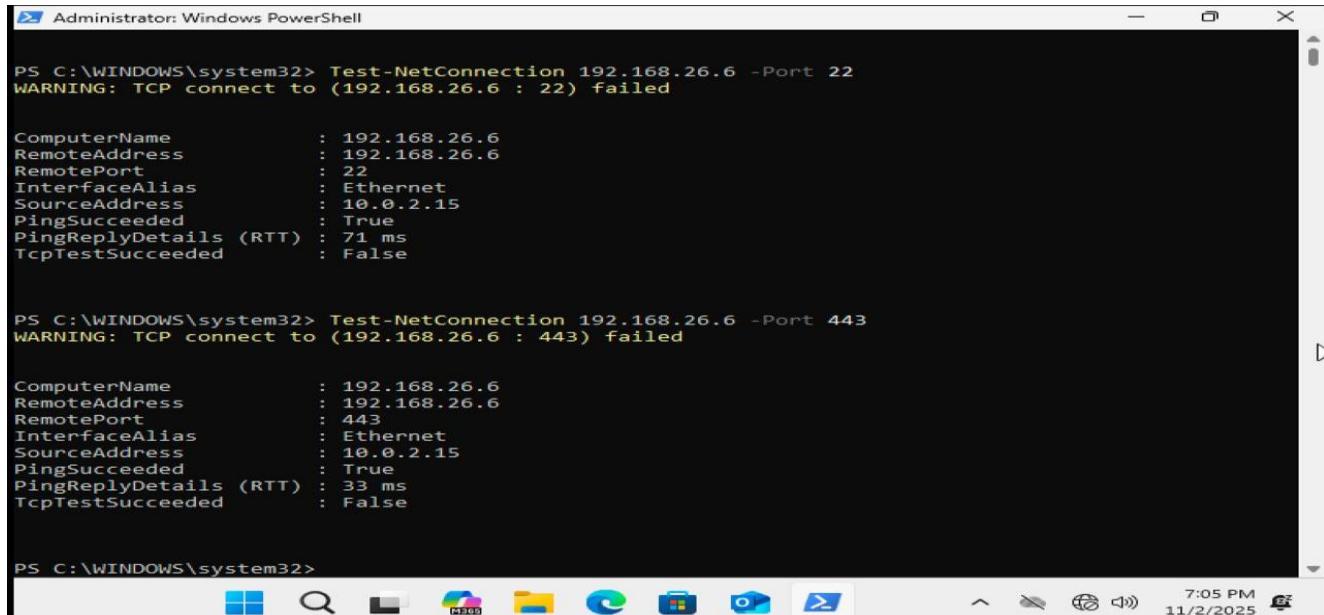
VPN Configuration and Connectivity

We managed to install WireGuard by downloading the latest installer and launching it in administrator mode and importing the tunnel configuration file that was provided to us. The VPN tunnel was enabled and displayed as an active one in the WireGuard application. PowerShell connectivity tests verified access to the network, pinging 192.168.26.6 was successful and Test-NetConnection to port 80 was successful indicating that a web server is available (Hoxha, 2020).

Figure 33

WireGuard

The screenshot shows the WireGuard application window. On the left, there's a sidebar with tabs for 'Tunnels' (selected) and 'Log'. Under 'Tunnels', a single tunnel named 'TeamCVPN' is listed. The main pane shows two sections: 'Interface: TeamCVPN' and 'Peer'. The 'Interface' section displays the following details: Status: Active, Public key: KbRLwaoY9/+mRthUGGV1ZMPOMeGI24HBgbfwx12Hngs=, Listen port: 61463, Addresses: 10.10.5.4/32, DNS servers: 10.10.5.1, and a 'Deactivate' button. The 'Peer' section shows a peer entry with the following details: Public key: IEIUJeM6IWsw8BZWkRAet6Q5q8iGNVd/nplzYUZx0w=, Allowed IPs: 0.0.0.0/0, 192.168.0.32/32, Endpoint: 47.24.144.65:51824, Persistent keepalive: 25, Latest handshake: 3 seconds ago, and Transfer: 331.50 KiB received, 301.07 KiB sent. At the bottom, there are buttons for 'Add Tunnel' and 'Edit'.

Figure 34*Port 22 Connection Testing*

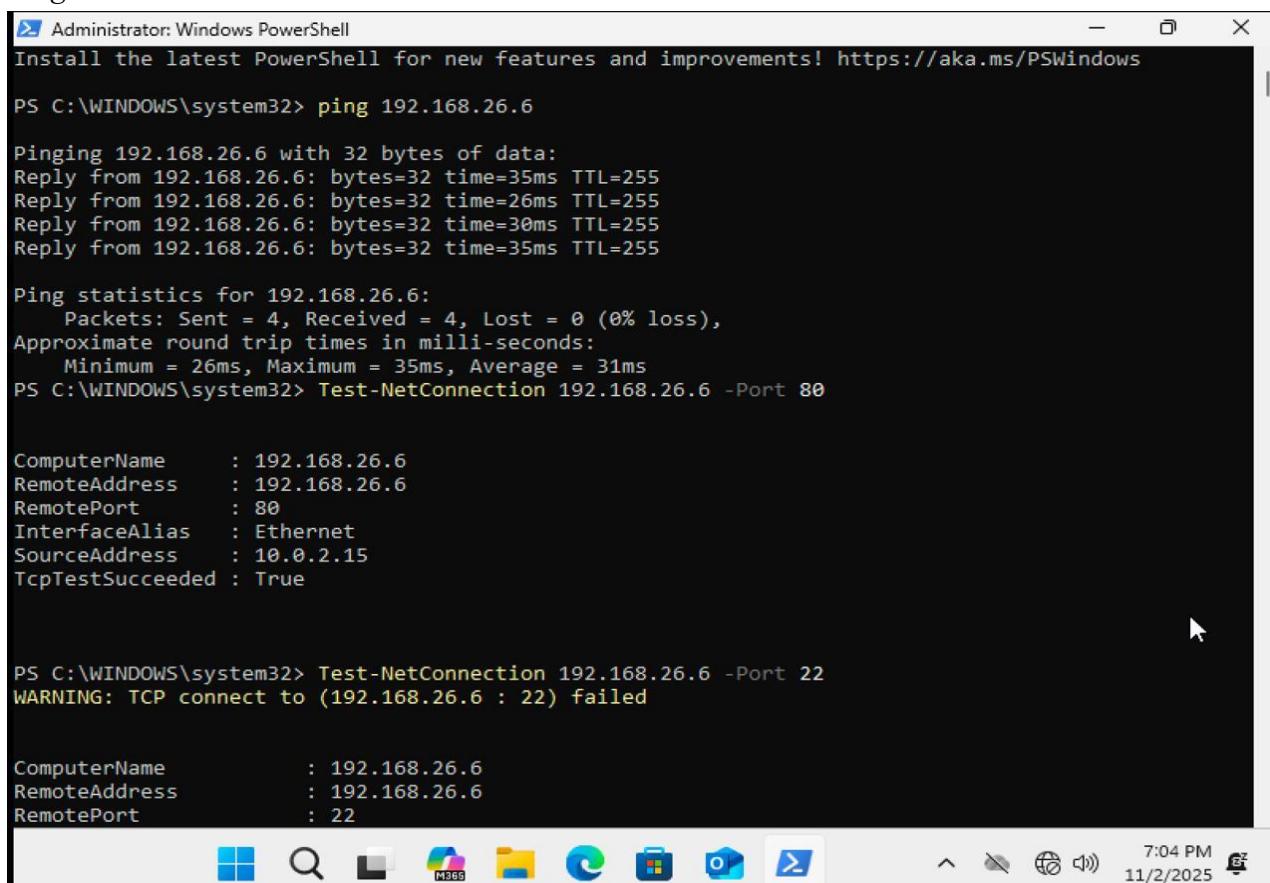
```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> Test-NetConnection 192.168.26.6 -Port 22
WARNING: TCP connect to (192.168.26.6 : 22) failed

ComputerName      : 192.168.26.6
RemoteAddress     : 192.168.26.6
RemotePort        : 22
InterfaceAlias    : Ethernet
SourceAddress     : 10.0.2.15
PingSucceeded     : True
PingReplyDetails (RTT) : 71 ms
TcpTestSucceeded   : False

PS C:\WINDOWS\system32> Test-NetConnection 192.168.26.6 -Port 443
WARNING: TCP connect to (192.168.26.6 : 443) failed

ComputerName      : 192.168.26.6
RemoteAddress     : 192.168.26.6
RemotePort        : 443
InterfaceAlias    : Ethernet
SourceAddress     : 10.0.2.15
PingSucceeded     : True
PingReplyDetails (RTT) : 33 ms
TcpTestSucceeded   : False

PS C:\WINDOWS\system32>
```

Figure 35*Ping*

```
Administrator: Windows PowerShell
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> ping 192.168.26.6

Pinging 192.168.26.6 with 32 bytes of data:
Reply from 192.168.26.6: bytes=32 time=35ms TTL=255
Reply from 192.168.26.6: bytes=32 time=26ms TTL=255
Reply from 192.168.26.6: bytes=32 time=30ms TTL=255
Reply from 192.168.26.6: bytes=32 time=35ms TTL=255

Ping statistics for 192.168.26.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 26ms, Maximum = 35ms, Average = 31ms
PS C:\WINDOWS\system32> Test-NetConnection 192.168.26.6 -Port 80

ComputerName      : 192.168.26.6
RemoteAddress     : 192.168.26.6
RemotePort        : 80
InterfaceAlias    : Ethernet
SourceAddress     : 10.0.2.15
TcpTestSucceeded   : True

PS C:\WINDOWS\system32> Test-NetConnection 192.168.26.6 -Port 22
WARNING: TCP connect to (192.168.26.6 : 22) failed

ComputerName      : 192.168.26.6
RemoteAddress     : 192.168.26.6
RemotePort        : 22
```

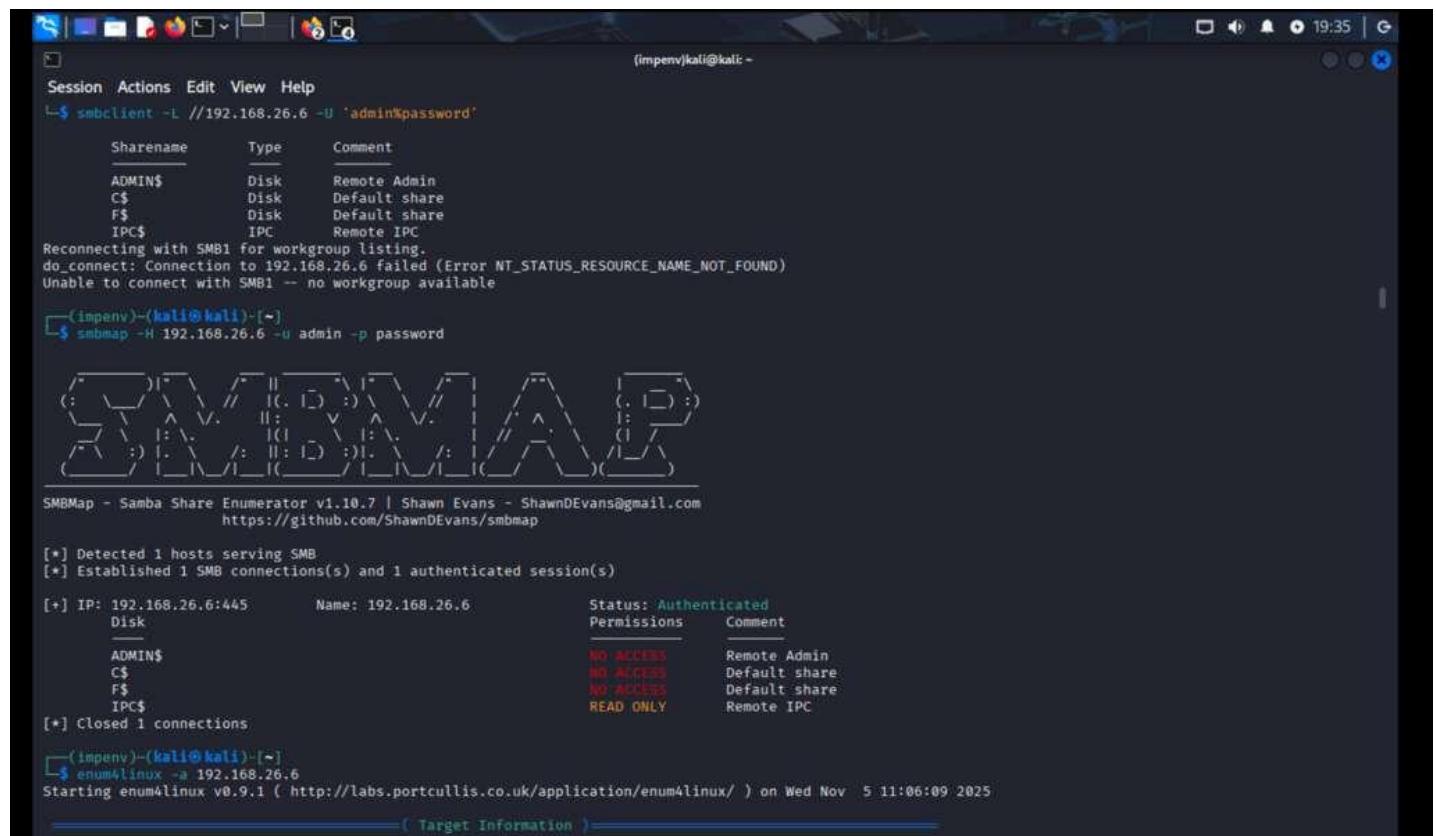
Red Team Offensive Simulation

Finding 1: SMB Access Achieved

It was possible to gain access to SMB on 192.168.26.6 with authenticated credentials having a chance to enumerate shares including ADMIN\$, C\$, F\$ and IPC\$ shares and no guest or null connections allowed. The severity of this concern is a High severity (CVSS 8) one because the compromised credentials can be used to move laterally, and it must be addressed by imposing strong access controls and tracking the SMB activity (Inbal, 2021).

Figure 36

SMB



The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal session is as follows:

```
(impenv㉿kali:~)
└─$ smbclient -L //192.168.26.6 -U 'adminKpassword'
Session Actions Edit View Help
[+] Sharename      Type      Comment
[+] ADMIN$        Disk      Remote Admin
[+] C$            Disk      Default share
[+] F$            Disk      Default share
[+] IPC$          IPC       Remote IPC
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 192.168.26.6 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
└─$ (impenv㉿kali:~)
└─$ smbmap -H 192.168.26.6 -u admin -p password
SMBMap - Samba Share Enumerator v1.10.7 | Shawn Evans - ShawnDEvans@gmail.com
https://github.com/ShawnDEvans/smbmap
[*] Detected 1 hosts serving SMB
[*] Established 1 SMB connection(s) and 1 authenticated session(s)
[*] IP: 192.168.26.6:445      Name: 192.168.26.6      Status: Authenticated
      Permissions      Comment
      NO ACCESS      Remote Admin
      NO ACCESS      Default share
      NO ACCESS      Default share
      READ ONLY      Remote IPC
[*] Closed 1 connections
└─$ (impenv㉿kali:~)
└─$ enum4linux -a 192.168.26.6
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Wed Nov  5 11:06:09 2025
----- ( Target Information ) -----
```

Figure 37

SMB

```
(impriv)kali㉿kali: ~
Session Actions Edit View Help
└─$ python3 /usr/share/doc/python3-impacket/examples/smbclient.py admin:password@192.168.26.6
Impacket v0.13.0 - Copyright Fortra, LLC and its affiliated companies

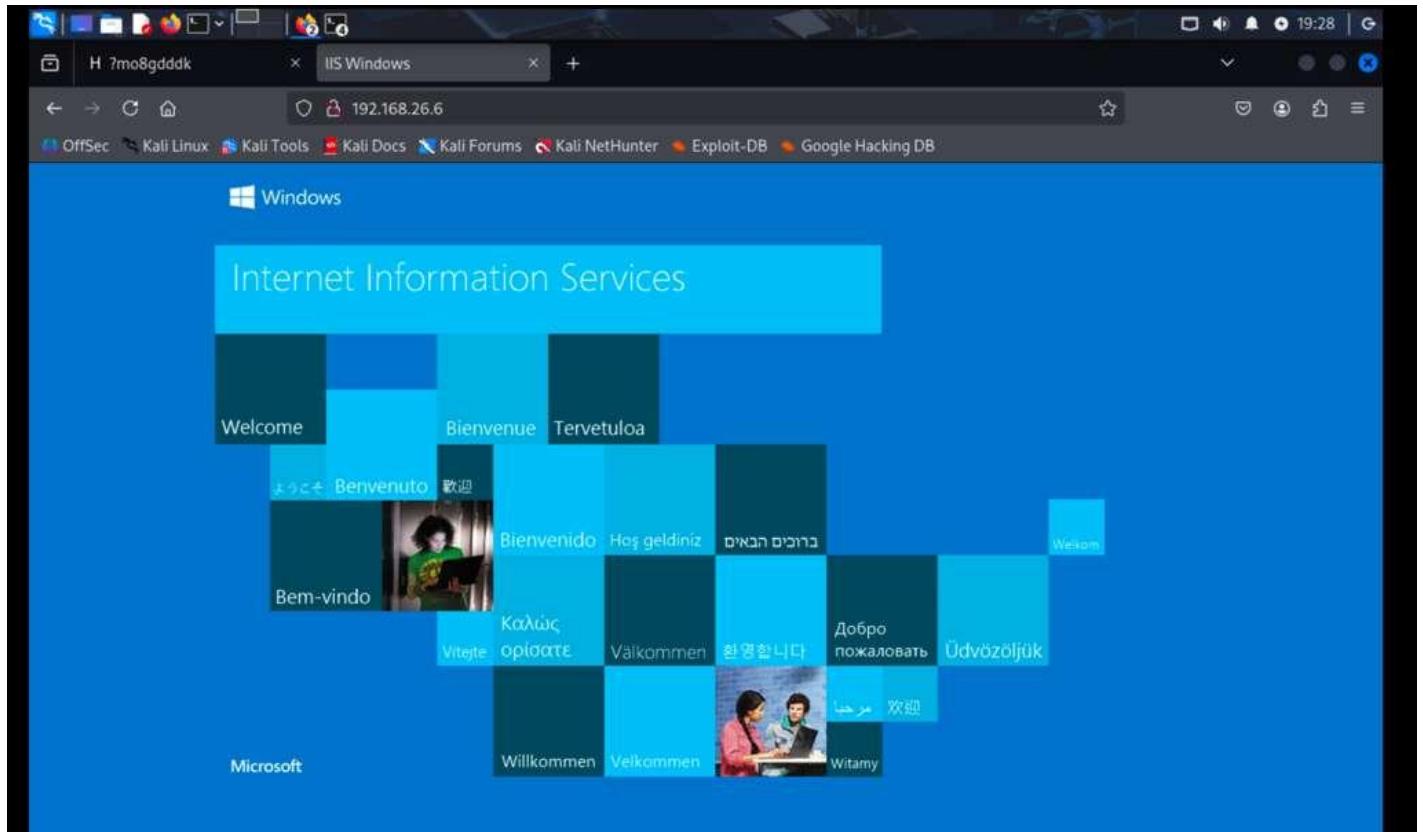
Type help for list of commands
# put /tmp/payload.vbs payload.vbs
[+] No share selected
# shares
ADMIN$
C$
F$
IPC$
# use C$
[+] SMB SessionError: code: 0xc0000002 - STATUS_ACCESS_DENIED - {Access Denied} A process has requested access to an object but has not been granted those
access rights.
# use IPC$
# ls
-rw-rw-rw-    3 Sun Dec 31 18:09:24 1600 InitShutdown
-rw-rw-rw-    4 Sun Dec 31 18:09:24 1600 lsass
-rw-rw-rw-    3 Sun Dec 31 18:09:24 1600 msavcs
-rw-rw-rw-    3 Sun Dec 31 18:09:24 1600 scspc
-rw-rw-rw-    1 Sun Dec 31 18:09:24 1600 Winsock2\CatalogChangeListener-2f0~0
-rw-rw-rw-    1 Sun Dec 31 18:09:24 1600 Winsock2\CatalogChangeListener-3e8~0
-rw-rw-rw-    3 Sun Dec 31 18:09:24 1600 epmapper
-rw-rw-rw-    1 Sun Dec 31 18:09:24 1600 Winsock2\CatalogChangeListener-248~0
-rw-rw-rw-    3 Sun Dec 31 18:09:24 1600 LSM_API_service
-rw-rw-rw-    1 Sun Dec 31 18:09:24 1600 Winsock2\CatalogChangeListener-1c4~0
-rw-rw-rw-    3 Sun Dec 31 18:09:24 1600 eventlog
-rw-rw-rw-    1 Sun Dec 31 18:09:24 1600 Winsock2\CatalogChangeListener-448~0
-rw-rw-rw-    3 Sun Dec 31 18:09:24 1600 atsvc
-rw-rw-rw-    1 Sun Dec 31 18:09:24 1600 Winsock2\CatalogChangeListener-5c0~0
-rw-rw-rw-    3 Sun Dec 31 18:09:24 1600 spools
-rw-rw-rw-    1 Sun Dec 31 18:09:24 1600 Winsock2\CatalogChangeListener-914~0
-rw-rw-rw-    4 Sun Dec 31 18:09:24 1600 crypt
-rw-rw-rw-    3 Sun Dec 31 18:09:24 1600 trkwks
-rw-rw-rw-    1 Sun Dec 31 18:09:24 1600 Winsock2\CatalogChangeListener-2d4~0
-rw-rw-rw-    4 Sun Dec 31 18:09:24 1600 svrsvc
-rw-rw-rw-    3 Sun Dec 31 18:09:24 1600 ROUTER
-rw-rw-rw-    1 Sun Dec 31 18:09:24 1600 PIPE_EVENTROOT\CMIV2SCM EVENT PROVIDER
-rw-rw-rw-    3 Sun Dec 31 18:09:24 1600 MsFteWds
-rw-rw-rw-    1 Sun Dec 31 18:09:24 1600 SearchTextHarvester
-rw-rw-rw-    1 Sun Dec 31 18:09:24 1600 Sessions\1\AppContainerNamedObjects\$1-15-2-536077884-713174666-1066051701-3219990555-339840825-1966734348
-1611281757\mojo.6248.6892.320534047620253564
-rw-rw-rw-    2 Sun Dec 31 18:09:24 1600 Sessions\1\AppContainerNamedObjects\$1-15-2-536077884-713174666-1066051701-3219990555-339840825-1966734348
```

Finding 2: IIS Running with Default Configuration & TRACE Enabled

The host 192.168.26.6 has been discovered to be running IIS with its default settings open on a port 80 and TRACE HTTP method is also available, which may be used to facilitate cross-site tracing attacks. The level of this problem is Medium (CVSS 5), and the solution will be disabling TRACE in IIS, enabling HTTPS, and deleting default IIS pages (MDN contributors, 2025).

Figure 38

IIS



Finding 3: Network Reconnaissance Identified Open Services

Network reconnaissance identified that host 192.168.26.6 had some open ports and running service information found with Nmap and Legion, but there were no direct vulnerabilities available at the time, as the available service information may be used in targeted attack (GeeksforGeeks, 2020). This is Low severity (CVSS 3), and the remediation would involve a reduction in unnecessary exposed services, periodic scanning of open ports, host-based firewall rules, and detection of suspicious scanning activity (Wattuhewa, 2023).

Figure 39

Nmap

```
(impenv)-(kali㉿kali)-[~]
└─$ nmap -sT -p 22,139,135,139,445,3389,5985,5986,1433,5988,5989,5900 192.168.26.6
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-05 10:46 CST
Nmap scan report for 192.168.26.6
Host is up (0.0071s latency).

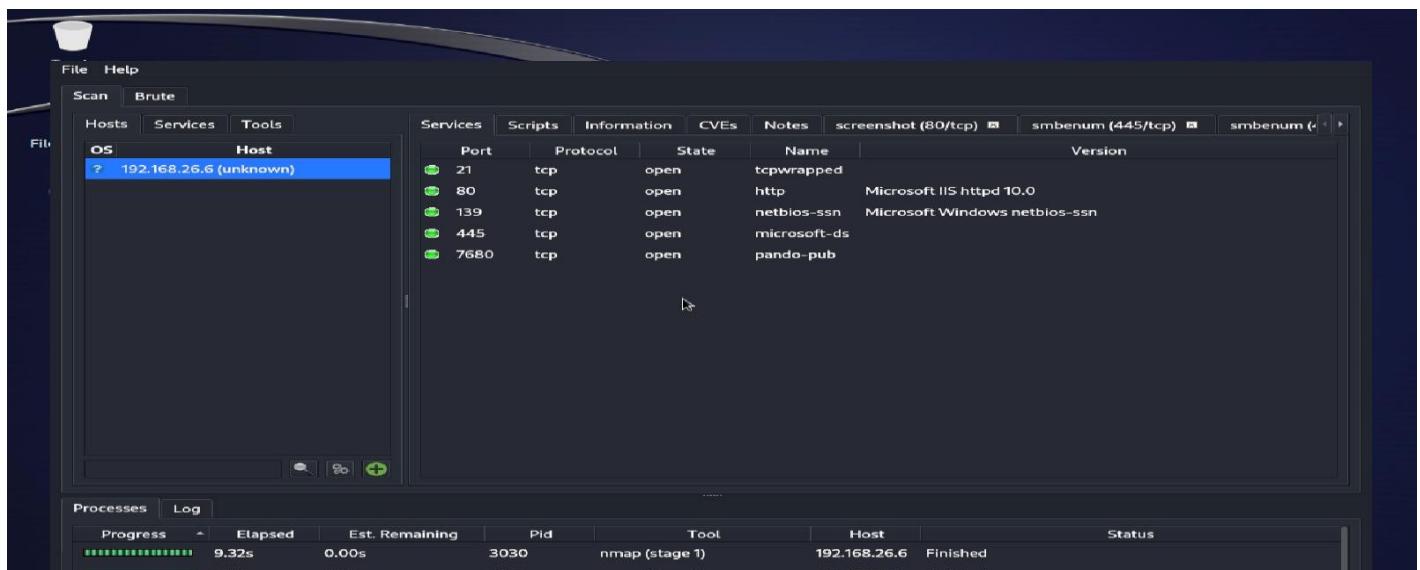
PORT      STATE SERVICE
22/tcp    filtered ssh
135/tcp   filtered msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
1433/tcp  filtered ms-sql-s
3389/tcp  filtered ms-wbt-server
5900/tcp  filtered vnc
5985/tcp  filtered wsman
5986/tcp  filtered wsman
5988/tcp  filtered wbem-http
5989/tcp  filtered wbem-https

Nmap done: 1 IP address (1 host up) scanned in 7.84 seconds

(impenv)-(kali㉿kali)-[~]
└─$ nmap -F 192.168.26.6
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-05 10:56 CST
Nmap scan report for 192.168.26.6
Host is up (0.014s latency).
Not shown: 96 filtered tcp ports (no-response)
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 8.50 seconds

(impenv)-(kali㉿kali)-[~]
└─$ nmap -p 192.168.26.6
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-05 10:56 CST
Nmap scan report for 192.168.26.6
Host is up (0.0030s latency).
Not shown: 60559 filtered tcp ports (net-unreach), 4972 filtered tcp ports (no-response)
PORT      STATE SERVICE
```

Figure 40*Legion***Finding 4: SMB Login Brute-Force Attempts Blocked**

A brute-force attack attempt on host 192.168.26 based on the smb_login Metasploit module and using several usernames and passwords to brute-force into host was unsuccessful, indicating account protection measures or lockout policies are in place. It is a Medium-severity vulnerability (CVSS 5), and its mitigation measures involve having policies of strong passwords and lockouts, attending to

recurrent unsuccessful logins, and the possibility of having SMB signing and multifactor authentication of sensitive accounts (Faturrohman et al., 2023).

Figure 41

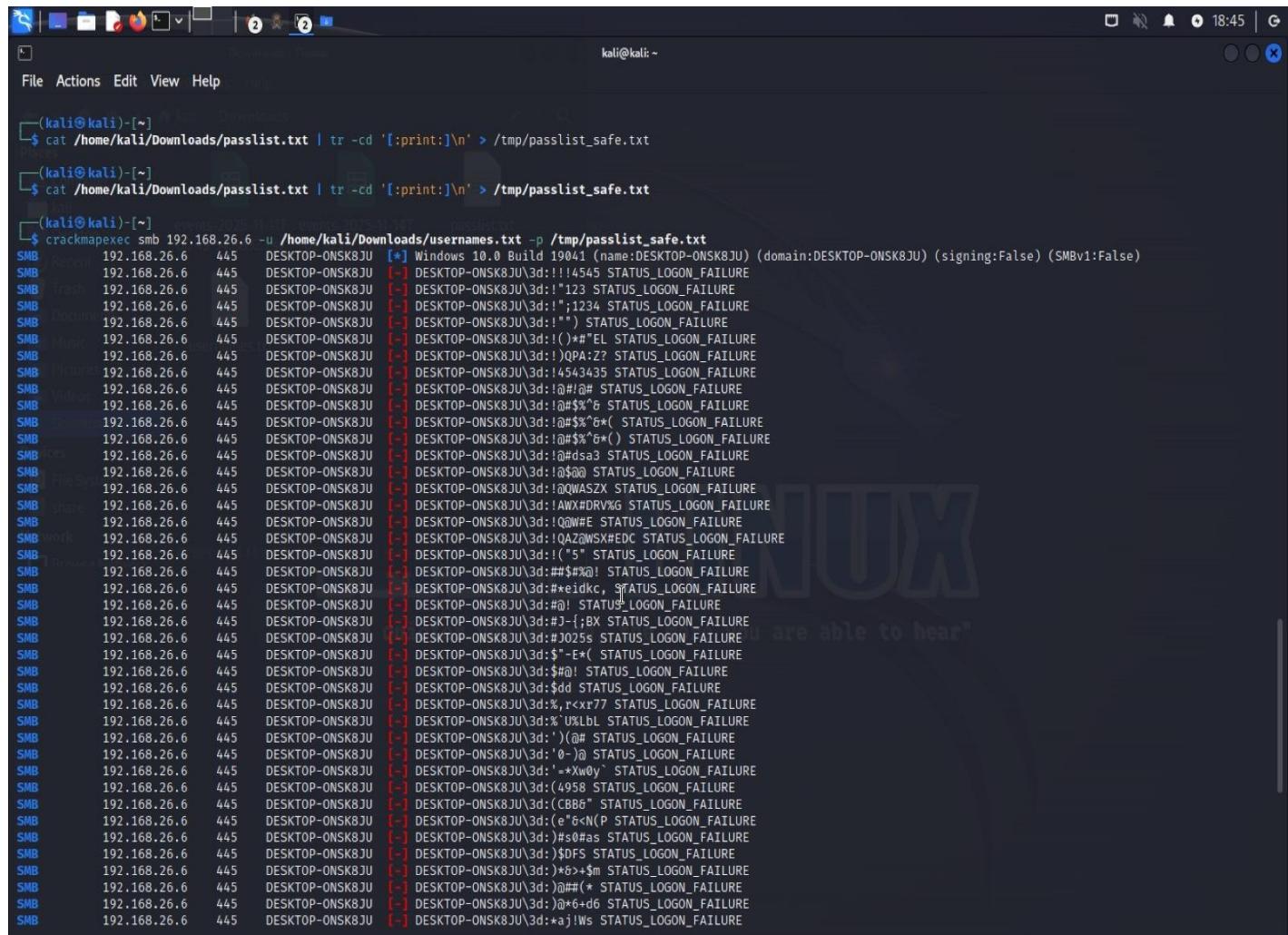
Brute-Force

```
File Actions Edit View Help
Metasploit Documentation: https://docs.metasploit.com/
msf6 > use auxiliary/scanner/smb/smb_login
msf6 auxiliary(scanner/smb/smb_login) > set RHOSTS 192.168.26.6
RHOSTS => 192.168.26.6
msf6 auxiliary(scanner/smb/smb_login) > set USER_FILE /home/kali/Downloads/username.txt
USER_FILE => /home/kali/Downloads/username.txt
msf6 auxiliary(scanner/smb/smb_login) > set PASS_FILE /tmp/passlist_safe.txt
PASS_FILE => /tmp/passlist_safe.txt
msf6 auxiliary(scanner/smb/smb_login) > run
[*] 192.168.26.6:445 - 192.168.26.6:445 - Starting SMB login bruteforce
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:!!14545', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d::"123", STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d::"1234', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d::"12345', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d::!()*#`EL', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d::!QPAZ?', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d::!4543435', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d::@#!#%', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d::@#$%6', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d::@#$%6(', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d::@#$%6(*)', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d::@#dsaz3', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:@#$%', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:@QWASZX', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:AWX#DRV%', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:!QGW#', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:IAZ@WSX#EDC', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:(5', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:##$%#!', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:#eidkc,', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:#!', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:#J:{;BX', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:#J025s', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:$-E*', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:$@!', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:$dd', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:%,rkrx77', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:%'U_lbl', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:(#@', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:O-)@', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:=Xw0Y', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:(4958', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:(CBB6', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:(e`6KN(P', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:#$0#$', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:)$DFS', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:)*$>$#$', STATUS_LOGON_FAILURE
[-] 192.168.26.6:445 - 192.168.26.6:445 - Failed: '\3d:@##(*', STATUS_LOGON_FAILURE
```

Finding 5: SMB Signing Disabled

The host is using windows SMB service, which does not have SMB signing enabled, which means that no SMB traffic has been validated, and is susceptible to Man-in-the-Middle attacks, message tampering, or authentication relays. Though no credentials were compromised the same undermines

lateral-movement protection. It is a medium-level language problem (CVSS 6), and the remedies involve the ability to use SMB signed communication via Group Policy, limit SMB traffic using host-based firewalls, and proper management of the credential to prevent credential reuse (Nairuz, 2023).

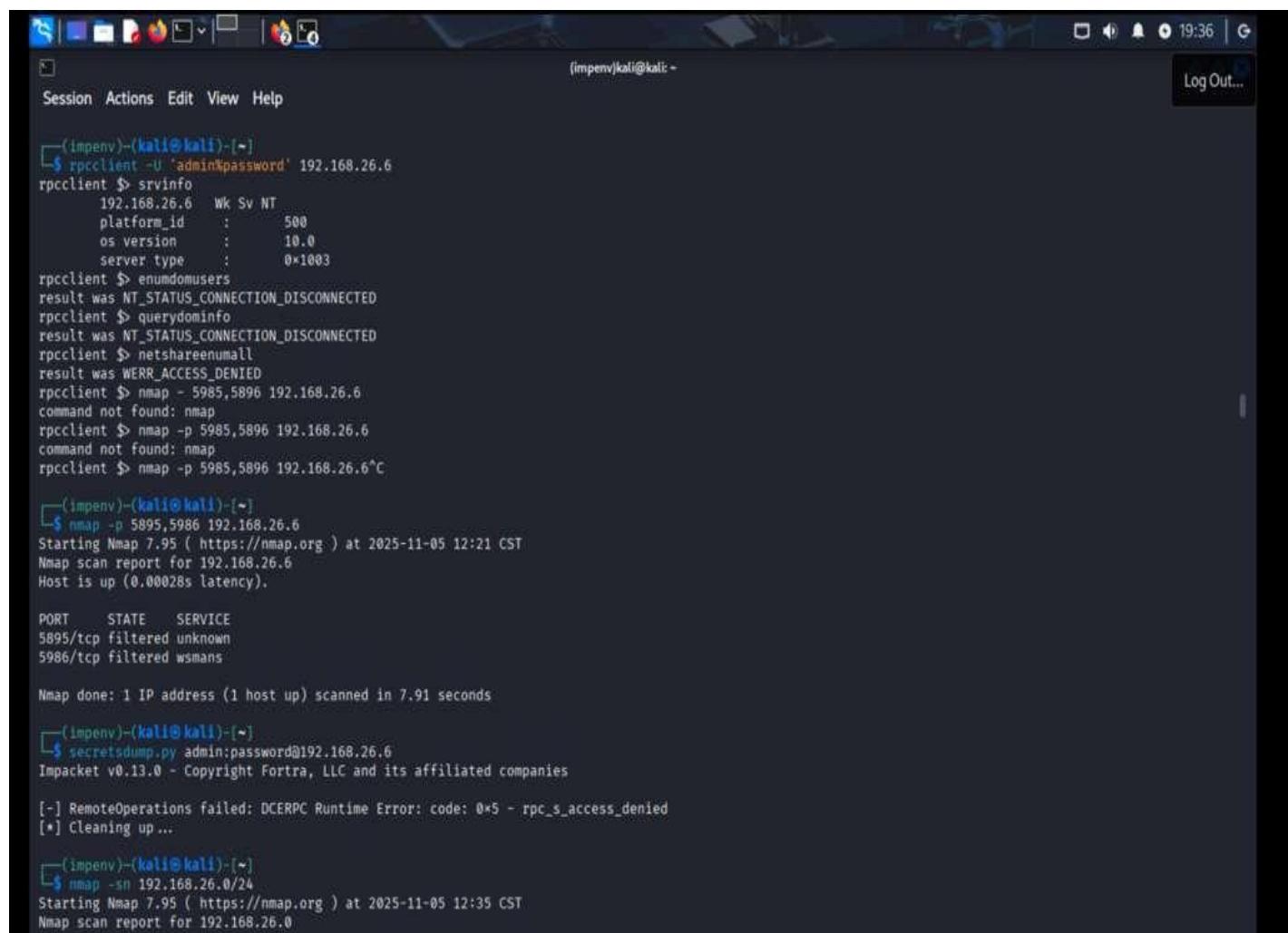
Figure 42*Crackmap*


The screenshot shows a terminal window titled 'kali@kali: ~' running on a Kali Linux system. The user has run the command 'crackmapexec smb 192.168.26.6 -u /home/kali/Downloads/usernames.txt -p /tmp/passlist_safe.txt'. The output of the command is displayed in the terminal, listing numerous logon failures for various users on the target machine.

```
(kali㉿kali)-[~]
└─$ cat /home/kali/Downloads/passlist.txt | tr -cd [:print:]\\n' > /tmp/passlist_safe.txt
(kali㉿kali)-[~]
└─$ cat /home/kali/Downloads/passlist.txt | tr -cd [:print:]\\n' > /tmp/passlist_safe.txt
(kali㉿kali)-[~]
└─$ crackmapexec smb 192.168.26.6 -u /home/kali/Downloads/usernames.txt -p /tmp/passlist_safe.txt
SMB Reconnect 192.168.26.6 445 DESKTOP-ONSK8JU [*] Windows 10.0 Build 19041 (name:DESKTOP-ONSK8JU) (domain:DESKTOP-ONSK8JU) (signing:False) (SMBv1:False)
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d!!!4545 STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*123 STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*1234 STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*"
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*#*EL STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*QPA:Z? STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*4543435 STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*#/# STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*#*$^& STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*#*$^&*() STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*#*$^&*() STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*#*dsaa3 STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*#$@@ STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*@WASZX STATUS_LOGON_FAILURE
SMB share 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*AWXDRVNG STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*@WHE STATUS_LOGON_FAILURE
SMB work 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*QZ0WSX#EDC STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*("5 STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*##$@! STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*#*idkc, STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*#@ STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*BX STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*J-{BX STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*J025s STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:*$-E( STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:*$#@! STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:$dd STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:*,r>r77 STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:;%`U%lbl STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:*)(@# STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d: 0-)@ STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:!*xw0y` STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:(4958 STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:(CBB6 STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:(e`&NP STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:#s0#as STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:)${0FS STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:)*6+*$ STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:)*#(* STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:)*@*6+d6 STATUS_LOGON_FAILURE
SMB 192.168.26.6 445 DESKTOP-ONSK8JU [-] DESKTOP-ONSK8JU\3d:*aj!ws STATUS_LOGON_FAILURE
```

Finding 6: Remote Service Information Disclosure via RPC

Host revealed the operating systems information through the `srvinfo` RPC command even though most of the enumeration was blocked which meant that attackers could customise exploits or password attacks based on the OS and configuration. It is a Medium-level problem (CVSS 6), and its solution involves disabling the unnecessary SMB/RPC services, permitting SMB/RPC traffic to trusted hosts, using the latest windows security patches, and implementing firewall policies to restrict RPC discovery traffic.

Figure 43*RPC*

The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal session is running under a user named 'impenv' with the password 'password'. The session starts with some basic system information and then proceeds to enumerate the host using the `rpcclient` tool. The user runs `rpcclient -U 'admin%password' 192.168.26.6` and then `rpcclient $ srvinfo`. The output shows the host is a Windows 7 (W7) system with a platform ID of 500, OS version 10.0, and server type 0x1003. The user then runs `rpcclient $ enumdomusers`, which fails with `result was NT_STATUS_CONNECTION_DISCONNECTED`. They then run `rpcclient $ querydominfo`, `rpcclient $ netshareenumall`, and `rpcclient $ netshareenumall`, all of which fail with `result was WERR_ACCESS_DENIED`. Finally, they run `rpcclient $ nmap -p 5985,5896 192.168.26.6` and `rpcclient $ nmap -p 5985,5896 192.168.26.6`, both of which fail with `command not found: nmap`. The user then runs `rpcclient $ nmap -p 5985,5896 192.168.26.6^C` to exit the `rpcclient` session. The terminal then shows the user running `nmap -p 5895,5986 192.168.26.6`, which starts an Nmap scan report for the host. The report shows the host is up with 0 latency. It lists two ports: 5895/tcp filtered unknown and 5986/tcp filtered wsmans. The user then runs `secretsdump.py admin:password@192.168.26.6`, which is part of the Impacket tool. The output shows a failed attempt to dump secrets due to a DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied. The user then runs `nmap -sn 192.168.26.0/24` to perform a quick scan of the subnet.

```
(impenv)-(kali㉿kali)-[~]
$ rpcclient -U 'admin%password' 192.168.26.6
rpcclient $ srvinfo
192.168.26.6  Wk Sv NT
platform_id   :      500
os version    : 10.0
server type   : 0x1003
rpcclient $ enumdomusers
result was NT_STATUS_CONNECTION_DISCONNECTED
rpcclient $ querydominfo
result was NT_STATUS_CONNECTION_DISCONNECTED
rpcclient $ netshareenumall
result was WERR_ACCESS_DENIED
rpcclient $ nmap -p 5985,5896 192.168.26.6
command not found: nmap
rpcclient $ nmap -p 5985,5896 192.168.26.6
command not found: nmap
rpcclient $ nmap -p 5985,5896 192.168.26.6^C

(impenv)-(kali㉿kali)-[~]
$ nmap -p 5895,5986 192.168.26.6
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-05 12:21 CST
Nmap scan report for 192.168.26.6
Host is up (0.00028s latency).

PORT      STATE      SERVICE
5895/tcp  filtered  unknown
5986/tcp  filtered  wsmans

Nmap done: 1 IP address (1 host up) scanned in 7.91 seconds

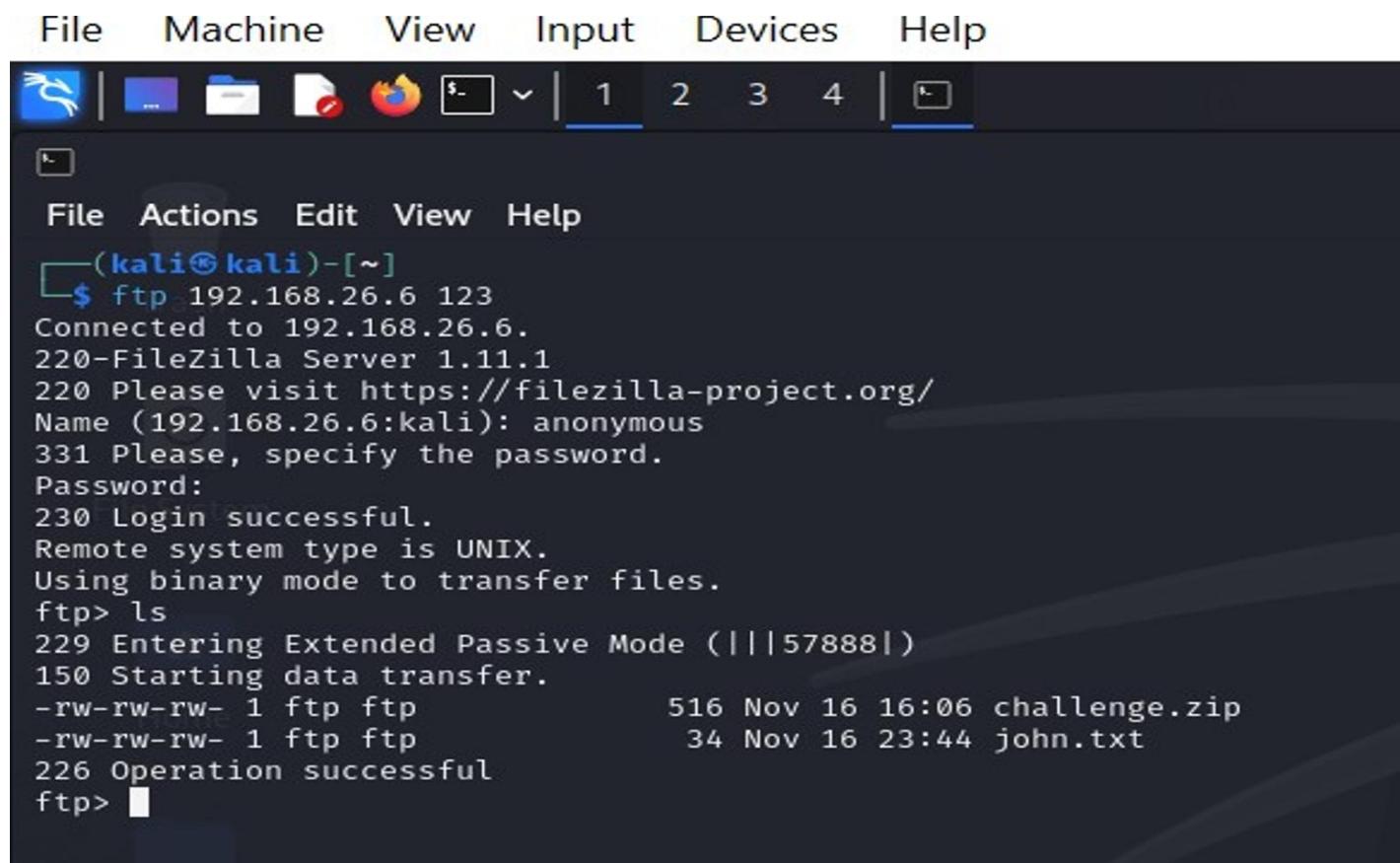
(impenv)-(kali㉿kali)-[~]
$ secretsdump.py admin:password@192.168.26.6
Impacket v0.13.0 - Copyright Fortra, LLC and its affiliated companies

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Cleaning up ...

(impenv)-(kali㉿kali)-[~]
$ nmap -sn 192.168.26.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-05 12:35 CST
Nmap scan report for 192.168.26.0
```

Finding 7: FTP Access on Port 123

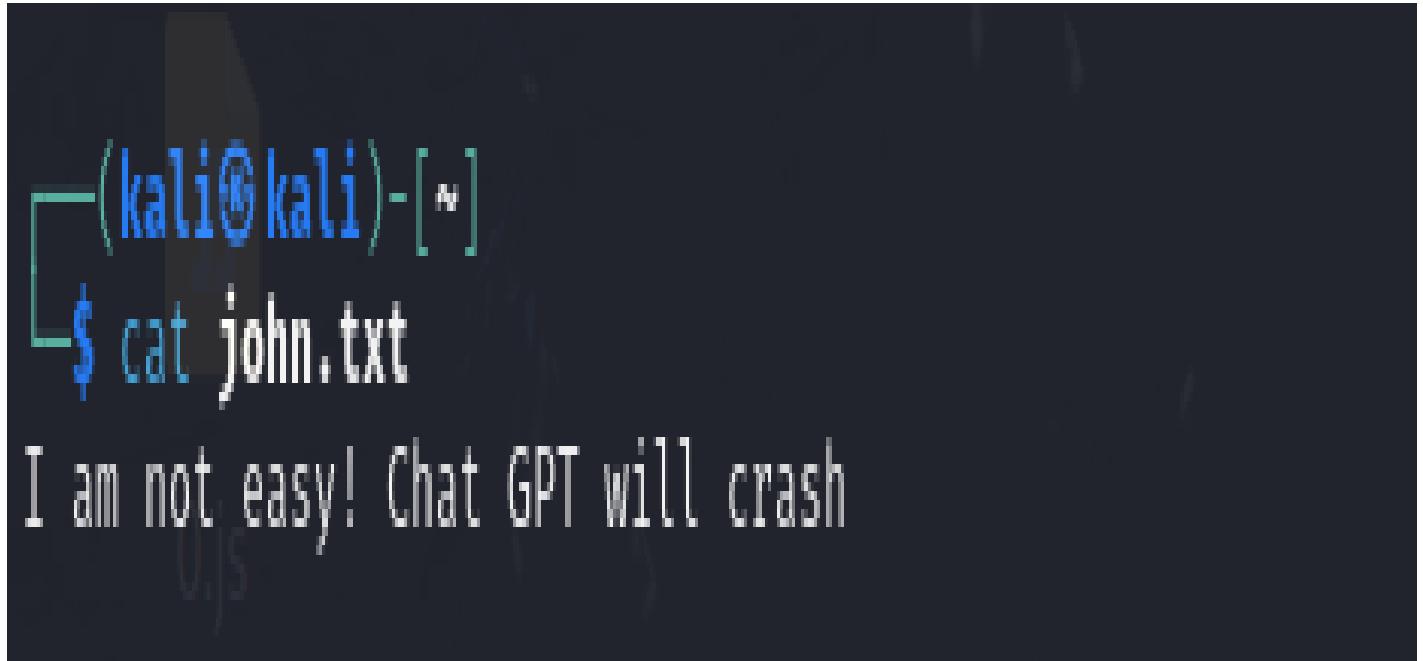
FTP service was found open on the nonstandard port 123 and the search of the FTP directory showed that there were two files: john.txt, which merely had the teasing text line “I am not easy! Chat GPT will crash,” and the user input to Challenge.zip, which decompresses to a main.c program that will XOR-transform the input and compare it with a hard-coded XOR-encoded array, is XOR-transformed and then compared before the program produces the success message.

Figure 44*FTP*

The screenshot shows the FileZilla interface. The menu bar includes File, Machine, View, Input, Devices, and Help. The toolbar has icons for file operations like Open, Save, and Copy. The status bar at the bottom shows '(kali㉿kali)-[~]'. The main window displays an FTP session:

```
$ ftp 192.168.26.6 123
Connected to 192.168.26.6.
220-FileZilla Server 1.11.1
220 Please visit https://filezilla-project.org/
Name (192.168.26.6:kali): anonymous
331 Please, specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||57888|)
150 Starting data transfer.
-rw-rw-rw- 1 ftp ftp          516 Nov 16 16:06 challenge.zip
-rw-rw-rw- 1 ftp ftp          34 Nov 16 23:44 john.txt
226 Operation successful
ftp> █
```

Figure 45*Contents of john.txt*



(kali㉿kali)-[~]\$ cat john.txt
I am not easy! Chat GPT will crash

Figure 46

main.c

```
(kali㉿kali)-[~]$ cat main.c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

void transform(char *s) {
    for (int i = 0; s[i]; i++) {
        s[i] = s[i] ^ (0x5A - (i % 3)); // XOR + pattern shift
    }
}

int main() {
    char input[64];

    unsigned char encoded[] = {
        0x08, 0x37, 0x1F, 0x08, 0x2E, 0x3B, 0x35, 0x08,
        0x0F, 0x3A, 0x1F, 0x08, 0x2E, 0x3B, 0x35, 0x00
    };

    for (int i = 0; encoded[i] != 0; i++) {
        encoded[i] ^= (0x5A - (i % 3)); // decode at runtime
    }

    printf("Enter the flag: ");
    fgets(input, sizeof(input), stdin);
    input[strcspn(input, "\n")] = 0;

    char check[64];
    strcpy(check, input);
    transform(check);

    if (strcmp(check, encoded) == 0) {
        printf("Correct! Flag: %s\n", input);
    } else {
        printf("Wrong flag!\n");
    }
}

return 0;
}
```

Blue Team Defensive Simulation

The Blue Team used the Wazuh threat-hunting dashboards that have agent-level granularity, where the registry, Sysmon, and command-execution activity were combined to reveal the following key indicators identified: suspicious service-creation event (rule ID 92307), Sysmon-detected outbound network connections to external IPs out of C:Program Files (rule ID 61605), and reconnaissance or account-discovery attempts forwarded with net.exe (rule ID 92039). They were set to moderate and high (level 3-5) alert thresholds to facilitate active triage and dashboards provided timestamped and correlated visibility by multiple kill-chain phases such as persistence, C2 communication and discovery so that they can easily detect and respond effectively.

Figure 47

Wazuh Dashboard

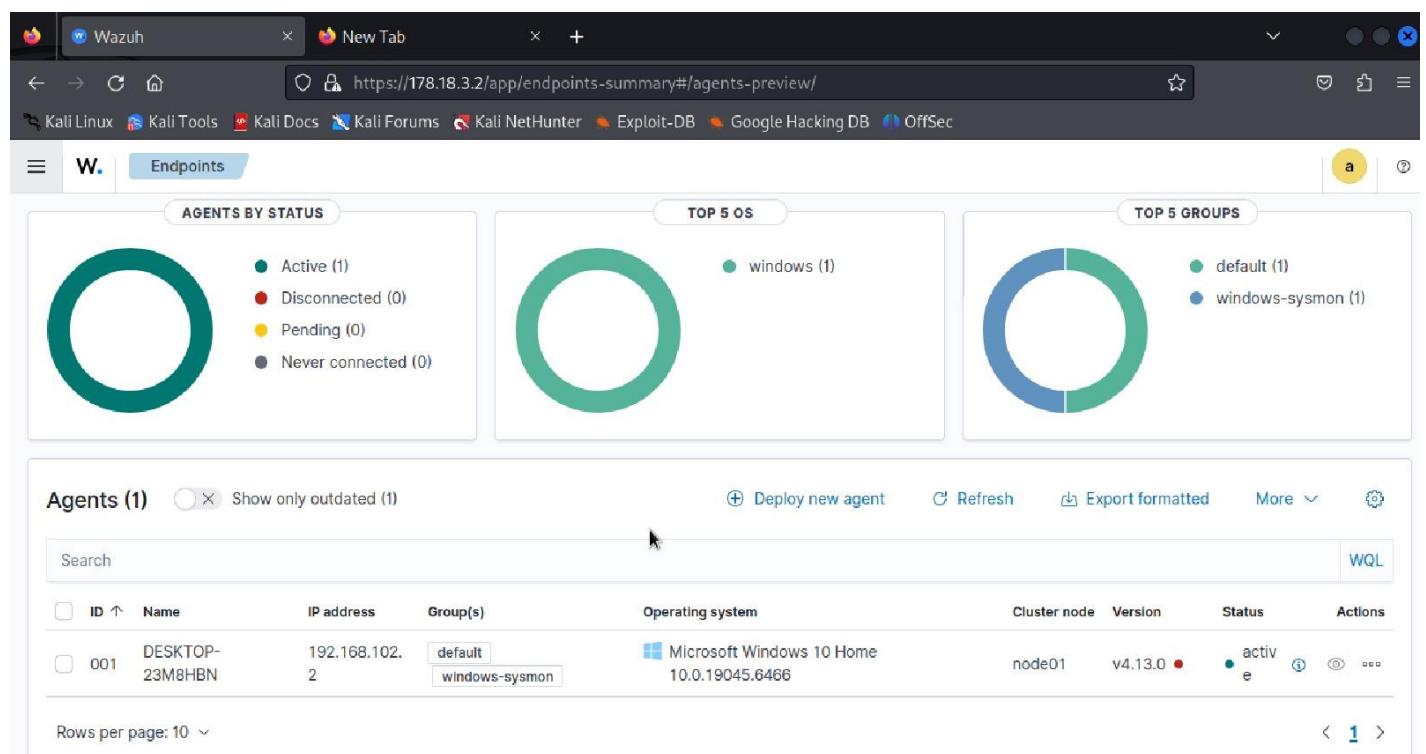
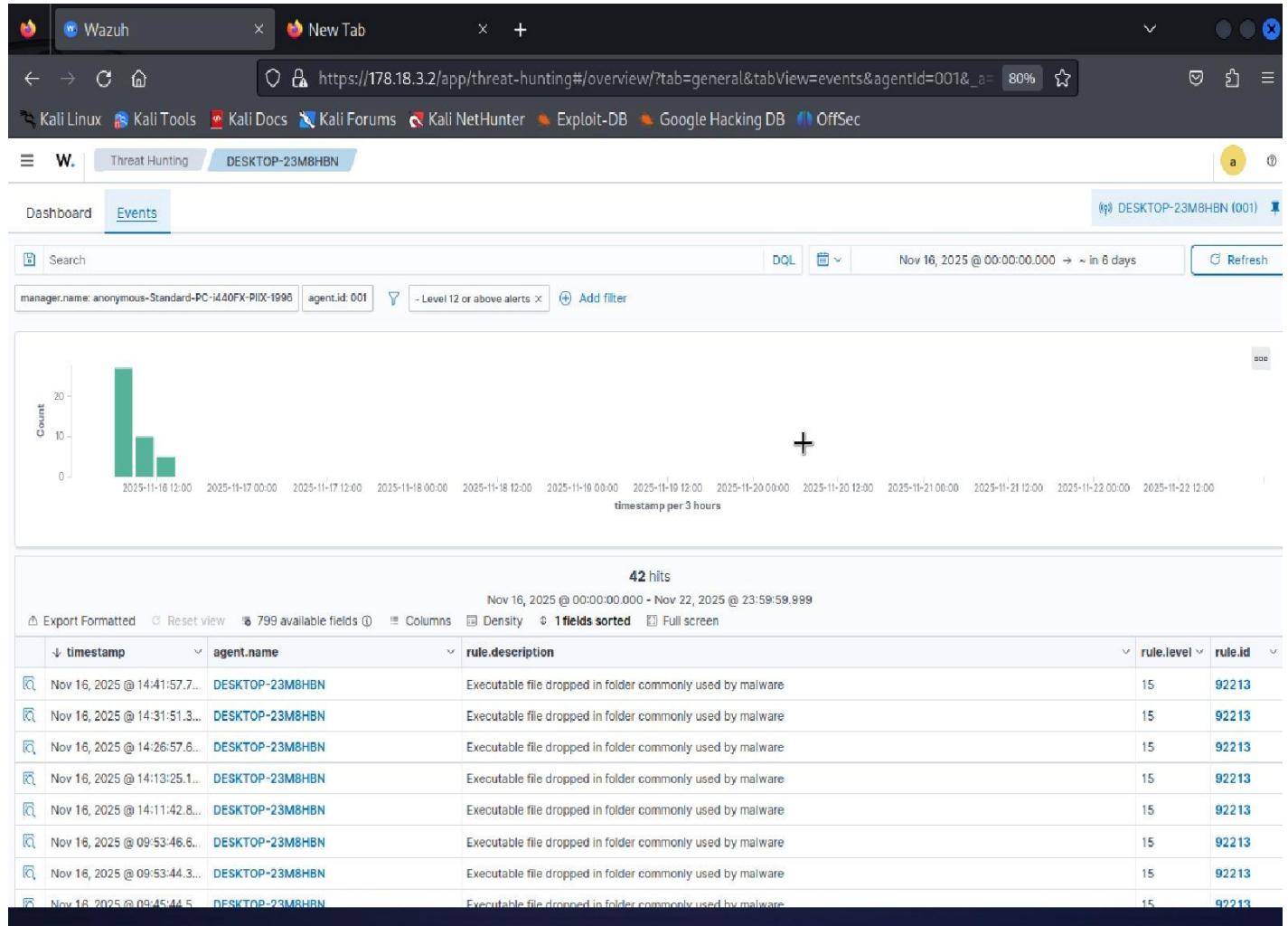


Figure 48*Wazuh Events*

We can observe repetitive activity of command shells and auto /scheduled processes, malware drops of high severity (level 15) on common folders, 689 SSH logon failures and 110 SSH connections by 192.168.78.9 indicating a possibility of brute-force attack or unauthorized access. Other anomalies are system time, service startup as well as registry integrity. Most network activity is safe (Microsoft Defender endpoints), but 192.168.78.9:22 is unusual, and the CVEs 59505 (occasional) suggested the need to patch. These trends lead to potential enduring threats, signs of malware and target attacks during the period under investigation.

Figure 49*Wazuh rule.id 92307*

timestamp	agent.name	rule.description	rule.level	rule.id
Nov 16, 2025 @ 14:25:58.3...	DESKTOP-23M8HBN	Evidence of new service creation found in registry under HKLM\System\ControlSet001\Services	3	92307
Nov 16, 2025 @ 14:25:58.6...	DESKTOP-23M8HBN	Evidence of new service creation found in registry under HKLM\System\ControlSet001\Services	3	92307
Nov 16, 2025 @ 14:25:58.6...	DESKTOP-23M8HBN	Evidence of new service creation found in registry under HKLM\System\ControlSet001\Services	3	92307
Nov 16, 2025 @ 14:25:58.6...	DESKTOP-23M8HBN	Evidence of new service creation found in registry under HKLM\System\ControlSet001\Services	3	92307
Nov 16, 2025 @ 14:25:58.5...	DESKTOP-23M8HBN	Evidence of new service creation found in registry under HKLM\System\ControlSet001\Services	3	92307
Nov 16, 2025 @ 14:25:58.5...	DESKTOP-23M8HBN	Evidence of new service creation found in registry under HKLM\System\ControlSet001\Services	3	92307
Nov 16, 2025 @ 14:25:58.4...	DESKTOP-23M8HBN	Evidence of new service creation found in registry under HKLM\System\ControlSet001\Services	3	92307
Nov 16, 2025 @ 14:25:58.4...	DESKTOP-23M8HBN	Evidence of new service creation found in registry under HKLM\System\ControlSet001\Services	3	92307
Nov 16, 2025 @ 14:25:58.4...	DESKTOP-23M8HBN	Evidence of new service creation found in registry under HKLM\System\ControlSet001\Services	3	92307
Nov 16, 2025 @ 14:25:58.4...	DESKTOP-23M8HBN	Evidence of new service creation found in registry under HKLM\System\ControlSet001\Services	3	92307
Nov 16, 2025 @ 14:25:58.4...	DESKTOP-23M8HBN	Evidence of new service creation found in registry under HKLM\System\ControlSet001\Services	3	92307
Nov 16, 2025 @ 14:25:58.3...	DESKTOP-23M8HBN	Evidence of new service creation found in registry under HKLM\System\ControlSet001\Services	3	92307
Nov 16, 2025 @ 14:25:58.3...	DESKTOP-23M8HBN	Evidence of new service creation found in registry under HKLM\System\ControlSet001\Services	3	92307
Nov 16, 2025 @ 14:25:58.2...	DESKTOP-23M8HBN	Evidence of new service creation found in registry under HKLM\System\ControlSet001\Services	3	92307
Nov 16, 2025 @ 14:25:58.2...	DESKTOP-23M8HBN	Evidence of new service creation found in registry under HKLM\System\ControlSet001\Services	3	92307

Figure 50*Wazuh rule.id 61605*

timestamp	agent.name	rule.description	rule.level	rule.id
Nov 16, 2025 @ 21:18:12.7...	DESKTOP-23M8HBN	Sysmon - Event 3: Network connection to 52.123.129.14:443 by C:\Program Files\Windows Resource Kits\Tools\NETCAT...	5	61605
Nov 16, 2025 @ 21:13:12.5...	DESKTOP-23M8HBN	Sysmon - Event 3: Network connection to 52.123.129.14:443 by C:\Program Files\Windows Resource Kits\Tools\NETCAT...	5	61605
Nov 16, 2025 @ 21:08:12.2...	DESKTOP-23M8HBN	Sysmon - Event 3: Network connection to 52.123.129.14:443 by C:\Program Files\Windows Resource Kits\Tools\NETCAT...	5	61605
Nov 16, 2025 @ 21:03:12.2...	DESKTOP-23M8HBN	Sysmon - Event 3: Network connection to 52.123.129.14:443 by C:\Program Files\Windows Resource Kits\Tools\NETCAT...	5	61605
Nov 16, 2025 @ 21:01:59.9...	DESKTOP-23M8HBN	Sysmon - Event 3: Network connection to 52.123.129.14:443 by C:\Program Files\Windows Resource Kits\Tools\NETCAT...	5	61605
Nov 16, 2025 @ 20:55:56.2...	DESKTOP-23M8HBN	Sysmon - Event 3: Network connection to 20.42.73.31:443 by C:\Program Files\Windows Resource Kits\Tools\NETCAT...	5	61605
Nov 16, 2025 @ 20:25:56.2...	DESKTOP-23M8HBN	Sysmon - Event 3: Network connection to 20.189.173.9:443 by C:\Program Files\Windows Resource Kits\Tools\NETCAT...	5	61605
Nov 16, 2025 @ 20:01:59.0...	DESKTOP-23M8HBN	Sysmon - Event 3: Network connection to 52.123.129.14:443 by C:\Program Files\Windows Resource Kits\Tools\NETCAT...	5	61605
Nov 16, 2025 @ 19:56:58.4...	DESKTOP-23M8HBN	Sysmon - Event 3: Network connection to 52.123.129.14:443 by C:\Program Files\Windows Resource Kits\Tools\NETCAT...	5	61605
Nov 16, 2025 @ 19:55:57.0...	DESKTOP-23M8HBN	Sysmon - Event 3: Network connection to 20.189.173.9:443 by C:\Program Files\Windows Resource Kits\Tools\NETCAT...	5	61605
Nov 16, 2025 @ 19:51:59.2...	DESKTOP-23M8HBN	Sysmon - Event 3: Network connection to 52.123.129.14:443 by C:\Program Files\Windows Resource Kits\Tools\NETCAT...	5	61605
Nov 16, 2025 @ 19:46:58.6...	DESKTOP-23M8HBN	Sysmon - Event 3: Network connection to 52.123.129.14:443 by C:\Program Files\Windows Resource Kits\Tools\NETCAT...	5	61605
Nov 16, 2025 @ 19:45:46.4...	DESKTOP-23M8HBN	Sysmon - Event 3: Network connection to 52.123.129.14:443 by C:\Program Files\Windows Resource Kits\Tools\NETCAT...	5	61605
Nov 16, 2025 @ 19:25:56.1...	DESKTOP-23M8HBN	Sysmon - Event 3: Network connection to 104.46.162.225:443 by C:\Program Files\Windows Resource Kits\Tools\NETCAT...	5	61605
Nov 16, 2025 @ 19:55:57.0...	DESKTOP-23M8HBN	Sysmon - Event 3: Network connection to 104.46.162.225:443 by C:\Program Files\Windows Resource Kits\Tools\NETCAT...	5	61605

Figure 51

Wazuh rule.id 92039

timestamp	agent.name	rule.description	rule.level	rule.id
Nov 16, 2025 @ 14:26:27.1...	DESKTOP-23M8HBN	A net.exe account discovery command was initiated	3	92039
Nov 16, 2025 @ 14:25:52.8...	DESKTOP-23M8HBN	A net.exe account discovery command was initiated	3	92039
Nov 16, 2025 @ 14:25:52.8...	DESKTOP-23M8HBN	A net.exe account discovery command was initiated	3	92039
Nov 16, 2025 @ 14:25:52.7...	DESKTOP-23M8HBN	A net.exe account discovery command was initiated	3	92039
Nov 16, 2025 @ 14:25:52.7...	DESKTOP-23M8HBN	A net.exe account discovery command was initiated	3	92039
Nov 16, 2025 @ 14:25:52.7...	DESKTOP-23M8HBN	A net.exe account discovery command was initiated	3	92039
Nov 16, 2025 @ 14:25:52.6...	DESKTOP-23M8HBN	A net.exe account discovery command was initiated	3	92039
Nov 16, 2025 @ 14:25:52.6...	DESKTOP-23M8HBN	A net.exe account discovery command was initiated	3	92039
Nov 16, 2025 @ 14:25:52.0...	DESKTOP-23M8HBN	A net.exe account discovery command was initiated	3	92039
Nov 16, 2025 @ 10:00:11.6...	DESKTOP-23M8HBN	A net.exe account discovery command was initiated	3	92039
Nov 16, 2025 @ 09:59:32.9...	DESKTOP-23M8HBN	A net.exe account discovery command was initiated	3	92039
Nov 16, 2025 @ 09:59:32.9...	DESKTOP-23M8HBN	A net.exe account discovery command was initiated	3	92039
Nov 16, 2025 @ 09:59:32.9...	DESKTOP-23M8HBN	A net.exe account discovery command was initiated	3	92039
Nov 16, 2025 @ 09:59:32.9...	DESKTOP-23M8HBN	A net.exe account discovery command was initiated	3	92039
Nov 16, 2025 @ 09:59:32.9...	DESKTOP-23M8HBN	A net.exe account discovery command was initiated	3	92039
Nov 16, 2025 @ 09:59:32.9...	DESKTOP-23M8HBN	A net.exe account discovery command was initiated	3	92039

The detection rules were registry service-create and registry service-modify messages (rule ID 92307) due to new entries under HKLM\System\CurrentControlSet\Services, Sysmon-based network connection messages (rule ID 61605) based on traffic to external IPs over port 443 and account-discovery-based net.exe command messages (rule ID 92039). The Dashboards facilitated triage using Events tab which was filtered by agent name and high severity rule and sorted by timestamp and rule description to enable easy investigation.

Suggestions are to investigate the 228 malware-drop events with the immediate scanning of suspicious folders and isolating the host, hardening the security with multi-factor authentication of SSH/RDP, block or monitor 192.168.78.9, and restrict the use of cmd.exe through policy. CVEs such as

2025-59505 should be patched using Windows and Defender. It is recommended to perform continuous monitoring and issue notifications in case of repeated executions of cmd's and logon failures and monitor the logs to identify parent processes that triggered an abnormal cmd activity. There is a possibility of further action that would include correlation with other logs or a complete system scan during incident response.

Learning Outcomes

The project required me to build an entire enterprise environment from scratch while learning its design and security configuration methods. The combination of Proxmox and VyOS and pfSense and Ubuntu and Windows 10 systems demonstrated how each infrastructure level depends on the previous one. The entire network collapses when you make a single mistake between incorrect gateway settings and absent NAT rules. The process of resolving these problems enabled me to understand actual network operations better while learning about the critical infrastructure configuration requirements.

The offensive and defensive simulation activities provided me with essential data which helped me learn new information. The combination of Legion and Hydra and CrackMapExec and Nmap and Metasploit tools provided me with hands-on experience of attacker tools used for enumeration and brute-force attacks and vulnerability testing. The failed attempts demonstrated to me how secure systems function when attackers try to access them while showing me the value of proper system configurations and secure passwords and network segmentation. The defensive training demonstrated to me how Wazuh and Sysmon systems operate together to identify security threats. The Wazuh dashboard displayed process creation events and login failures and registry changes and network activity and brute-force attempts which helped me understand how to track threats in real-time.

Troubleshooting became one of my biggest learning areas. I spent time fixing Wazuh manager issues, improving Sysmon logging, validating pfSense rules, and resolving connectivity errors. This taught me to slow down and confirm assumptions before moving forward. I also learned how important teamwork is in a project like this. Every change one person made affected the entire environment, so we had to communicate constantly, verify each other's work, and solve problems together. By the end, I felt more confident not just technically, but also in project management, collaboration, and staying organized under pressure.

Reflection

This project was the most challenging and rewarding thing I've done in the entire program. There were times when nothing worked - routing didn't pass traffic, pfSense blocked connections we needed, Wazuh agents would not register, and Sysmon logs were missing. I learned not to panic and instead break the problem down layer by layer: check the IPs, check the routes, check the firewall rules, check the agent, check the logs. Every time I solved an issue, I understood the environment a little better.

The Wazuh package mistake alone cost us almost two weeks of repeated uninstalls and cleanups - frustrating, but it taught me how one tiny command can derail everything and why attention to detail is everything.

My experience with Red Team and Blue Team activities provided me with a complete purple team perspective. Running offensive tools like Hydra, CrackMapExec, Legion, Nmap, and Metasploit made me realize how difficult it is to successfully attack a well-configured system. The best moment was seeing another team's real attacks instantly trigger high-severity alerts in our Wazuh dashboard with their source IPs. It proved our setup worked under live pressure. Attacking another lab while defending

our own showed me how quickly misconfigurations get exploited and how powerful proper logging and detection are.

The project required me to acquire essential skills which I needed to learn. I gained technical expertise through working with SIEM tools and learning to manage firewalls and routes and monitor endpoints and solve technical problems. The project allowed me to build my abilities in working with teams and developing my skills in professional communication. The group members encountered multiple instances of incorrect setup which affected everyone, so we needed to stay connected for fast technical problem resolution. It felt like working in a real SOC where collaboration is critical.

If I were to do this project again, I would verify “wazuh-one” installation, automate more of the Windows hardening and Sysmon setup, and keep a shared change-log to avoid accidental misconfigurations. But overall, I’m proud of how much I learned and how well our team worked together. This project gave me real hands-on experience with the same tools, challenges, and workflows used by cybersecurity professionals every day - and it made me even more confident in the path I’m pursuing.

References

- Faturrohman, M., Salsabila, A., Mardiah, Z., & Rosadi, A. (2023). Attack in to The Server Message Block (CVE-2020-0796) Vulnerabilities in Windows 10 using Metasploit Framework. JEEMECS (Journal of Electrical Engineering, Mechatronic and Computer Science). 6. 37-44.
- https://www.researchgate.net/publication/368965977_Attack_in_to_The_Server_Message_Block_CVE-2020-0796_Vulnerabilities_in_Windows_10_using_Metasploit_Framework
- Ford, D. (2019, June 30). Proxmox VE – How to home lab part 1. dlford.io. <https://www.dlford.io/how-to-home-lab-part-1/>
- GeeksforGeeks. (2020). Legion Tool in Kali Linux. <https://www.geeksforgeeks.org/linux-unix/legion-tool-in-kali-linux/>
- Hoxha, D. (2020). Using WireGuard VPN.
https://www.researchgate.net/publication/345641555_Using_WireGuard_VPN
- Inbal, A. (2021). Lateral Movement. <https://www.cynet.com/blog/lateral-movement/>
- Jiang, Y., Meng, Q., Shang, F., Oo, N., Minh, L., Lim, H., & Sikdar, B. (2025). MITRE ATT&CK Applications in Cybersecurity and The Way Forward.
https://www.researchgate.net/publication/389090450_MITRE_ATTCK_Applications_in_Cybersecurity_and_The_Way_Foward
- MDN contributors. (2025). TRACE request method. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Methods/TRACE>

Nairuz, A. (2023). CrackMapExec in Action: Enumerating Windows Networks (Part 1).

<https://medium.com/r3d-buck3t/crackmapexec-in-action-enumerating-windows-networks-part-1-3a6a7e5644e9>

Rodríguez, A. (2017, May 30). Using Wazuh to monitor Sysmon events. Wazuh.

<https://wazuh.com/blog/using-wazuh-to-monitor-sysmon-events/>

Samuel, O. (2018, July 11). How to configure Nat the cisco and Vyos Way. Expert Network Consultant.

<https://www.expertnetworkconsultant.com/configuring/how-to-configure-nat-the-cisco-and-vyos-way/>

VyOS maintainers and contributors. (2024). NAT. In VyOS 1.3 documentation. VyOS.

<https://docs.vyos.io/en/1.3/configuration/nat/index.html>

Wattuhewa, S. (2023). Network Scanning with Nmap.

https://www.researchgate.net/publication/374135016_Network_Scanning_with_Nmap