Design and Implementation of a Segmented Enterprise Network

Simon Kuester

CYBR 4950

December 3, 2025

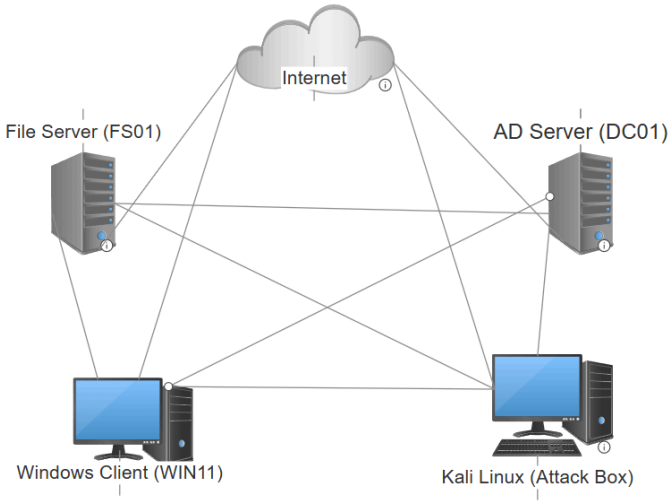
Table of Contents

Abstract

This capstone project presents the design, implementation, and hardening of a segmented enterprise network using tools and best practices. In the beginning of the semester, I deployed and integrated a unsecure flat virtual network that mimics a small business network environment using VMware, and then performed various attacks to highlight the dangers of an unsegmented network. Following these attacks, the network and virtual machines were then strengthened and hardened using a variety of industry standard cybersecurity tools and technology such as pfSense, Active Directory policies, Snort, and Wazuh. After each strengthening phase, attacks and validation tests were conducted to compare the security effectiveness of the baseline network to the improved network. Overall, this project has found that network segmentation and applying best cybersecurity practices like the principle of least privilege helps to dramatically decrease the risk of large swaths of cyberattacks and malicious reconnaissance efforts.

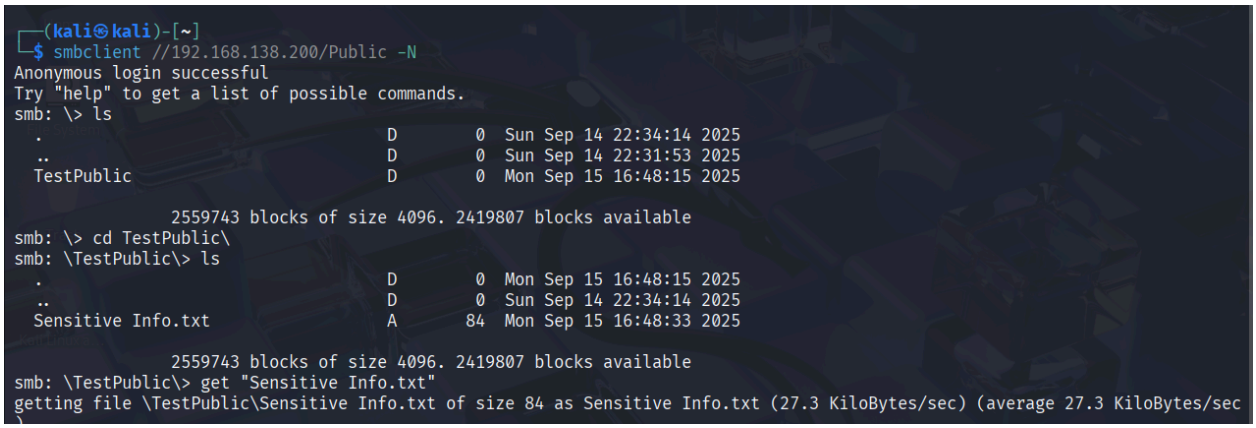
Part One : Implementation of Unsecure Flat Network

In order to fully evaluate the effectiveness of network segmentation, it is critical to set a baseline network environment to showcase the potential dangers of having an unsecure, unsegmented network. To achieve this, I used VMware as my hypervisor on my desktop machine. The first network design includes two Windows servers (Active Directory Server and File Server), a Windows 11 machine as a client, and a Kali Linux machine all within the same subnet. Below is a screenshot of the network design of the flat network.



One major issue with this network design is that there is free communication between all client and server machines. Although this might not seem like an issue, unrestricted communication between all machines is an incredibly large security risk, as there is zero filtering and no policy enforcement of traffic. This can escalate into an even larger issue when a machine on an unsegmented network is compromised. To showcase this issue, I performed an SMB share enumeration attack on the File Server using the Kali Linux machine (assuming the kali machine has been compromised). I started by performing an nmap scan on the entire network and was able to discover the File Server (192.168.138.200) has SMB (port 445). After further analysis

using enum4linux, a vulnerability is found within the SMB service that allows users to log into the SMB client using anonymous sessions. The anonymous sessions allow the user to essentially need no credentials to access and extract sensitive information from the SMB shares.

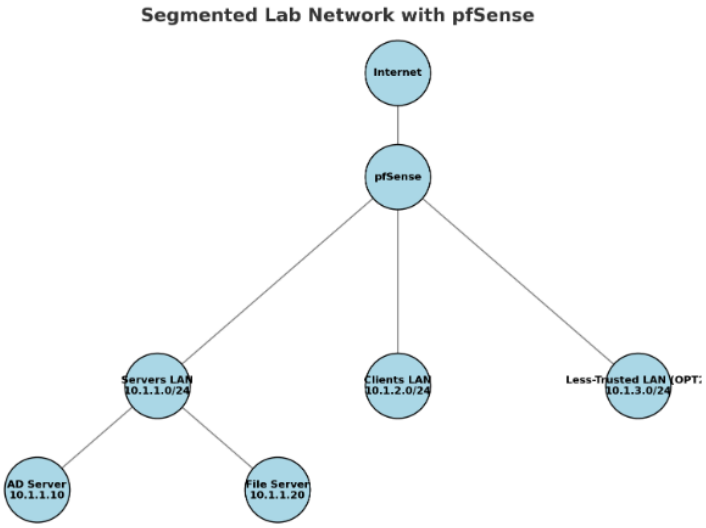


This sample attack highlights the critical issues that unsecure flat networks have. When a single machine is compromised on the network, any machine is at risk.

Part Two : Network Segmentation and pfSense Integration

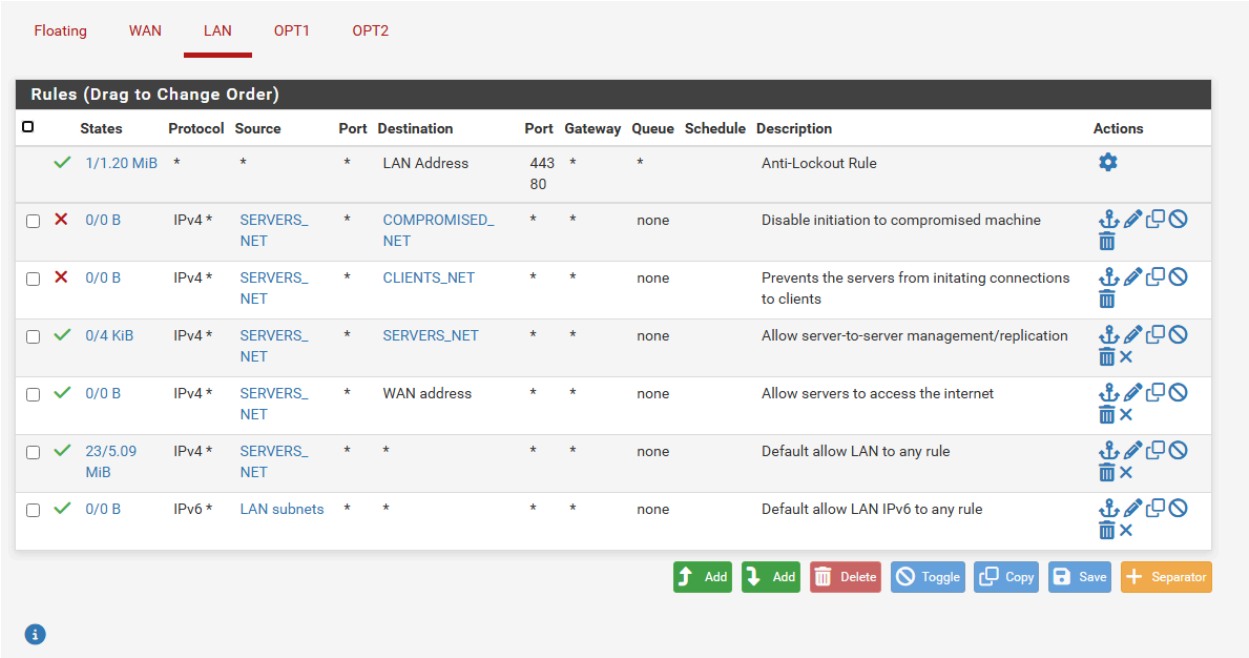
After identifying these critical issues with flat networks, there are a multitude of solutions to help reduce the risks of cyberattacks. One solution is network segmentation; network segmentation helps to isolate client machines from critical systems such as the file server and active directory server, preventing unauthorized access from reaching critical services. In order to achieve network segmentation in VMware, I split the network into 3 distinct subnets. The first subnet (10.1.1.0) was dedicated for the two servers within the network. The second subnet (10.1.2.0) was for normal client machines such as the Windows 11 virtual machine. These machines are treated as normal employee machines that would be found in departments such as HR or Sales. The third subnet (10.1.3.0) is for less trusted devices such as guests or contractors for a company and this is where the kali linux machine will reside. After designing how I wanted

my network to be divided, I needed a virtual router and firewall platform to allow my subnets to communicate with each other. In order to satisfy this, I decided on using pfSense for this project. pfSense is an open-source, FreeBSD-based firewall and router platform that includes an intuitive web GUI design. I chose pfSense as it is a platform I have used previously in a network security class and I enjoyed the large amounts of packages and customizability that it brings to the table. Below is the improved network design that showcases the 3 subnets created and the pfSense VM.

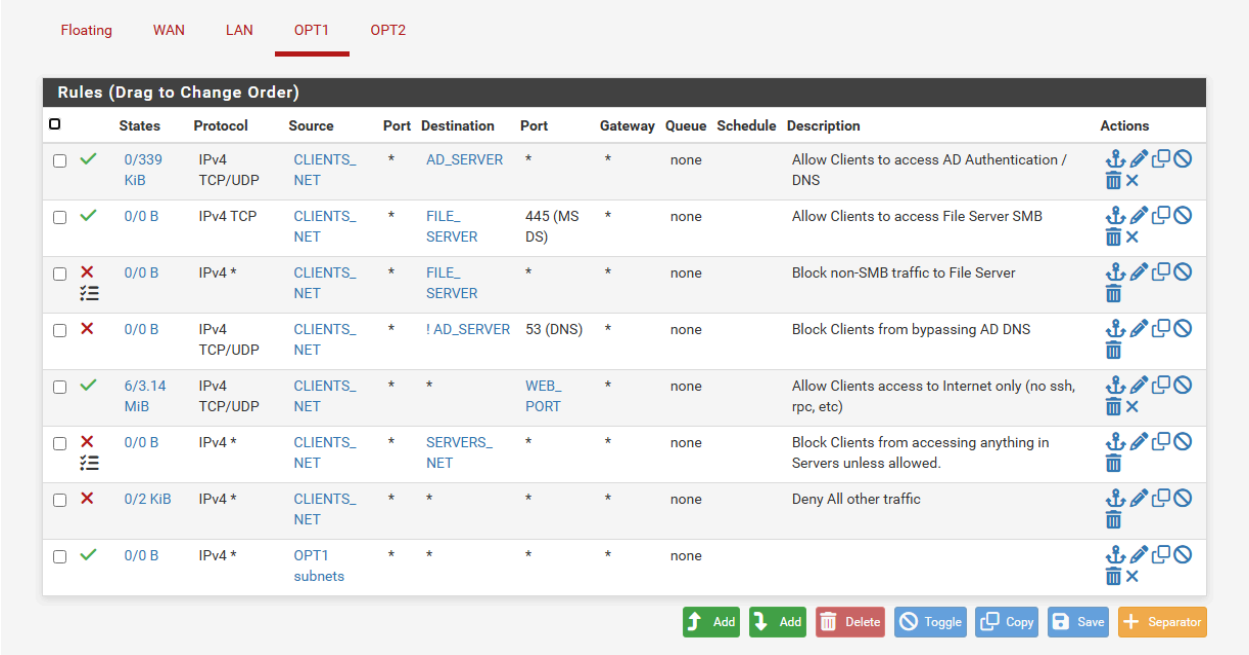


In order for the LANs to communicate with each other and the internet, their traffic will now need to be routed to pfSense. The next step in this process is to set up the pfSense firewall rules. When designing the firewall, there were a few best practices that I wanted to abide by in order to achieve a secure network. The first and most important practice is to enforce the principle of least privilege. The principle of least privilege states that you should only give devices only the minimum access and permissions needed to perform their required tasks. This concept helps to reduce the attack surface and mitigate damage that can occur if a machine is compromised. Another important concept for firewalls is the philosophy of deny by default, and explicitly allow. This concept is crucial for ensuring the principle of least privilege is upheld. With these

concepts in mind, I designed the server interface rules to allow for intra-server communication, allow the servers to access the internet (for updates), and explicitly deny initiation of connections to client machines.



For the client interface rules, I wanted the clients to be able to access the internet for general use, and allow clients to connect to specific Active Directory services, but block client machines from contacting servers for unrelated services.



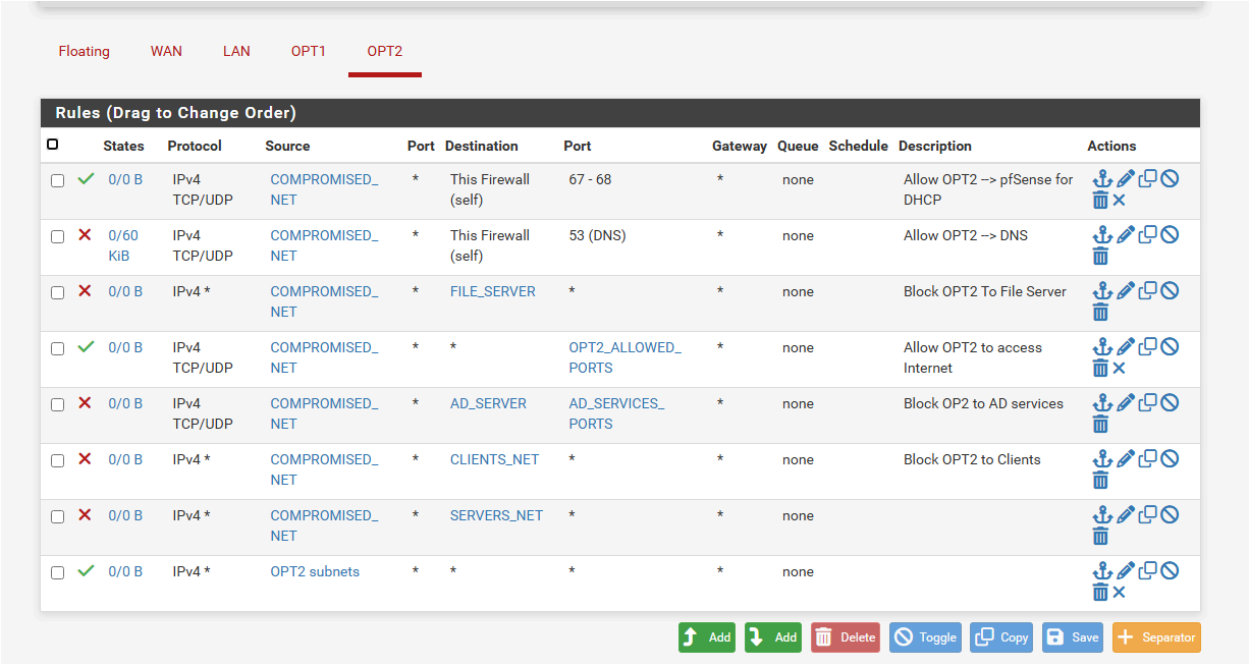
Floating WAN LAN OPT1 OPT2

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓	0/339 KiB	IPv4 TCP/UDP	CLIENTS_NET	*	AD_SERVER	*	*	none		Allow Clients to access AD Authentication / DNS	
✓	0/0 B	IPv4 TCP	CLIENTS_NET	*	FILE_SERVER	445 (MS DS)	*	none		Allow Clients to access File Server SMB	
✗	0/0 B	IPv4 *	CLIENTS_NET	*	FILE_SERVER	*	*	none		Block non-SMB traffic to File Server	
✗	0/0 B	IPv4 TCP/UDP	CLIENTS_NET	*	! AD_SERVER	53 (DNS)	*	none		Block Clients from bypassing AD DNS	
✓	6/3.14 MiB	IPv4 TCP/UDP	CLIENTS_NET	*	*	WEB_PORT	*	none		Allow Clients access to Internet only (no ssh, rpc, etc)	
✗	0/0 B	IPv4 *	CLIENTS_NET	*	SERVERS_NET	*	*	none		Block Clients from accessing anything in Servers unless allowed.	
✗	0/2 KiB	IPv4 *	CLIENTS_NET	*	*	*	*	none		Deny All other traffic	
✓	0/0 B	IPv4 *	OPT1 subnets	*	*	*	*	none			

Add Add Delete Toggle Copy Save Separator

For the Less Trusted Interface, I explicitly wanted it to have a restricted experience, allowing the LAN to only perform basic internet browsing but block critical services (like SMB) in the Servers LAN.



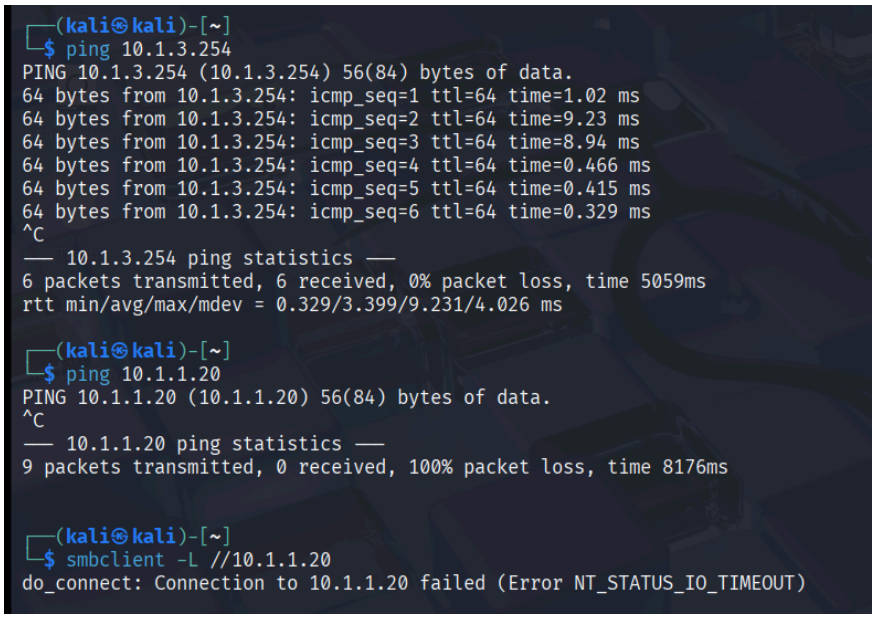
Floating WAN LAN OPT1 OPT2

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓	0/0 B	IPv4 TCP/UDP	COMPROMISED_NET	*	This Firewall (self)	67 - 68	*	none		Allow OPT2 –> pfSense for DHCP	
✗	0/60 KiB	IPv4 TCP/UDP	COMPROMISED_NET	*	This Firewall (self)	53 (DNS)	*	none		Allow OPT2 –> DNS	
✗	0/0 B	IPv4 *	COMPROMISED_NET	*	FILE_SERVER	*	*	none		Block OPT2 To File Server	
✓	0/0 B	IPv4 TCP/UDP	COMPROMISED_NET	*	*	OPT2_ALLOWED_PORTS	*	none		Allow OPT2 to access Internet	
✗	0/0 B	IPv4 TCP/UDP	COMPROMISED_NET	*	AD_SERVER	AD_SERVICES_PORTS	*	none		Block OP2 to AD services	
✗	0/0 B	IPv4 *	COMPROMISED_NET	*	CLIENTS_NET	*	*	none		Block OPT2 to Clients	
✗	0/0 B	IPv4 *	COMPROMISED_NET	*	SERVERS_NET	*	*	none			
✓	0/0 B	IPv4 *	OPT2 subnets	*	*	*	*	none			

Add Add Delete Toggle Copy Save Separator

After creating all the necessary rules for the firewall, I began doing validation tests to confirm that the firewall was working. I performed the same SMB enumeration attack shown in part one to test if it would be successful after the pfSense integration.

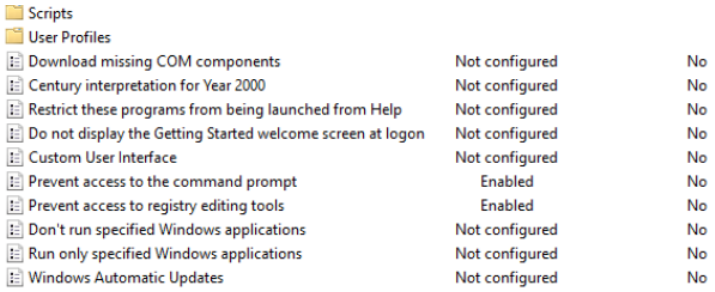


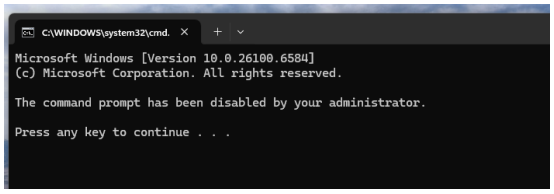
The SMB enumeration attack that was performed on the file server was unsuccessful due to the firewall rules, as the smbclient command shows a connection failure to the shares. In this part, segmentation and firewall integrations helped to not only create a more organized network environment, but also helped to prevent a cyberattack that previously worked on the flat network.

Part Three : AD Hardening and IDS Integration with Snort

After successfully integrating the pfSense firewall, I wanted to focus on strengthening the services within the virtual machines themselves, specifically Active Directory. Active Directory is Microsoft's centralized identity and access management system that stores critical information about users, computers, and network resources. When first setting up the Windows server and unsecure network, I started by allowing all settings to be set to default. However, the default

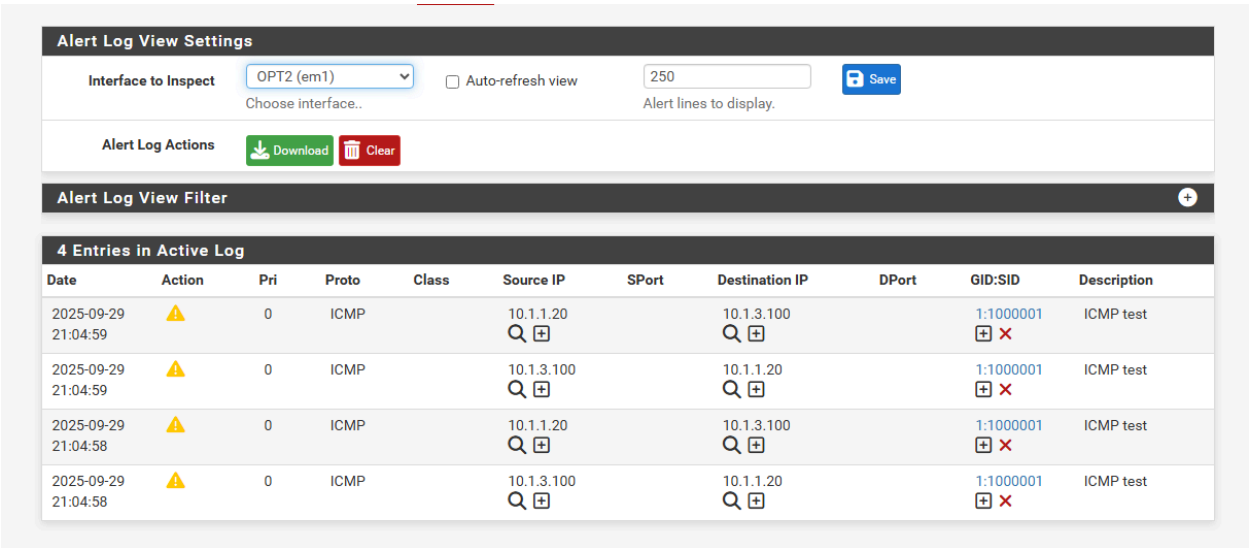
settings and domain structures are not enough for a secure AD environment. I first started hardening the environment by restructuring the organizational units (OU) within Active Directory, separating it by clients, servers, and the domain controllers. I then proceeded to create OU-specific security policies that align with my security goals for each group. For example, the client security policy would disable local administrator accounts, enforce windows defender to be always on, and disable tools such as control panel and the command prompt for non admins. Here is an example of configuring the command prompt to be disabled in the Group Policy management editor. This part was very interesting to me as there were so many customizable settings within the Group Policy editor to look and choose from.





After strengthening the security of my Active Directory services, I wanted to integrate an intrusion detection system to further enhance the security of the network. An intrusion detection system is a security tool that helps to monitor network traffic to detect signs of malicious behavior and policy violations. After researching a large variety of IDS's, I chose Snort for this project as it is a widely used open-source IDS that is incredibly easy to install on pfSense. It has a wide range of rulesets that is maintained by large corporations such as Cisco and is able to integrate well with SIEM tools such as Wazuh which will be discussed later in the report. In

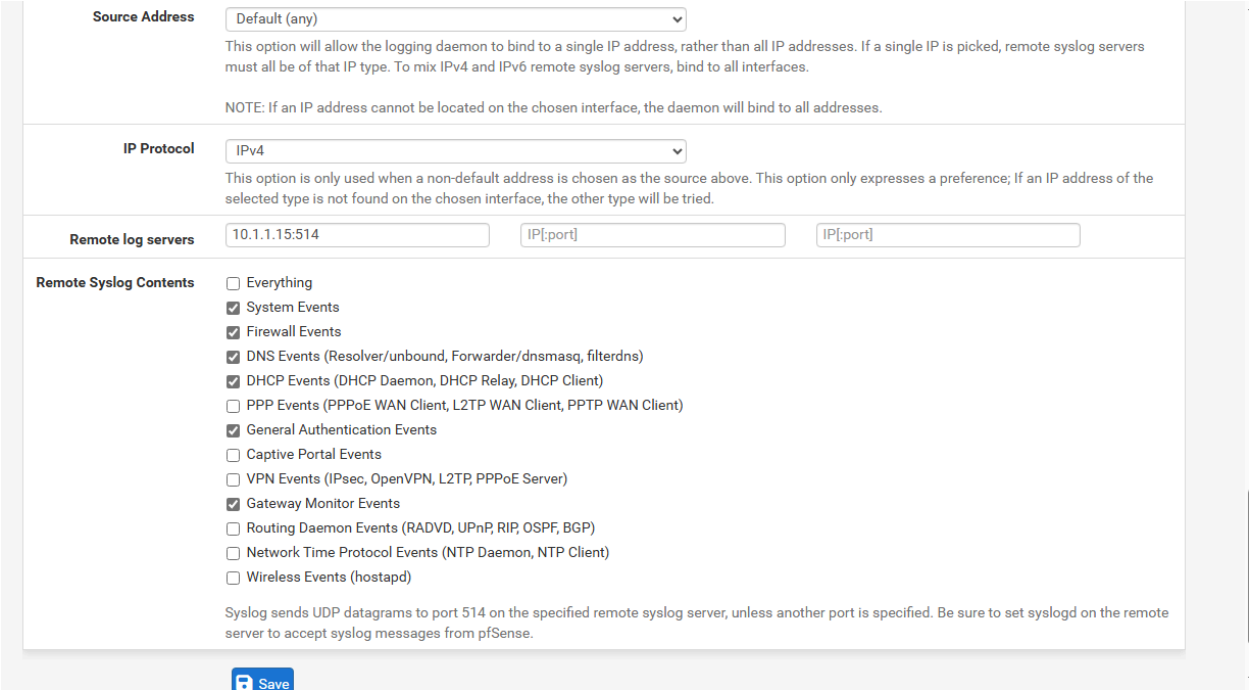
pfSense, I was able to easily install the Snort package and enable one of their many rulesets. To test if the alerts were working properly, I sent a few ping requests from the kali machine to the AD server. Although from the Kali Linux side it showed that the ping requests weren't reaching the AD server (due to the firewall rules established in part two), alerts were being still being generated by Snort.



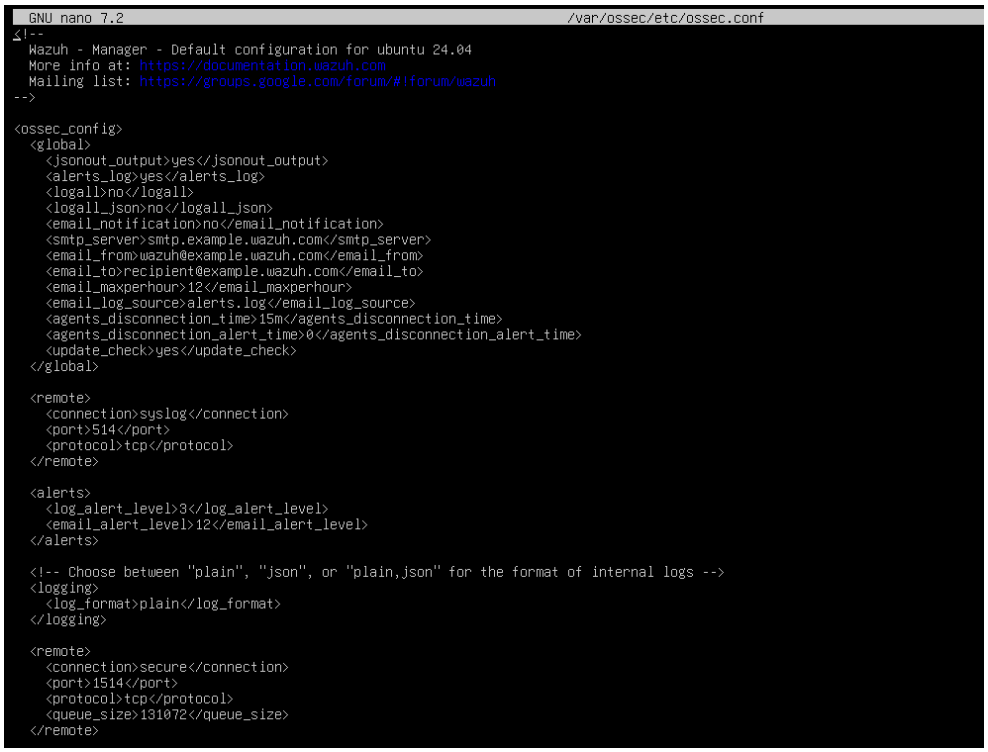
Part Four : SIEM Integration using Wazuh

Now that there are active alerts and logs being generated by pfSense and Snort, I wanted a monitoring tool to help aggregate, collect, and analyze all these new logs coming in. One way to achieve this centralization is by using a SIEM (Security Information and Event Management). The purpose of a SIEM is to give a full view of what's happening inside each machine and help detect threats that individual devices alone would miss. I chose Wazuh for my SIEM as it was free and has been known to integrate well with pfSense and Snort logs. After configuring and setting up a standalone Wazuh server on VMware, I needed to forward the logs from pfSense to

the server for analysis. In order to achieve this, I had to first configure the log forwarding system on pfSense to route to the remote log server (the Wazuh server).



On the Wazuh VM, I also needed to configure it to accept the log traffic coming from port 514.



Part Five : Powershell Script Automation

Nearing the end of the semester and satisfied with the state of the security of the lab environment, I wanted to expand on my limited powershell knowledge to help learn more about Active Directory which I enjoyed working on. I worked on a few scripts that helped to automate some of the mundane AD tasks that I was needing to perform. The first script I created was CreateNewUserFromCSV, which allowed me to create new AD users based on the given structure “Name,GivenName,Surname,Username,UserPrincipalName,Password,OU”. The second script built on top of the first script and allowed the user to add the newly created user to an AD group of their choice. Although this was the shortest section of my project I worked on, it was very interesting to see how powerful Powershell can be for automating active directory tasks.

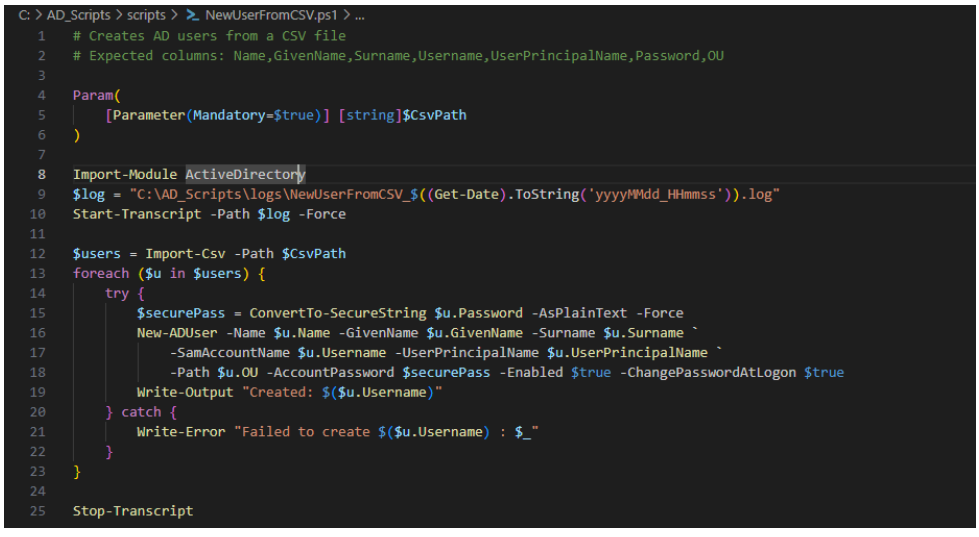
Challenges and Conclusion

Over the course of the project, I encountered many different challenges and difficulties when creating this lab environment. The first and most pressing issue was restructuring the flat network to become a secure segmented network. When dividing the network into 3 different LANs, I needed to completely reconfigure all networking information such as IP address, DNS server, DHCP server, etc. On top of this, I had to learn how interface setups worked for pfSense and how to get the LANs to receive internet. It took countless hours reading the pfSense documentation and troubleshooting for me to receive internet and LAN communication for all three interfaces. Another difficulty I had during this project was understanding the intricacies of Active Directory and how groups, organizational units, and security policies all interact with each other.

Overall, this capstone project provided me incredibly valuable hands-on experience in designing and securing a virtual network using a variety of different tools and technologies. Starting with a vulnerable flat network to highlight the dangers and issues with this network design, I was able to transform the environment into a significantly more secure and manageable network that could withstand attacks the flat network couldn't. Each phase of the project development cycle helped to showcase how segmentation, proper access control, traffic monitoring, and scripting automation help to reduce attack surfaces and overall security. This project was by far the most enjoyable and interactive learning experience I have had during my time at UNG.

Extra Screenshots

Below are a few extra screenshots of the lab network that I felt was important to include. For more screenshots and diagrams, I included all 5 write-ups that document large parts of my process to securing and hardening the virtual network that contain lots of screenshots of my development process.



NewUserFromCSV source code

```powershell
$SamAccountName = Read-Host "Enter the user's SamAccountName"

# Validate user exists
$User = Get-ADUser -Filter "SamAccountName -eq '$SamAccountName'" -ErrorAction SilentlyContinue
if (-not $User) {
    Write-Host "❌ User '$SamAccountName' not found in Active Directory." -ForegroundColor Red
    exit
}

Write-Host "`n✅ Found user: $($User.Name)`n" -ForegroundColor Green

# Option 1: Display group list (filtered to security groups)
$Groups = Get-ADGroup -Filter * | Sort-Object Name | Select-Object -ExpandProperty Name

# Show top 10 to avoid clutter
Write-Host "Top 10 groups in AD (you can still enter a custom name):" -ForegroundColor Cyan
$Groups | Select-Object -First 10 | ForEach-Object { Write-Host " - $_" }

# Ask user to type group name
$GroupName = Read-Host "`nEnter the group name to add the user to"

# Validate group exists
$Group = Get-ADGroup -Filter "Name -eq '$GroupName'" -ErrorAction SilentlyContinue
if (-not $Group) {
    Write-Host "Group '$GroupName' not found in Active Directory." -ForegroundColor Red
    exit
}

# Add the user to the group
try {
    Add-ADGroupMember -Identity $GroupName -Members $SamAccountName -ErrorAction Stop
    Write-Host "`n Successfully added '$SamAccountName' to '$GroupName'." -ForegroundColor Green
}
catch {
```

Add-UserToGroup source code

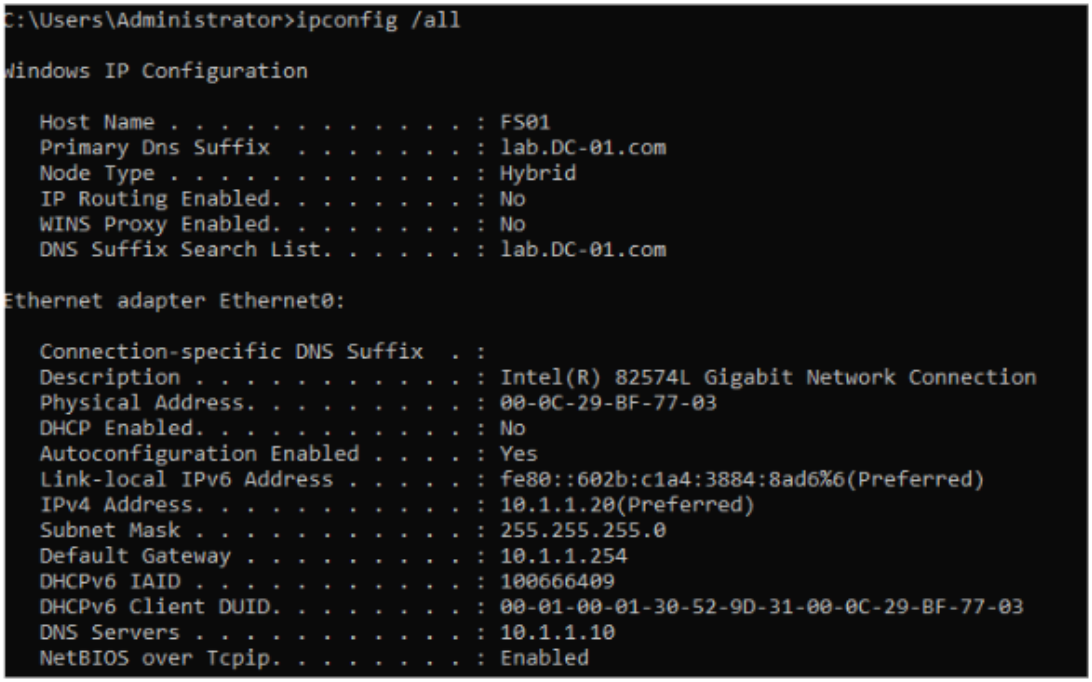
AD Server (10.1.1.10):

- Static IP, gateway set to pfSense (10.1.1.254), DNS = self (127.0.0.1).

```
Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) 82574L Gigabit Network Connection
   Physical Address. . . . . . . . . : 00-0C-29-CB-52-5B
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::2db8:b2a4:575c:802e%12(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.1.1.10(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.1.1.254
   DHCPv6 IAID . . . . . . . . . . . : 100666409
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-30-07-6D-08-00-0C-29-CB-52-5B
   DNS Servers . . . . . . . . . . . : ::1
                                       10.1.1.10
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

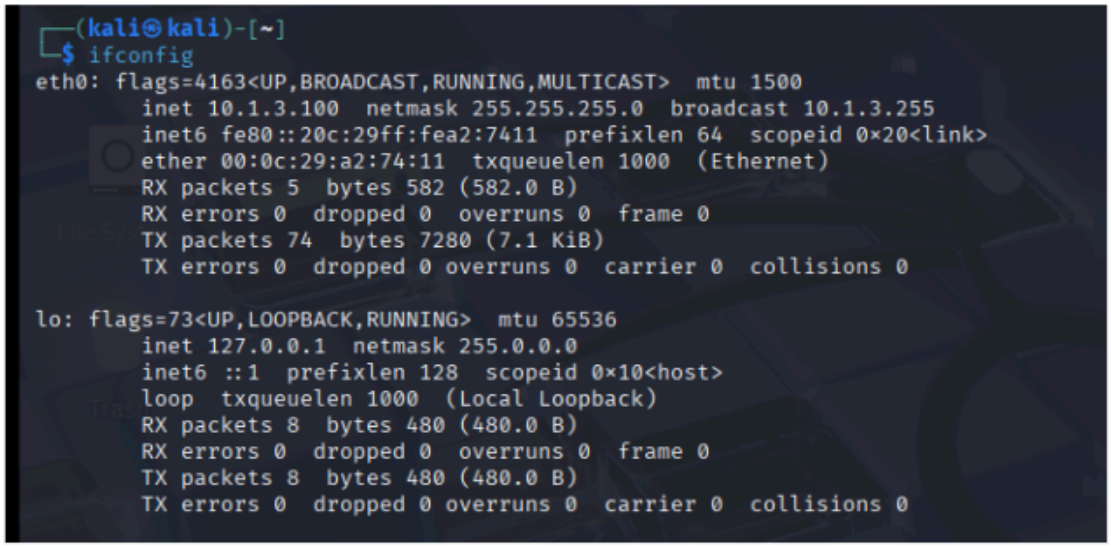
File Server (10.1.1.20):

- Static IP, gateway set to pfSense (10.1.1.254), DNS = AD (10.1.1.10).



```
C:\Users\Administrator>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : FS01
   Primary Dns Suffix  . . . . . . . : lab.DC-01.com
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : lab.DC-01.com

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) 82574L Gigabit Network Connection
   Physical Address. . . . . . . . . : 00-0C-29-BF-77-03
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::602b:c1a4:3884:8ad6%6(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.1.1.20(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.1.1.254
   DHCPv6 IAID . . . . . . . . . . . : 100666409
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-30-52-9D-31-00-0C-29-BF-77-03
   DNS Servers . . . . . . . . . . . : 10.1.1.10
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Kali Linux (DHCP – 10.1.3.101):



```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.3.100  netmask 255.255.255.0  broadcast 10.1.3.255
        inet6 fe80::20c:29ff:fea2:7411  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:a2:74:11  txqueuelen 1000  (Ethernet)
        RX packets 5  bytes 582 (582.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 74  bytes 7280 (7.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 480 (480.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	9/30/2025 5:29:12 PM	Microsoft Windows sec...	4634	Logoff
Audit Success	9/30/2025 5:29:12 PM	Microsoft Windows sec...	4624	Logon
Audit Success	9/30/2025 5:28:32 PM	Microsoft Windows sec...	4634	Logoff
Audit Success	9/30/2025 5:28:32 PM	Microsoft Windows sec...	4634	Logoff
Audit Success	9/30/2025 5:28:29 PM	Microsoft Windows sec...	4634	Logoff
Audit Success	9/30/2025 5:28:29 PM	Microsoft Windows sec...	4634	Logoff
Audit Success	9/30/2025 5:28:21 PM	Microsoft Windows sec...	5140	File Share
Audit Success	9/30/2025 5:28:21 PM	Microsoft Windows sec...	4624	Logon
Audit Success	9/30/2025 5:28:12 PM	Microsoft Windows sec...	4634	Logoff
Audit Success	9/30/2025 5:28:12 PM	Microsoft Windows sec...	4624	Logon
Audit Failure	9/30/2025 5:27:43 PM	Microsoft Windows sec...	4673	Sensitive Privilege Use
Audit Failure	9/30/2025 5:27:43 PM	Microsoft Windows sec...	4673	Sensitive Privilege Use
Audit Failure	9/30/2025 5:27:43 PM	Microsoft Windows sec...	4673	Sensitive Privilege Use
Audit Failure	9/30/2025 5:27:43 PM	Microsoft Windows sec...	4673	Sensitive Privilege Use
Audit Failure	9/30/2025 5:27:43 PM	Microsoft Windows sec...	4673	Sensitive Privilege Use
Audit Failure	9/30/2025 5:27:43 PM	Microsoft Windows sec...	4673	Sensitive Privilege Use
Audit Failure	9/30/2025 5:27:43 PM	Microsoft Windows sec...	4673	Sensitive Privilege Use
Audit Failure	9/30/2025 5:27:43 PM	Microsoft Windows sec...	4673	Sensitive Privilege Use

Event Logs found in Active Directory Event Manager

References

https://learn.microsoft.com/en-us/windows-server/identity/identity-and-access

https://www.snort.org/documents

https://learn.microsoft.com/en-us/powershell/

https://docs.netgate.com/pfsense/en/latest/

https://documentation.wazuh.com/current/index.html

https://www.vmware.com/resources/resource-center

https://learn.microsoft.com/en-us/windows-server/

https://www.kali.org/docs/