Simulated Cloud Environment for Network Defense and Monitoring

Purpose

- Gain hands-on experience with provisioning machines and installing operating systems.
- Configure network interfaces, implement DHCP ranges, and apply subnetting to build a segmented environment.
- Enforce security through firewall rules, IDS/IPS, and centralized monitoring with SIEM.
- Work with Active Directory to apply domain and group policy management.
  Strengthen troubleshooting skills by diagnosing and resolving connectivity issues.

Network Layout/Topology

Device Table:

Host	Private IP	Interface	Purpose
Home Router	192.168.1.xxx (masked for privacy)	WAN	Forwards traffic from the internal network to the internet
Switch	192.168.2.xxx (masked for privacy)	WAN	Allows devices on a physical local network to forward traffic to one another based on MAC addresses. For lab purposes, it forwards the traffic from our local environment to the home router
Physical Desktop	192.168.2.xxx (masked for privacy)	WAN	The primary host allocates storage for machines, and the hypervisor is controlled through this machine. This machine also bridges the connection from the host's network and the WAN, so that our internal virtual network has internet connectivity
WAN	192.168.2.xxx (masked for privacy)	WAN	The network interface that allows our internal network to send and receive traffic. Bridged connection with the host
DMZ	192.168.20.1	DMZ	Restricted access zone for web server, public-facing
Web Server	192.168.20.101	DMZ	Hosts a resource page for file transfers over HTTPS/TLS
LAN	192.168.10.1	LAN	Local Area Network, heavily restricted, internal access only

Windows Client	192.168.10.101	LAN	Mock client, only here so we have something to enforce AD policy upon
Domain Controller	192.168.10.10	LAN	Admin, DHCP server, Domain Controller, Active Directory Domain Services, etc., the mothership
SIEM	192.168.10.50	LAN	Server for centralized log monitoring of all endpoints

Topology:

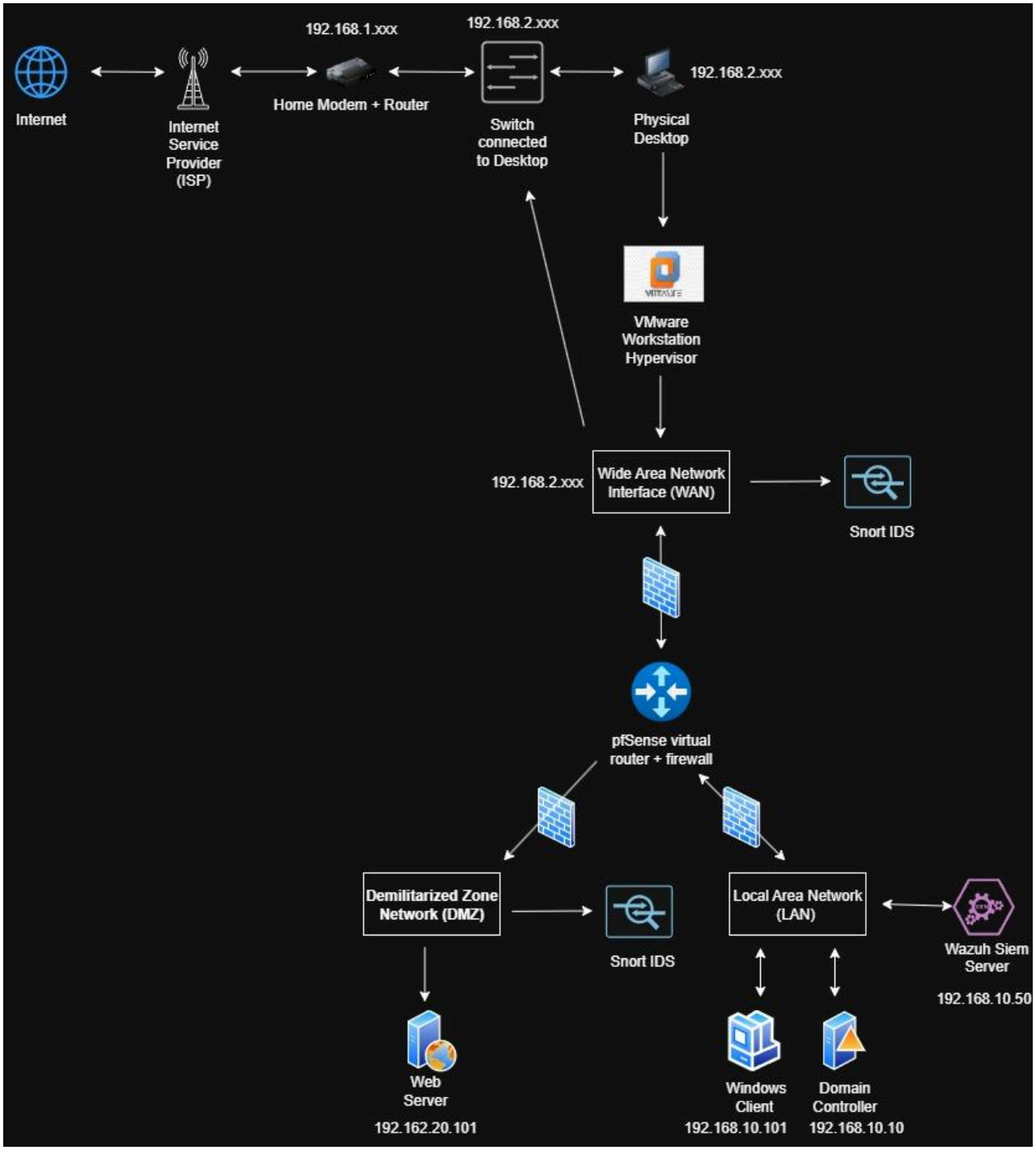


Figure: Network Topology of Environment

Instantiations and Configs

pfSense:

What is pfSense? pfSense is an open-source operating system that serves as a firewall and routing software. In this lab, I used pfSense to create three virtual NICs to host my WAN, LAN, and DMZ network interfaces.

How to install? Download ISO image from pfSense[.]org, attach to VM, then follow instructions.

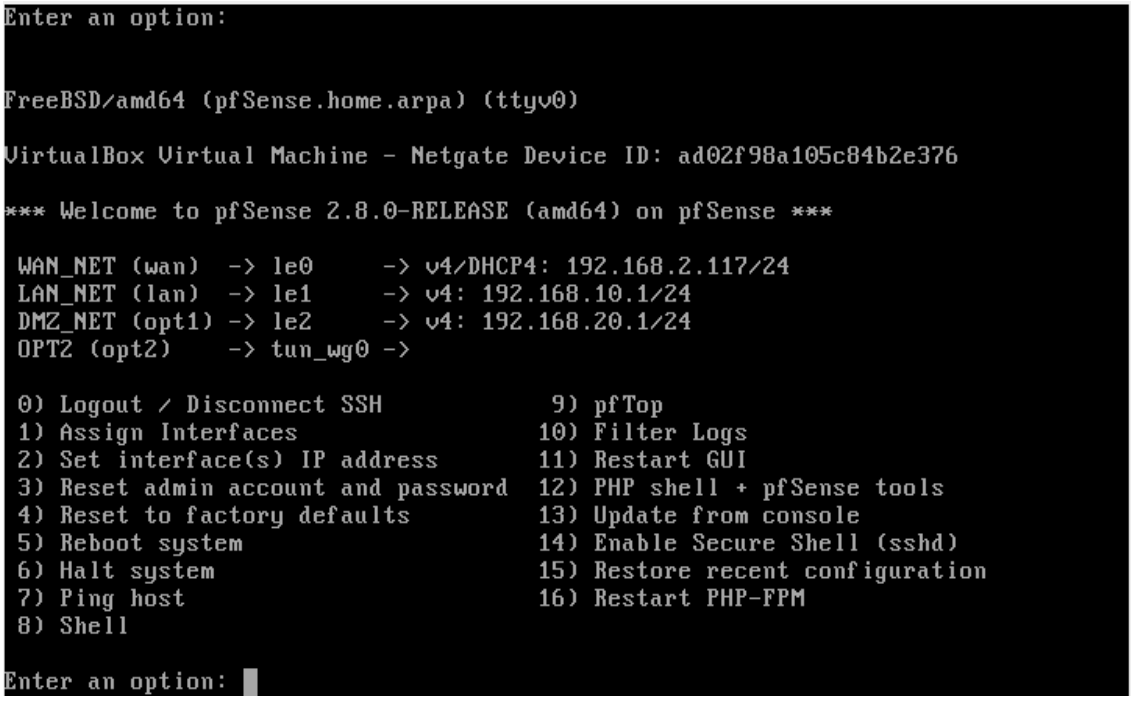


Figure: pfSense shell

As you can see, my WAN is assigned a private IP from my desktop's private network via DHCP. I bridged the network adapter within VirtualBox network settings to achieve this. My LAN and DMZ are both internal networks that have been statically assigned their respective IP addresses.

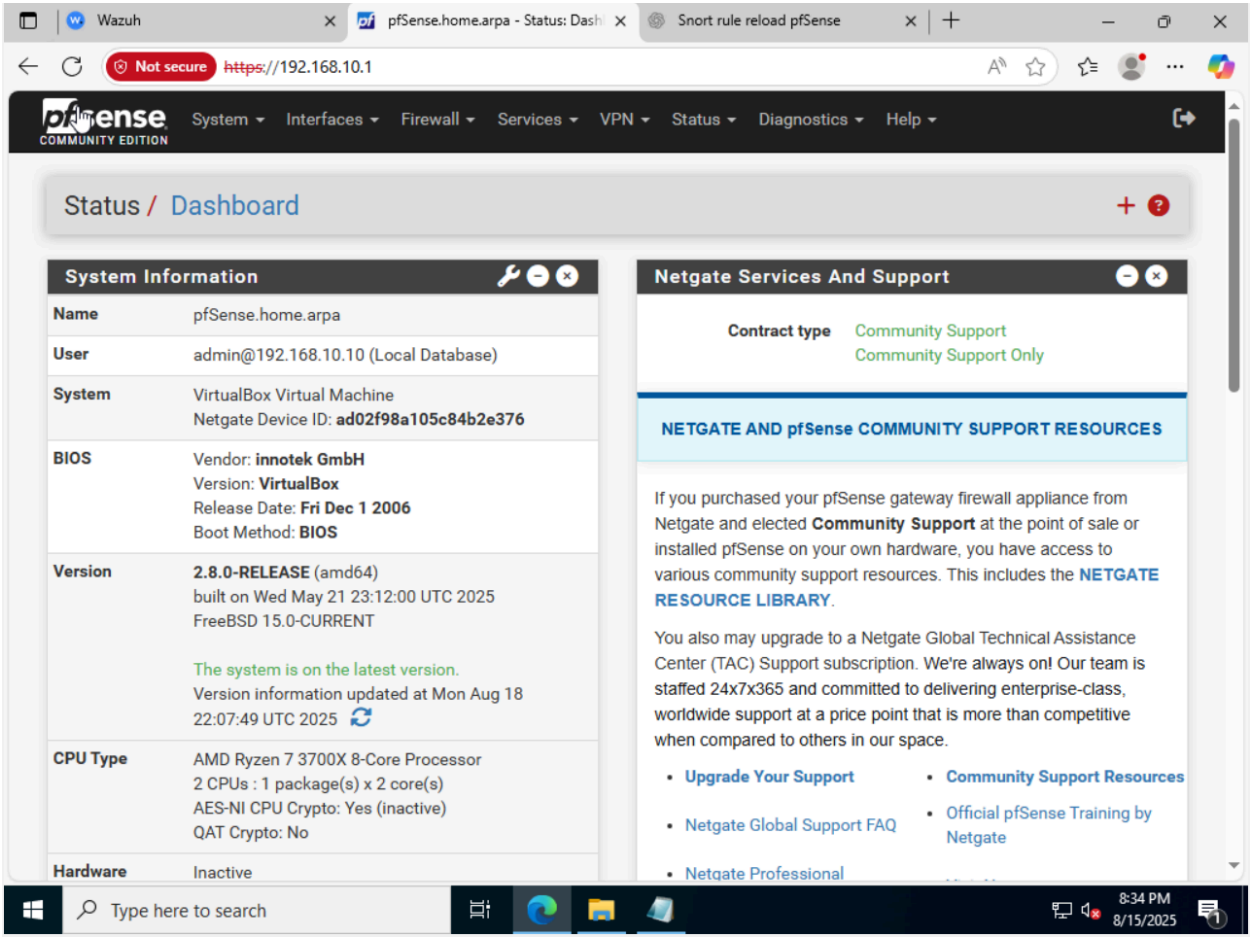


Figure: pfSense web interface dashboard

DMZ Web Server:

What is a DMZ? DMZ, or Demilitarized Zone, is a network that acts as a buffer zone between an external network (Internet/WAN) and a private, internal network (LAN). These are utilized to segment and isolate public-facing services (like web or email servers) from internal hosts and systems.

How to Instantiate a Web Server (Apache2)? Use apt to install Apache2 package and start and enable the service. Then, to deploy the website, find the /html/ directory and place your index.html and other application files ther,e and reload the service.

Figure: Apache Web server being spun up on port 8443

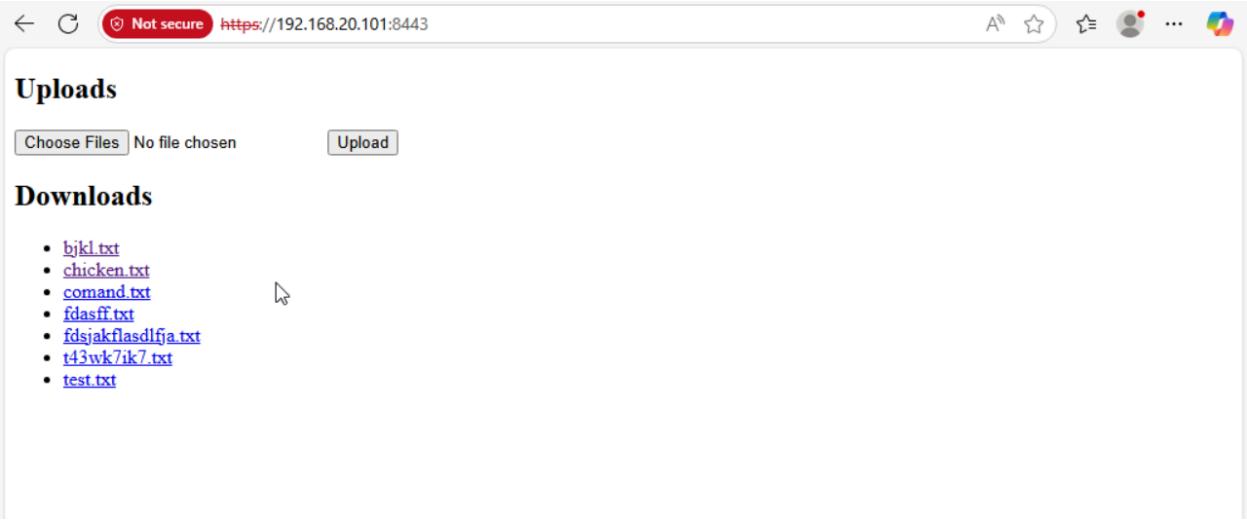


Figure: Web Page with Uploads option and Downloads option

Domain Controller and Active Directory:

What is AD or Active Directory? Active Directory is a way for an administrator to manage the users, groups, services, devices, and permissions on a network. It is the hub for authentication, authorization, and resource sharing/management in Windows environments. The DC, or Domain Controller, acts as the server that physically manages resources and services on a network. It is the central point where identity and network access are managed.

How do you implement AD? Once you've installed Windows Server 2022 (or a similar version), go to Server Manager, add roles and features, and go through the wizard's instructions.

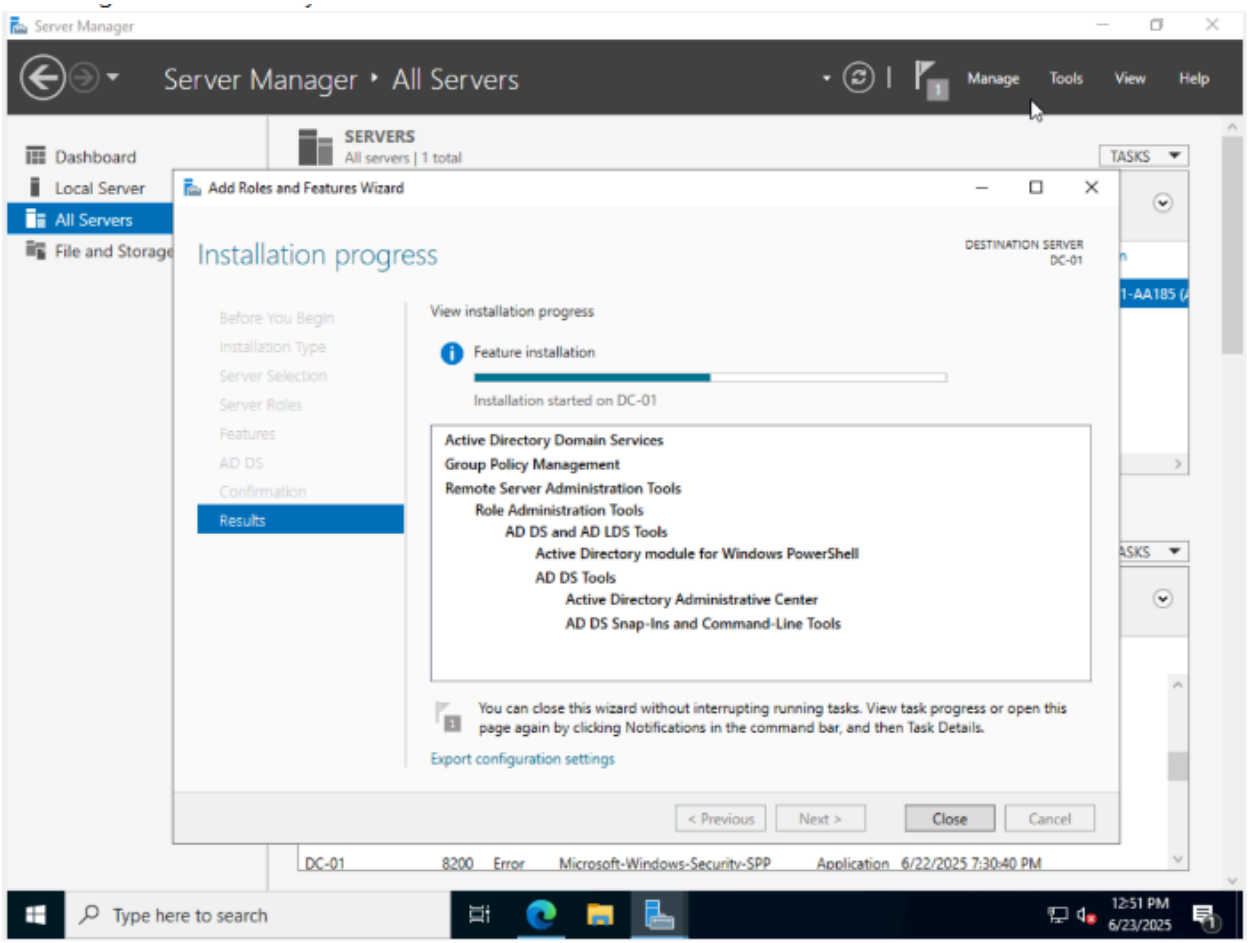


Figure: Instantiating Active Directory

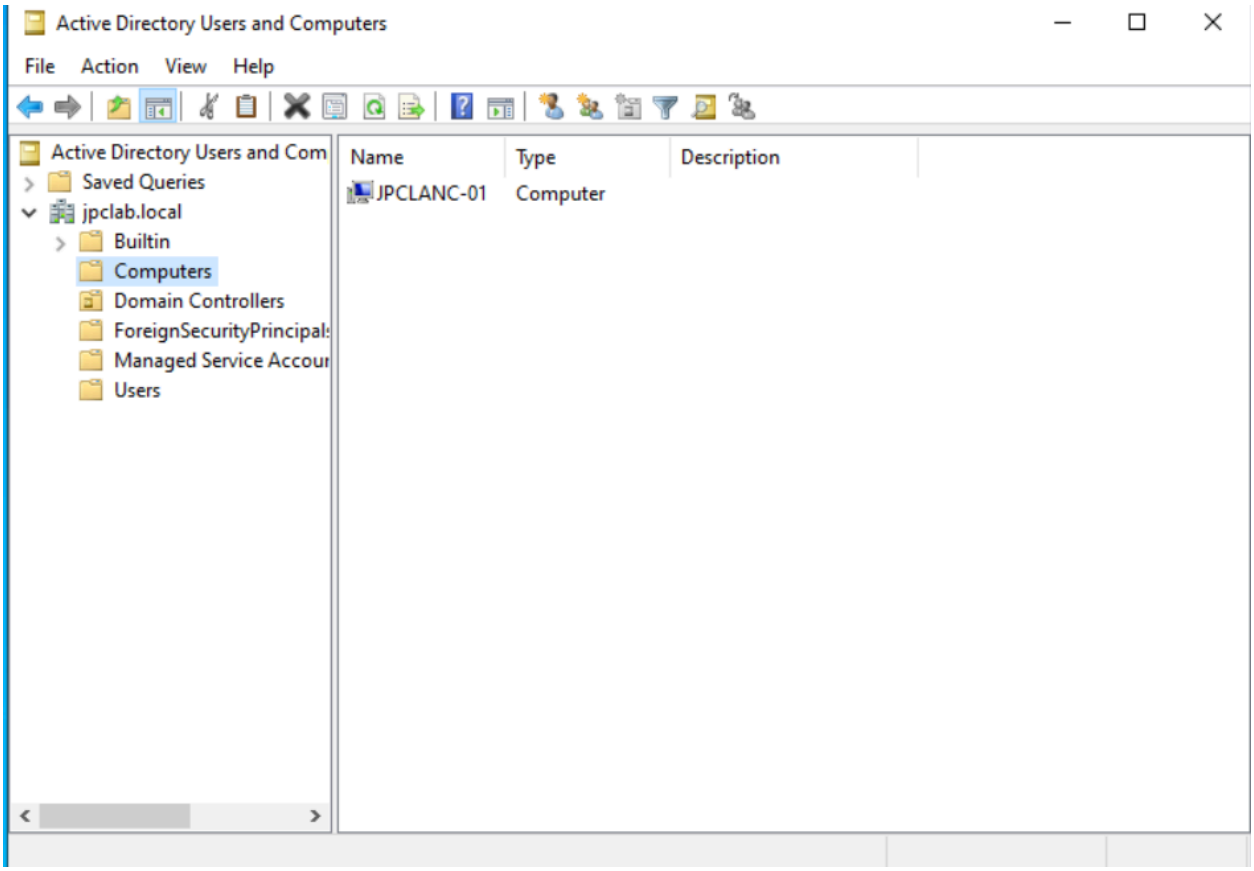


Figure: My Windows LAN client in my jpclab.local domain

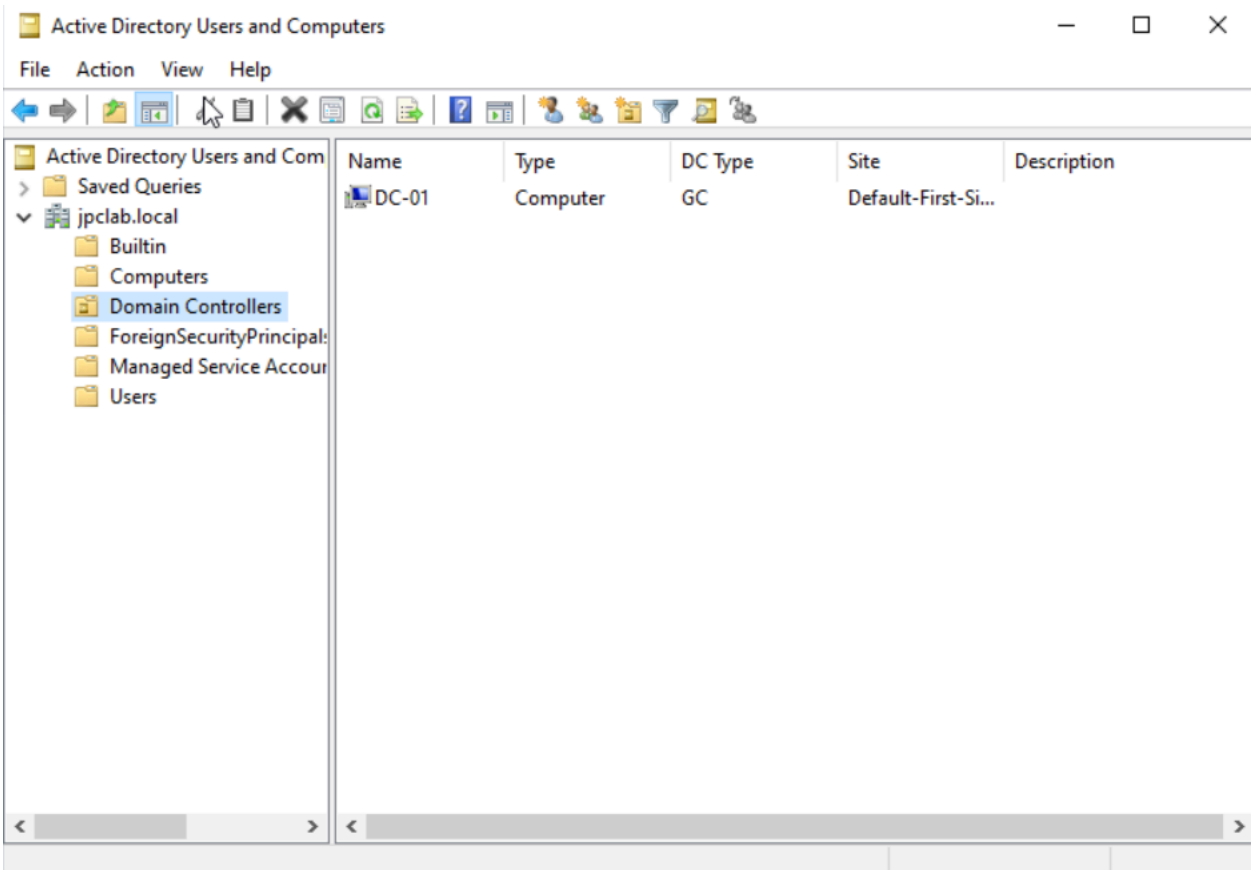


Figure: My Windows DC in my jpclab.local domain

Snort Intrusion Detection and Prevention System:

What is IDPS? An IDPS or Intrusion Detection and Prevention System is a security solution that monitors network traffic and either flags or prevents perceived threats from interacting with an environment. Snort is an open source anomaly-based IDPS that supports heuristic-based detections too.

How to install? Snort is a built-in package within pfSense. Go to System > Package Manager and find Snort. Once you install, go to Services > Snort and add Snort to the NICs you wish to protect.

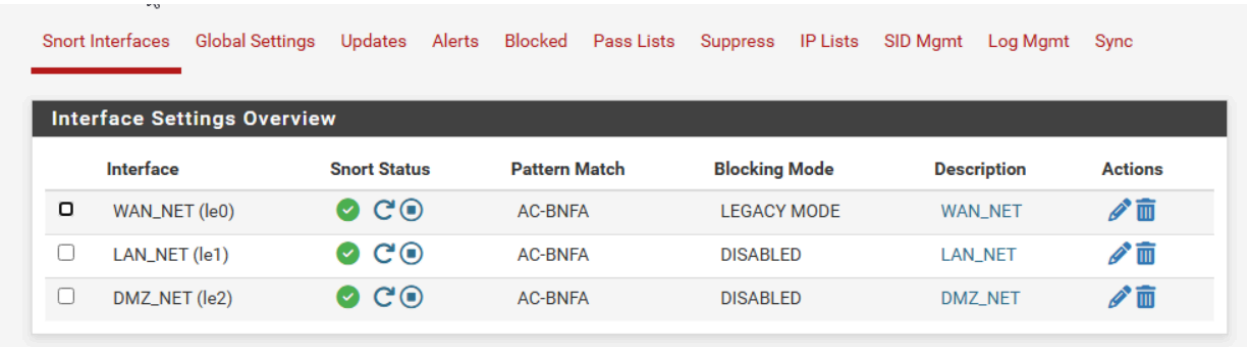


Figure: Snort software deployed on all interfaces, only actually preventing perceived malicious traffic on the WAN interface

Wazuh Security and Information Event Manager and XDR Platform:

What is a SIEM? A SIEM or Security Information and Event Manager is a security solution designed to aid in threat detection through log aggregation. SIEMs take log data from multiple sources, including applications, network devices, other security tools, etc., and aggregate them to a centralized server or dashboard. The data collected is then analyzed for anomalies or malicious patterns using a combination of predefined rules, machine learning, and behavioral analytics. Wazuh is a SIEM/XDR platform that allows an administrator to deploy an agent on endpoints they would like to monitor. The platform provides free endpoint security, insight into threat intelligence, as well as other robust features like ensuring compliance through proper endpoint configuration and cloud integration support.

How to Install? Visit Wazuh[.]com and follow the instructions for implementing the central components: Wazuh indexer, Wazuh server, and Wazuh dashboard. Once your manager is up, follow the instructions for installing agents onthe endpoints you wish to monitor.

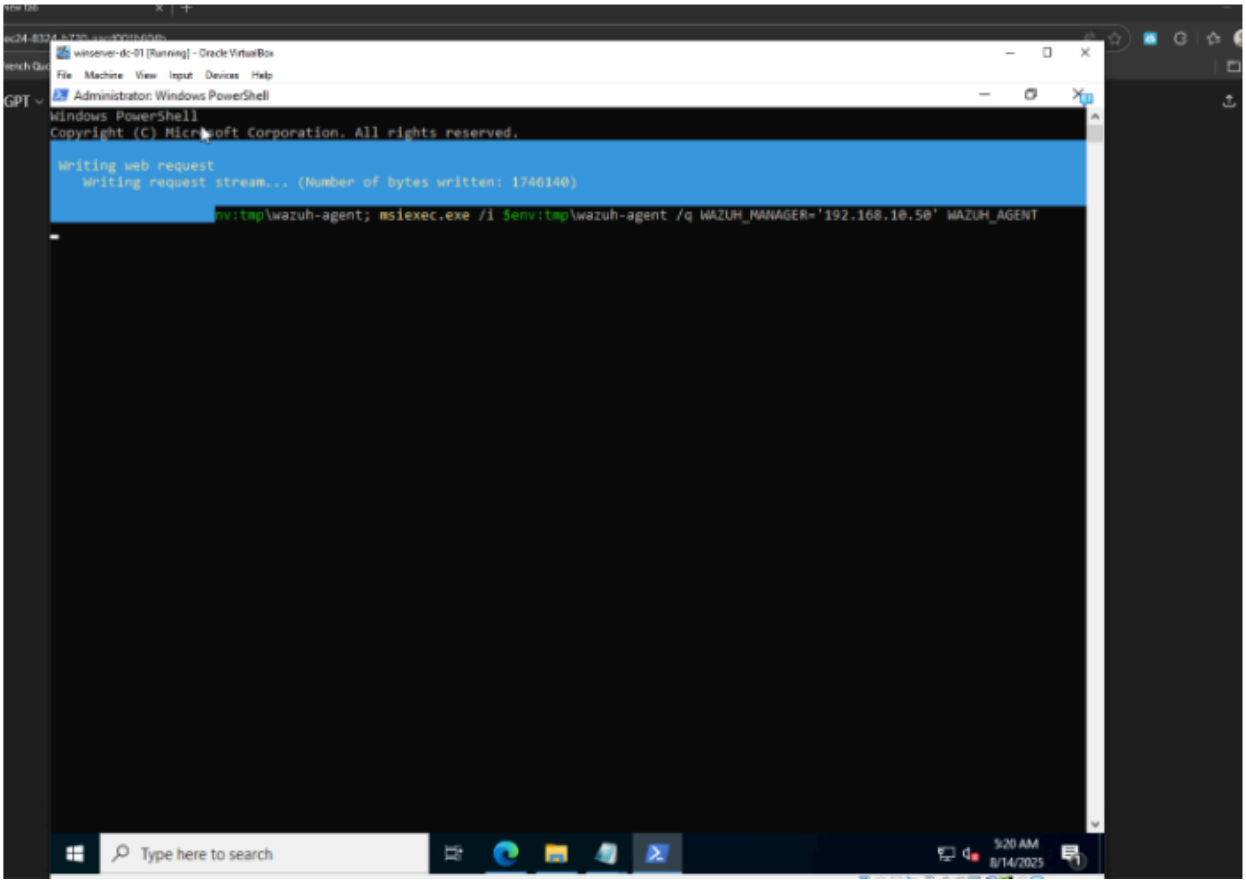


Figure: Deploying Wazuh Agent on DC

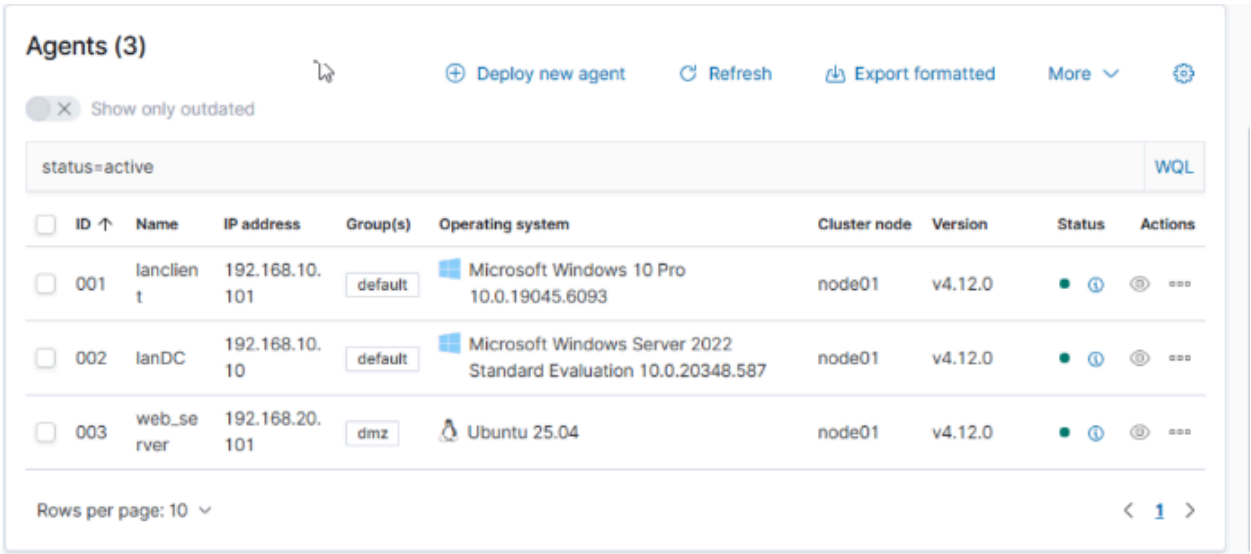


Figure: all agents deployed in the environment for our three endpoints (DC, Web Server, and LAN client)

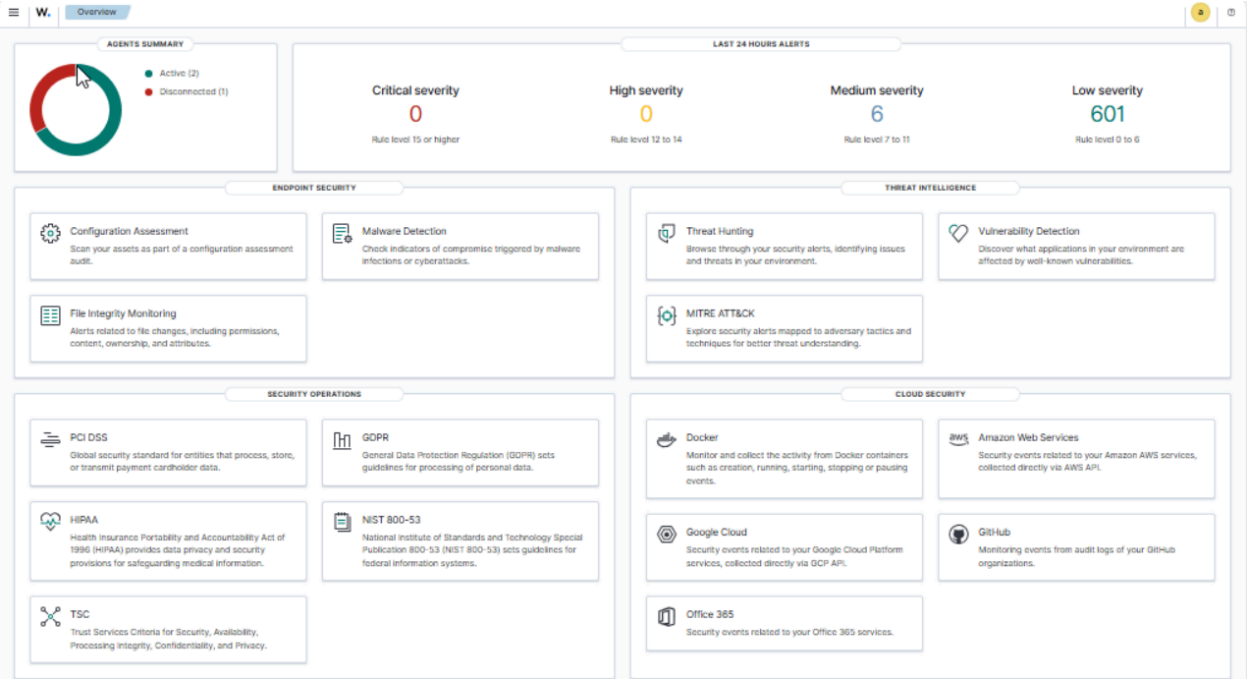


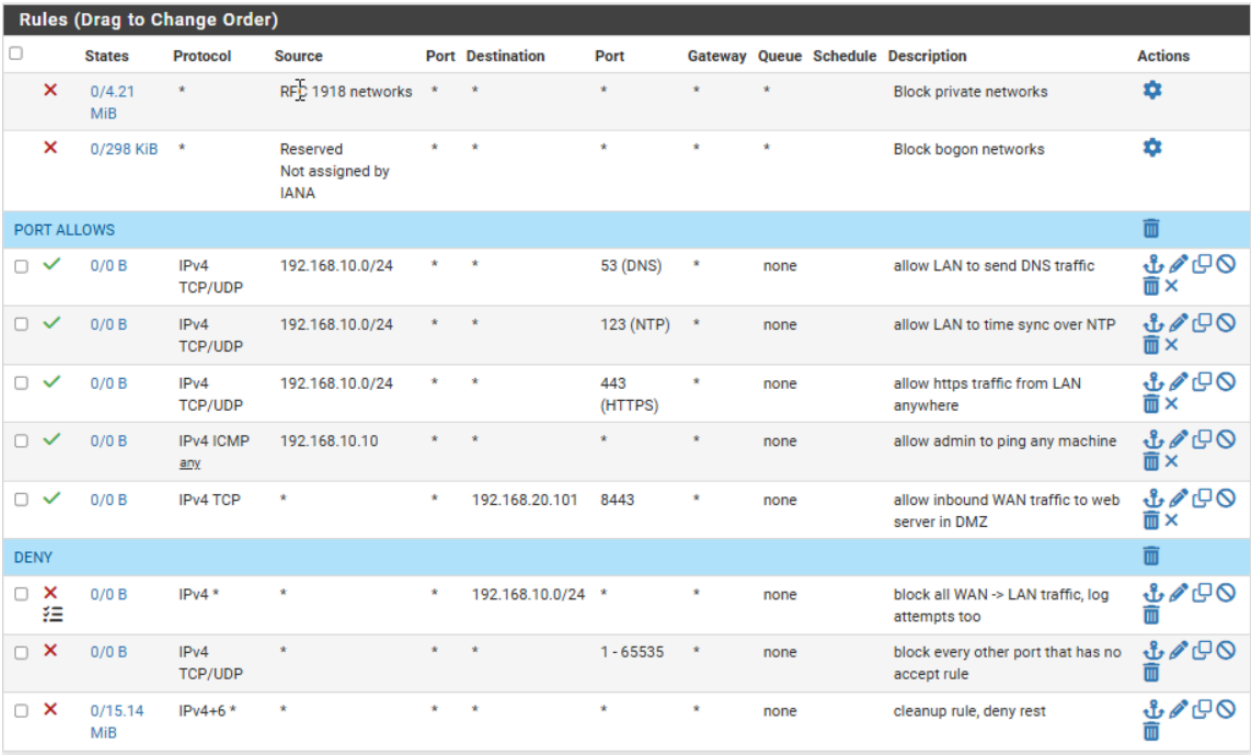
Figure: Main Wazuh Dashboard

Security Implementations

pfSense Firewall Rules:

Not only does pfSense allow us to set up network interfaces for communication, but it also allows us to predefine what traffic should be allowed to enter and exit these interfaces via Firewalls.

WAN



Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✕	0/4.21 MiB	*	RFC 1918 networks	*	*	*	*	*		Block private networks	
✕	0/298 KiB	*	Reserved Not assigned by IANA	*	*	*	*	*		Block bogon networks	
PORT ALLOWS											
✓	0/0 B	IPv4 TCP/UDP	192.168.10.0/24	*	*	53 (DNS)	*	none		allow LAN to send DNS traffic	
✓	0/0 B	IPv4 TCP/UDP	192.168.10.0/24	*	*	123 (NTP)	*	none		allow LAN to time sync over NTP	
✓	0/0 B	IPv4 TCP/UDP	192.168.10.0/24	*	*	443 (HTTPS)	*	none		allow https traffic from LAN anywhere	
✓	0/0 B	IPv4 ICMP any	192.168.10.10	*	*	*	*	none		allow admin to ping any machine	
✓	0/0 B	IPv4 TCP	*	*	192.168.20.101	8443	*	none		allow inbound WAN traffic to web server in DMZ	
DENY											
✕	0/0 B	IPv4 *	*	*	192.168.10.0/24	*	*	none		block all WAN -> LAN traffic, log attempts too	
✕	0/0 B	IPv4 TCP/UDP	*	*	*	1 - 65535	*	none		block every other port that has no accept rule	
✕	0/15.14 MiB	IPv4+6 *	*	*	*	*	*	none		cleanup rule, deny rest	

Figure: Firewall rule table for WAN Interface

What does a WAN do? A WAN acts as the intermediary for communications between our internal network and other internal networks. For example, the biggest WAN known to man is the internet, which connects a bunch of internal networks into one big network where communication is seamless.

Rules for WAN:

- By Default, pfSense blocks private networks from communicating with my WAN because everybody coming in from the internet should have a public IP
- Block Bogon networks or bogus networks. Traffic from hosts with an IP of 999.999.999.999, for example, is blocked
- Rules 3-7 are ports I want to allow for my LAN., I want them to be able to send out DNS, NTP, and HTTPS traffic, as well as have my admin be able to ping anything for network troubleshooting (I can accept that risk)
- Block WAN traffic from accessing my LAN and log attempts. I should never have traffic from the internet initiating connections with hosts in my LAN.
    - Security risks, exposure of internal services, unsolicited access to hosts, RCE, Brute Force, Lateral Movement, etc.
- Blocking every other port that has no accept rule unnecessarily increases the attack surface.
- The cleanup rule ensures that the rest of the traffic that doesn't match an accept rule is dropped.

LAN

✓	1/8.25 MiB	*	*	*	LAN_NET Address	443 80	*	*	Anti-Lockout Rule	⚙
✓	0/0 B	IPv4 TCP/UDP	192.168.10.10	*	192.168.10.0/24	22 (SSH)	*	none	allow DC admin to SSH to LAN	
✓	0/388 KiB	IPv4 TCP/UDP	192.168.10.10	*	192.168.20.0/24	22 (SSH)	*	none	allow DC admin to SSH to DMZ	
✓	0/424 KiB	IPv4 TCP/UDP	192.168.10.0/24	*	*	53 (DNS)	*	none	allow LAN to send DNS traffic	
✓	0/0 B	IPv4 TCP/UDP	192.168.10.0/24	*	192.168.10.0/24	88	*	none	allow Kerberos for AD authentication	
✓	0/4 KiB	IPv4 TCP/UDP	192.168.10.0/24	*	*	123 (NTP)	*	none	allow LAN to time sync over NTP	
✓	0/0 B	IPv4 TCP/UDP	192.168.10.0/24	*	192.168.10.0/24	135	*	none	allow RPC communication over LAN	
✓	0/0 B	IPv4 TCP	192.168.10.0/24	*	192.168.10.0/24	389 (LDAP)	*	none	allow LDAP, directory services	
✓	12/58.58 MiB	IPv4 TCP	192.168.10.0/24	*	*	443 (HTTPS)	*	none	allow https traffic from LAN anywhere	
✓	0/0 B	IPv4 TCP	192.168.10.0/24	*	192.168.10.0/24	445 (MS DS)	*	none	allow SMB traffic within LAN	
✓	0/0 B	IPv4 TCP	192.168.10.0/24	*	192.168.10.0/24	464	*	none	allow kerberos password changes	
✓	0/68.91 MiB	IPv4 TCP	192.168.10.0/24	*	192.168.20.101	8443	*	none	allow lan to access web server on port 8443	
✓	0/480 B	IPv4 ICMP any	192.168.10.10	*	*	*	*	none	allow admin to ping any machine	
	SIEM RULES									
✓	0/0 B	IPv4 TCP	192.168.10.0/24	*	192.168.10.50	1514 - 1515	*	none	allow siem agents to forward logs to manager and agent registration on 1515, remember dmz cannot make outbound so make temp change whenever	
✓	0/0 B	IPv4 TCP	192.168.10.10	*	192.168.10.50	8443	*	none	allow DC admin to access wazuh dashboard	
	BAD RULE									
✓	0/0 B	IPv4 TCP	192.168.10.0/24	*	192.168.10.0/24	*	*	none	let LAN devices talk to each other, not necessarily needed but I will need this later when I exploit this network	
	DENY RULES									
✗	0/1.01 MiB	IPv4+6 *	*	*	*	*	*	none	cleanup rule, deny rest	
✗	0/0 B	IPv4 TCP/UDP	*	*	*	1 - 65535	*	none	block every other port that has no accept rule	

Figure: Firewall rule table for LAN interface

What is a LAN? A LAN or Local Area Network is a network that connects devices in a local area, such as a home, office, or building. LANs enable these devices to share data and other resources, such as files, printers, and even internet access.

Rules for LAN:

- The pfSense default anti-lockout rule prevents me (the administrator in this case) from locking myself out of the pfSense GUI interface
- Rule 2-12 are all the ports I am allowing,
    - DC admin can SSH to any device on the JPC Network
    - DNS traffic anywhere (ideally in a corporate environment have a DNS sinkhole or trap somewhere to catch malicious domains, but this is okay for home lab)
    - Kerberos between other local devices for authentication
    - NTP for time sync
    - RPC enables a program to use a service located on another computer on the network without network details
    - Allow LDAP for internal AD services
    - HTTPS anywhere (for now)
    - SMB traffic
    - 464 for password changes
    - 8443 to access the web server in the DMZ
- Rule 13, allow admin to ping anywhere (for network troubleshooting)
- SIEM RULES:
    - Allow devices to communicate with 10.50 on 1514 (log forwarding for Wazuh) and 1515 (agent registration for Wazuh)
    - Allow LAN DC to connect to the Wazuh web interface
- Cleanup Rule: Block everything else and every other port not accepted

DMZ

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
☐	✓ 0/0 B	IPv4 TCP	*	*	192.168.20.101	8443	*	none		allow inbound WAN traffic to web server in DMZ	
☐	✗ 0/0 B	IPv4 *	192.168.20.0/24	*	192.168.10.0/24	*	*	none		any connection initiated from dmz to lan is blocked	
☐	✓ 0/35 KiB	IPv4 TCP/UDP	192.168.20.0/24	*	192.168.10.50	1514 - 1515	*	none		allow siem agents in dmz to forward logs to manager and agent registration on 1515, remember dmz cannot make outbound so make temp change whenever	
☐	✓ 0/0 B	IPv4 ICMP any	192.168.10.10	*	192.168.20.0/24	*	*	none		allow dc admin to ping dmz network	
☐	✗ 0/1.30 MiB	IPv4 *	192.168.20.0/24	*	*	*	*	none		block outbound dmz connections, DMZ shouldn't be instantiating any outbound connections, only accepting inbound requests	
☐	✗ 0/0 B	IPv4 TCP/UDP	*	*	*	1 - 65535	*	none		block every other port that has no accept rule	
☐	✗ 0/6 KiB	IPv4+6 *	*	*	*	*	*	none		cleanup rule, deny rest	

Figure: Firewall rule table for DMZ interface

What is a DMZ? A DMZ or Demilitarized Zone acts as a buffer zone in an environment that acts as an intermediary between external networks (internet) and a private internal network (LAN). This interface typically hosts web-facing services like web servers or email servers that we wish to have on our network, but not on the same one that hosts all of our internal connections. After all, if a DMZ gets compromised, since communication with internal devices is restricted, it makes lateral movement for the attacker much more difficult.

Rules for DMZ:

- Allow inbound WAN traffic to reach the web server on port 8443(drastically increases attack surface, but for our needs, we are accepting this risk)
- Block any instantiated connection for the DMZ subnet to the LAN subnet (no need for that to happen)
- Allow DMZ devices to forward logs to the manager on 1514 and perform agent registration on 1515
- Allow the DC admin to ping the subnet
- Block all outbound connections from DMZ (shouldn't be establishing traffic, only replying to it)
- Block every other port, and deny the rest via clean-up rule

DMZ Web Server Security:

Why serve a File transfer over TLS?

- Encrypts traffic to prevent sniffing of credentials and files
- Authenticates the server to stop man-in-the-middle attacks
- Protects the integrity to avoid tampering or file injection

Secure Code Integrations

```
UPLOAD_FOLDER = 'uploads'
ALLOWED_EXTENSIONS = {'txt', 'pdf', 'png', 'jpg', 'jpeg', 'gif'}
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
app.config['MAX_CONTENT_LENGTH'] = 16 * 1024 * 1024  #16MB
```

Figure: Limit file types and upload size limit

```
from werkzeug.utils import secure_filename
@app.route('/upload', methods=['POST'])
def upload_file():
    if 'file' not in request.files:
        return "No file part", 400

    files = request.files.getlist('file')
    saved = 0
    for f in files:
        if not f or f.filename == '':
            continue
        if not allowed_file(f.filename):
            return "File type not allowed", 400
        filename = secure_filename(f.filename)
        f.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
```

Figure: secure_filename method sanitizes filenames to prevent directory traversal

```
@app.after_request
def set_secure_headers(resp):
    resp.headers['X-Content-Type-Options'] = 'nosniff'
    resp.headers['X-Frame-Options'] = 'DENY'
    resp.headers['X-XSS-Protection'] = '1; mode=block'
    resp.headers['Content-Security-Policy'] = "default-src 'self'"
    return resp
```

Figure: Secure headers, prevent MIME sniffing, prevent clickjacking by disabling ifram embedding, add XSS protection, and restrict the loading of scripts/styles and resources to this domain

```python
if __name__ == '__main__':
    app.run(host='192.168.20.101', port=8443, ssl_context=('cert.pem', 'key.pem'))
```

Figure: TLS encryption using generated cert and key PEM files

```python
def download_file(filename):
    filename = os.path.basename(filename)  #sanity check
    fullpath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
    if not os.path.isfile(fullpath):
        abort(404)
    return send_from_directory(app.config['UPLOAD_FOLDER'], filename, as_attachment=False)
```

Figure: os.path.basename strips the directory from filenames to prevent traversal, downloads can only be done from the uploads folder, and HTTP 404 error codes are used to avoid details of filesystem from being exposed. Also, send from directory is used instead of string concatenation for further traversal prevention

Tight Folder Permissions

```
drwxrwxr-x 5 jpcsftp jpcsftp 4096 Aug 19 16:07 .
drwxr-x--- 6 jpcsftp jpcsftp 4096 Aug 14 02:25 ..
-rw-rw-r-- 1 jpcsftp jpcsftp 1350 Aug  5 23:09 cert.pem
-rw------- 1 jpcsftp jpcsftp 1708 Aug  5 23:09 key.pem
-rw-rw-r-- 1 jpcsftp jpcsftp 2137 Aug 19 15:57 main.py
-rw-rw-r-- 1 jpcsftp jpcsftp  109 Aug  5 22:42 requirements.txt
drwxrwxr-x 2 jpcsftp jpcsftp 4096 Aug 18 19:51 templates
drwx------ 2 jpcsftp jpcsftp 4096 Aug 19 16:08 uploads
drwxrwxr-x 5 jpcsftp jpcsftp 4096 Aug  5 22:31 venv
jpcsftp@sftp-01:~/webup$
```

Figure: Tight rwx permissions over the uploads folder

Wireshark Captures TCP Stream Showing Encrypted Web Traffic

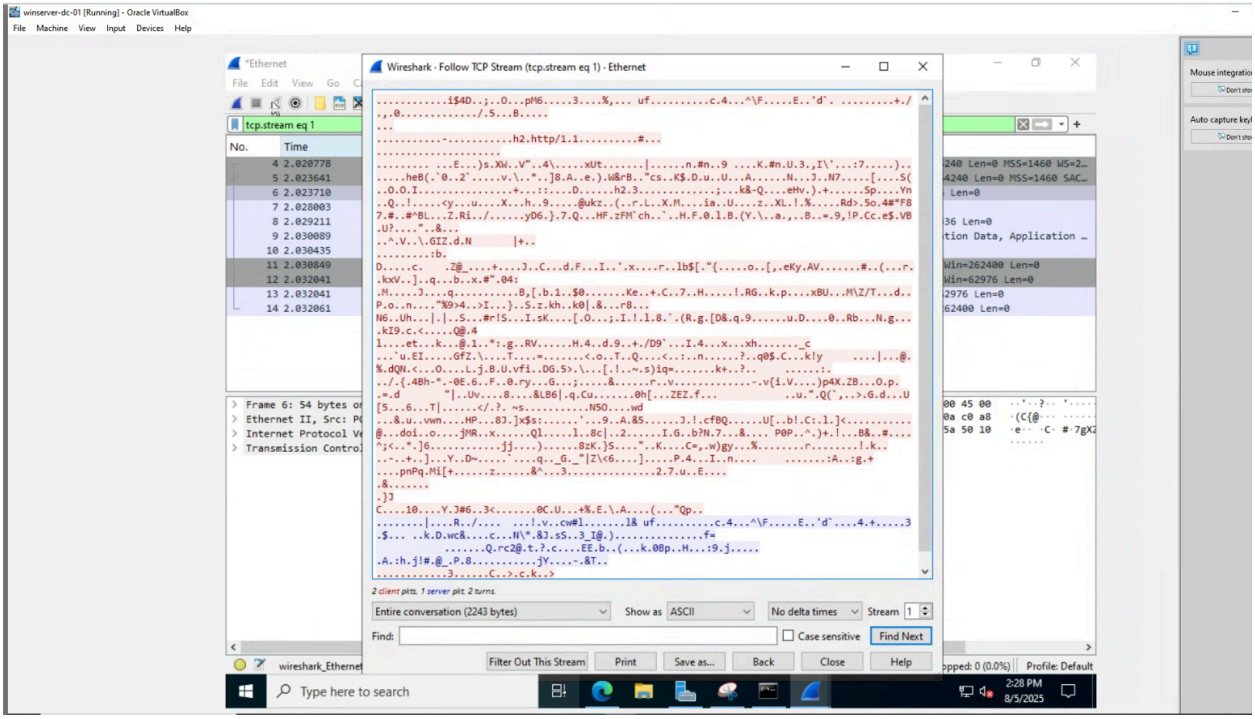


Figure: Wireshark capture of encrypted packet between client and web server

Domain Controller and Active Directory Configurations:

Why configure Active Directory?

- Centralize user authentication and permissions
- Apply consistent security and system policies via Group Policy
- Control access to shared resources (files, apps)
- Scale management of users and computers in large networks
- Improve security with auditing, strong policies, and Kerberos authentication

It makes managing many users and systems easier, more secure, and consistent.

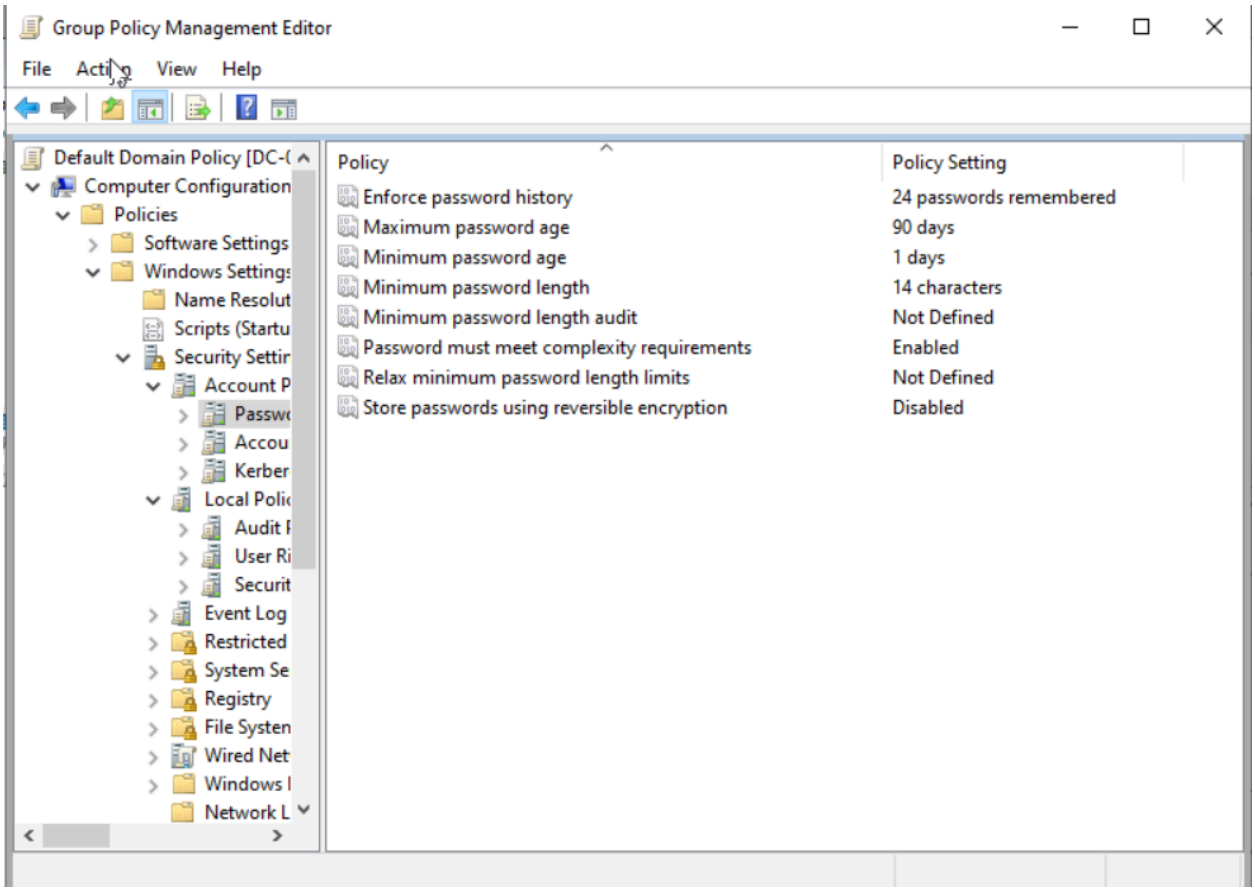


Figure: Password policy

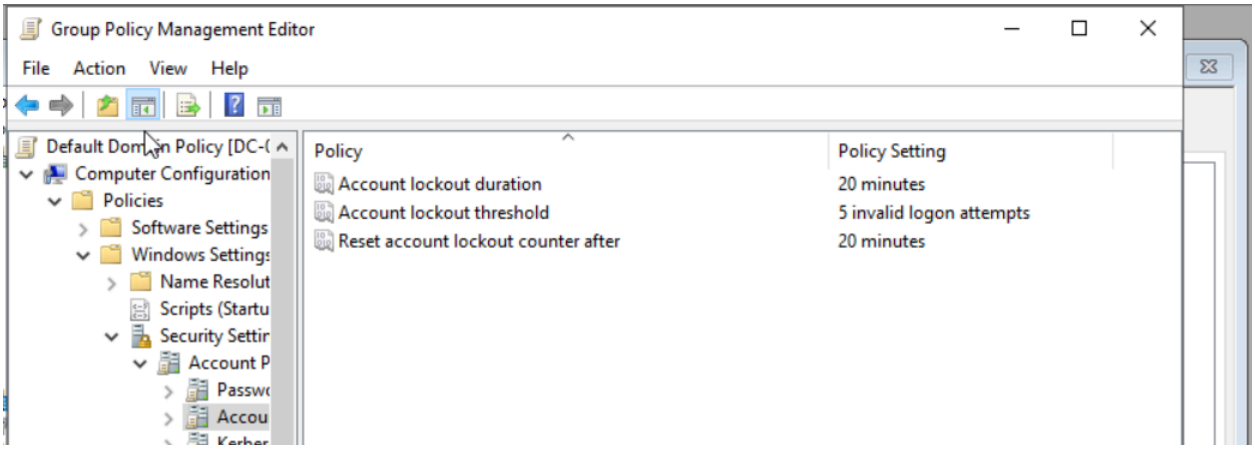


Figure: Account lockout

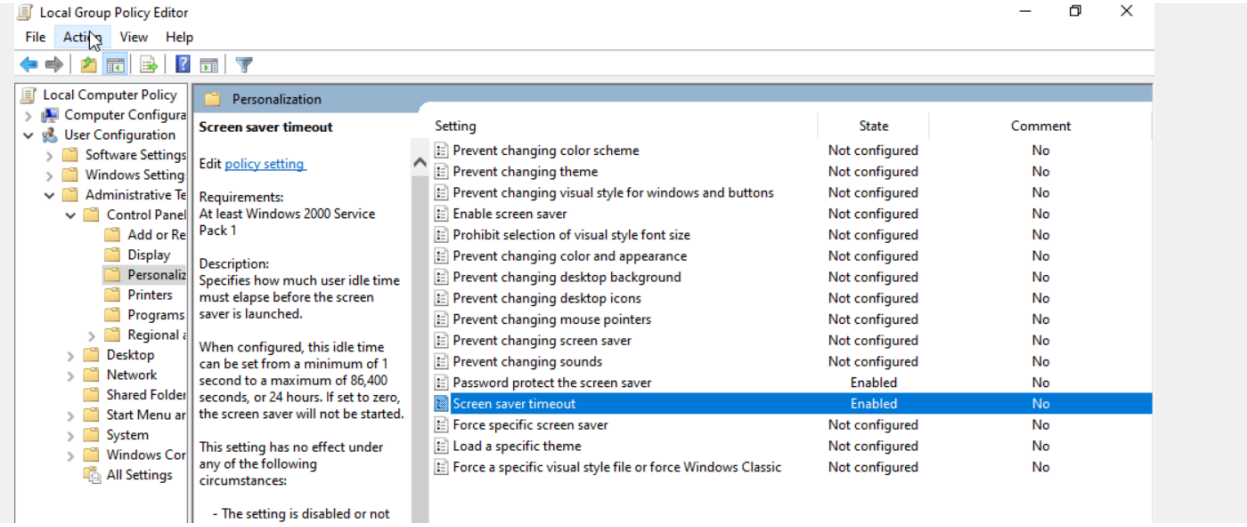


Figure: Screen lock after 15 minutes of inactivity requires you to re-enterthe password

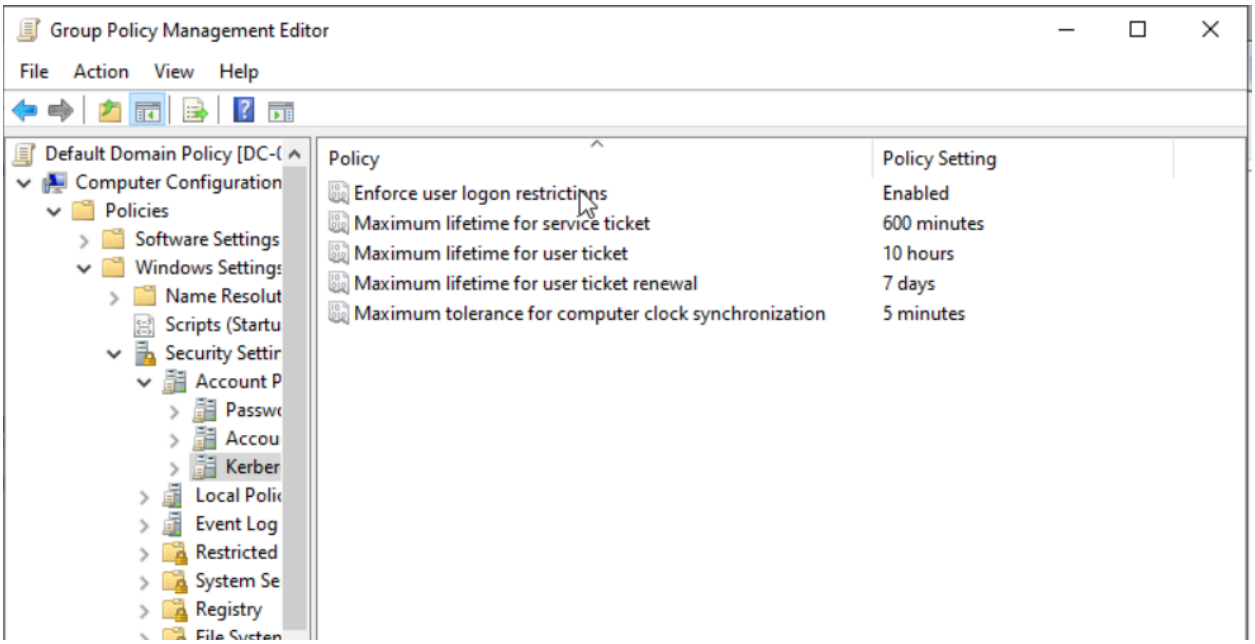


Figure: Kerberos authentication policy

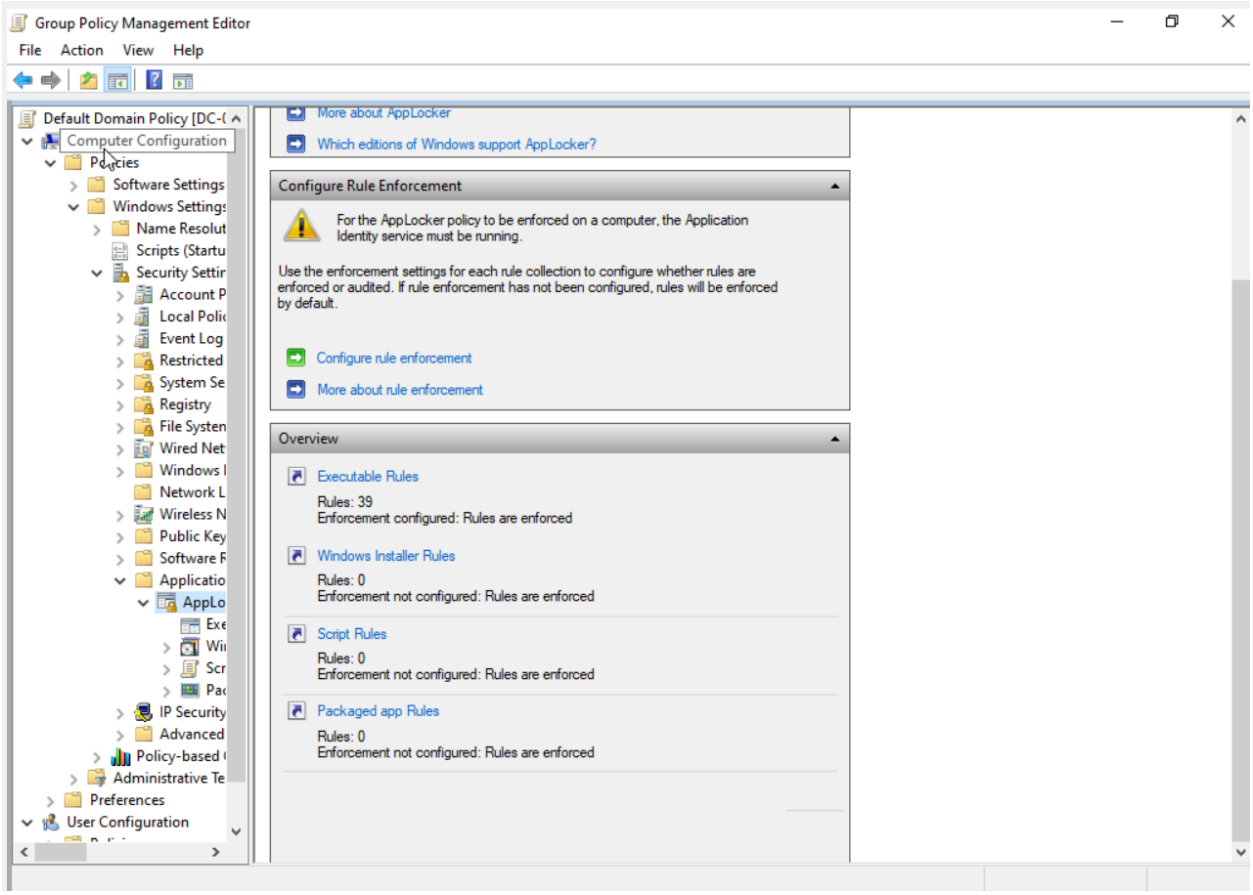


Figure: Limit users to only being able to execute signed programs

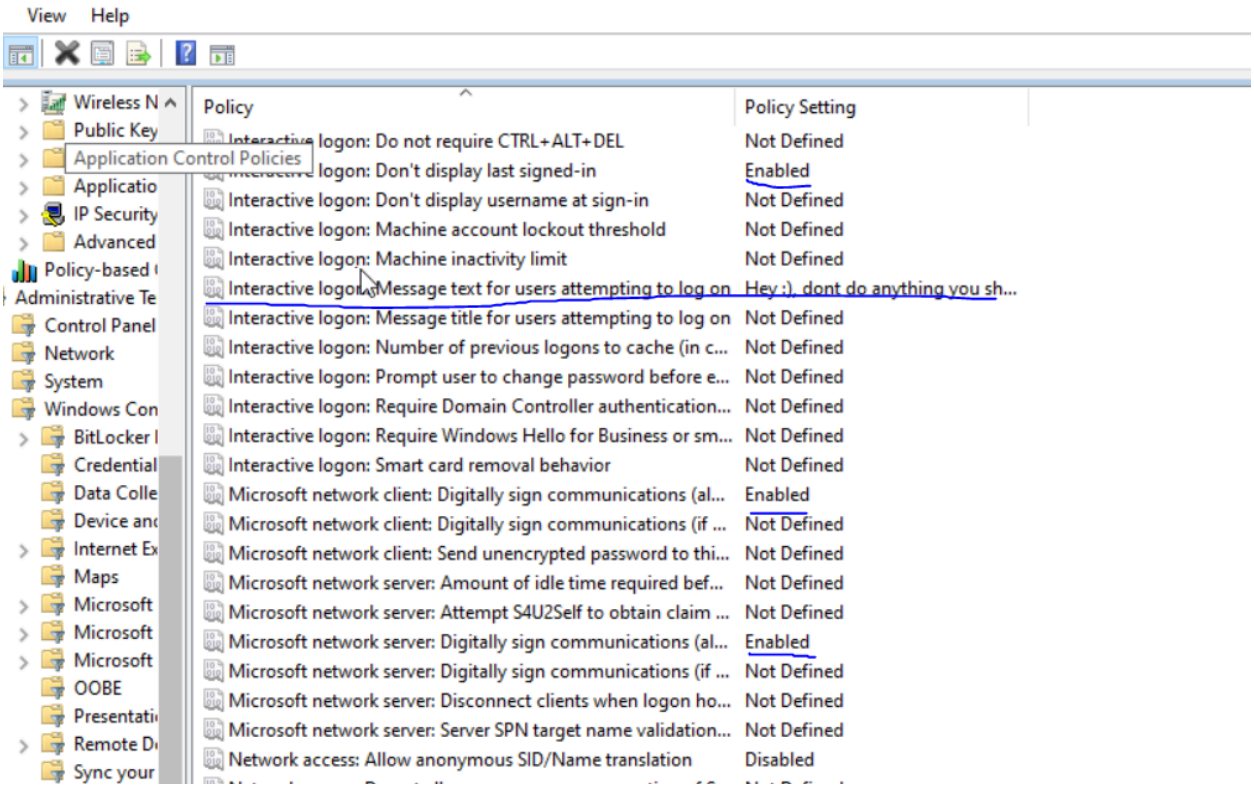


Figure: Print message at login, digitally sign client and server communications, and don't display the last user signed in

Figure: Log successful and failed logon attempts and log successful logoff attempts

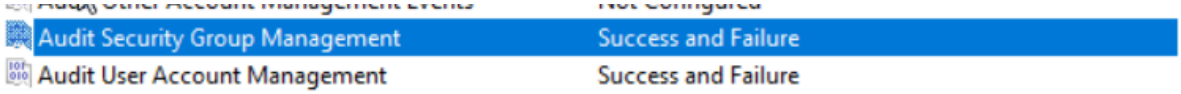


Figure: Log changes to both management consoles

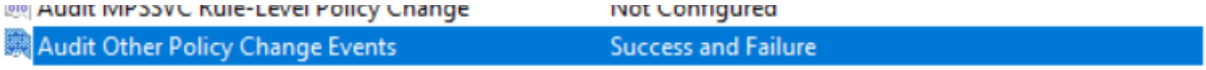


Figure: Log changes to group policy

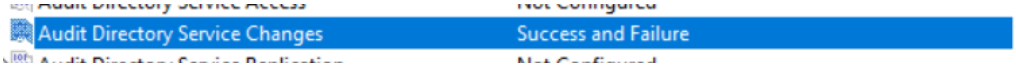


Figure: Log changes to ADDS

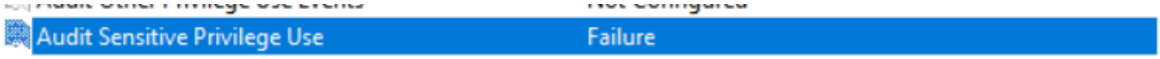


Figure: Log failed privilege use attempts

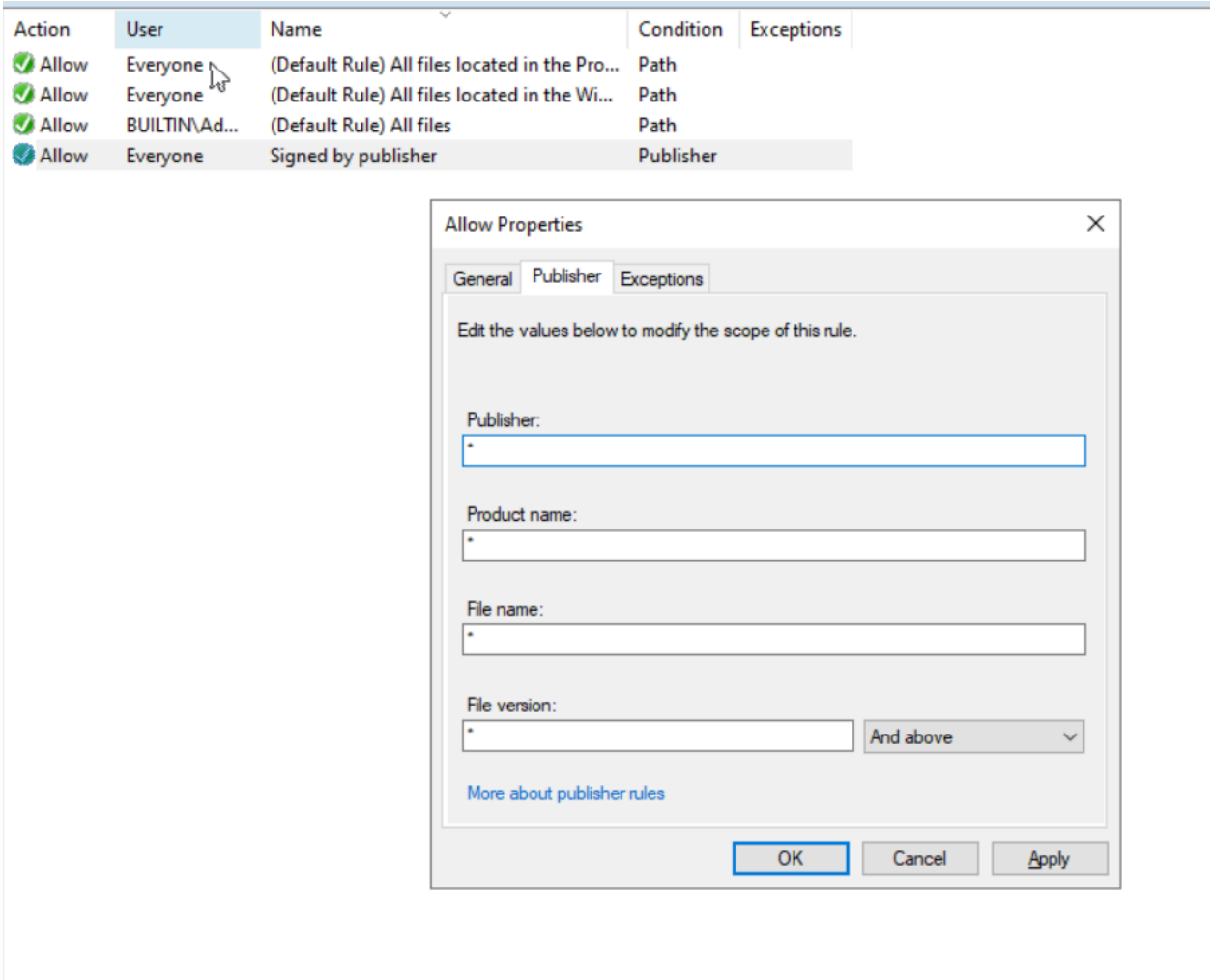


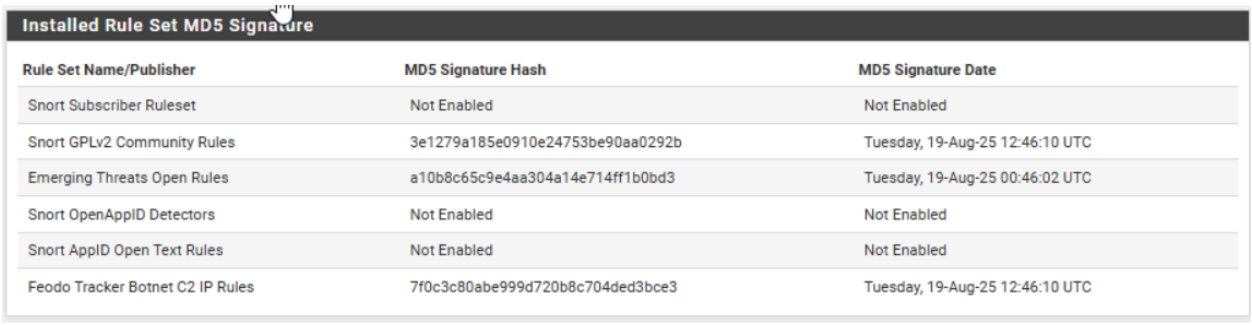
Figure: Allow program executions from only digitally signed software

Importing Rules for Snort IDPS:

Why use an IPS or IDS?

- Monitor network and host activity for malicious behavior
- Detect and alert on suspicious or unauthorized access attempts
- Block or prevent attacks in real time (e.g., malware, exploits, scans)
- Log events for auditing, compliance, and forensic analysis
- Strengthen overall network security by reducing exposure to threats

It helps detect and stop attacks before they compromise systems.



Installed Rule Set MD5 Signature

Rule Set Name/Publisher	MD5 Signature Hash	MD5 Signature Date
Snort Subscriber Ruleset	Not Enabled	Not Enabled
Snort GPLv2 Community Rules	3e1279a185e0910e24753be90aa0292b	Tuesday, 19-Aug-25 12:46:10 UTC
Emerging Threats Open Rules	a10b8c65c9e4aa304a14e714ff1b0bd3	Tuesday, 19-Aug-25 00:46:02 UTC
Snort OpenAppID Detectors	Not Enabled	Not Enabled
Snort AppID Open Text Rules	Not Enabled	Not Enabled
Feodo Tracker Botnet C2 IP Rules	7f0c3c80abe999d720b8c704ded3bce3	Tuesday, 19-Aug-25 12:46:10 UTC

Figure: Imported rule sets from Snort

pfSense allows us to install predefined rules directly from the Snort package manager. I installed the Snort GPLv2 Community Rules because they are free and cover a decent number of threats. Overall, it can't hurt to have it.

I also have Emerging Threat Open Rules installed. This provides me with signatures to malware, botnets, exploits, port/network scans, and other reported malicious activity. This is also updated frequently and will keep me somewhat protected as new threats emerge.

Lastly, I imported the Feodo Botnet Tracker, a large dataset of known Command and Control servers. I have this to prevent any known C2 communications from establishing connections with my network.

Installing Wazuh Agent on Endpoints and XDR capabilities:

Why do I like Wazuh? Wazuh combines the powerful functionalities of a SIEM and an XDR platform into one centralized solution. Not only can I monitor the logs and activity of all my endpoints, but I can also scan and interact with them due to my XDR capabilities.

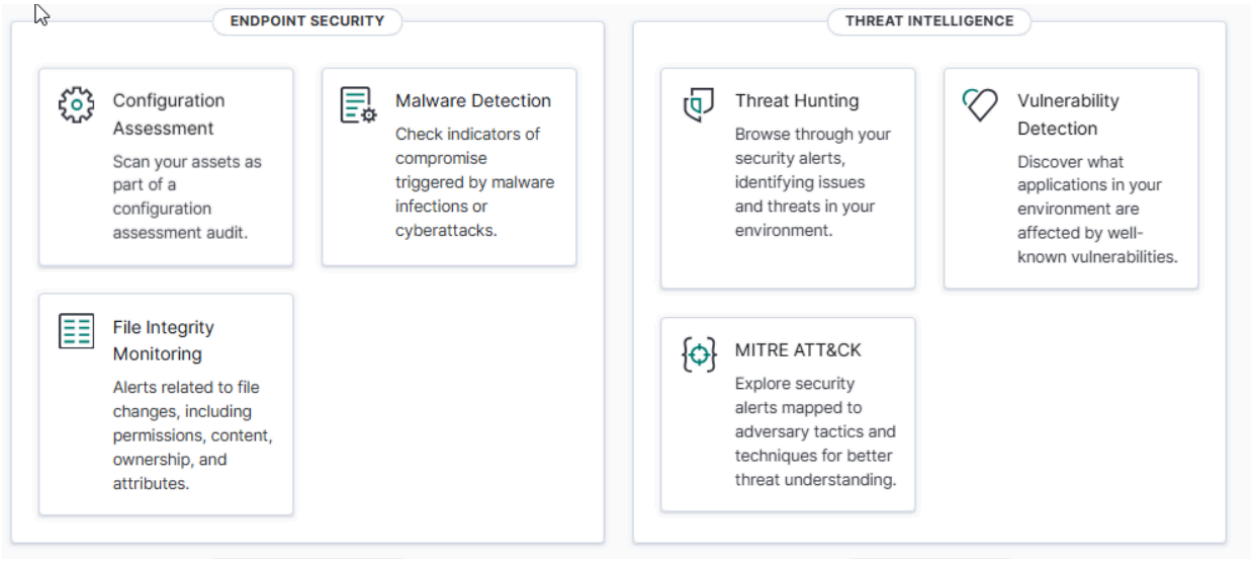


Figure: Endpoint Security and Threat Intelligence options for XDR

Configuration Assessment: allows us to scan devices with a deployed agent to ensure consistent and secure configuration across devices.
Malware Detection: Check for IOCs on devices with deployed agents.

File Integrity Monitoring: Alerts concerning changes to host files.

Threat Hunting: Allows me to parse through security alerts and identify threats.

Vulnerability Detection: Vulnerability scan on devices to assess the environment for well-known vulnerabilities.

MITRE ATT&CK: Maps security alerts to known ATT&CK techniques.

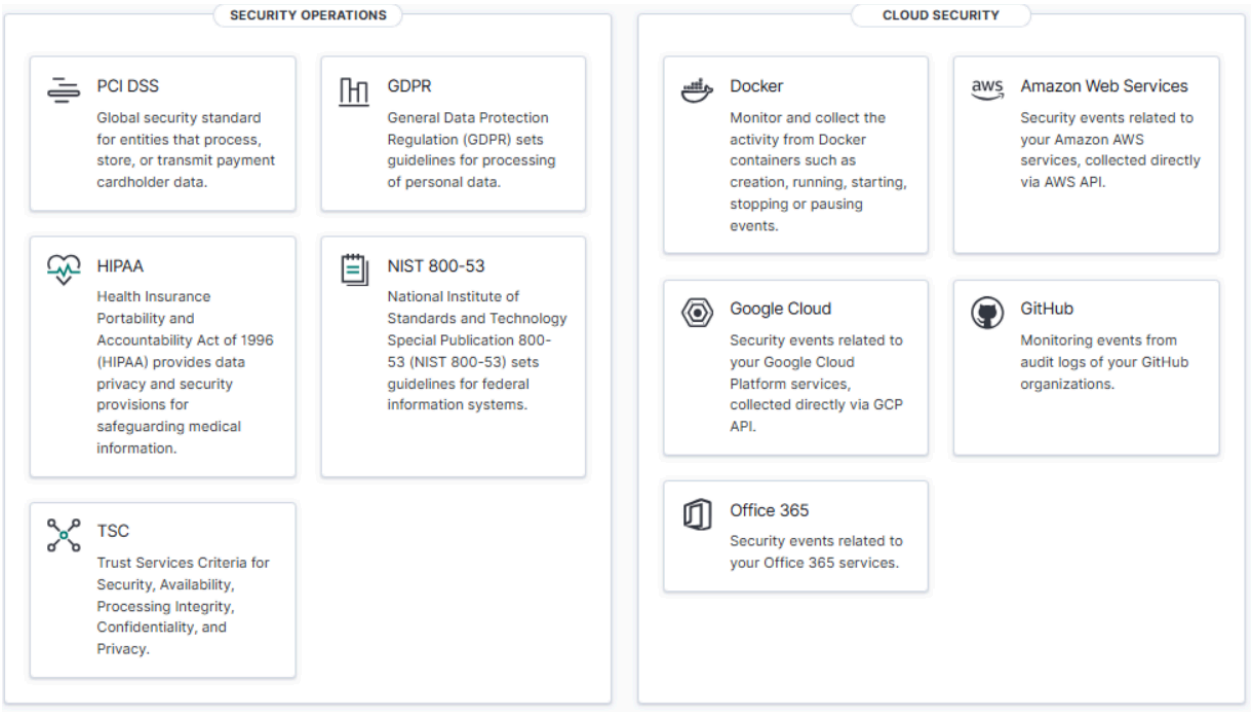


Figure: Sec Ops options and Cloud Security Options for XDR

Security Operations: Wazuh helps to ensure you remain compliant with standards and regulations like PCI DSS for handling card data, GDPR for handling data native to Europeans, HIPAA for health records, and NIST for general security posture

Cloud Security: Allows you to extend Wazuh capabilities to CSPs, Docker and Github environments.

Challenges and Resolutions

I encountered a wide range of issues during this lab. Some were resolved quickly, while others took days of troubleshooting. Each challenge deepened my understanding of system administration and network security.

- **Windows Domain Join Issue**
  - When instantiating a new VM to join my Active Directory environment, I mistakenly used Windows Home Edition instead of Pro. I learned that Windows Home cannot join domains, and that Windows Pro or Education Pro are the best options for most purposes.
- **Linux Server Connectivity in DMZ**
  - My first Linux server (intended as an SFTP host) couldn't reach the WAN interface despite having the correct IP (192.168.20.100), gateway (192.168.20.1), and subnet connection. A tcpdump on the DMZ gateway showed traffic reaching the subnet, but not routing properly.
  - To test, I spun up a Linux client, which connected without issue. Eventually, I reinstalled using a new ISO, assigned a static IP address and gateway, and everything worked. The lesson: sometimes the most straightforward fix is to restart and rebuild.
- **SFTP Server Failure, Switched to Web Server over TLS**
  - I attempted to configure an SFTP server in the DMZ with a chroot jail to restrict users to a single directory. Despite configuring sshd_config to match by group (sftpuzers with user anoncvs), connections failed repeatedly. I even switched it to match users (just anoncvs) and had no luck.
  - After repeated attempts, I pivoted and deployed a file upload web server over TLS instead.
- **WireGuard VPN Setup Issues**

  I wanted to test VPN access into my internal network from my physical desktop. I installed WireGuard on pfSense, created an interface, configured rules, and set up my desktop client. However, I couldn't reach the WAN from my desktop.
  - The issue was twofold:
    1. My WAN interface (bridged to my home LAN) could reach the internet outbound, but was not configured for inbound connections. My home router required explicit routes for devices, so inbound pings and VPN requests were dropped.
    2. pfSense was using NAT for outbound traffic only and lacked proper port forwarding for inbound services.
  - I learned that for services like my web server (8443) and WireGuard VPN (51820), I needed to configure port forwarding in both pfSense and VirtualBox to route inbound connections properly. This reinforced my understanding of NAT, port forwarding, and VPNs.
- **Wazuh Deployment Problems**
  - Initially, I attempted to deploy the Wazuh Indexer, Server, and Dashboard using Docker. I ran out of disk space during container composition, so I mounted a new drive.
  - Next, I hit an issue where Wazuh's authentication certificates and keys were generated as directories instead of files, while Docker expected file arguments. After regenerating them, I still ran into service issues.
  - Ultimately, I abandoned Docker and followed the step-by-step manual installation, which proved to be much more reliable.

Improvements to Further Harden Security

It should be known that I will be using this environment for red teaming practice later on, so I didn't want to make it invincible. Just good enough so I could learn how everything is put in place so that I can break it later. Once I know how to break it, I can learn how to secure it.

But here are some general improvements:

pfSense:

- DNS Sinkhole: Redirects users from malicious domains
- Port Forwarding to internal services (Apache Web server, WireGuard VPN service, etc.) - more accessibility than security
- IPv6?? (Not necessarily security, but something to consider)

DMZ Web Server:

- Reverse Proxy - more security, reliability, and efficiency
- WAF (Web Application Firewall)
- Log failed upload attempts
- Authentication before accessingthe page (Login page)

Domain Controller and Active Directory:

- RBAC
- More installation, script, and package rules

Snort Intrusion Detection and Prevention System:

- More Rule tuning, get rid of redundant or unneeded ones for efficiency
- Pay for better rule sets
- Design your own rules that are optimized for threats to your environment

Wazuh Security and Information Event Manager and XDR Platform:

- Addressing scanned vulnerabilities on agents
- Customize the dashboard for viewing
- Implement SOAR (acts based on SIEM decisions)

Accomplishments

- Instantiated and configured virtual machines to emulate a realistic enterprise network.
- Implemented **secure network segmentation** with least-privilege access rules (ex., DMZ services exposed to WAN whilst LAN shielded from DMZ).
- Confirmed **IDS effectiveness** by generating and detecting test intrusion traffic with Snort across multiple network interfaces
- Demonstrated the **principle of least privilege (POLP)** by preventing DMZ-hosted services from initiating unauthorized communication with LAN resources.

- Achieved **centralized security visibility** by integrating Wazuh SIEM with endpoint agents, enabling real-time alerts, log correlation, and XDR capabilities.
- Hardened DMZ Web Server through:
    - Restricting user access with **tight file and directory permissions** (uploads folder locked down with rwx restrictions).
    - Enforcing **secure web application configurations** such as file type whitelisting, upload size limits, sanitized filenames, secure response headers, and TLS encryption.
- Simulated **enterprise environment** with Windows AD/DC and client machines, reinforcing understanding of identity, authentication, and endpoint monitoring.
- Applied **compliance-aligned practices** (logging, segmentation, encrypted transport) relevant to frameworks such as **CMMC, NIST 800-171 / 800-53, ISO 27001, PCI DSS,** and more.
- Strengthened **troubleshooting and monitoring skills** using tools such as **tcpdump**, **nc**, **Wireshark**, and **ping** to validate firewall enforcement.

Conclusion

This lab reinforced my ability to design and secure a segmented enterprise-style network and deepened my understanding of segmentation, firewalls, IDS/IPS, SIEM integration, and system hardening. The experience tied together core infrastructure skills with real-world compliance practices, preparing me for larger-scale cloud and enterprise security challenges.