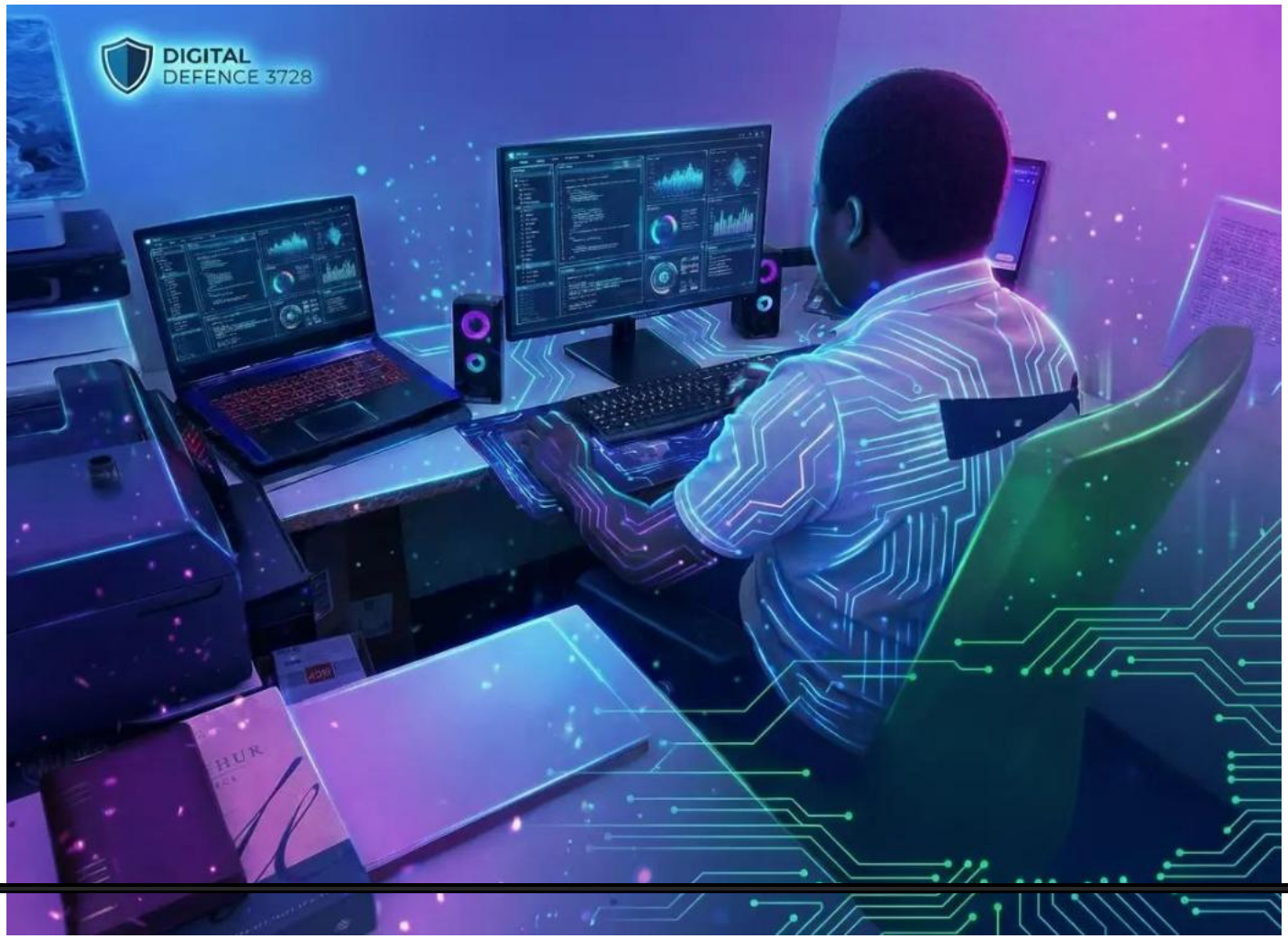
Enterprise Active Directory + SIEM Cybersecurity Home-Lab

Puporse:
Build hands-on SOC analyst skills

Prepared by:
Blessed Muteswa



Digital Defence 3728 — On-Prem SOC Analyst Home Lab

Author: Blessed Muteswa

Introduction

Digital Defence 3728 is an on-premises, virtualized SOC analyst lab designed to demonstrate **Blue-Team** skills: centralized logging. The lab runs on VMware Workstation Pro (host Windows 11) and uses a tightly controlled network with pfSense as gateway and a single AD authoritative DNS server.

Key components

pfSense firewall, Windows Server 2022 (Domain Controller + DNS), Ubuntu Server 24.04 LTS running Splunk Enterprise, a hardened Jumpbox (Windows 11 LTSC), a domain-joined Windows client, and two exercise systems (Kali, Metasploitable2).
All configuration, troubleshooting and verification steps are included, so the lab is fully reproducible and suitable.

Hardware Optimization for my MSI Laptop

- Host System: Windows 11 (Hypervisor Host)
- CPU: Intel i5 6 cores, 12 Logical Processors
- RAM: 40GB Total
- Storage: 1TB SSD + 500GB SSD NVMe
- Storage: 500GB SSD (Primary for VMs),
- 500GB NVMe (Host OS & Applications)

Goal: stable, repeatable SOC work; minimal risk of host lag.

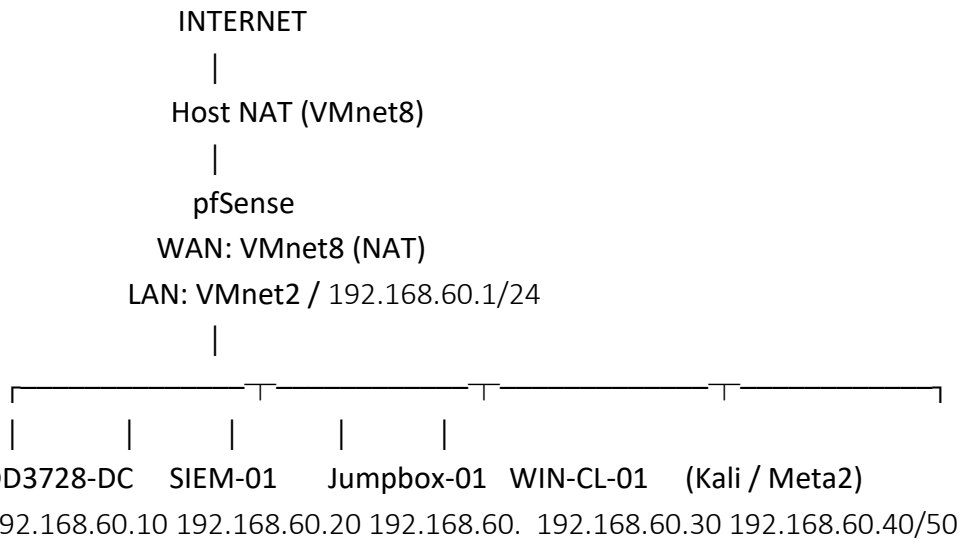
Host reserve: 8 GB RAM; reserve 4 logical processors for host → 32 GB RAM and 8 logical processors available to VMs.

1 — Summary / Outcomes

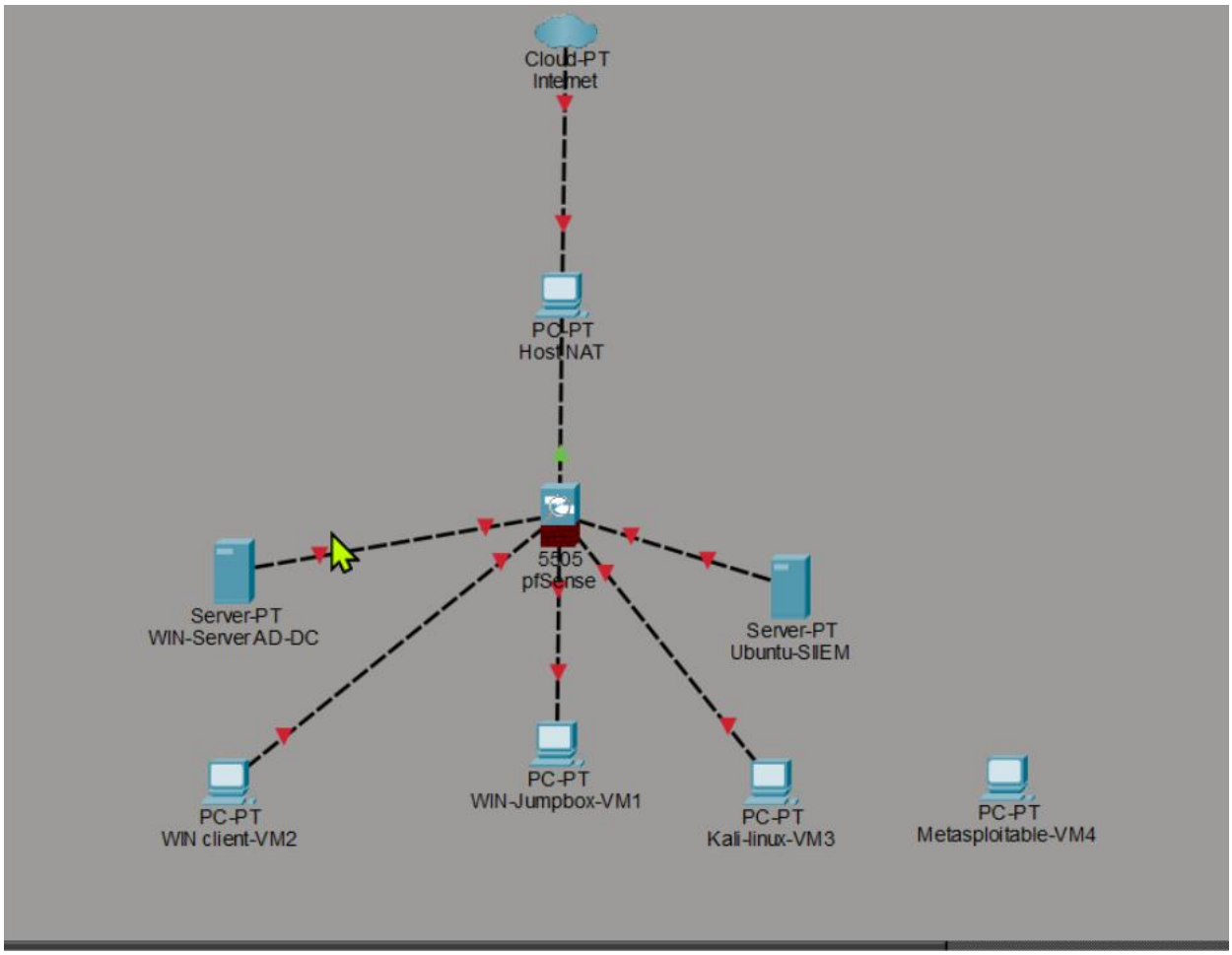
- Built a Blue-Team SOC lab named **Digital Defence 3728**.
- Achieved: AD + DNS health, pfSense gateway + NAT, Splunk Enterprise running and accepting logs, Jumpbox management access (SSH/RDP), Windows UF and Sysmon telemetry ingestion, and controlled attacker/test hosts.
- Captured and resolved real faults: DC rename → stale DNS/glue A record; Ubuntu Netplan/cloud-init override; incorrect subnet assignment; RDP profile & firewall issues; all fixes and verification commands are documented.

2 — Architecture & VM sizing

Network: single isolated lab LAN 192.168.60.0/24, pfSense LAN 192.168.60.1, DC/DNS 192.168.60.10, SIEM 192.168.60.20



[**screenshot** 👇🏾]



VM sizing (final):

- pfSense: 1 vCPU / 1 GB RAM / 10 GB disk
- Windows Server 2022 (DC): 2 vCPU / 4 GB RAM / 80 GB disk
- Ubuntu Server (SIEM): 4 vCPU / 12 GB RAM / 200 GB disk
- Jumpbox (analyst): 2 vCPU / 8 GB RAM / 80 GB disk
- Windows Client: 2 vCPU / 4 GB RAM / 80 GB disk
- Kali: 2 vCPU / 2 GB RAM / 40 GB disk
- Metasploitable2: 1 vCPU / 1 GB RAM / 20 GB disk

3 — Files / ISOs (official sources)

Downloaded official images only (evaluation or community builds):

- pfSense: https://www.pfsense.org/download/
- Ubuntu Server 24.04 LTS: https://ubuntu.com/download/server
- Splunk Enterprise (Linux): https://www.splunk.com/en_us/download.html
- Splunk Universal Forwarder (Windows): https://www.splunk.com/en_us/download/universal-forwarder.html
- Windows Server Evaluation / Windows 11 Enterprise LTSC: Microsoft Evaluation Center
- Kali Linux: https://www.kali.org/get-kali/
- Metasploitable2: Rapid7 / official archived images

4 — Step-by-step build — overview

High-level order of execution:

1. Created VMware networks (VMnet8 NAT, VMnet1 Host-only 192.168.60.0/24), disable host DHCP on the host-only network.
2. Created VMs with sizing above (attach ISOs).
3. Installed pfSense (assigned WAN = VMnet8, LAN = VMnet1), set LAN IP 192.168.60.1, enabled DHCP for lab.
4. Installed Windows Server 2022, configured static IP, installed AD DS & DNS, promoted to domain digitaldefence3728.lab (DC).
5. Installed Ubuntu Server, configured static network to 192.168.60.20, hardened netplan/cloud-init, installed Splunk Enterprise.
6. Configured pfSense syslog to forward to Splunk, set firewall rules.
7. Created Jumpbox VM (Windows 11 LTSC), set DNS to DC, installed tools.
8. Created Windows client, set DNS to DC, joined domain, installed Splunk UF and Sysmon.

9. Added Kali & Metasploitable2, kept powered off when not testing.
10. Validated logs in Splunk and ran sample searches.

Each major section below pairs steps with commands and validation checks.

5 — pfSense installation & configuration

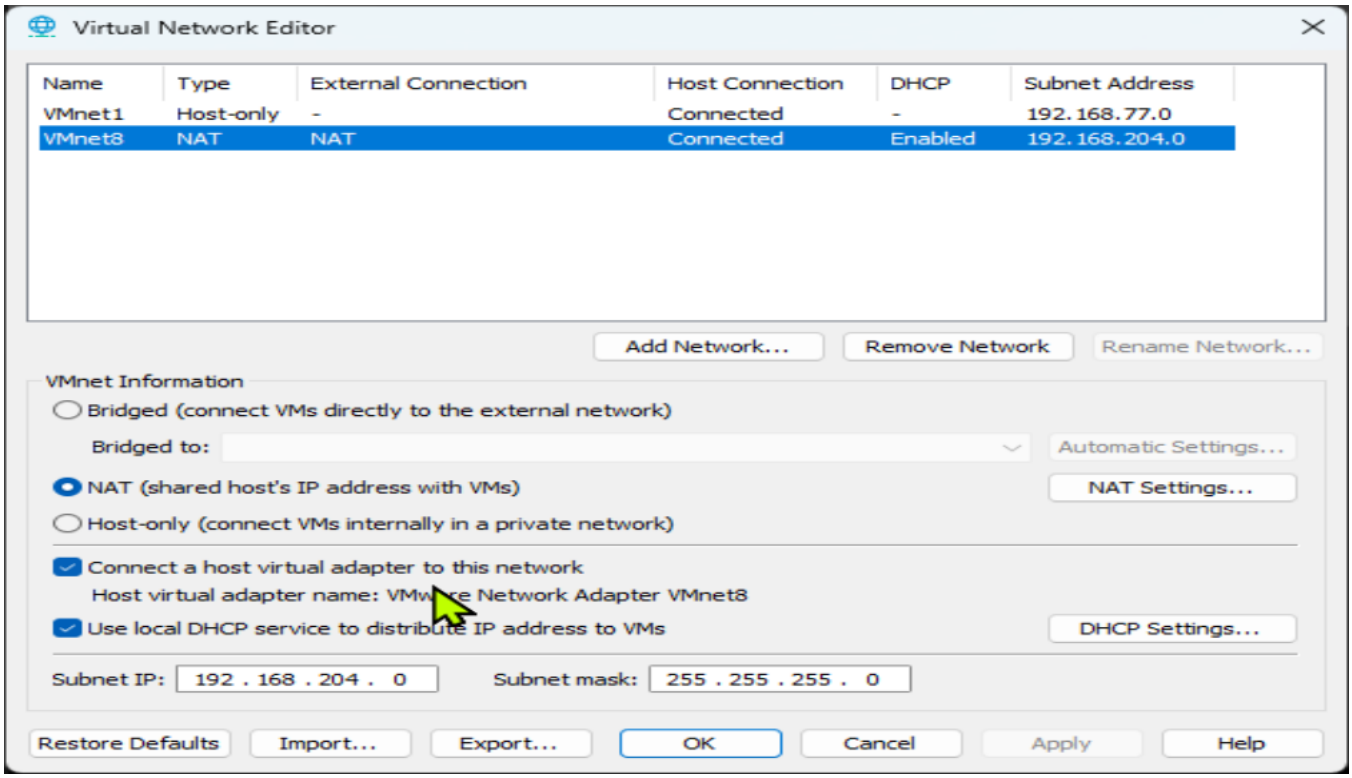
Goal: pfSense provides a safe gateway for internet access and isolates the lab LAN, while forwarding logs to the Splunk SIEM.

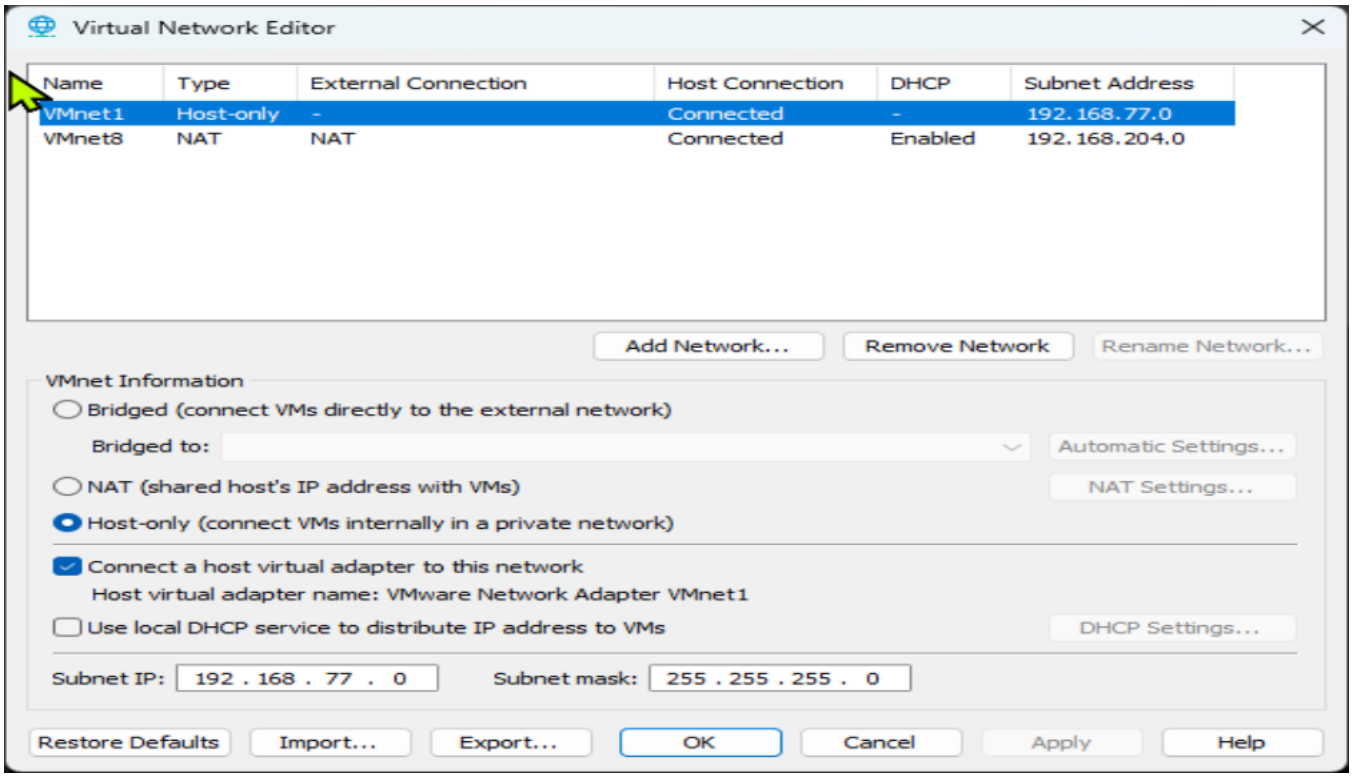
A. VMware network preparation (host)

- Opened VMware Workstation → Edit → Virtual Network Editor (Ran as Administrator).
- VMnet8: NAT (defaults) — used for pfSense WAN.
- Added VMnet1: Host-only, Subnet 192.168.60.0, Mask 255.255.255.0 — Disabled VMware DHCP (pfSense will provide DHCP).
- Saved configuration.

[**screenshots** 👆🏽👇🏽]

VMnet8: NAT | WAN | DHCP Enabled



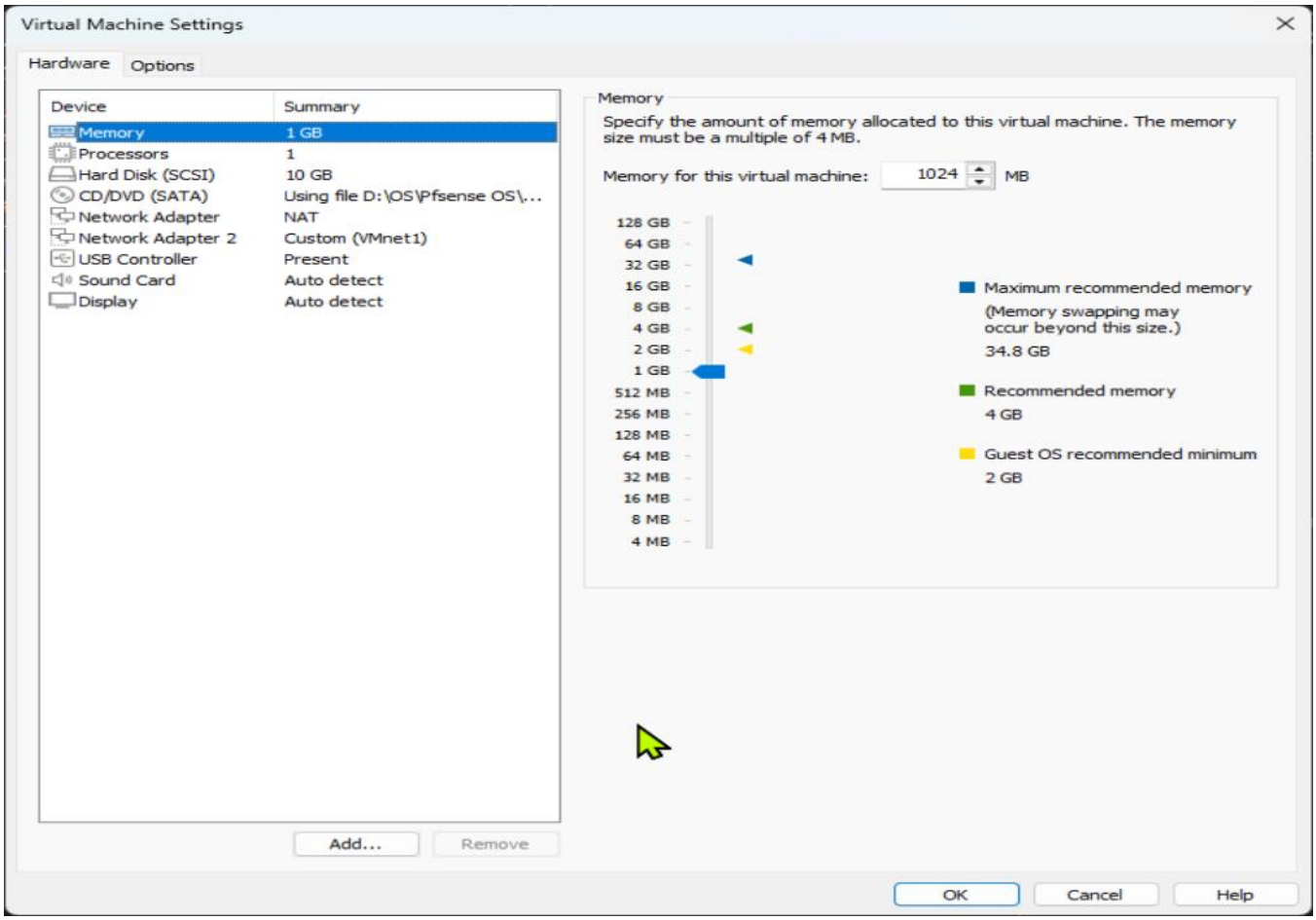


VMnet1: LAN | DHCP Disabled

B. Created pfSense VM

- Guest OS: Netgate-Installer v1.1
- Disk: 10GB, thin provisioned.
- Memory: 1 GB.
- CPU: 1 vCPU.
- Network adapters:
    - Adapter 1 → VMnet8 (WAN).
    - Adapter 2 → VMnet1 (LAN).

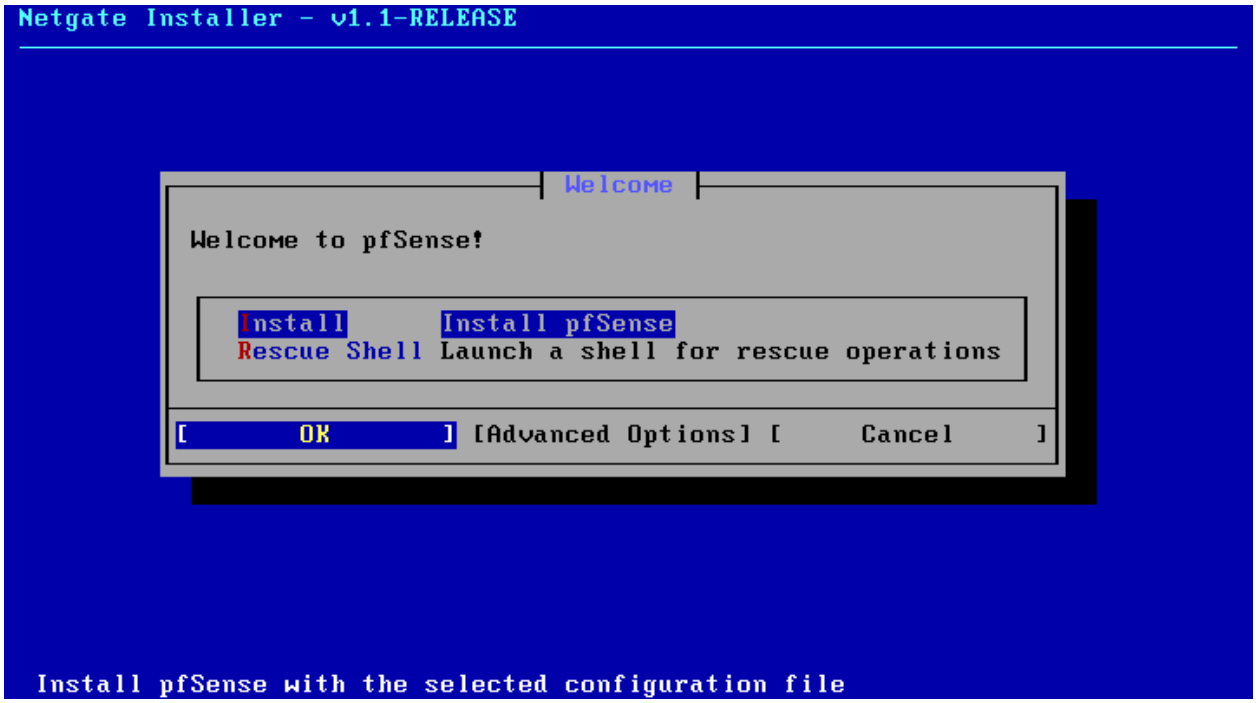
[**screenshots** 👆🏾👇🏾]

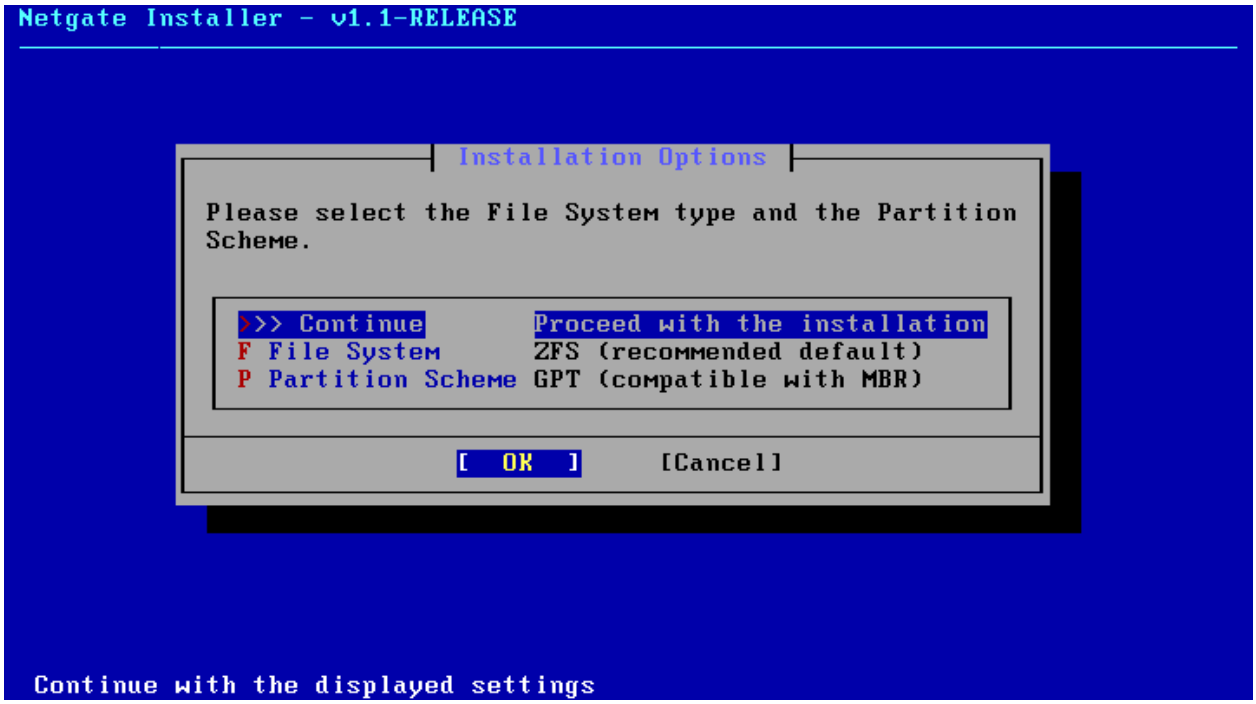


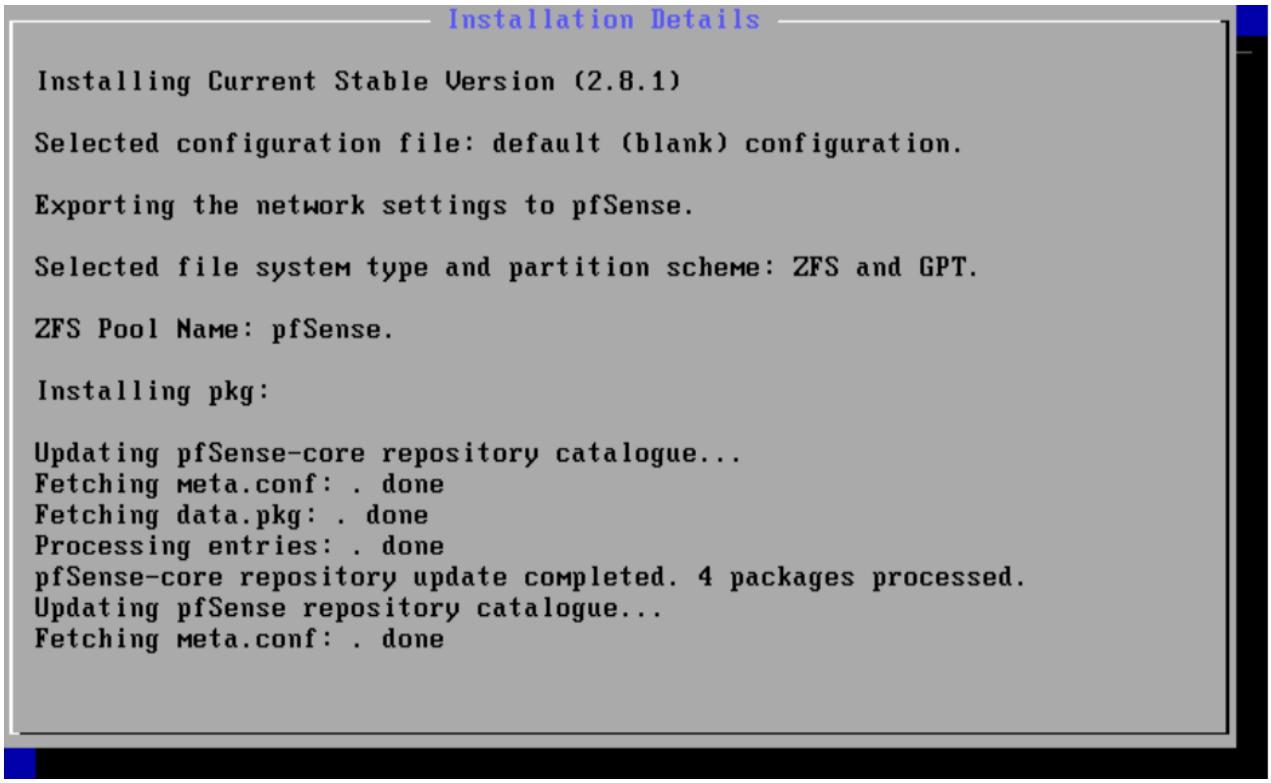
C. Installed pfSense (console)

- Booted from pfSense ISO and followed installer defaults.
- Accepted partitioning and standard options; rebooted into installed pfSense.

[**screenshots** 👆🏾👇🏾]


Netgate Installer - v1.1-RELEASE
Welcome
Welcome to pfSense!
Install Install pfSense
Rescue Shell Launch a shell for rescue operations
[OK] [Advanced Options] [Cancel]
Install pfSense with the selected configuration file


Netgate Installer - v1.1-RELEASE
Installation Options
Please select the File System type and the Partition Scheme.
>>> Continue Proceed with the installation
F File System ZFS (recommended default)
P Partition Scheme GPT (compatible with MBR)
[OK] [Cancel]
Continue with the displayed settings

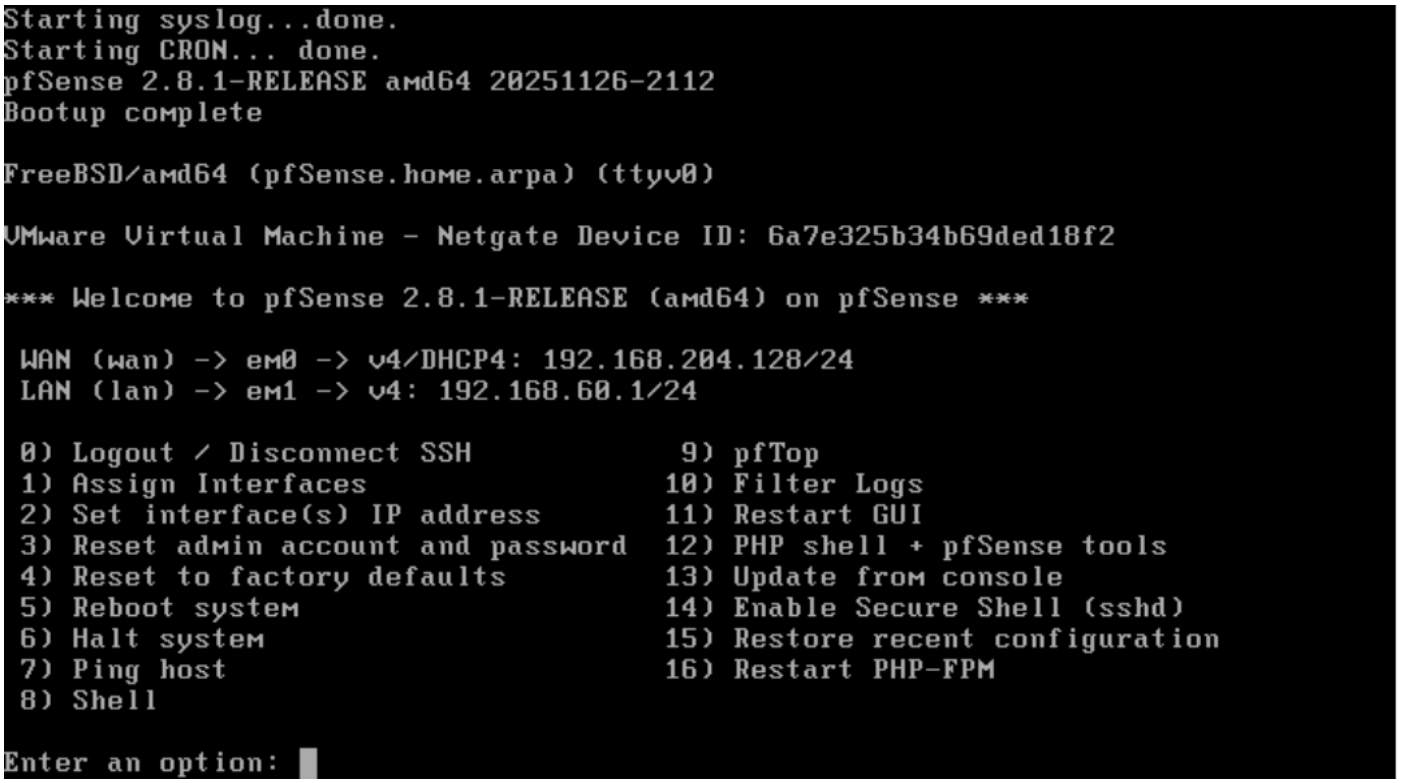


D. Assigned interfaces (console)

At pfSense console:

- Option 1: Assign interfaces — chose NIC mapped to VMnet8 [em0] as WAN and NIC mapped to VMnet1 [em1] as LAN.
- Option 2: Set LAN IP to `192.168.60.1` with prefix /24. Enabled DHCP.

[**screenshot** 👆👇]

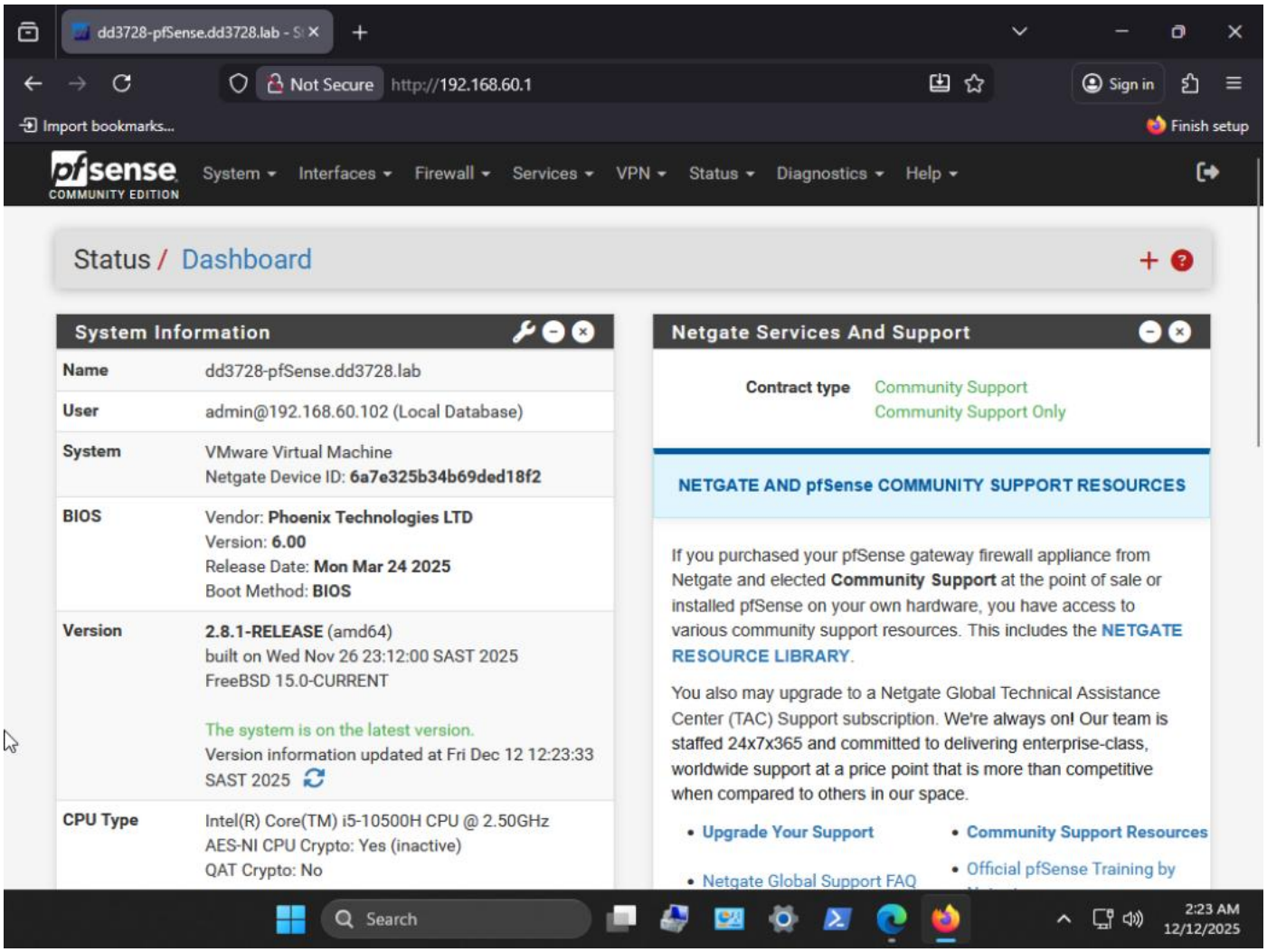


E. GUI setup

From Jumpbox on LAN:

- Browsed to https://192.168.60.1
- Ran setup wizard:
  - Hostname: dd3728-pfsense.
  - Domain: dd3728.lab.
  - WAN: DHCP (VMnet8).
  - LAN: `192.168.60.1/24`.
  - Admin password: **********

[**screenshots** 👆👇]

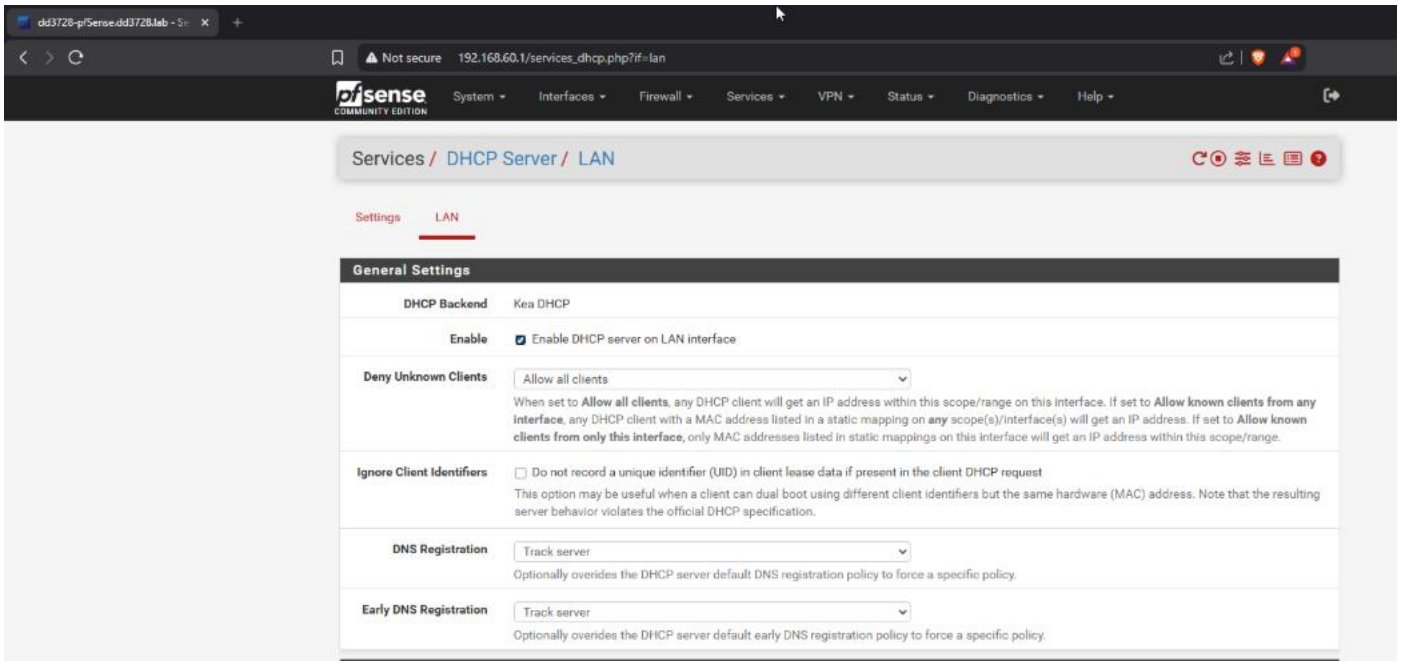


F. DHCP & DNS settings

In pfSense GUI:

- Navigated to: Services → DHCP Server → LAN.
- Enabled DHCP (optional but convenient).
- Range: `192.168.60.100 → 192.168.60.254`.
- **DNS servers:** set `192.168.60.10` (the Domain Controller) as DNS server, so clients use AD DNS for name resolution (instead of pfSense).

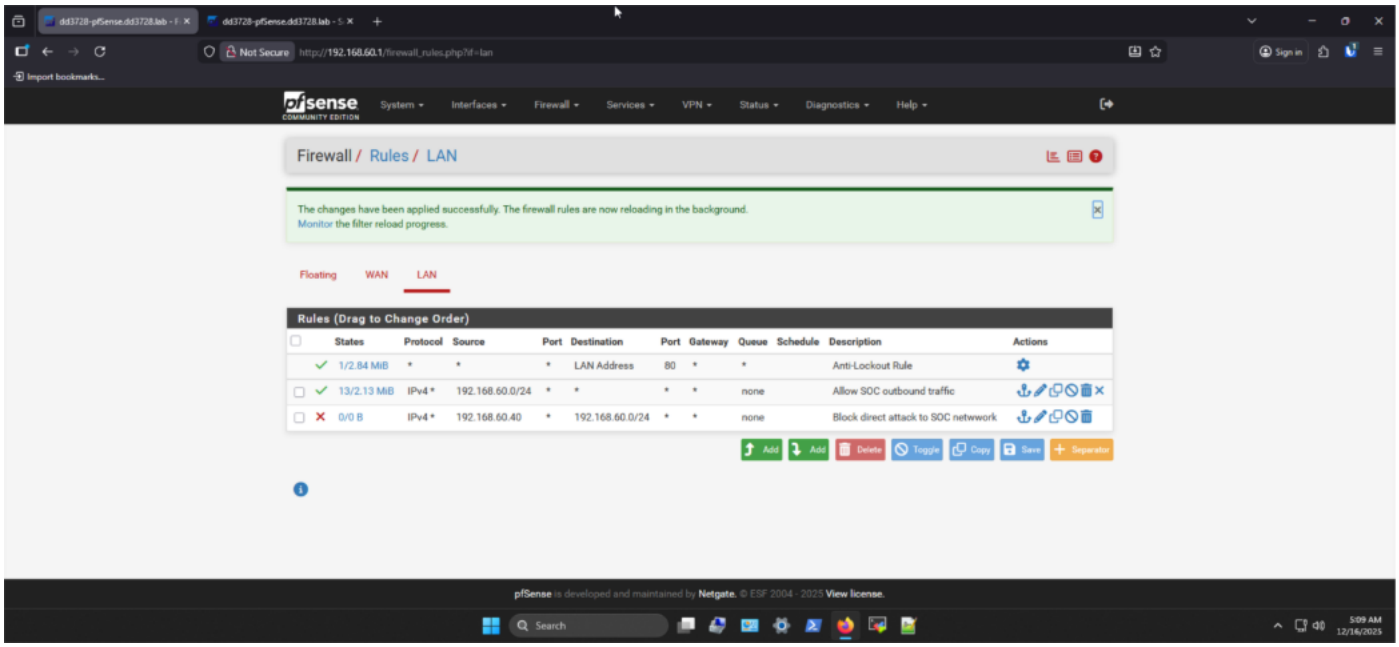
[**screenshots** 👆👇]



G. Firewall rules

- Firewall → Rules → LAN: added baseline “Allow LAN net → any” rule for lab convenience.
- Kept default WAN deny rules.
- Enabled logging on reject/deny rules to feed meaningful traffic into Splunk.

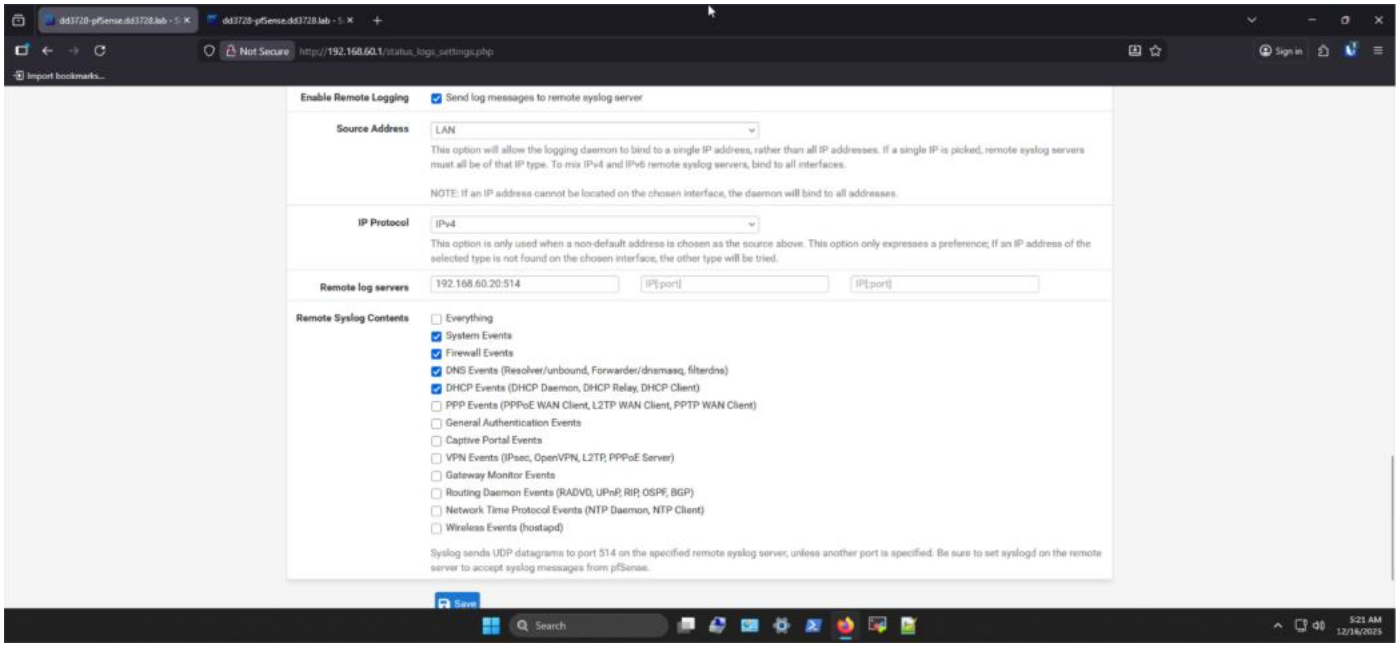
[**screenshots** 👆🏾👇🏾]



H. Remote syslog forwarding to Splunk

- Status → System Logs → Settings → Remote Logging Options.
- Remote log server: `192.168.60.20`.
- Port: `514` (UDP) .
- Applied and saved.

[**screenshots** 👆🏽👇🏽]



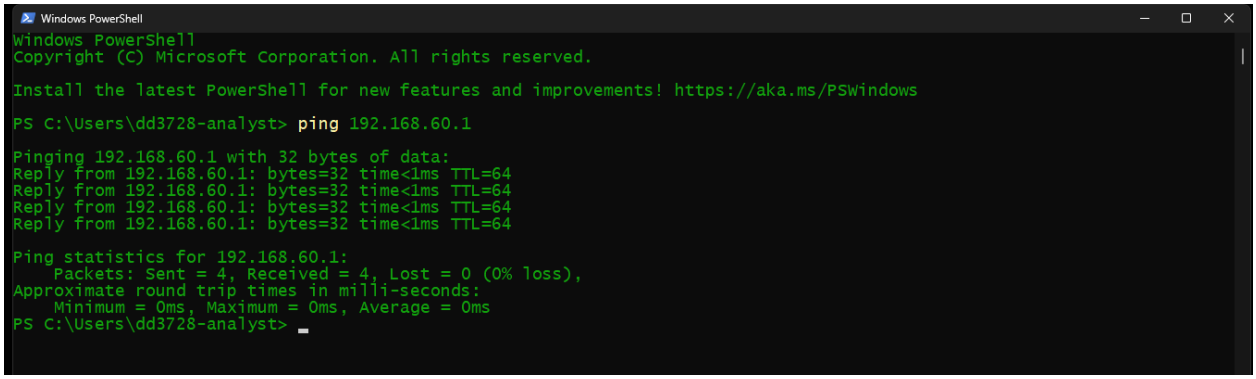
I. Validation

From Jumpbox (PowerShell):

`#powershell`

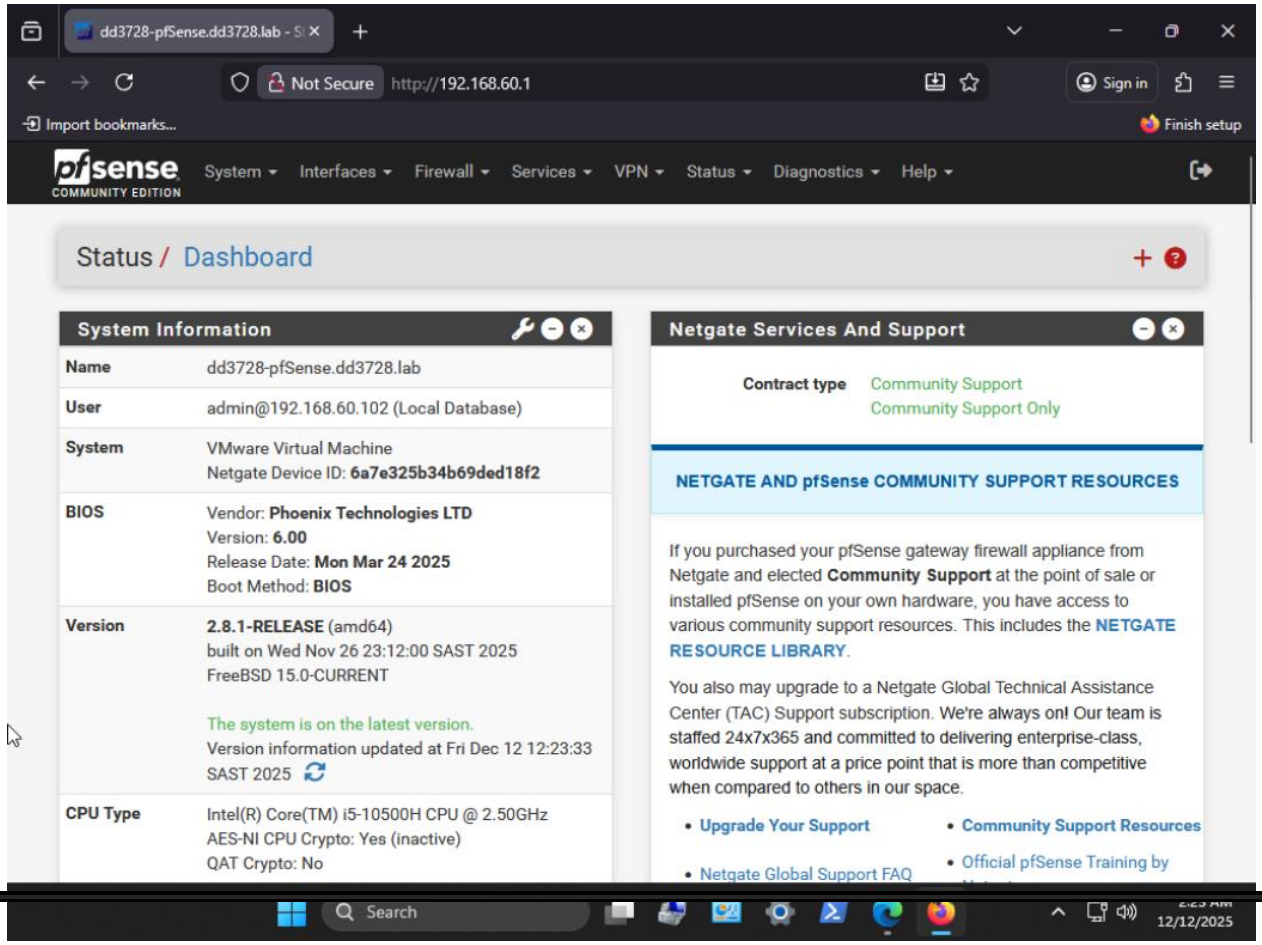
`pinged 192.168.60.1`

[**screenshots** 👆👇]



- Browse to https://192.168.60.1 and confirmed pfSense GUI loads.

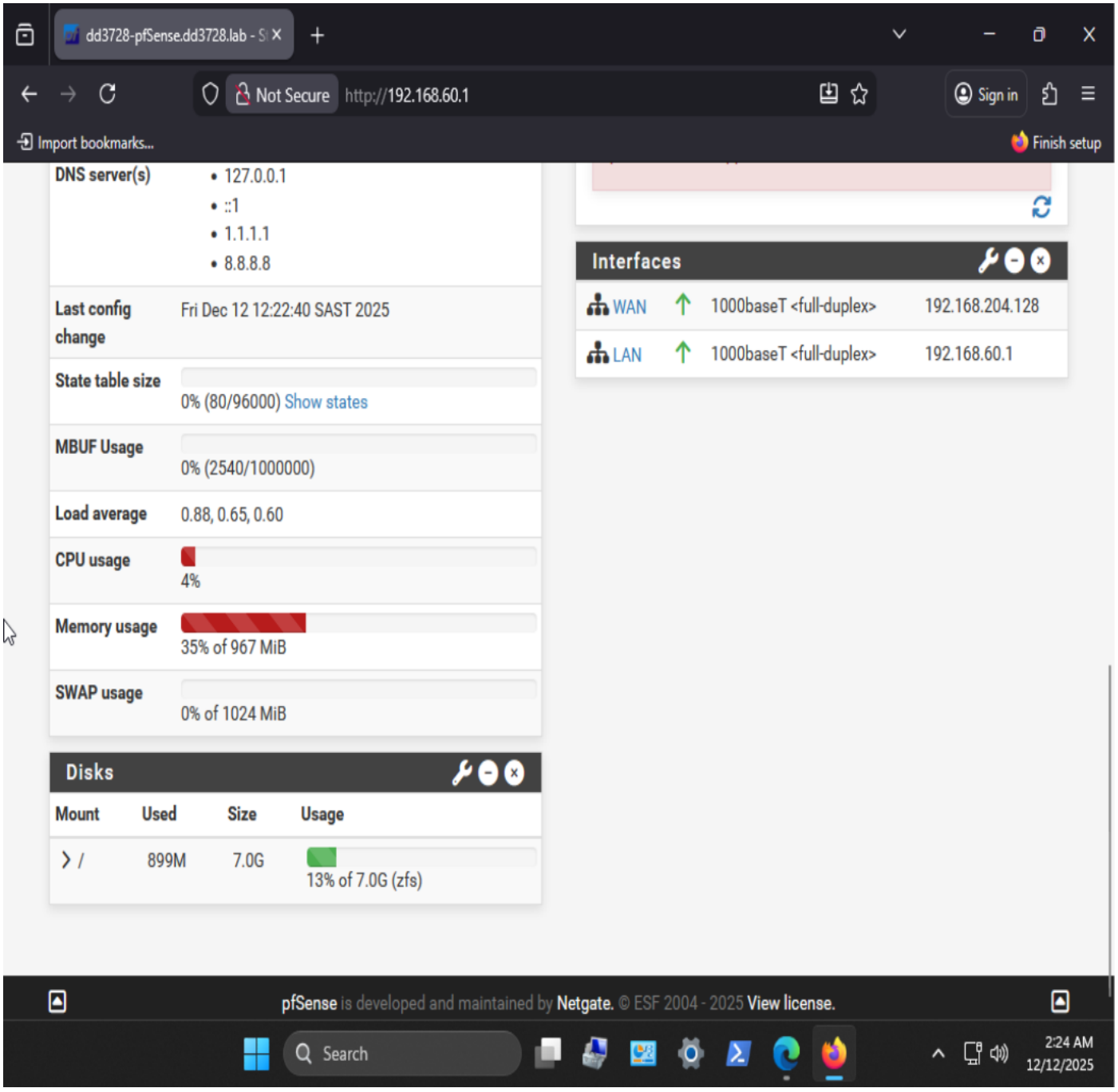
[**screenshots** 👆👇]



On pfSense GUI: Status → Interfaces:

- Confirmed WAN has an IP from VMnet8 (NAT).
- Confirmed LAN is `192.168.60.1/24`.

[**screenshots** 👆🏾👇🏾]

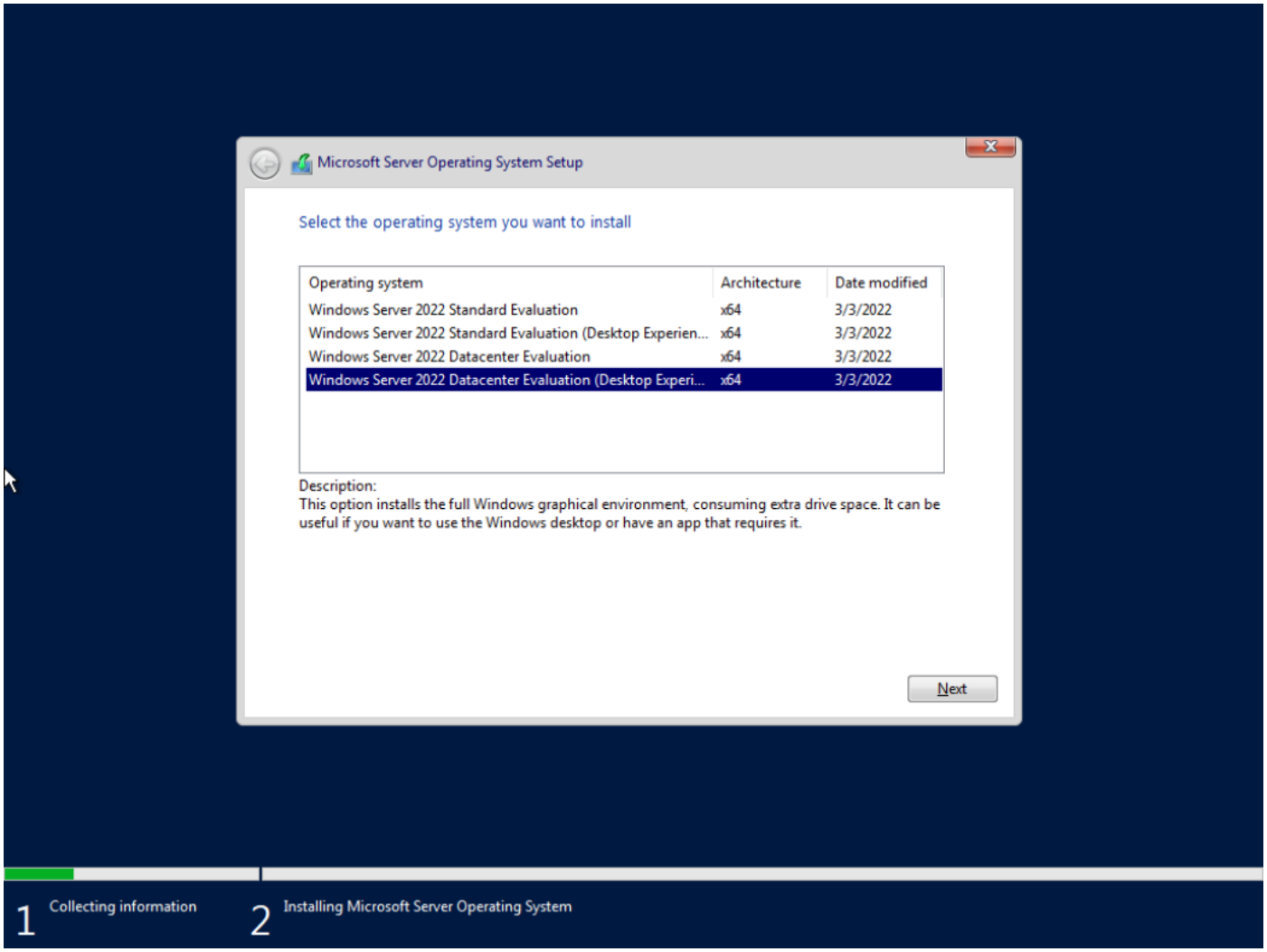


6 — Windows Server 2022: installed, DC promotion & DNS & DC troubleshooting ref: 6-(F)

Goal: Created digitaldefence3728.lab, made the DC authoritative for AD DNS, and validated Netlogon SRV registrations.

A. OS installation & static IP

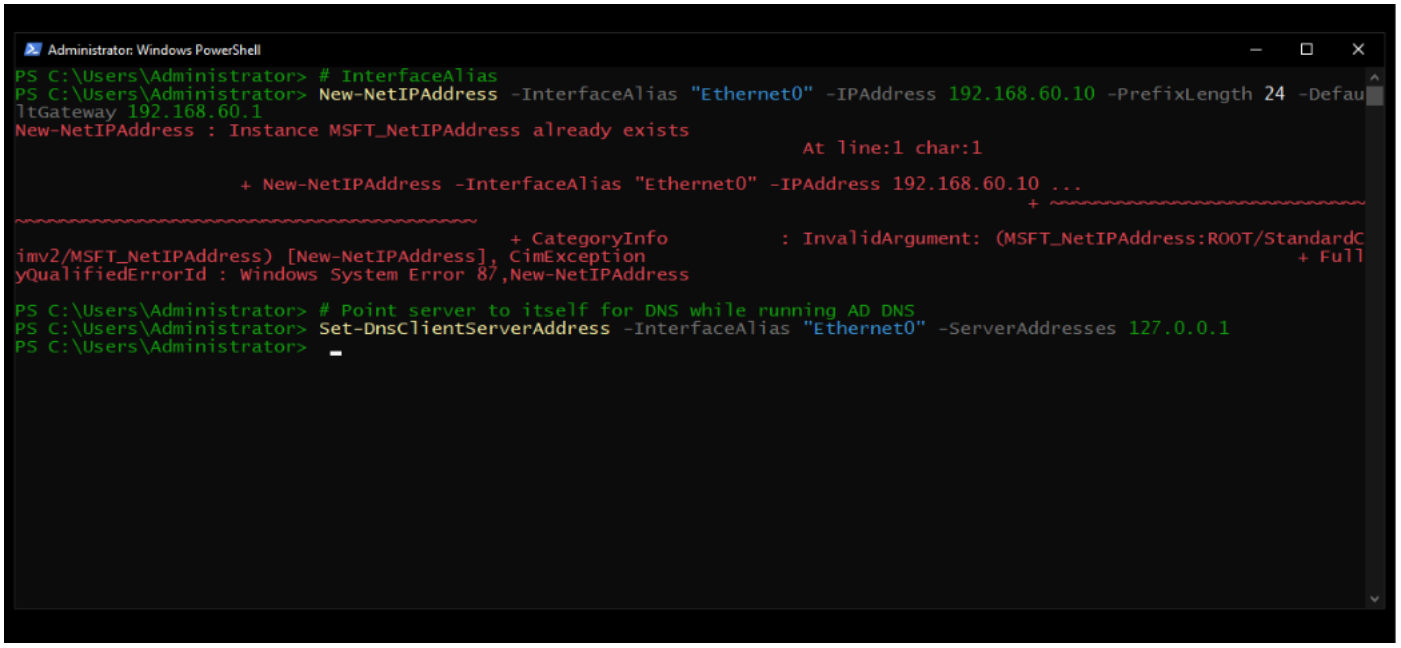
[**screenshots** 👆🏾👇🏾]



- After Windows Server installation and network driver readiness,
- configured static networking (ran as Administrator):

```
#powershell

New-NetIPAddress -InterfaceAlias "Ethernet0" -IPAddress
192.168.60.10 -PrefixLength 24 -DefaultGateway 192.168.60.1
Set-DnsClientServerAddress -InterfaceAlias "Ethernet0" -
ServerAddresses 127.0.0.1
```

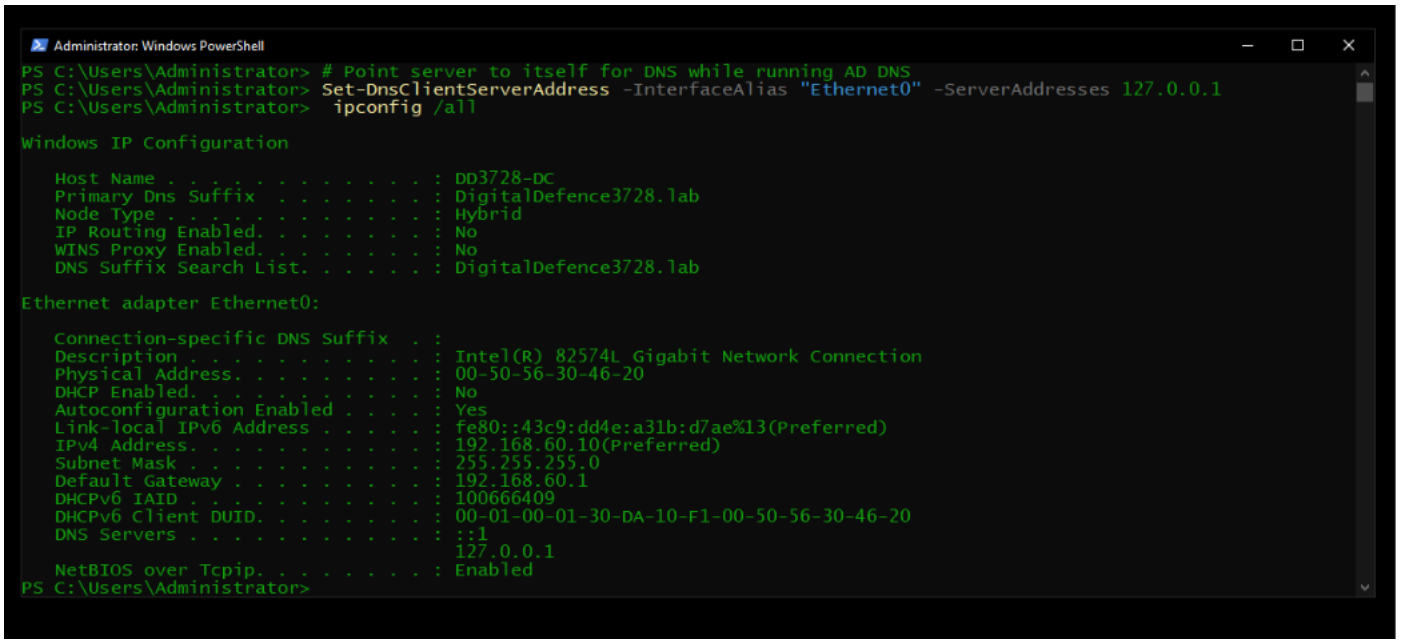
[**screenshots** 👆👇]



Verified:

```
#powershell

ipconfig /all
```

[**screenshots** 👆👇]



B. Installed AD DS role

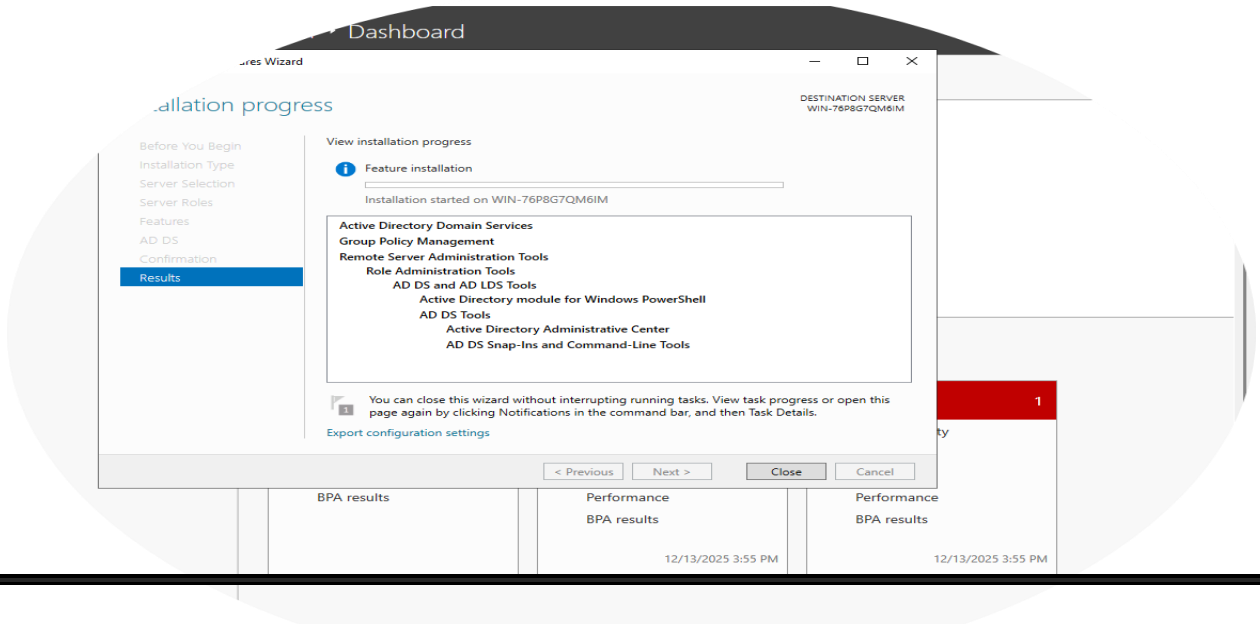
```
#powershell

Install-WindowsFeature AD-Domain-Services –IncludeManagementTools
```

Purpose

“By executing this command, the server will be prepared to function as a domain controller, allowing for centralized management of user and computer accounts in a domain environment.**”**

Note: AD DS was installed via GUI instead

[**screenshots** 👆👇]



C. Promote to forest root (new domain)

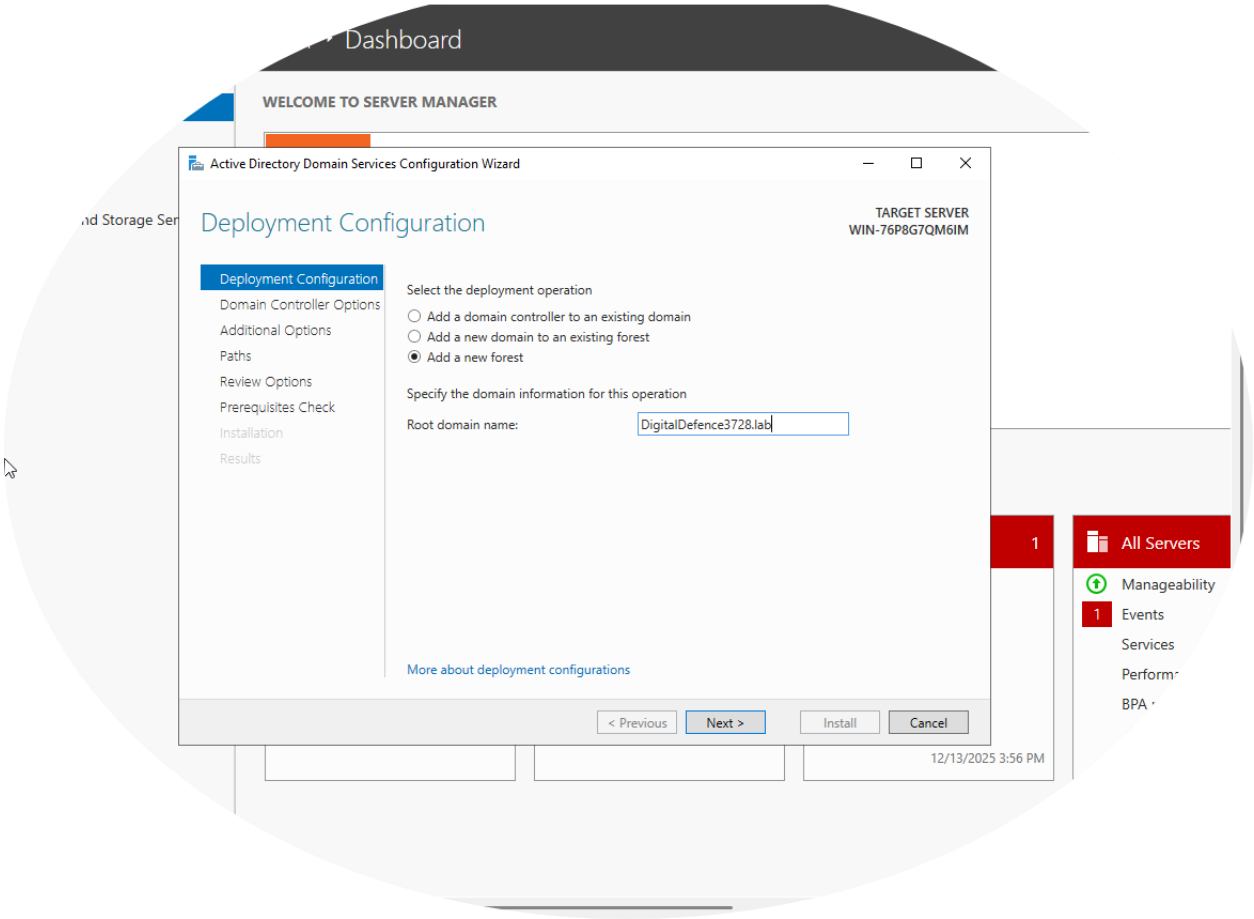
```
#powershell

Install-ADDSForest -DomainName "digitaldefence3728.lab" -
DomainNetbiosName "DD3728" -InstallDNS
```

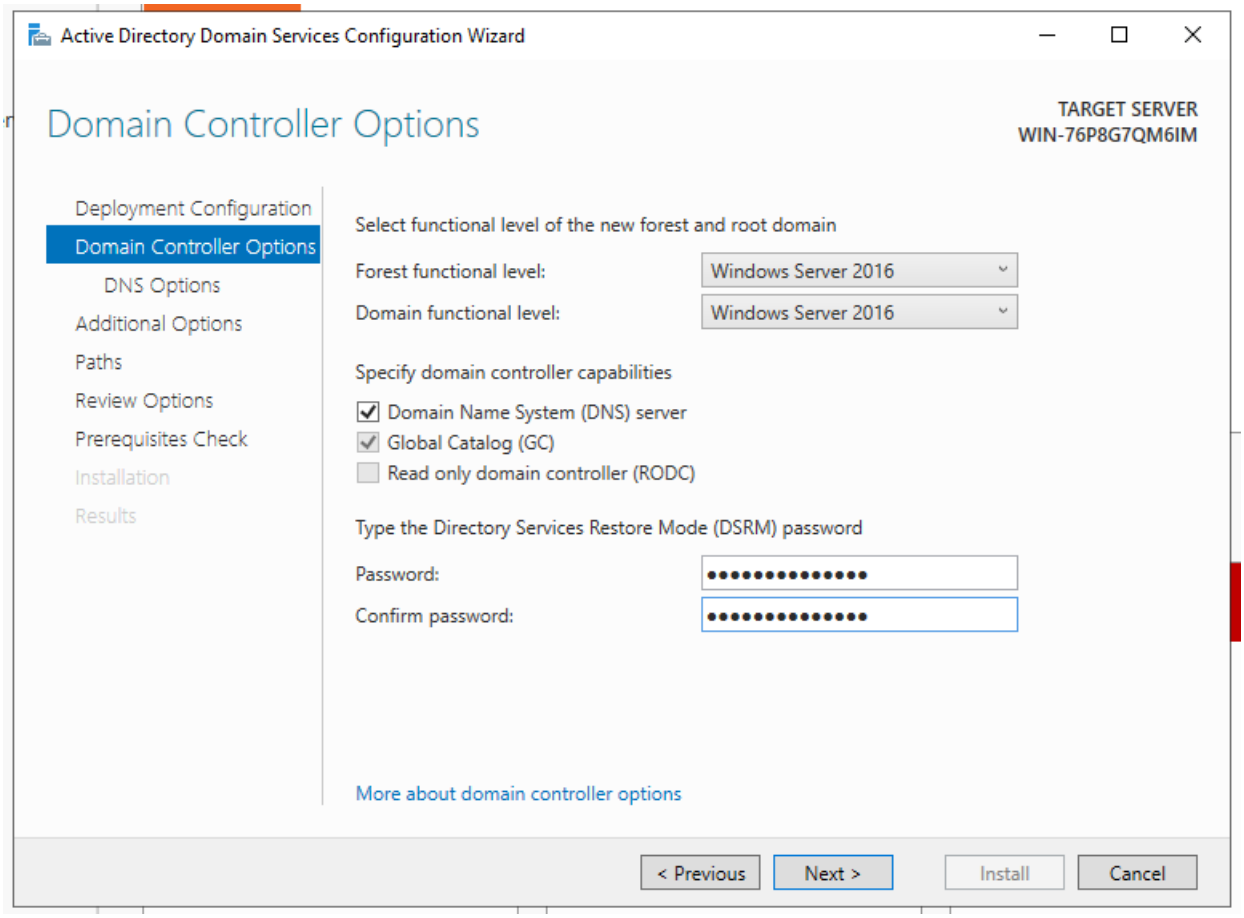
Purpose

”This command effectively sets up a new Active Directory forest named "digitaldefence3728.lab" with the NetBIOS name "DD3728" and installs the necessary DNS services needed for domain functionality. It serves as a foundational step in deploying Active Directory services.**”**

- Set Directory Services Restore Mode (DSRM) password when prompted.
- Rebooted server after promotion.

[**screenshots** 👆👇]





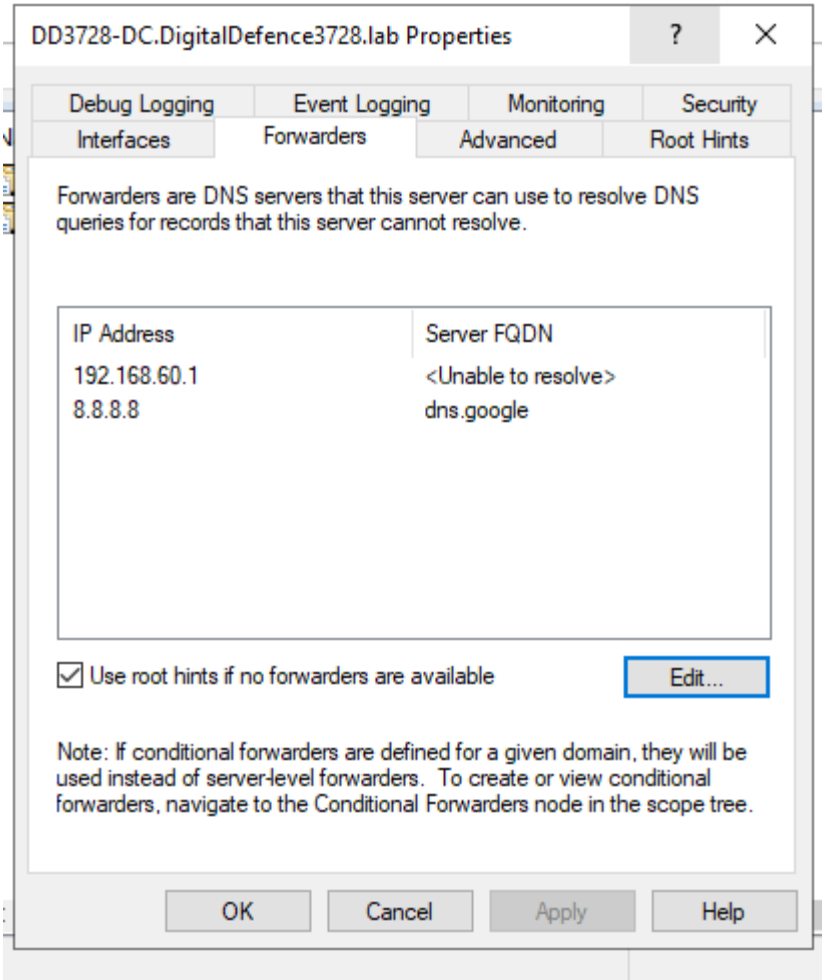
D. DNS forwarders

In DNS Manager → Server → Properties → Forwarders:

- Add pfSense `IP 192.168.60.1` and **FQDN** dns.google → `IP 8.8.8.8` a public resolver as a secondary forwarder for resilience.

“The Domain Controller is the authoritative DNS for the AD domain and therefore resolves all internal SRV/A records. For Internet name resolution, the DC forwards unknown queries to our gateway (pfSense) centralized outbound DNS, logging and policy enforcement. pfSense then forwards to public resolvers (e.g., dns.google). This split of responsibilities — DC = internal authority, pfSense = outbound resolver/filter — mirrors enterprise best practice and keeps Active Directory service discovery reliable and auditable.**”**

[**screenshots** 👆🏾👇🏾]



Server **FQDN**: <Unable to resolve> next to `192.168.60.1`

"That message simply says the DC cannot reverse-resolve the IP 192.168.60.1 to a hostname. It does not prevent forwarder functionality."

PowerShell equivalent:

```
#powershell

Add-DnsServerForwarder -IPAddress 192.168.60.1
Add-DnsServerForwarder -IPAddress 8.8.8.8
```

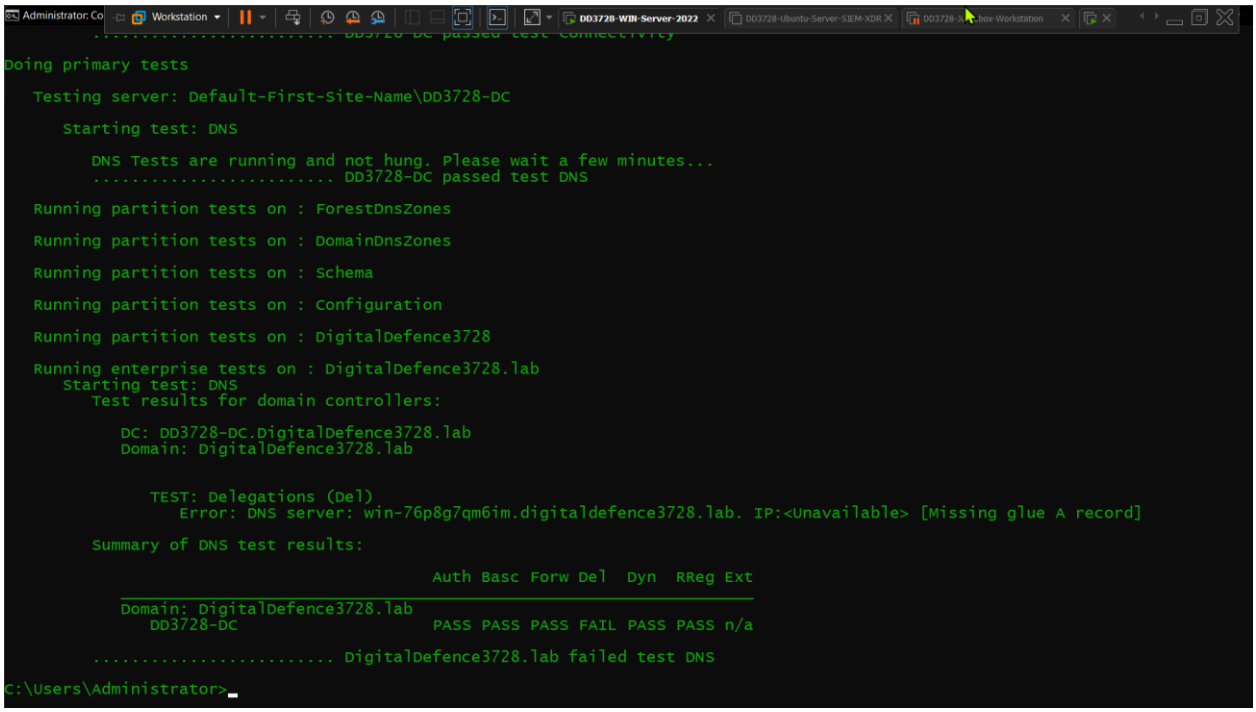
E. Netlogon / DNS registration & validation

`#powerShell [elevated]:`

```
ipconfig /registerdns
net stop netlogon
net start netlogon
nltest /dsregdns
dcdiag /test:dns
```

Expected: DNS tests **pass** with proper SRV records and no missing glue A records.

[**screenshot** 👆🏾👇🏾 **failed** ⚠️]



ℹ Reason why DNS test **failed** = ref: **DC troubleshoot** = **6_F** 👇🏾

I executed the following command sequence to ensure proper DNS registration and functionality related to the Netlogon service on the domain controller. This is critical for maintaining network authentication and domain operations.

Command Breakdown

ipconfig /registerdns: forces the computer to register its DNS records to the DNS server. Essential for allowing other devices to locate the domain controller.

net stop netlogon: Stops the Netlogon service, which is responsible for facilitating user authentication and accounts in the domain. Stopping it may be necessary to refresh its state.

net start netlogon: Restarts the Netlogon service, re-establishing connections and ensuring it can communicate effectively with the domain controller.

nltest /dsregdns: This command triggers registration of the domain controller's DNS records. It confirms that the domain controller is properly registered in the domain.

dcdiag /test:dns: This diagnostic tool tests the DNS health of the domain controller, ensuring that DNS queries and configurations are functioning correctly. It helps identify any configuration issues that may affect network operations.

Purpose

“This sequence of commands was executed to validate and refresh the DNS registration of the domain controller, ensuring optimal communication and authentication across the network.**”**

F. DC troubleshooting

DC hostname was changed after promotion:

1. Removed stale A and PTR records in DNS Manager for the old hostname

   (**WIN-78P8G7QM6IM**)

★ Through **DC event view** I managed to trace when the last was I logged off prior to hostname change.

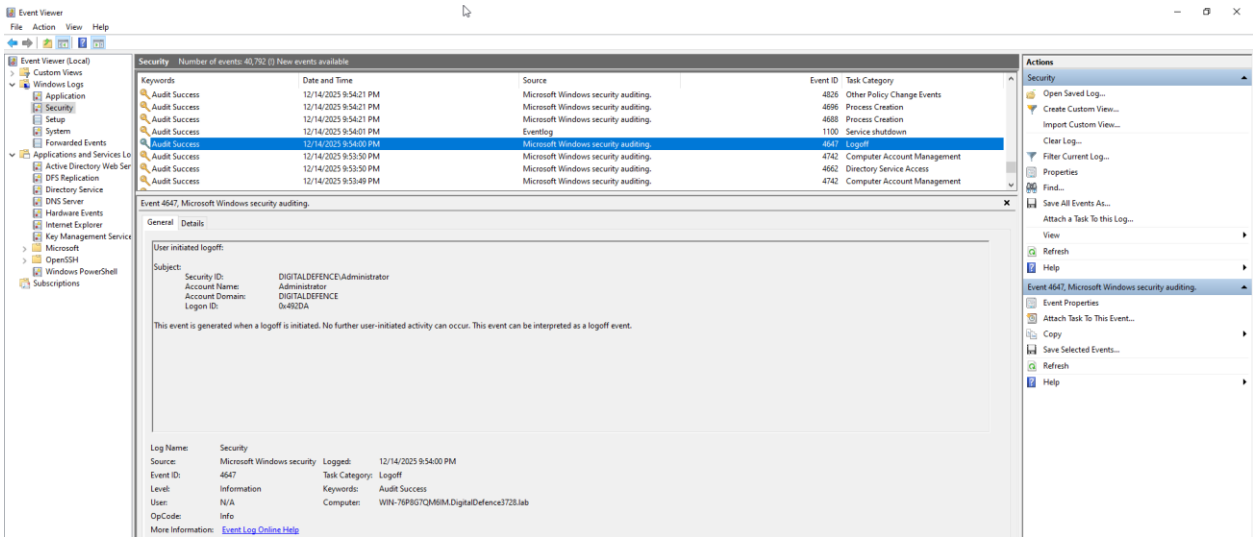
Channel: Security

Computer**: WIN-78P8G7QM6IM _ WIN76P8G7QM6IM.DigitalDefence3728.lab**

12/14/2025 9:54:00 PM

[**screenshots** 👆🏽👇🏽]

old host name → **WIN76P8G7QM6IM.DigitalDefence3728.lab**

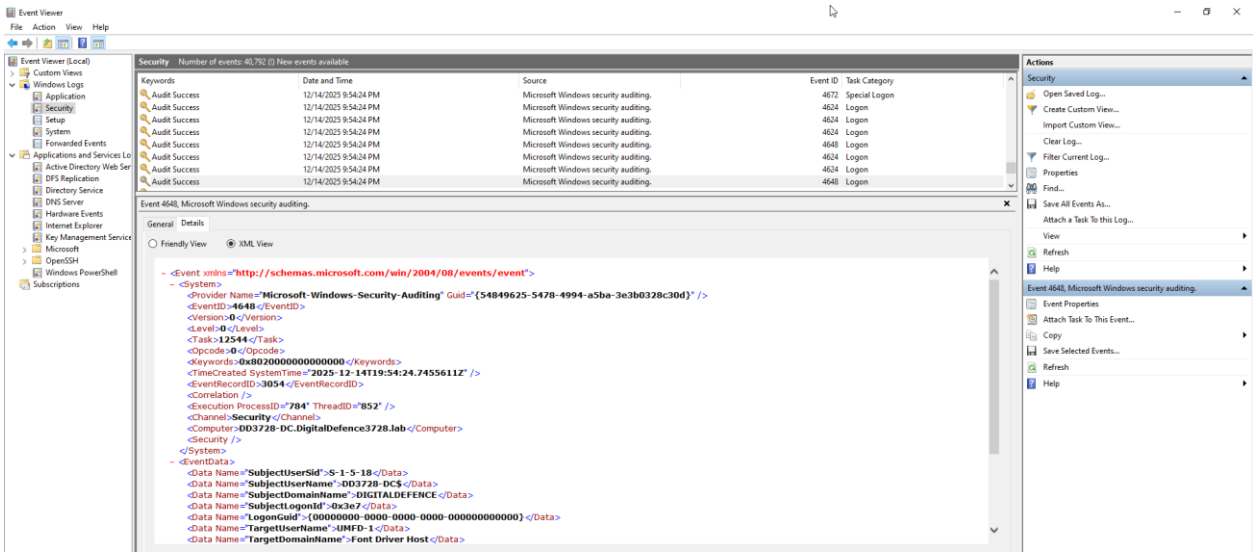


2. Created correct A and PTR records for the new hostname (`DD3728-DC → 192.168.60.10`).

   DD3728-DC → digitaldefence3728.lab

[screenshots 👆👇]

XML view

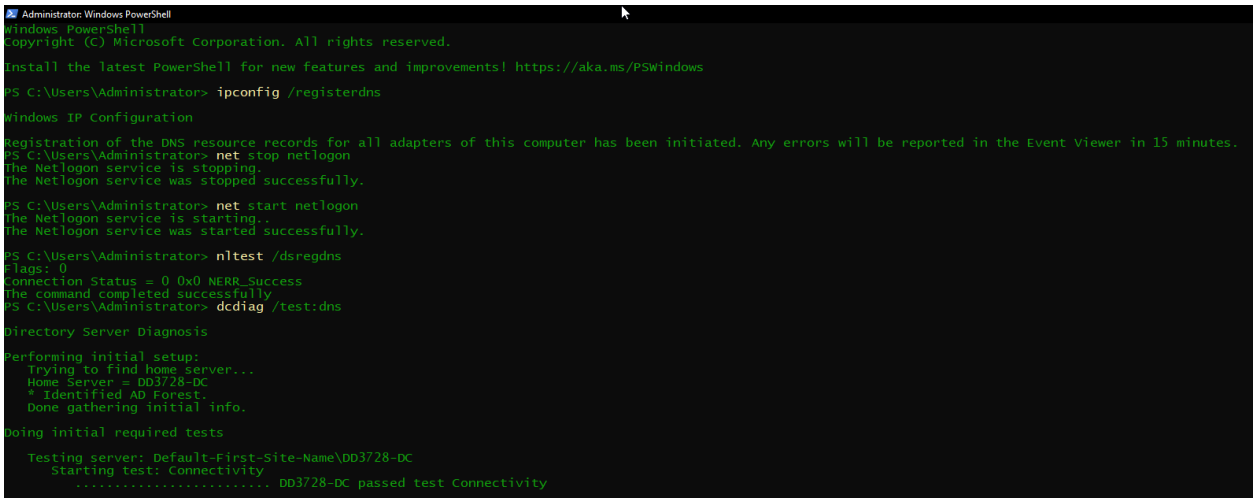


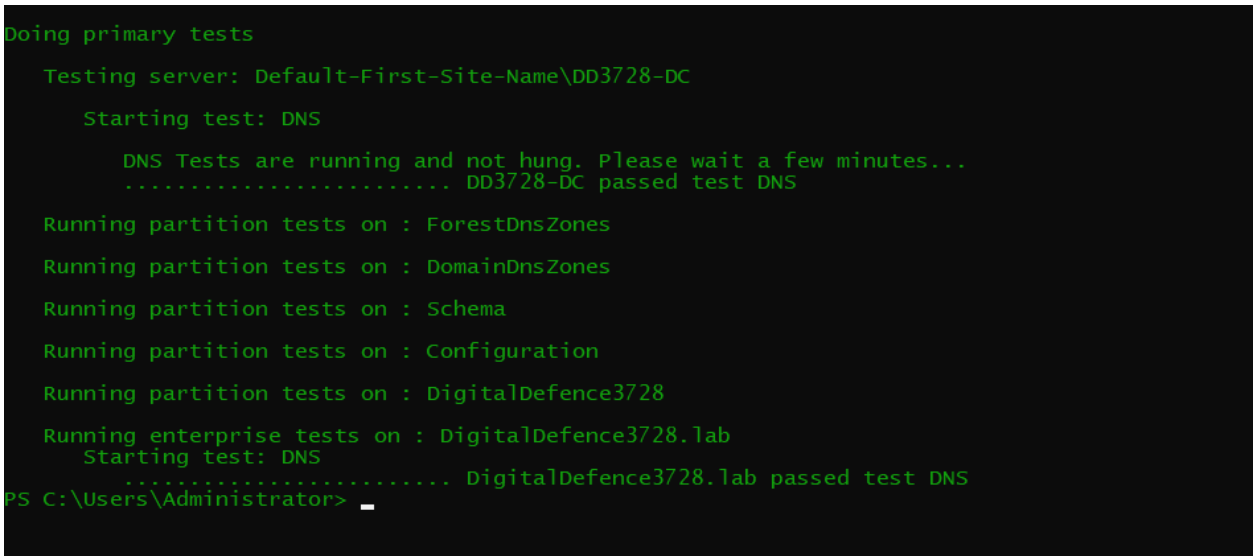
3. **Re-ran:**

```
#powershell

ipconfig /registerdns
net stop netlogon
net start netlogon
```

```
nltest /dsregdns
dcdiag /test:dns
```

[**screenshot** 👆🏽👇🏽] DNS tests **pass** ✅





G. Validation

```
#powershell

dcdiag /v
nslookup
set type=SRV
_ldap._tcp.dc._msdcs.digitaldefence3728.lab
```

Command Breakdown 👇🏽

```
dcdiag /v:
```

Purpose: This command runs the Domain Controller Diagnostic tool in verbose mode.

Function: It checks the health of the domain controller and provides detailed output about various aspects, such as replication, connectivity, and DNS configurations. This helps identify potential issues affecting Active Directory functionality.

`nslookup`:

Purpose: This command is a network administration command-line tool used for querying the Domain Name System (DNS).

Function: It allows you to obtain information about domain names and their corresponding IP addresses.

`set type=SRV`:

Purpose: This command sets the query type to SRV (Service) records.

Function: SRV records are used to locate specific services, including Active Directory services. By setting up the type to SRV, subsequent queries will focus on these service records.

`_ldap._tcp.dc._msdcs.digitaldefence3728.lab`:

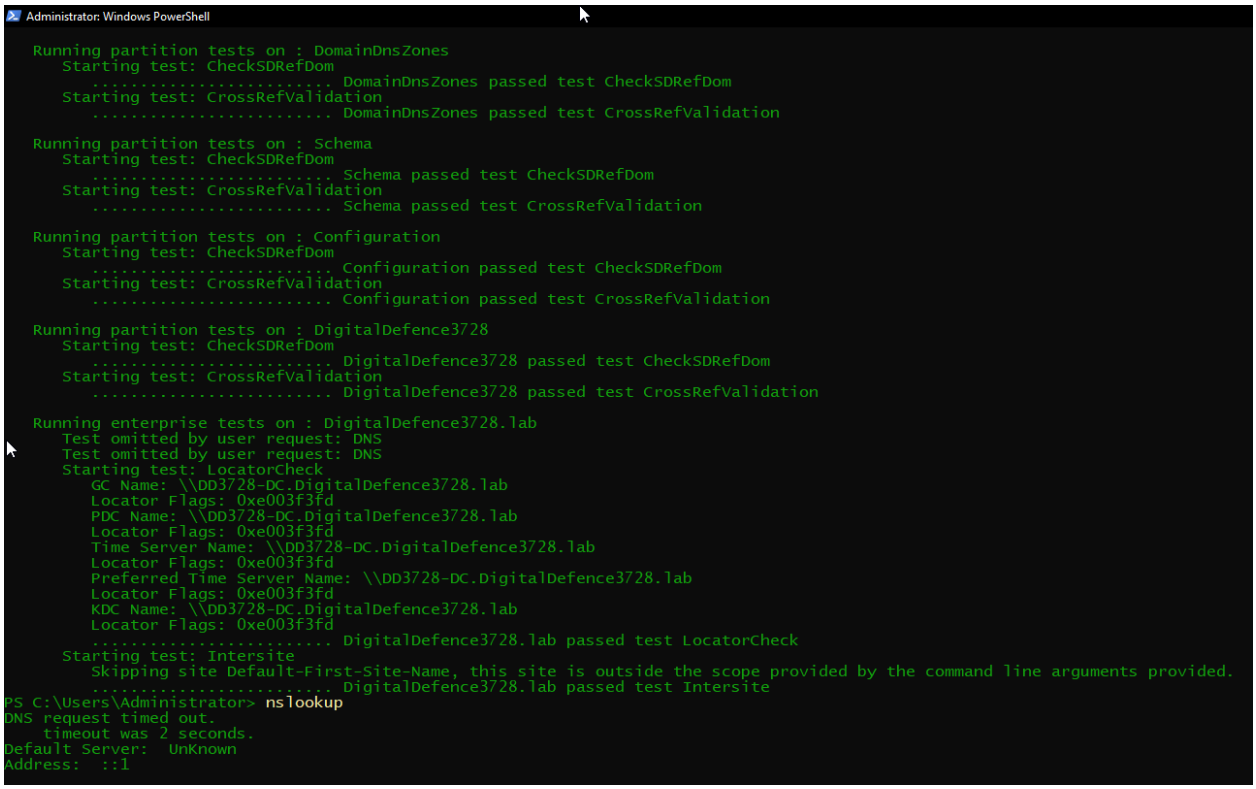
Purpose: This is a specific DNS query for locating the LDAP service associated with domain controllers in the "`digitaldefence3728.lab`" domain.

Function: It queries the DNS for SRV records that provide information about LDAP servers that can authenticate users or applications for the specified domain.

Purpose [overall]

Together, these commands are executed to diagnose the health of a domain controller and query the DNS for LDAP service records. This is essential for ensuring that Active Directory services are accessible and functioning correctly, which is critical for user authentication and domain-related tasks.

[**screenshot** 👇🏿]



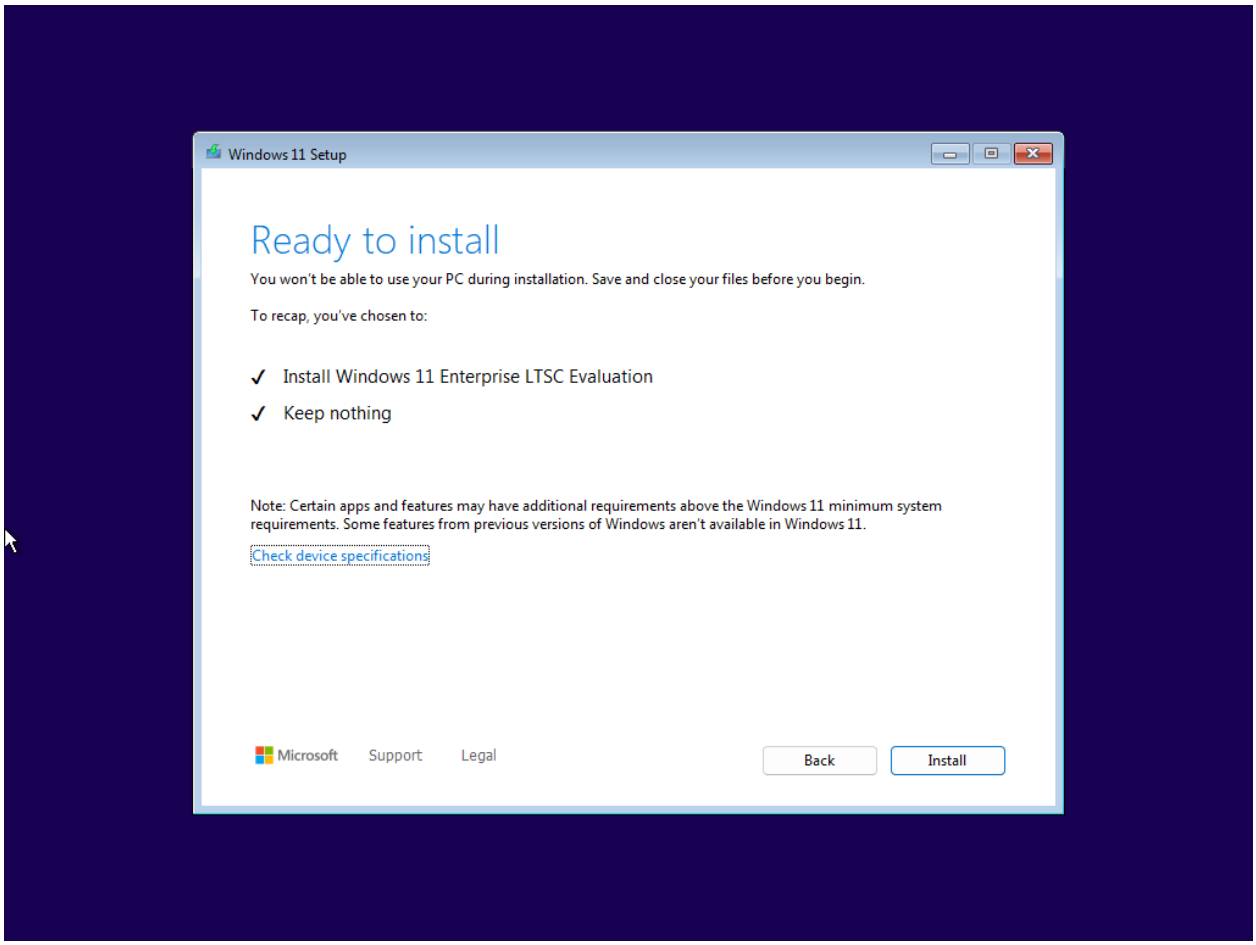
- Expected to see the **DD3728-DC** host in the SRV results.

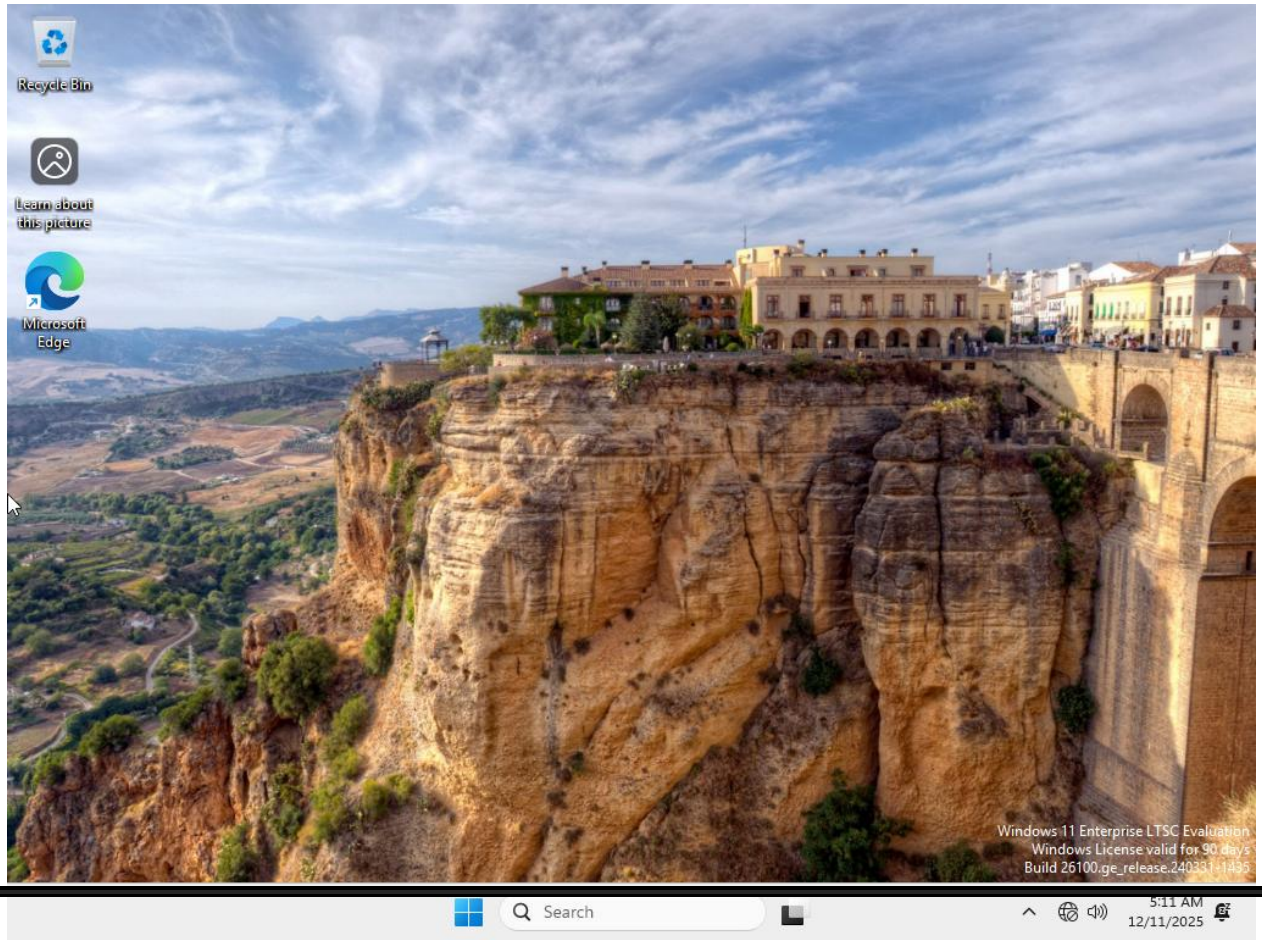
7 — Jumpbox (Windows 11 LTSC): install, hardening & tools

Goal: Analyst workstation for accessing Splunk (web), SSH to Linux, and RDP to Windows Server, with hardened baseline and investigation tools.

A. Installation:

[**screenshots** 👆🏽👇🏽]


Windows 11 Setup
Ready to install
You won't be able to use your PC during installation. Save and close your files before you begin.
To recap, you've chosen to:
✓ Install Windows 11 Enterprise LTSC Evaluation
✓ Keep nothing
Note: Certain apps and features may have additional requirements above the Windows 11 minimum system requirements. Some features from previous versions of Windows aren't available in Windows 11.
Check device specifications
Microsoft
Support
Legal
Back
Install


Recycle Bin
Learn about this picture
Microsoft Edge
Windows 11 Enterprise LTSC Evaluation
Windows License valid for 90 days
Build 26100.ge_release.240331-1435
Search
5:11 AM
12/11/2025

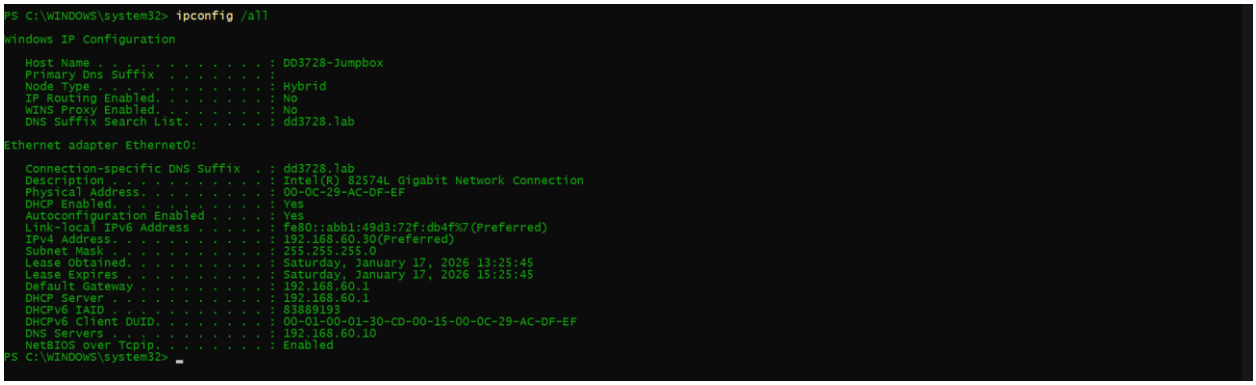
B. Networking

#powershell

```
New-NetIPAddress -InterfaceAlias "Ethernet" -IPAddress
192.168.60.30 -PrefixLength 24 -DefaultGateway 192.168.60.1
Set-DnsClientServerAddress -InterfaceAlias "Ethernet" -
ServerAddresses 192.168.60.10
```

- Assigned a static IP to the Jumpbox to ensure stable management access and predictable firewall rules.

[**screenshots** 👆🏾👇🏾]



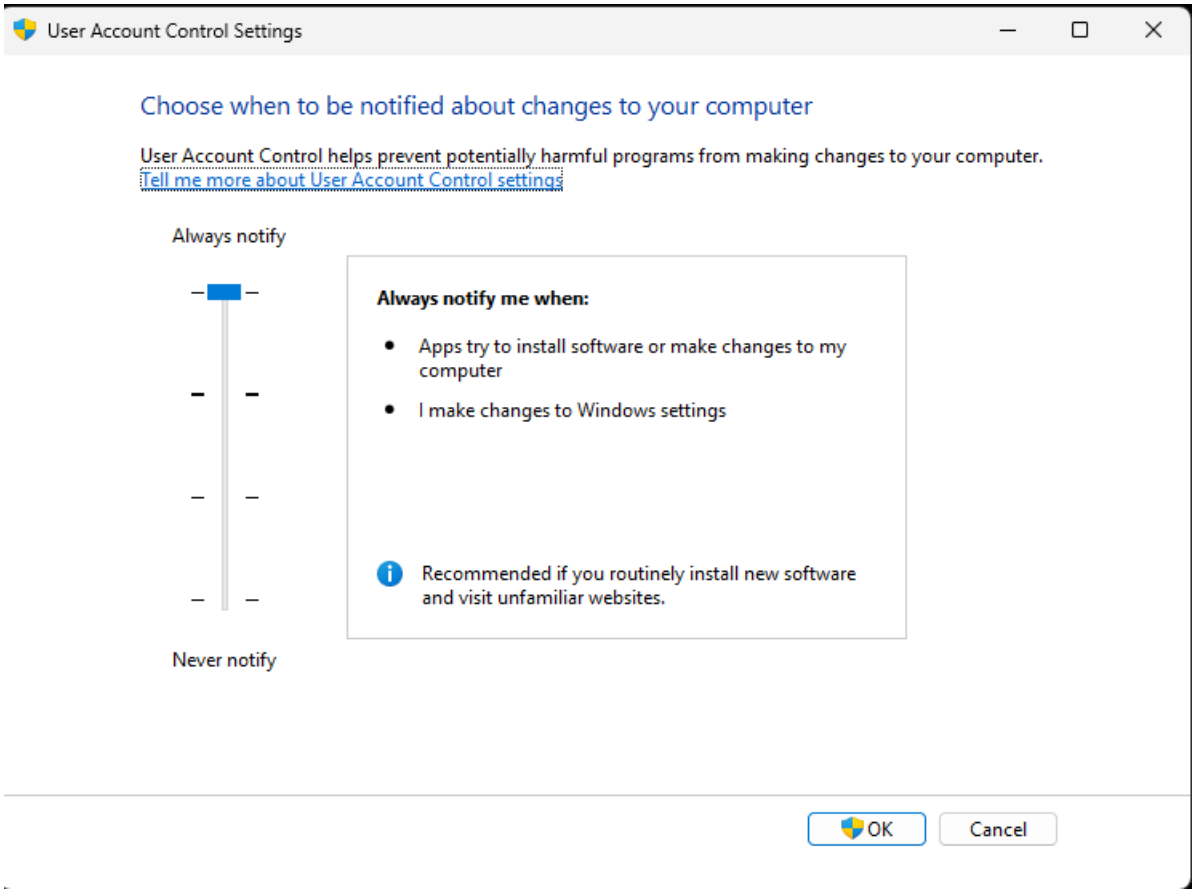
i. Hardening baseline

- Enabled Windows Defender and tamper protection.
- Prevents malware and stops local users from disabling security controls.

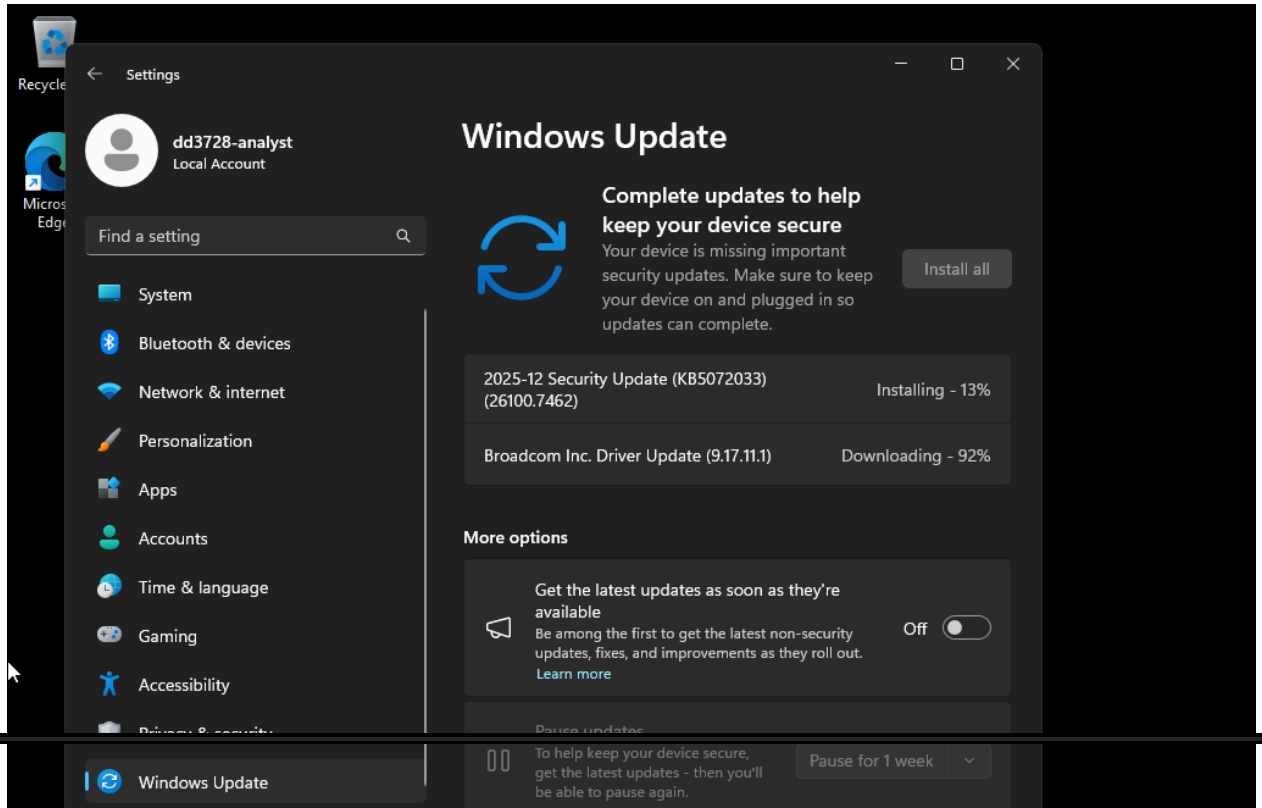
[**screenshots** 👆🏾👇🏾]

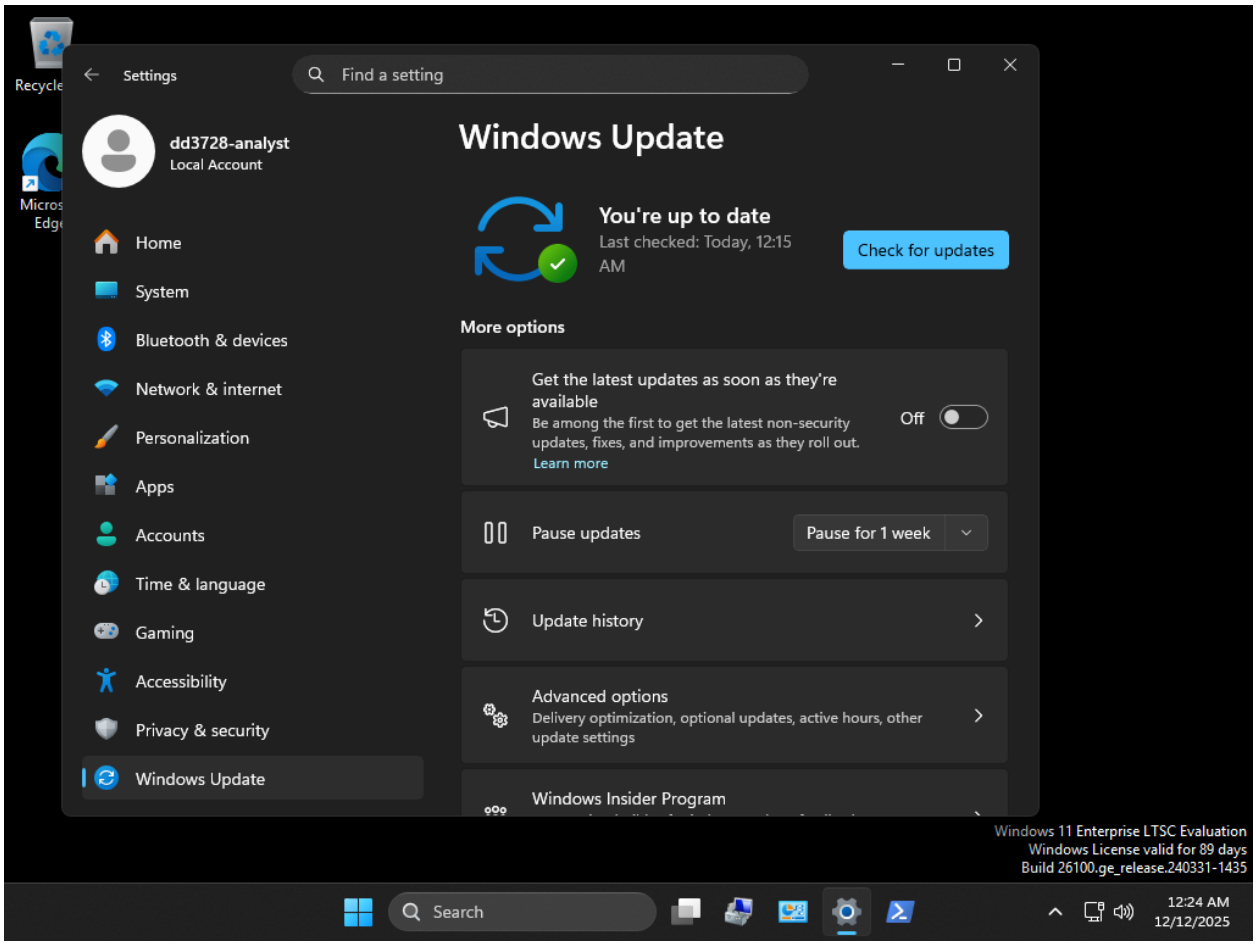
Set UAC to “Always notify”.

- Enforces explicit privilege escalation; reduces silent admin abuse.



- Windows updates



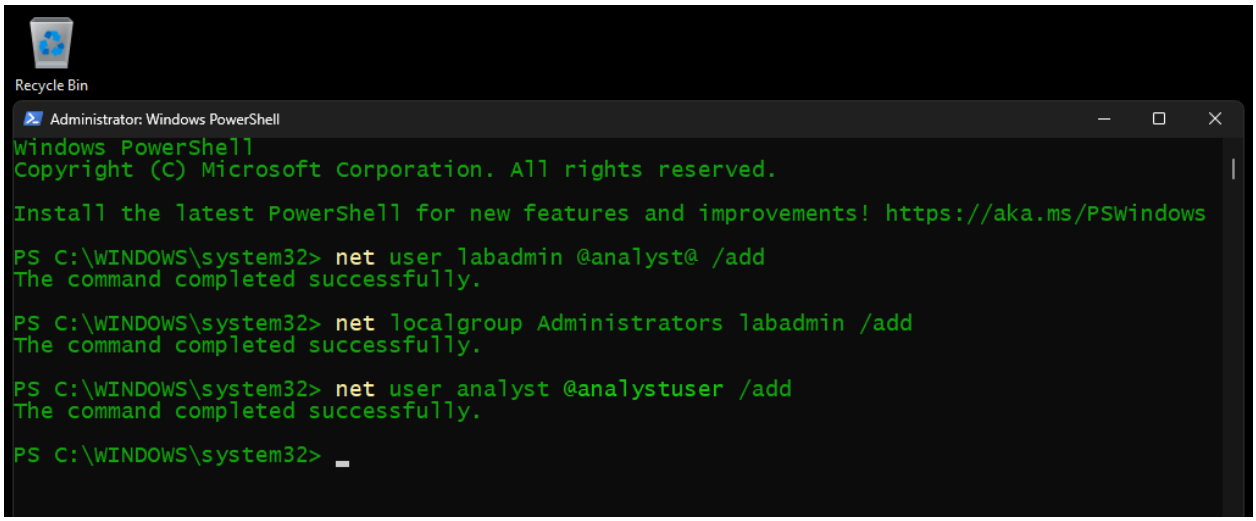


Create separate admin and analyst accounts:

```
#powershell

net user labadmin ************** /add
net localgroup Administrators labadmin /add
net user analyst ************** /add
```

- Creates a dedicated administrative account to avoid daily-use admin risk.
- Creates a low-privilege analyst account aligned with least-privilege principles.

[**screenshots** 👆🏾👇🏾]



C. Tools and rationale

Installed:

- MobaXterm — SSH/SFTP/X11 client to manage the Linux SIEM and transfer files.
- RDP (built-in) — for GUI admin to the DC and client.
- Browser (Firefox/Brave) — primary Splunk Web client for triage.
- Sysinternals Suite — process and autoruns analysis on Windows endpoints.
- Wireshark — packet capture and network triage.
- PowerShell 7 — modern scripting and remoting.
- KeePass — secure credential vault.
- Visual Studio Code (portable) — scripting and configuration editing.

Chocolatey installed:

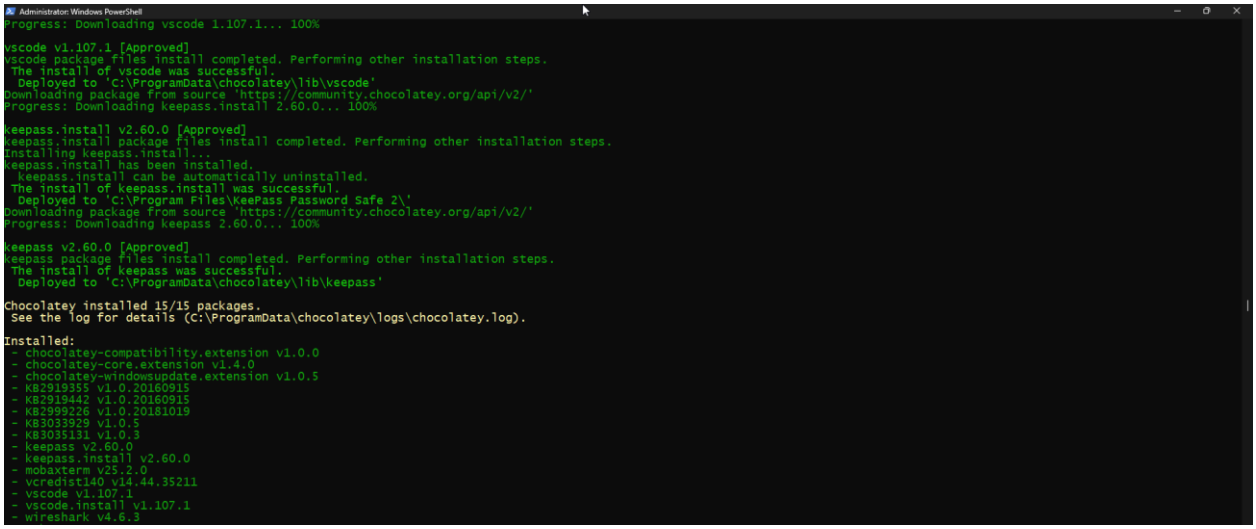
```
#powershell

Set-ExecutionPolicy Bypass -Scope Process -Force
iwr https://chocolatey.org/install.ps1 -UseBasicParsing | iex
```

- Temporarily allows script execution for package installation only.
- Installs Chocolatey for repeatable, auditable tool deployment.

```
choco install mobaxterm wireshark vscode keepass -y
```

- Installs core tools quickly and consistently.

[screenshots 👆👇]



D. Quick hardening commands

```
#powershell

Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled
True
```

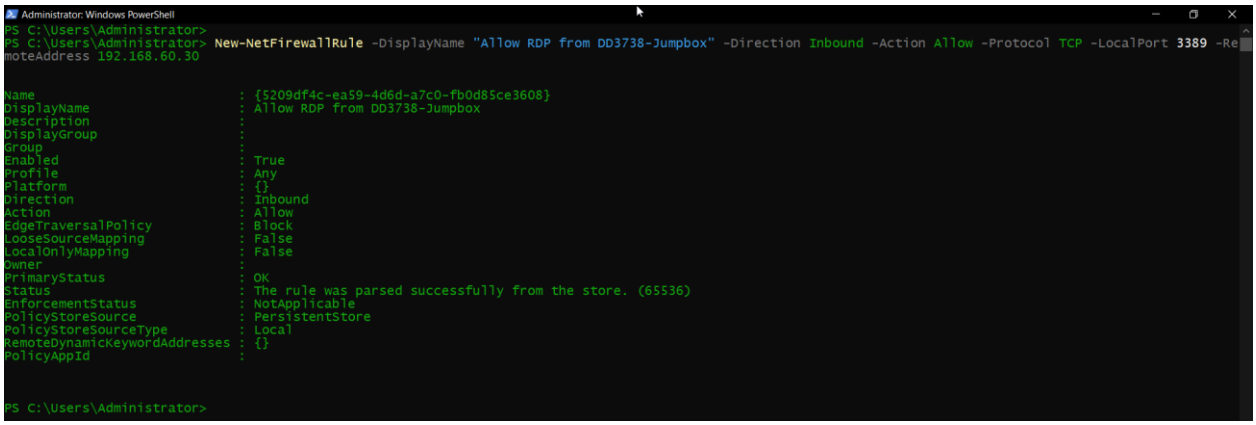
- Ensures that the Windows Firewall is active across all network profiles.

Firewall rule (ran on DC) to allow RDP only from the Jumpbox:

```
#powershell

New-NetFirewallRule -DisplayName "Allow RDP from DD3738-Jumpbox" -
Direction Inbound -Action Allow -Protocol TCP -LocalPort 3389 -
RemoteAddress 192.168.60.30
```

- Ensures that the Windows Firewall is active across all network profiles.
- Restricts RDP access only to the Jumpbox, reducing lateral-movement risk.

[screenshots 👆👇]

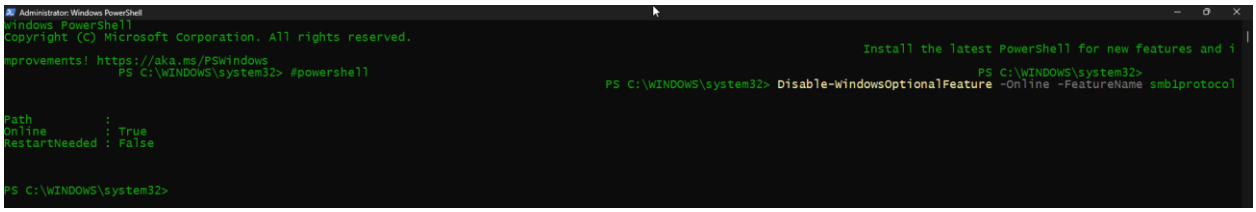


Disable SMBv1:

```
#powershell

Disable-WindowsOptionalFeature -Online -FeatureName smb1protocol
```

- Removes SMBv1, eliminating a known legacy attack vector (e.g., EternalBlue).

[**screenshots** 👆🏽👇🏽]



“These steps show deliberate network design, strict access control, security-first hardening, and disciplined tooling — exactly how a production-grade SOC or SIEM management environment is built.**”**

E. Access validation

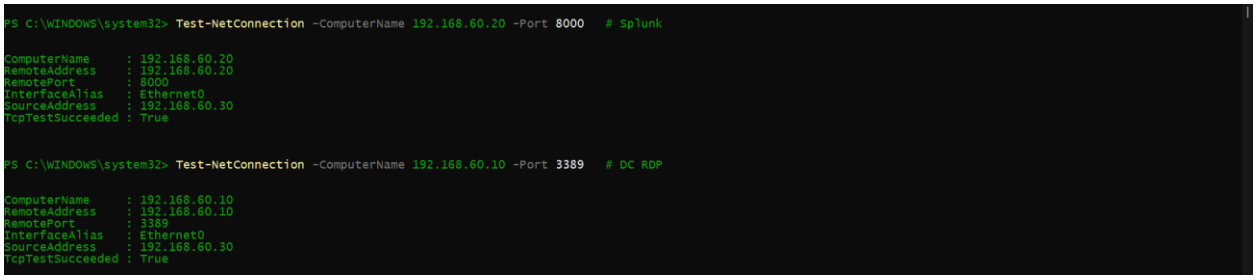
From Jumpbox:

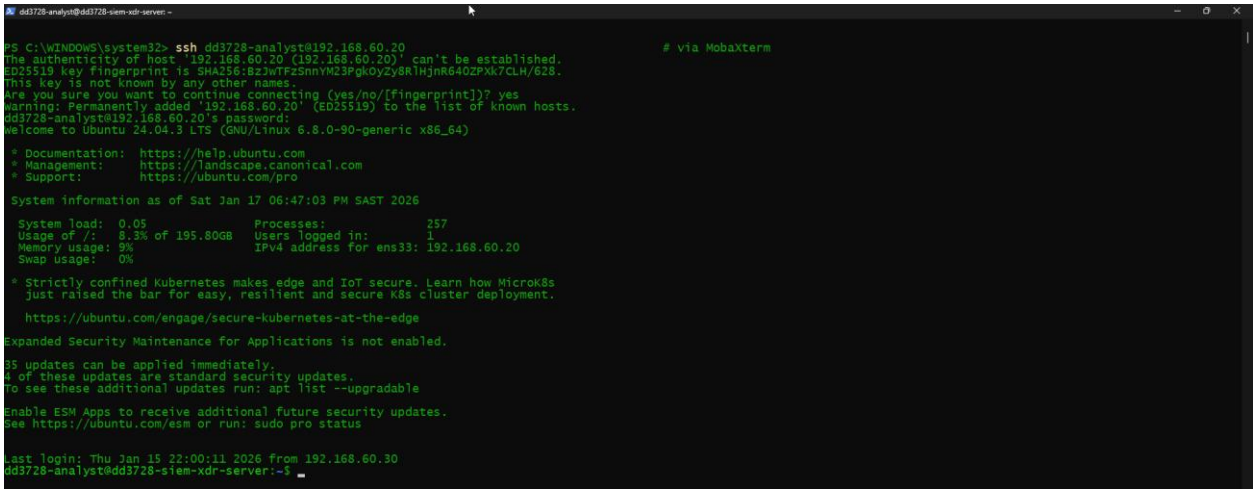
```
#powershell

Test-NetConnection -ComputerName 192.168.60.20 -Port 8000
#Splunk
Test-NetConnection -ComputerName 192.168.60.10 -Port 3389
#DC RDP
```

ssh dd3728-analyst@192.168.60.20

- Confirms Splunk Web is reachable from the Jumpbox.
- Validates controlled RDP access to the Domain Controller.
- Verifies secure SSH access to the Splunk server.

[**screenshots** 👆🏾👇🏾]





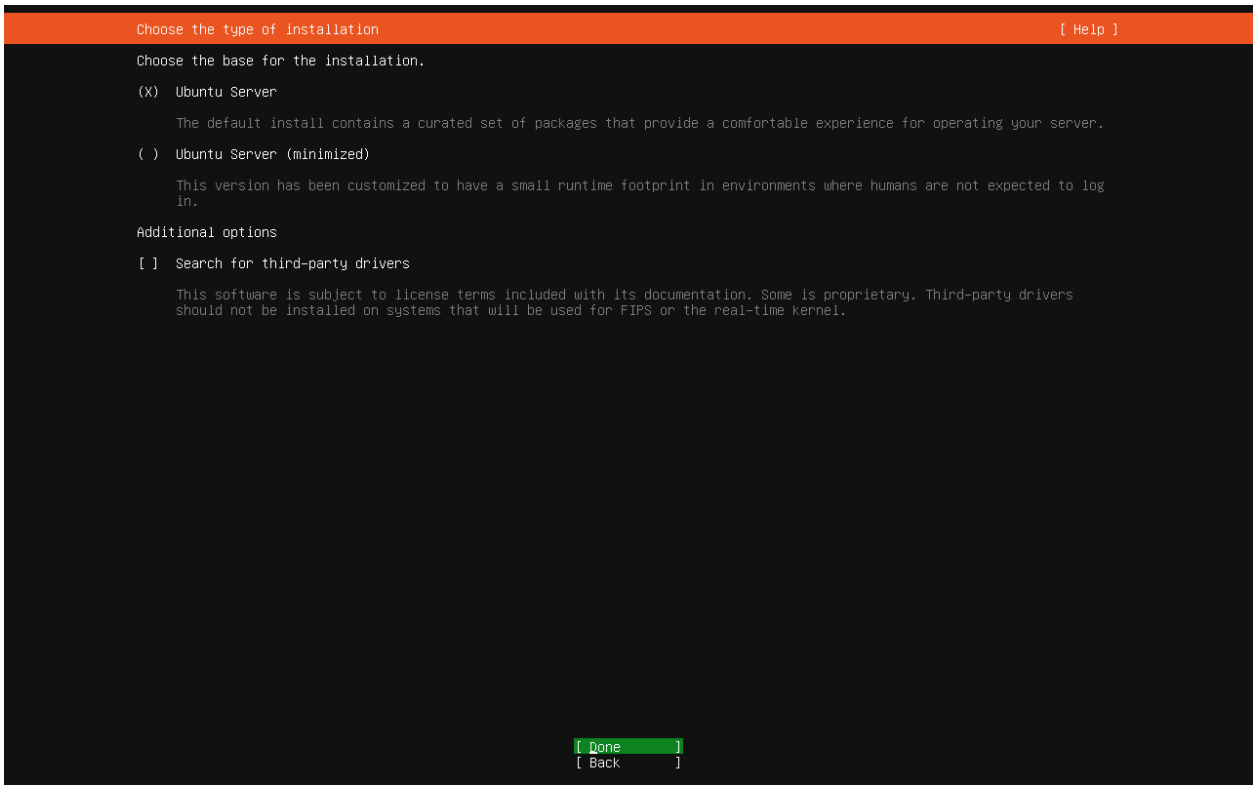
8 — Ubuntu Server 24.04 LTS: installation, networking, hardening & Splunk

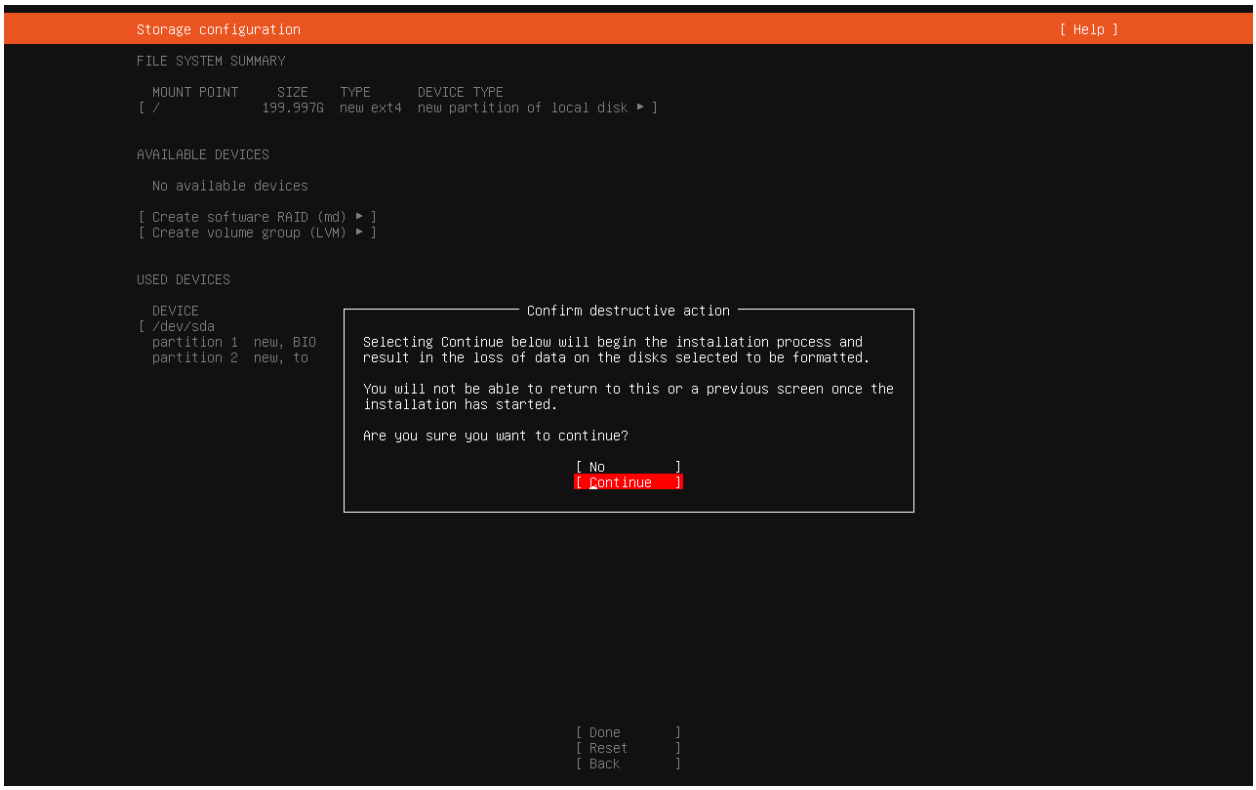
Goal: Minimal GUI-less server with static network, hardened baseline, and Splunk Enterprise configured to receive logs.

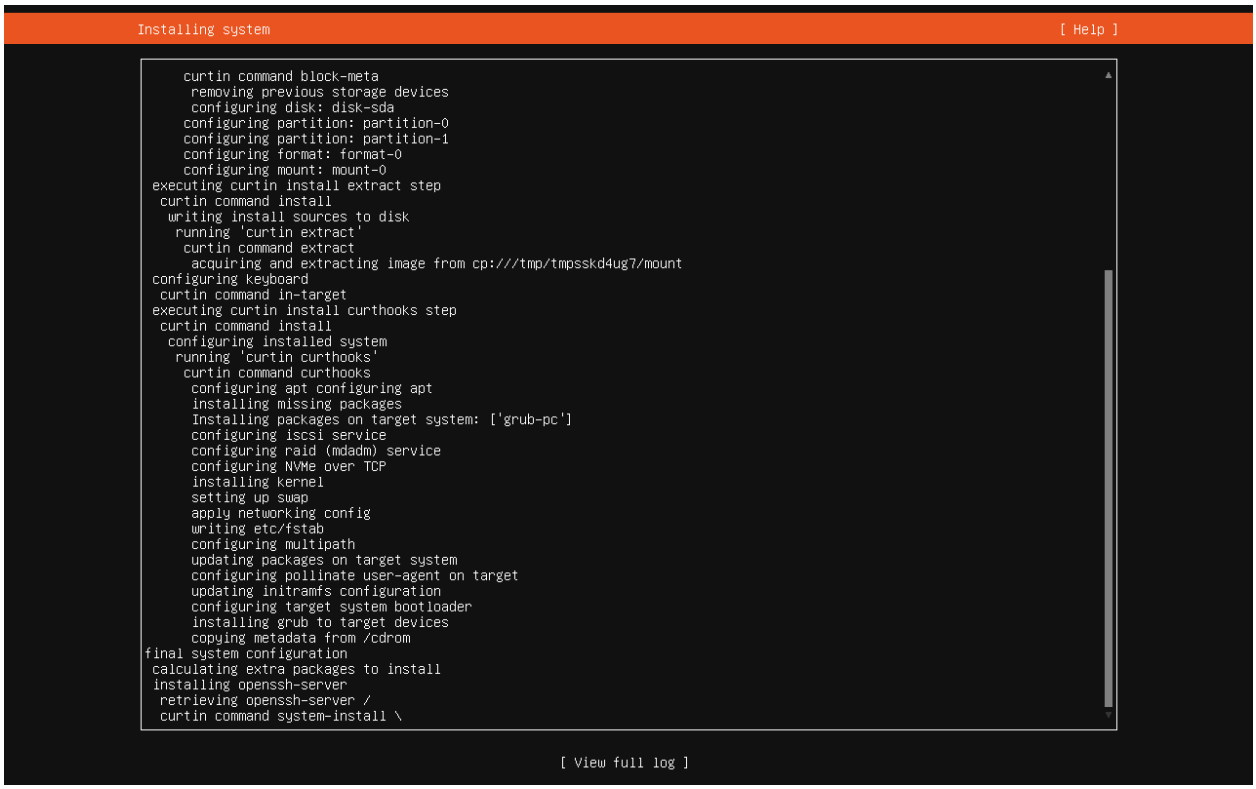
A. Install Ubuntu Server 24.04

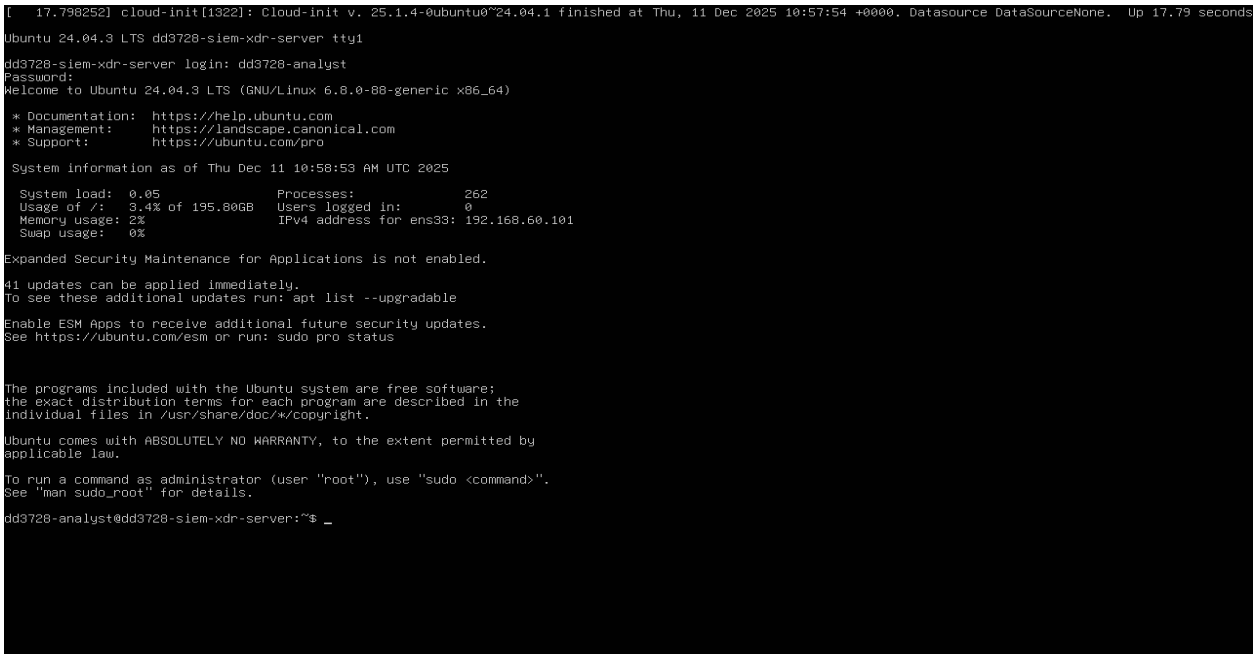
- Used Ubuntu Server ISO

[***screenshots*** 👇🏾 **selected** ✅]


Choose the type of installation
[Help]
Choose the base for the installation.
(X) Ubuntu Server
The default install contains a curated set of packages that provide a comfortable experience for operating your server.
( ) Ubuntu Server (minimized)
This version has been customized to have a small runtime footprint in environments where humans are not expected to log in.
Additional options
[ ] Search for third-party drivers
This software is subject to license terms included with its documentation. Some is proprietary. Third-party drivers should not be installed on systems that will be used for FIPS or the real-time kernel.
[Done]
[Back]


Storage configuration
[Help]
FILE SYSTEM SUMMARY
MOUNT POINT SIZE TYPE DEVICE TYPE
[/ 199.997G new ext4 new partition of local disk ▸]
AVAILABLE DEVICES
No available devices
[Create software RAID (md) ▸]
[Create volume group (LVM) ▸]
USED DEVICES
DEVICE
[/dev/sda
partition 1 new, BIO
partition 2 new, to
Confirm destructive action
Selecting Continue below will begin the installation process and result in the loss of data on the disks selected to be formatted.
You will not be able to return to this or a previous screen once the installation has started.
Are you sure you want to continue?
[No]
[Continue]
[Done]
[Reset]
[Back]

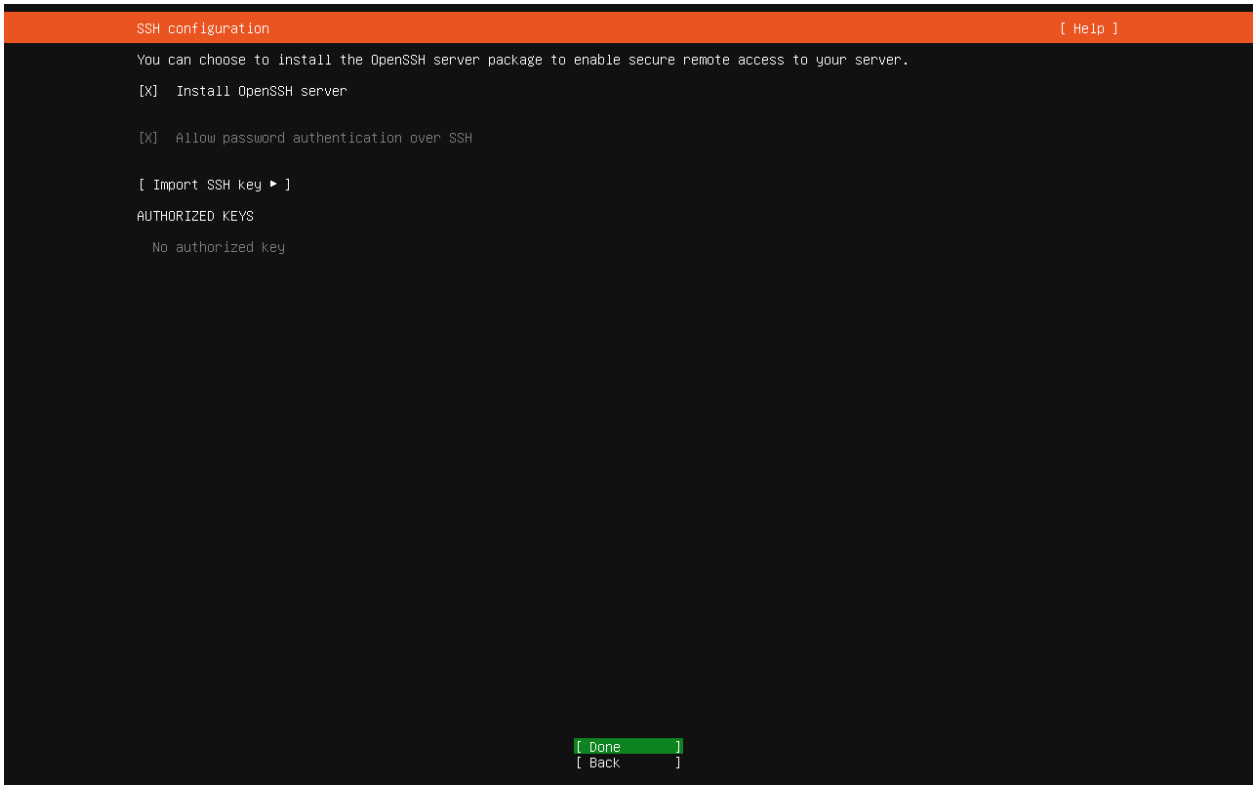




- enabled SSH during installation.
- **Install OpenSSH serve**r: This option allows you to install the OpenSSH server package, which is essential for remote access to the server via the SSH protocol.

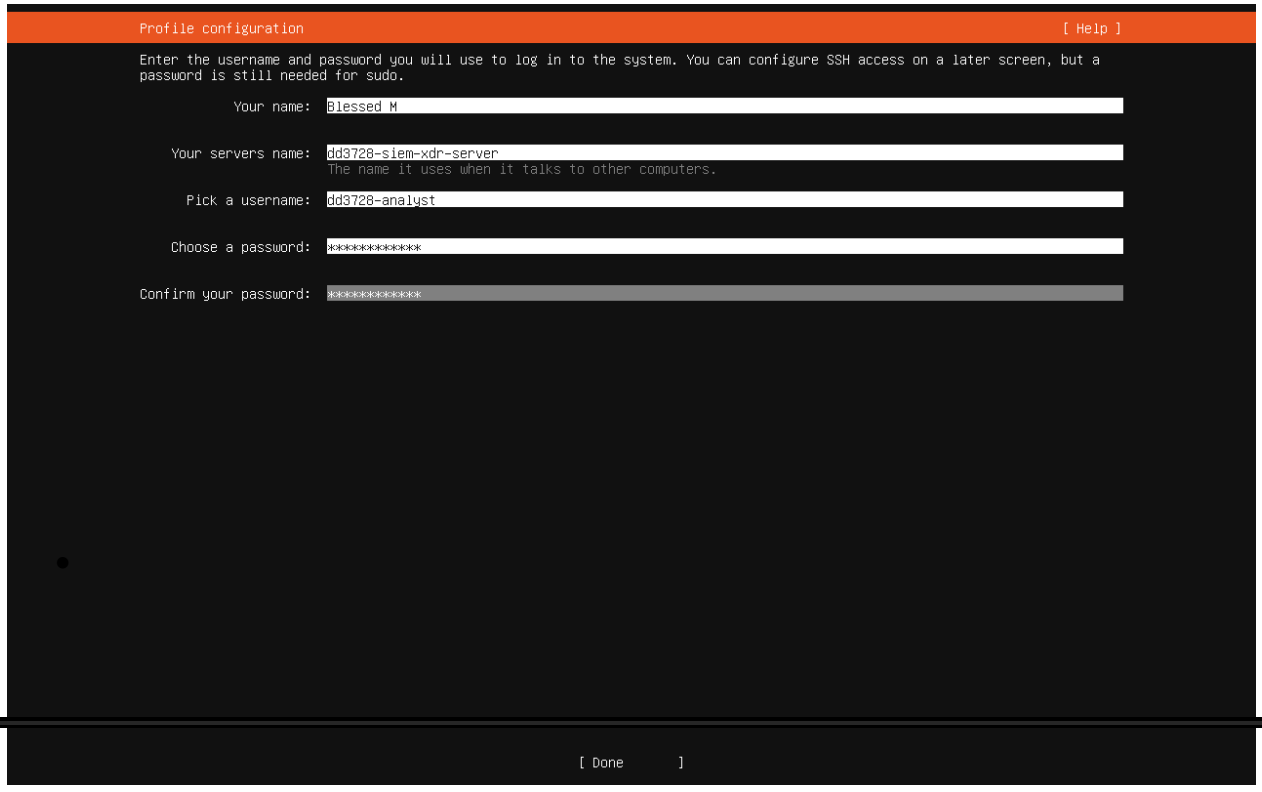
- **Allow password authentication over SSH**: This option permits users to connect to the server using password authentication.

[**screenshots** 👆👇]



- Created admin user →**dd3728-analyst**.

[**screenshots** 👆👇]



B. Static Netplan configuration (authoritative)

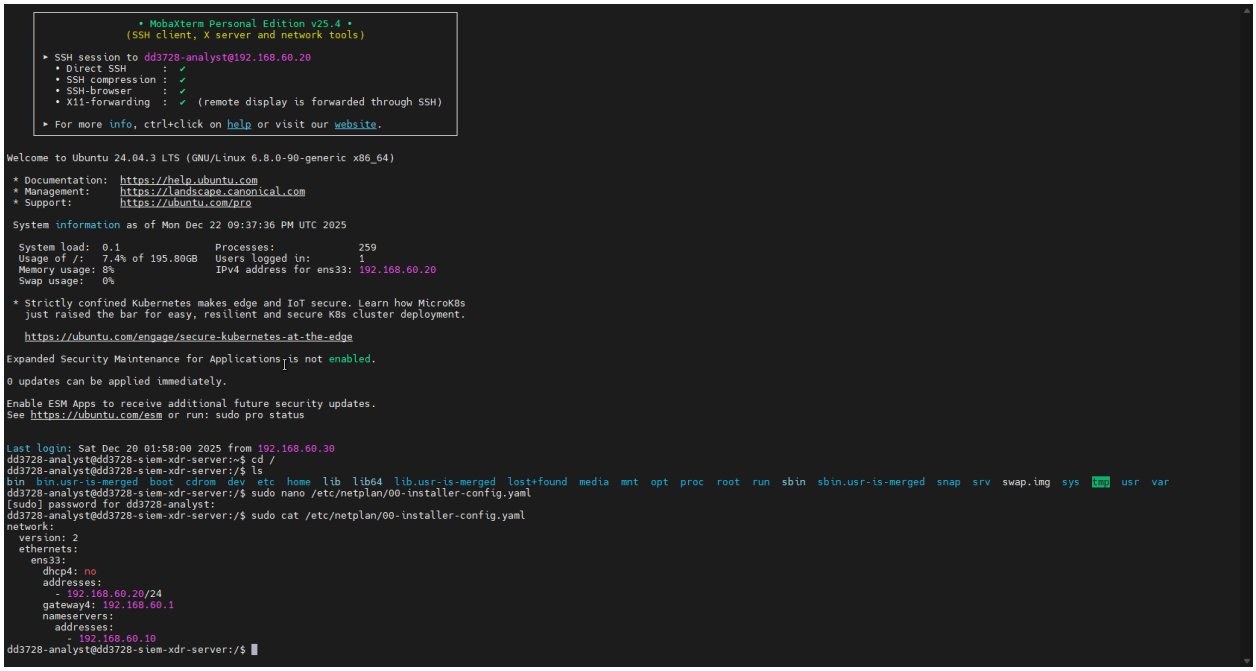
- post- [Splunk]install I did manual network configuration/adjust Netplan

Created /etc/netplan/00-installer-config.yaml:

```
#bash

sudo tee /etc/netplan/00-installer-config.yaml <<'EOF'
network:
  version: 2
  ethernets:
    ens33:
      dhcp4: no
      addresses:
        - 192.168.60.20/24
      routes:
        - to: default
          via: 192.168.60.1
      nameservers:
        addresses:
          - 192.168.60.10
EOF
```

- Defines a static IP, gateway, and AD-DNS server, so the SIEM server behaves like a real enterprise server, not a DHCP client.

[**screenshots** 👆👇]

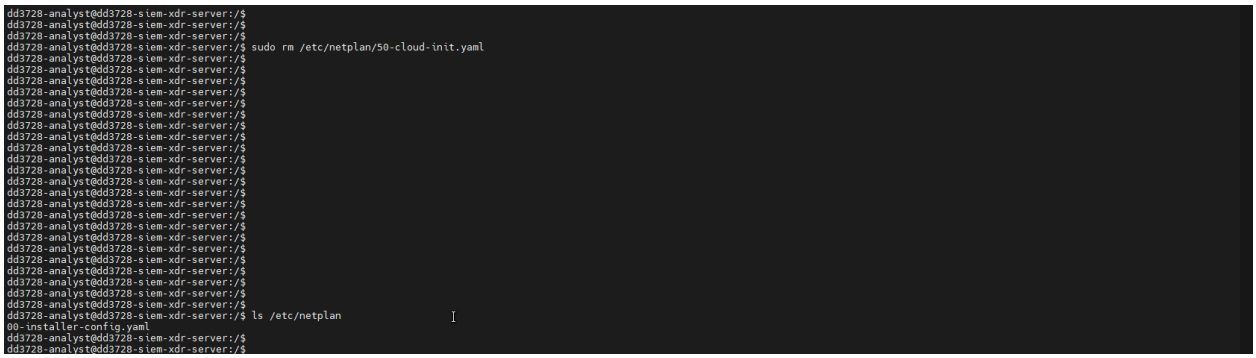


★ **Disable cloud-init networking and remove conflicting Netplan file**:

```
#bash

echo 'network: {config: disabled}' | sudo tee
/etc/cloud/cloud.cfg.d/99-disable-network-config.cfg
sudo rm -f /etc/netplan/50-cloud-init.yaml
```

- Prevents VMware / cloud-init from overwriting enterprise network settings after rebooting.

[**screenshots** 👆👇]



★ **Fix permissions and apply**:

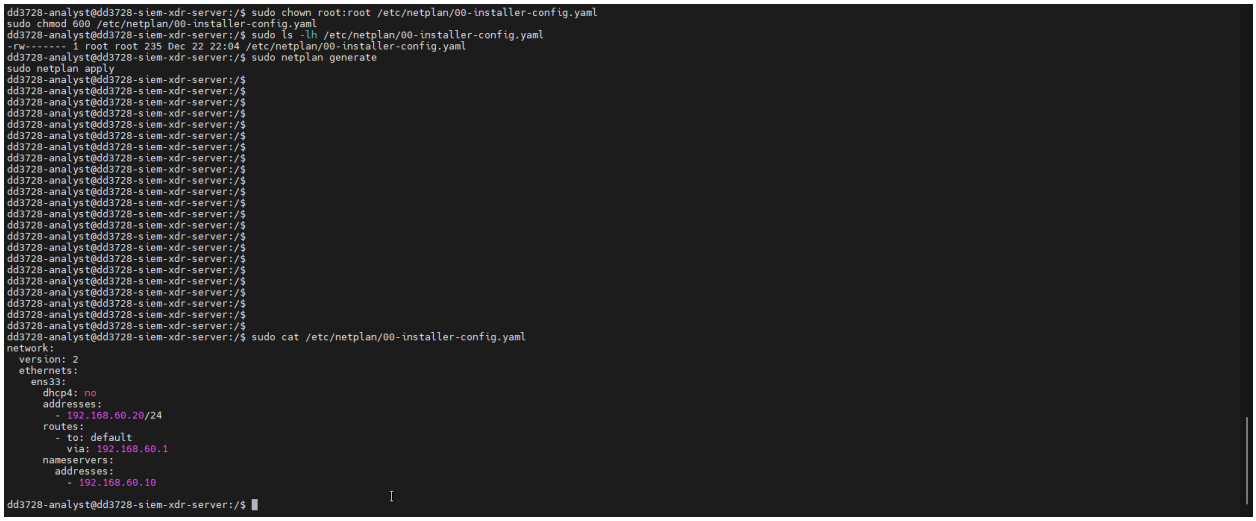
```
#bash

sudo chown root:root /etc/netplan/00-installer-config.yaml
sudo chmod 600 /etc/netplan/00-installer-config.yaml
sudo netplan generate
sudo netplan apply
sudo systemctl restart systemd-resolved
resolvectl status
```

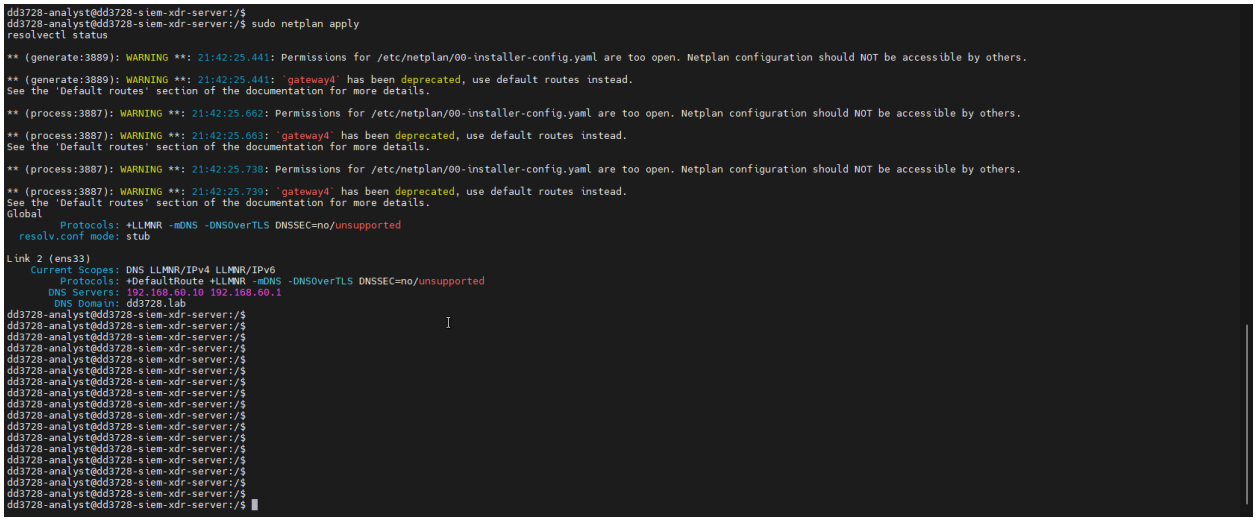
- The commands demonstrate essential operations for managing network configuration securely on a Linux system. They ensure that the Netplan

configuration file is owned and accessible only by root, apply new network settings, and verify DNS resolution status.

- Activates the static IP, gateway, and DNS and restarts the DNS resolver.

[**screenshots** 👆👇]





Validation:

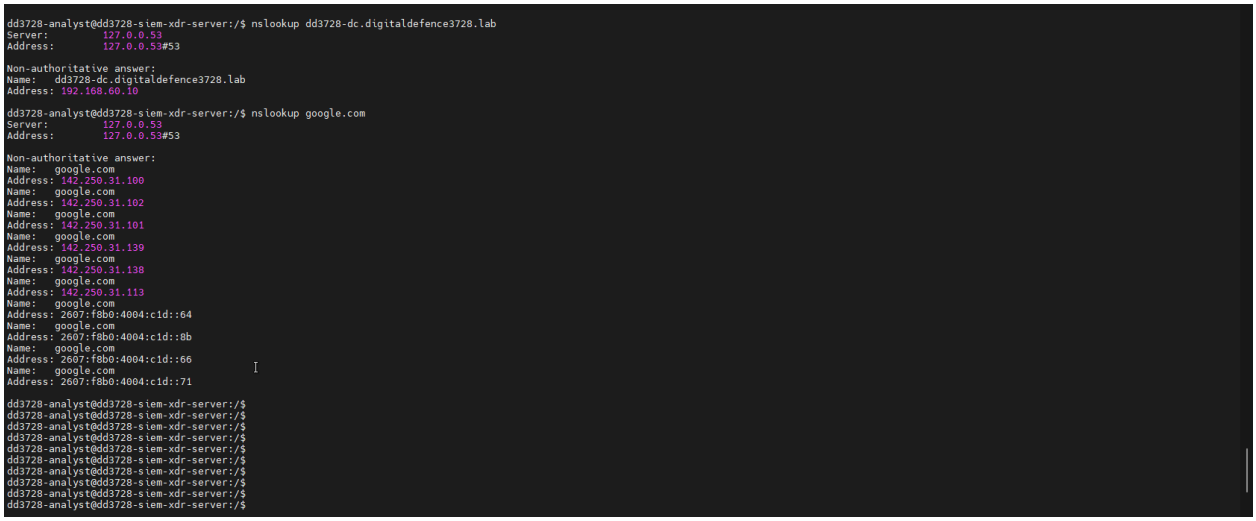
```
#bash

ip a
ip route
nslookup dd3728-dc.digitaldefence3728.lab
nslookup google.com
```

Confirms:

- Internal AD DNS resolution works
- Internet name resolution works

This verifies correct routing + DNS forwarding.

[**screenshots** 👆👇]



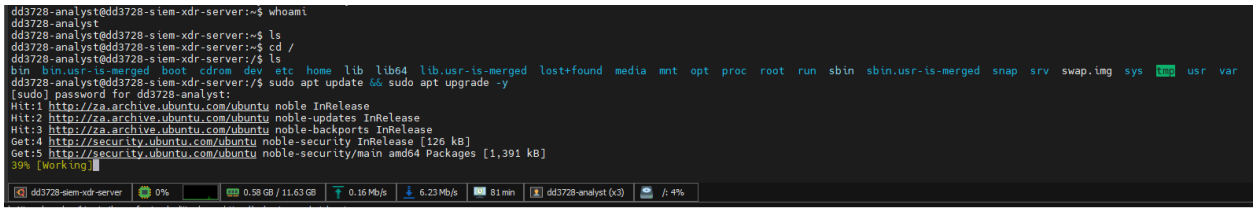
C. Hardening (baseline)

```
#Update and basic tools:

#bash

sudo apt update && sudo apt upgrade -y
```

Why right after install 👆🏽?

- New servers ship with baseline packages that may have known vulnerabilities; skipping this exposes your system to exploit until manually updated.
- Brings the server to a fully patched, vulnerability-reduced state before production use.

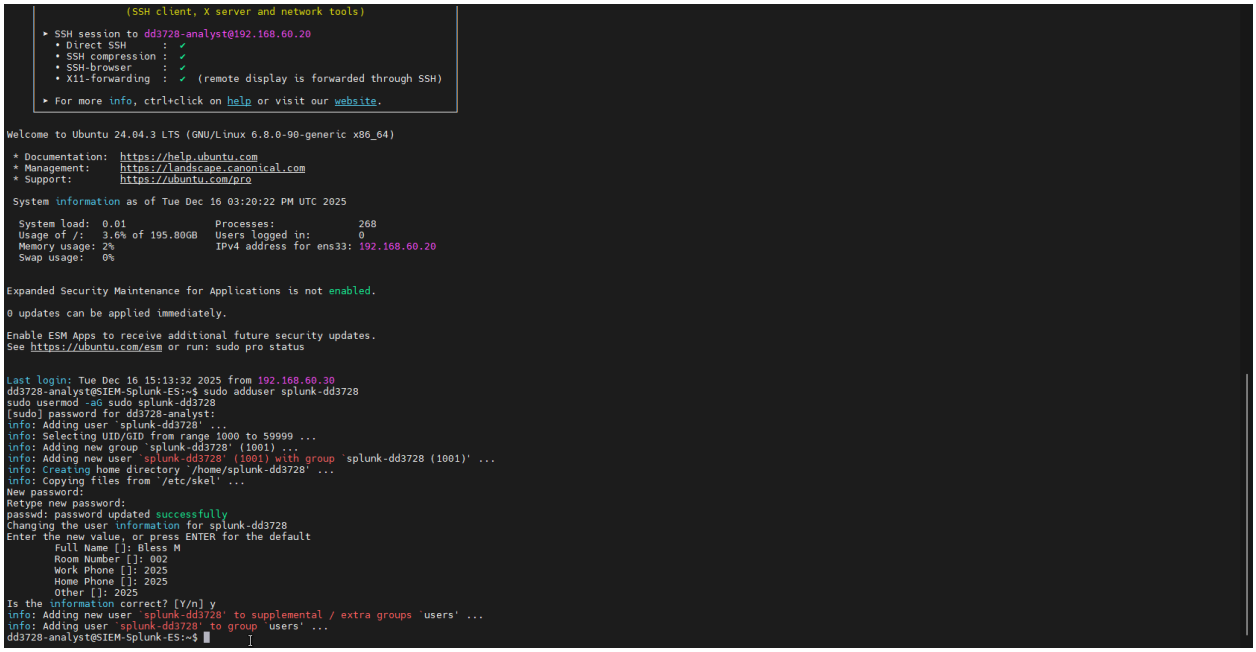
[**screenshots** 👆👇]



Create Splunk service user:

```
#bash

sudo adduser splunk-dd3728
sudo usermod -aG sudo splunk
```

[**screenshots** 👆👇]



What happened ? 👆

I created:

```
sudo adduser splunk-dd3728
```

- ★ When I should have done created **Splunk service user** like the following bash command 👇

```
sudo adduser --disabled-password --gecos "" splunk-dd3728
sudo usermod -aG sudo splunk
```

What --**disabled-password** --**gecos** "" does

Those two flags do **only one thing**:

Flag	**Purpose**
--disabled-password	Prevents **password login** for that account

--gecos ""	Skips full-name and metadata prompts

★ **That means**:

- A password was set
- The account is interactive
- Someone could `SSH` or `sudo` as that account

★ **Why this matters in a SOC environment**

- Splunk service accounts should be:
- Non-interactive
- Not usable for login
- Not usable for `SSH`

★ **Because**:

- If Splunk is exploited, the attacker should not get a shell.
- According to previous executed bash command, service accounts are slightly weaker than enterprise best practice.

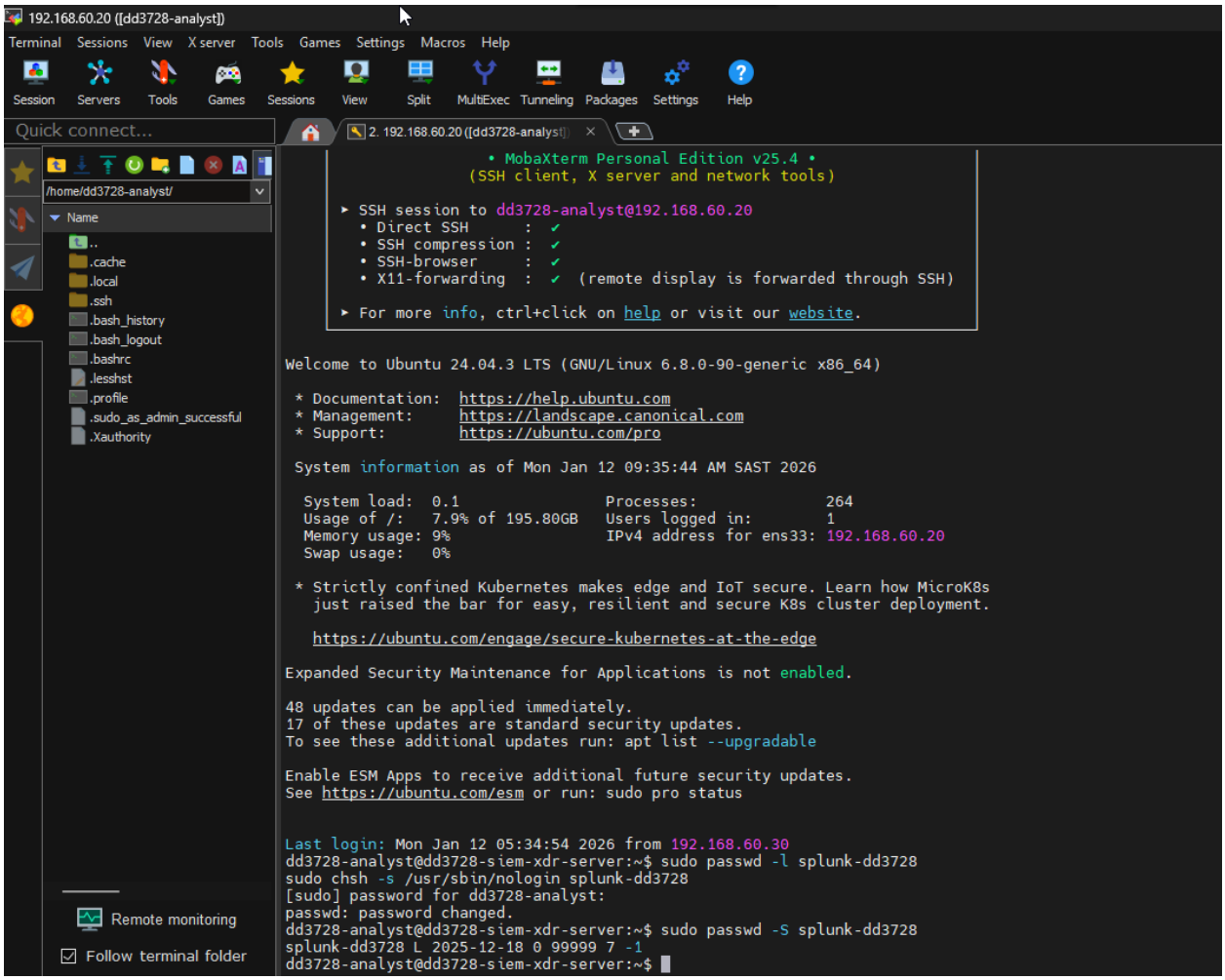
Correct fix [*no reinstallation required (Splunk)]*

```
#bash

sudo passwd -l splunk-dd3728
sudo chsh -s /usr/sbin/nologin splunk-dd3728
```

This:

- Locks the password
- Prevents shell login
- Converts the account into a true service account

[**screenshots** 👆👇]



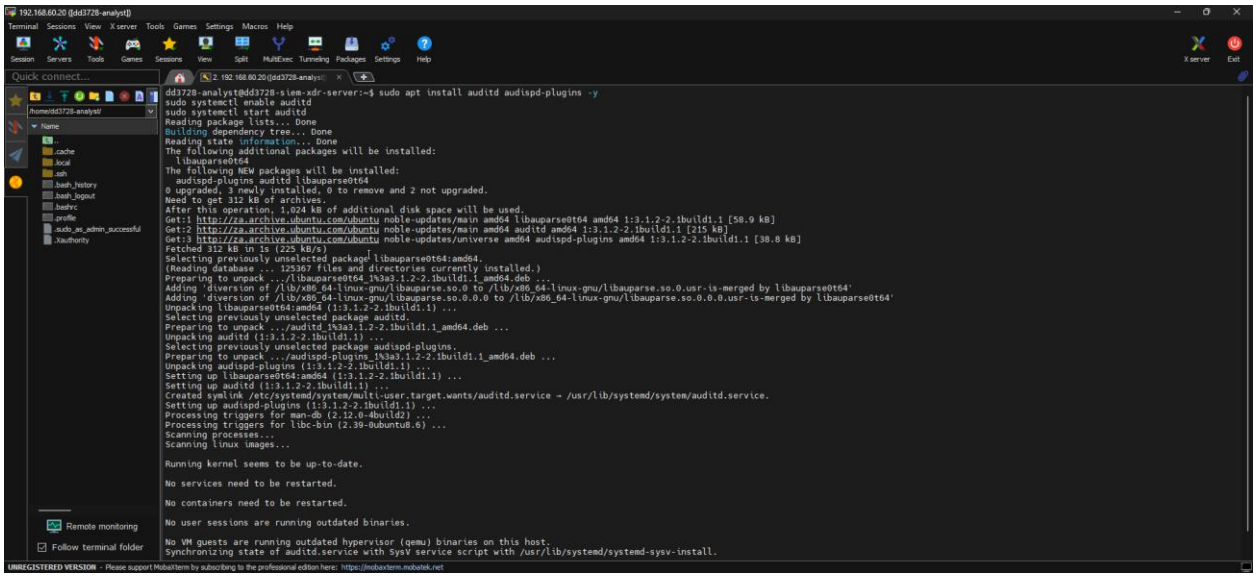
★ **Install auditd**:

```
#bash

sudo apt install -y auditd audispd-plugins
```

Enable Linux auditing:

- Provides kernel-level security logging for file access, user activity, and system changes — critical for SOC investigations.

[**screenshots** 👆👇]



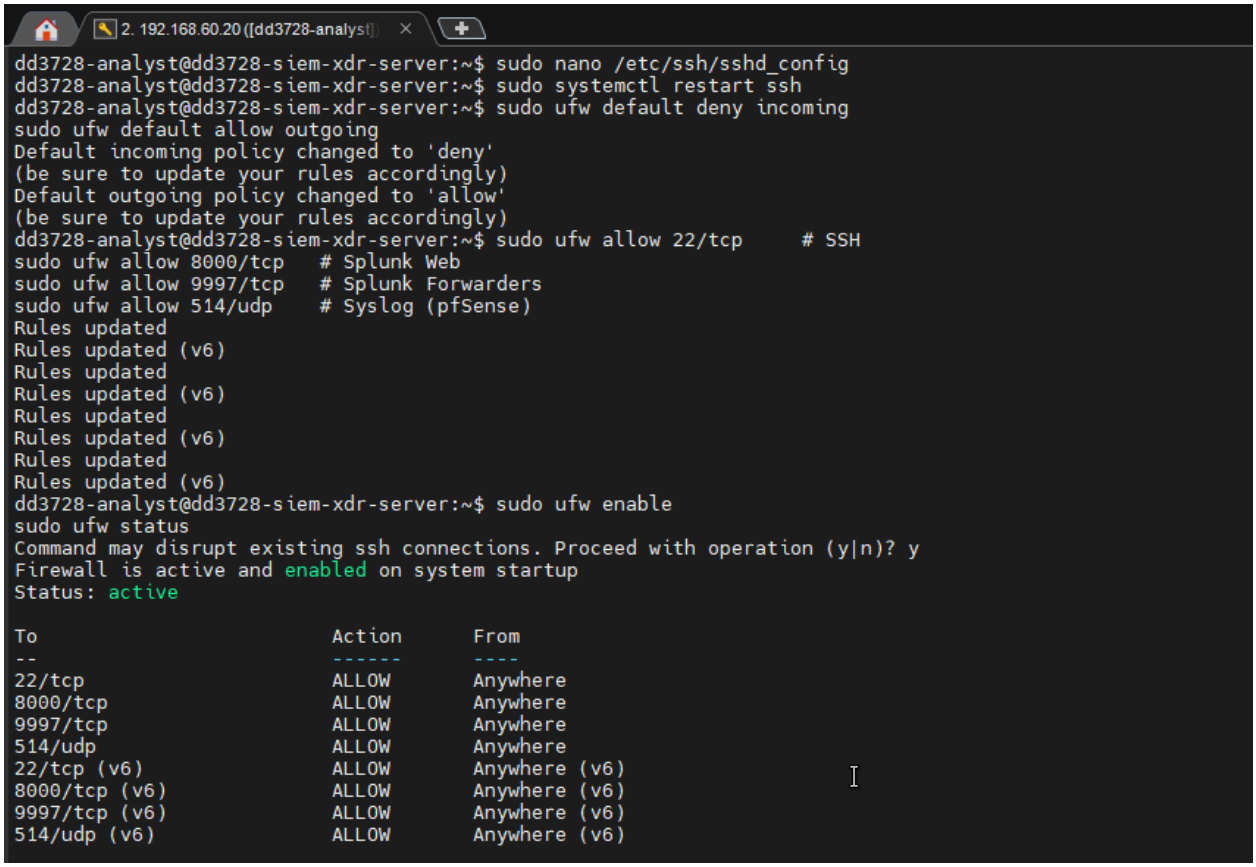
★ **UFW firewall**:

```
#bash

sudo ufw default deny incoming
sudo ufw default allow outgoing
sudo ufw allow 22/tcp
sudo ufw allow 8000/tcp
sudo ufw allow 9997/tcp
sudo ufw allow 514/udp
sudo ufw enable
```

Implements zero-trust host firewalling while allowing only:

- `22` → Admin access (`SSH`)
- `8000` → Splunk Web
- `9997` → Splunk forwarders
- `514` → pfSense firewall logs

[**screenshots** 👆👇]



Time sync:

```
#bash

sudo timedatectl set-timezone Africa/Johannesburg
sudo timedatectl set-ntp true
```

- Ensures accurate timestamps across logs, which is mandatory for threat correlation and investigations.

[**screenshot** 👇🏾]

```
dd3728-analyst@dd3728-siem-xdr-server:~$ timedatectl set-timezone Africa/Johannesburg
timedatectl set-ntp true
timedatectl status
==== AUTHENTICATING FOR org.freedesktop.timedate1.set-timezone ====
Authentication is required to set the system timezone.
Multiple identities can be used for authentication:
 1.  Blessed M (dd3728-analyst)
 2.  Blessed M,002,1055560,1055560,1055596 (splunk-dd3728)
Choose identity to authenticate as (1-2): 1-2
Invalid response `1-2'.
==== AUTHENTICATION CANCELED ====
Failed to set time zone: Access denied
==== AUTHENTICATING FOR org.freedesktop.timedate1.set-ntp ====
Authentication is required to control whether network time synchronization shall be enabled.
Multiple identities can be used for authentication:
 1.  Blessed M (dd3728-analyst)
 2.  Blessed M,002,1055560,1055560,1055596 (splunk-dd3728)
Choose identity to authenticate as (1-2): 2
Password:
==== AUTHENTICATION COMPLETE ====
               Local time: Thu 2025-12-18 04:50:56 UTC
           Universal time: Thu 2025-12-18 04:50:56 UTC
                 RTC time: Thu 2025-12-18 04:50:55
                Time zone: Etc/UTC (UTC, +0000)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
dd3728-analyst@dd3728-siem-xdr-server:~$
lobaXterm by subscribing to the professional edition here:  https://mobaxterm.mobatek.net
```

Configure resource limits for Splunk:

```
#bash

sudo tee -a /etc/security/limits.conf <<'EOF'
splunk soft nofile 65536
splunk hard nofile 65536
EOF
```

Raise Splunk file limits:

- Prevents Splunk from dropping logs during high-volume ingestion.

[**screenshots** 👆👇]

```
 GNU nano 7.2                                  /etc/security/limits.conf *
#
#<type> can have the two values:
#       - "soft" for enforcing the soft limits
#       - "hard" for enforcing hard limits
#
#<item> can be one of the following:
#       - core - limits the core file size (KB)
#       - data - max data size (KB)
#       - fsize - maximum filesize (KB)
#       - memlock - max locked-in-memory address space (KB)
#       - nofile - max number of open file descriptors
#       - rss - max resident set size (KB)
#       - stack - max stack size (KB)
#       - cpu - max CPU time (MIN)
#       - nproc - max number of processes
#       - as - address space limit (KB)
#       - maxlogins - max number of logins for this user
#       - maxsyslogins - max number of logins on the system
#       - priority - the priority to run user process with
#       - locks - max number of file locks the user can hold
#       - sigpending - max number of pending signals
#       - msgqueue - max memory used by POSIX message queues (bytes)
#       - nice - max nice priority allowed to raise to values: [-20, 19]
#       - rtprio - max realtime priority
#       - chroot - change root to directory (Debian-specific)
#
#<domain>      <type>  <item>         <value>
#

#*              soft    core            0
#root           hard    core            100000
#*              hard    rss             10000
#@student       hard    nproc           20
#@faculty       soft    nproc           20
#@faculty       hard    nproc           50
#ftp            hard    nproc           0
#ftp            -       chroot          /ftp
#@student       -       maxlogins       4

# End of file

splunk soft nofile 65536
splunk hard nofile 65536
                                [ Cancelled ]
^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location      M-U Undo         M-A Set Mark     M-] To Bracket   M-Q Previous     ^B Back          ^← Prev Word
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line    M-E Redo         M-6 Copy         ^Q Where Was     M-W Next         ^F Forward       ^→ Next Word
```

Why I disabled Transparent Huge Pages (THP)

- Transparent Huge Pages can cause unpredictable latencies, memory fragmentation and pause/stalls in high-throughput, low-latency services (Splunk indexers, etc.). Splunk's own guidance and real-world practice recommend disabling THP to avoid long GC-like pauses and to stabilize indexing/search performance. Disabling THP is a small OS-level change that materially improves reliability for a production SIEM.

```
#!/bin/bash
### BEGIN INIT INFO
# Provides:          disable-thp
# Required-Start:    $local_fs
# Required-Stop:
# X-Start-Before:    couchbase-server
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Disable THP
# Description:       Disables transparent huge pages (THP) on
boot, to improve
#                    Couchbase performance.
### END INIT INFO

case $1 in
  start)
    if [ -d /sys/kernel/mm/transparent_hugepage ]; then
      thp_path=/sys/kernel/mm/transparent_hugepage
    elif [ -d /sys/kernel/mm/redhat_transparent_hugepage ]; then
      thp_path=/sys/kernel/mm/redhat_transparent_hugepage
    else
      return 0
    fi
```

```
    echo 'never' > ${thp_path}/enabled
    echo 'never' > ${thp_path}/defrag

    re='^[0-1]+$'
    if [[ $(cat ${thp_path}/khugepaged/defrag) =~ $re ]]
    then
      # RHEL 7
      echo 0  > ${thp_path}/khugepaged/defrag
    else
      # RHEL 6
      echo 'no' > ${thp_path}/khugepaged/defrag
    fi

    unset re
    unset thp_path
    ;;
esac
```

What the script does (brief, line-by-line summary)

- The file shown is an init script placed in `/etc/init.d/disable_thp` so it runs at boot.
- It checks for the kernel THP sysfs path (`/sys/kernel/mm/transparent_hugepage`).
- It writes never to `${thp_path}/enabled` to turn THP off at runtime.
- It writes never or no to the `${thp_path}/khugepaged/defrag` file (depending on kernel variants) to stop the khugepaged defragmenter.
- The script contains ### BEGIN INIT INFO metadata so `init/system` scripts start it in runlevels 2–5 on boot.
- The result: THP and automatic hugepage defragmentation are disabled every boot, ensuring Splunk runs on a predictable memory model.

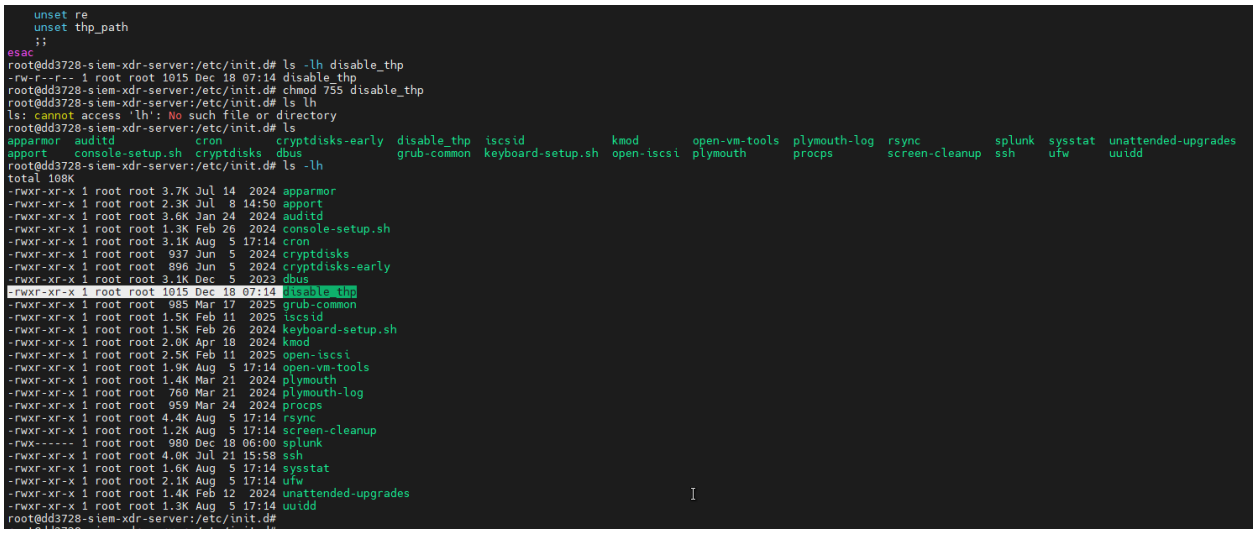
[**screenshot 👆🏾👇🏾 non-consistent configurations**]

```
-rwxr-xr-x 1 root root 2.1K Aug  5 17:14 ufw
-rwxr-xr-x 1 root root 1.4K Feb 12  2024 unattended-upgrades
-rwxr-xr-x 1 root root 1.3K Aug  5 17:14 uuidd
root@dd3728-siem-xdr-server:/etc/init.d# cat /sys/kernel/
cat: /sys/kernel/: Is a directory
root@dd3728-siem-xdr-server:/etc/init.d# cd /sys/kernel/
root@dd3728-siem-xdr-server:/sys/kernel# ls
address_bits  cgroup         crash_elfcorehdr_size  iommu_groups        kexec_crash_size  mm          profiling      reboot    software_nodes  uevent_seqnum
boot_params   config         debug                  irq                 kexec_loaded      notes       rcu_expedited  security  tracing         vmcoreinfo
btf           cpu_byteorder  fscaps                 kexec_crash_loaded  livepatch         oops_count  rcu_normal     slab      uevent_helper   warn_count
root@dd3728-siem-xdr-server:/sys/kernel# cat mm
cat: mm: Is a directory
root@dd3728-siem-xdr-server:/sys/kernel# cd mm
root@dd3728-siem-xdr-server:/sys/kernel/mm# ls
hugepages  ksm  lru_gen  numa  page_idle  swap  transparent_hugepage
root@dd3728-siem-xdr-server:/sys/kernel/mm# cd transparent_hugepage
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage# ls
defrag   hpage_pmd_size    hugepages-128kB  hugepages-2048kB  hugepages-32kB   hugepages-64kB  shmem_enabled
enabled  hugepages-1024kB  hugepages-16kB   hugepages-256kB   hugepages-512kB  khugepaged      use_zero_page
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage# cat enabled
always [madvise] never
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage# cat defrag
always defer defer+madvise [madvise] never
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
```

```
root@dd3728-siem-xdr-server:/sys/kernel# cd mm
root@dd3728-siem-xdr-server:/sys/kernel/mm# ls
hugepages  ksm  lru_gen  numa  page_idle  swap  transparent_hugepage
root@dd3728-siem-xdr-server:/sys/kernel/mm# cd transparent_hugepage
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage# ls
defrag   hpage_pmd_size    hugepages-128kB  hugepages-2048kB  hugepages-32kB   hugepages-64kB  shmem_enabled
enabled  hugepages-1024kB  hugepages-16kB   hugepages-256kB   hugepages-512kB  khugepaged      use_zero_page
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage# cat enabled
always [madvise] never
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage# cat defrag
always defer defer+madvise [madvise] never
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage# echo never > /sys/kernel/mm/transparent_hugepage/defrag
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage# cat  /sys/kernel/mm/transparent_hugepage/defrag
always defer defer+madvise madvise [never]
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
```

[screenshot 👆🏽 👇🏽 consistent configurations]

```
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage#
root@dd3728-siem-xdr-server:/sys/kernel/mm/transparent_hugepage# cd /etc/init.d
root@dd3728-siem-xdr-server:/etc/init.d# nano disable_thp
root@dd3728-siem-xdr-server:/etc/init.d# ls
apparmor  auditd            cron        cryptdisks-early  disable_thp  iscsid              kmod         open-vm-tools  plymouth-log  rsync           splunk  sysstat  unattended-upgrades
apport    console-setup.sh  cryptdisks  dbus              grub-common  keyboard-setup.sh  open-iscsi  plymouth       procps        screen-cleanup  ssh     ufw      uuidd
root@dd3728-siem-xdr-server:/etc/init.d# cat disable_thp
#!/bin/bash
### BEGIN INIT INFO
# Provides:          disable-thp
# Required-Start:    $local_fs
# Required-Stop:
# X-Start-Before:    couchbase-server
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Disable THP
# Description:       Disables transparent huge pages (THP) on boot, to improve
#                    Couchbase performance.
### END INIT INFO

case $1 in
  start)
    if [ -d /sys/kernel/mm/transparent_hugepage ]; then
      thp_path=/sys/kernel/mm/transparent_hugepage
    elif [ -d /sys/kernel/mm/redhat_transparent_hugepage ]; then
      thp_path=/sys/kernel/mm/redhat_transparent_hugepage
    else
      return 0
    fi

    echo 'never' > ${thp_path}/enabled
    echo 'never' > ${thp_path}/defrag

    re='^[0-1]+$'
    if [[ $(cat ${thp_path}/khugepaged/defrag) =~ $re ]]
    then
      # RHEL 7
      echo 0  > ${thp_path}/khugepaged/defrag
    else
      # RHEL 6
      echo 'no' > ${thp_path}/khugepaged/defrag
    fi

    unset re
    unset thp_path
    ;;
esac
root@dd3728-siem-xdr-server:/etc/init.d#
```



- ★ **Enable unattended security updates**:

#bash

sudo apt install -y unattended-upgrades
sudo dpkg-reconfigure --priority=low unattended-upgrades

Enable automatic security patching

- Keeps the SIEM continuously protected without manual intervention.

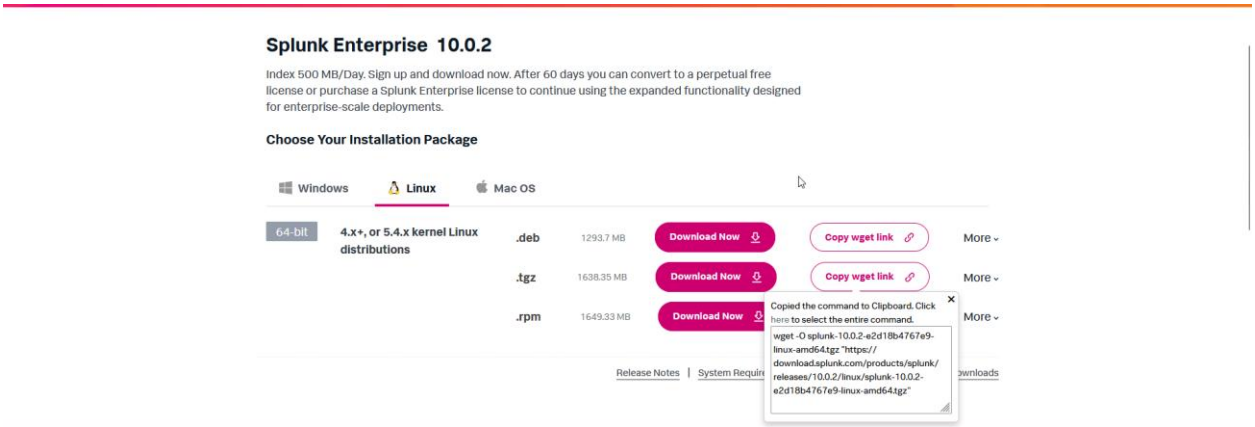
What I implemented 🔄:

- OS patching
- Least-privilege service accounts
- Host-based firewalling
- Kernel auditing
- Time integrity
- SIEM performance tuning
- Disabled Transparent Huge Pages (THP)
- Automated security maintenance

D. Splunk Enterprise installation

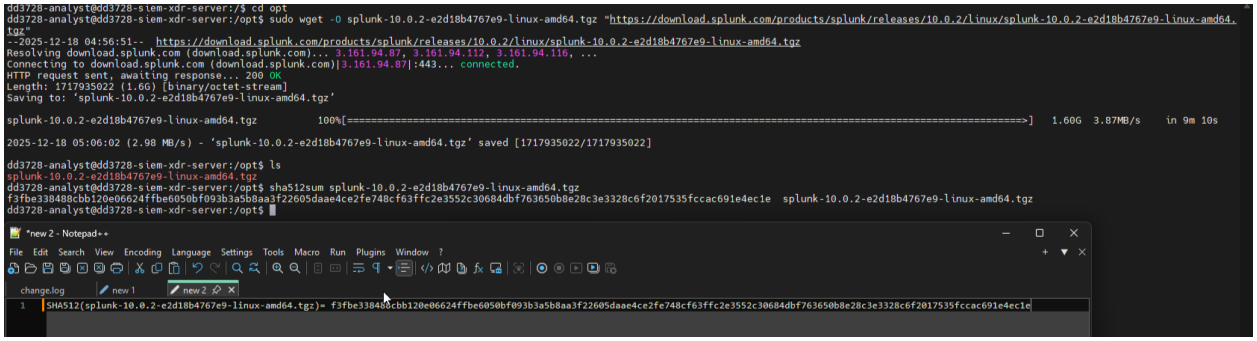
Copy and Paste Splunk `https://www.splunk.com/en_us/download.html` to `/opt` (via `MobiXterm SSH` from Jumpbox - browser), then:

[**screenshots** 👆👇]



Download:

Verify Hash SHA512:




```
#bash

cd /opt
sudo tar -xvzf splunk-splunk-10.0.2-e2d18b4767e9-linux-amd64.tgz-
linux.tgz
sudo chown -R splunk-dd3728:splunk-dd3728 /opt/splunk
```

- Prevents Splunk from running as root — standard SOC security practice.

[**screenshots** 👆👇]

```
Last login: Thu Dec 18 04:44:04 2025 from 192.168.60.30
dd3728-analyst@dd3728-siem-xdr-server:~$ su splunk-dd3728
Password:
splunk-dd3728@dd3728-siem-xdr-server:/home/dd3728-analyst$ cd /
splunk-dd3728@dd3728-siem-xdr-server:/$ cd opt
splunk-dd3728@dd3728-siem-xdr-server:/opt$ ls
splunk  splunk-10.0.2-e2d18b4767e9-linux-amd64.tgz
splunk-dd3728@dd3728-siem-xdr-server:/opt$ ll
total 1677688
drwxr-xr-x  3 root  root        4096 Dec 18 05:10 ./
drwxr-xr-x 23 root  root        4096 Dec 11 10:52 ../
drwxr-xr-x 11 10777 10777       4096 Oct 30 00:16 splunk/
-rw-r--r--  1 root  root  1717935022 Nov 14 03:39 splunk-10.0.2-e2d18b4767e9-linux-amd64.tgz
splunk-dd3728@dd3728-siem-xdr-server:/opt$ sudo chown -R splunk-dd3728:splunk-dd3728 /opt/splunk
[sudo] password for splunk-dd3728:
splunk-dd3728@dd3728-siem-xdr-server:/opt$ ll
total 1677688
drwxr-xr-x  3 root          root               4096 Dec 18 05:10 ./
drwxr-xr-x 23 root          root               4096 Dec 11 10:52 ../
drwxr-xr-x 11 splunk-dd3728 splunk-dd3728      4096 Oct 30 00:16 splunk/
-rw-r--r--  1 root          root          1717935022 Nov 14 03:39 splunk-10.0.2-e2d18b4767e9-linux-amd64.tgz
splunk-dd3728@dd3728-siem-xdr-server:/opt$
splunk-dd3728@dd3728-siem-xdr-server:/opt$
splunk-dd3728@dd3728-siem-xdr-server:/opt$
splunk-dd3728@dd3728-siem-xdr-server:/opt$
```

Start Splunk as **splunk-dd3728** user and enable boot-start:

```
#bash

sudo -u splunk /opt/splunk/bin/splunk start --accept-license
```

- Launches Splunk as a dedicated service account.

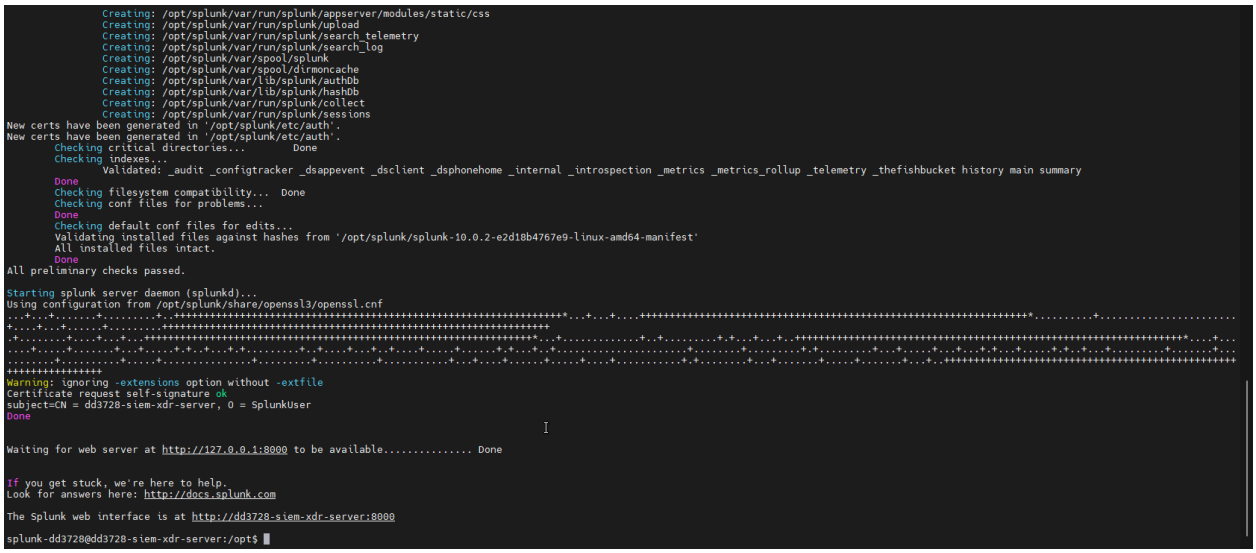
```
sudo /opt/splunk/bin/splunk enable boot-start -user splunk
```

- Ensures the SIEM survives reboots like a real SOC server.

[**screenshots** 👆👇]

```
splunk-dd3728@dd3728-siem-xdr-server:/opt$
splunk-dd3728@dd3728-siem-xdr-server:/opt$
splunk-dd3728@dd3728-siem-xdr-server:/opt$
splunk-dd3728@dd3728-siem-xdr-server:/opt$ sudo -u splunk-dd3728 /opt/splunk/bin/splunk start --accept-license

This appears to be your first time running this version of Splunk.

Splunk software must create an administrator account during startup. Otherwise, you cannot log in.
Create credentials for the administrator account.
Characters do not appear on the screen when you type in credentials.
```

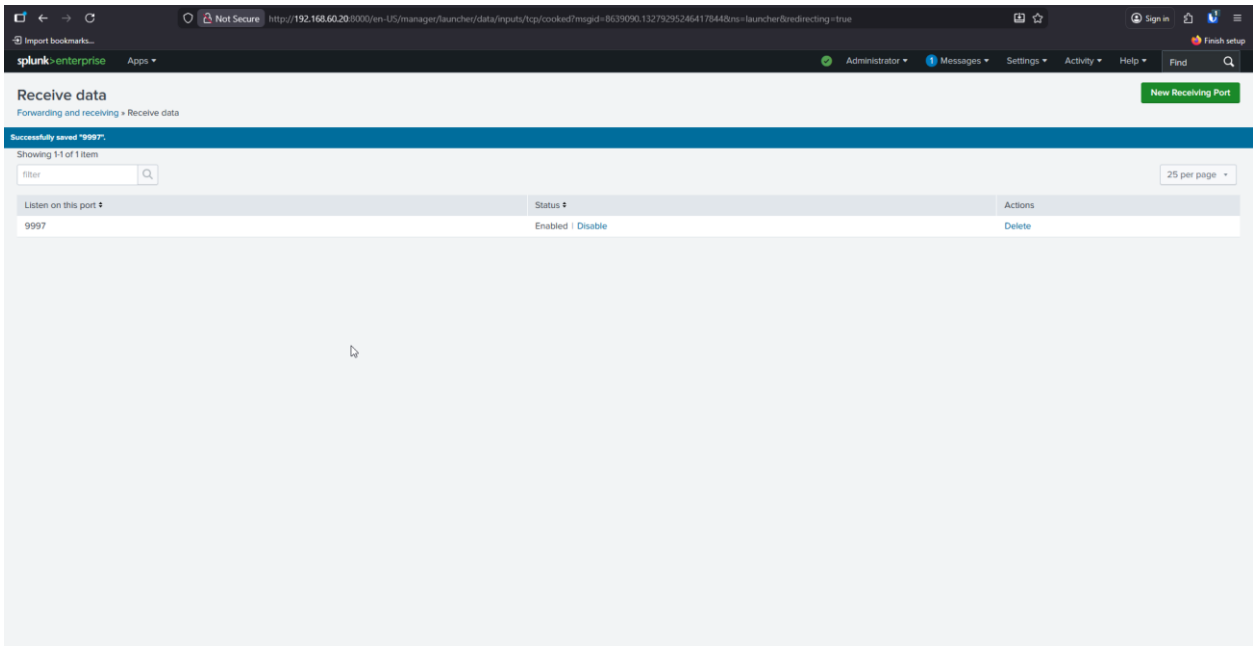
```
splunk-dd3728@dd3728-siem-xdr-server:/opt$ sudo /opt/splunk/bin/splunk enable boot-start -user splunk-dd3728
[sudo] password for splunk-dd3728:
Sorry, try again.
[sudo] password for splunk-dd3728:
Init script installed at /etc/init.d/splunk.
Init script is configured to run at boot.
splunk-dd3728@dd3728-siem-xdr-server:/opt$
splunk-dd3728@dd3728-siem-xdr-server:/opt$
splunk-dd3728@dd3728-siem-xdr-server:/opt$
splunk-dd3728@dd3728-siem-xdr-server:/opt$
```



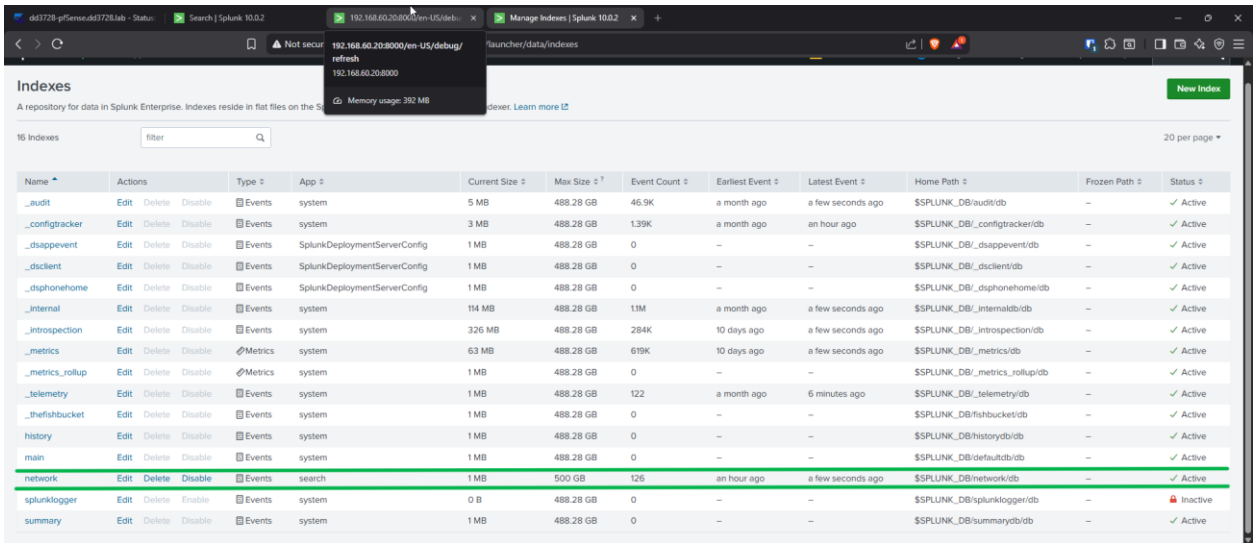
Configure receiving:

- Splunk Web: http://192.168.60.20:8000
- Settings → Forwarding and receiving → Add TCP input port 9997.

[**screenshots** 👆👇]



★ **Created index named**: network.



What you configured (quick list)

- Splunk to listen for pfSense syslog at `UDP 514` (privileged port).
- Changed Splunk input to `UDP 1514` (non-privileged).
- Restarted Splunk.
- Opened the host firewall for `UDP 1514`.
- Verified Splunk is listening on `1514`.

"I initially configured Splunk to listen on UDP 514 (standard syslog), but Splunk runs as a non-root service and cannot bind privileged ports. After observing zero tcpdump packets and bind errors in splunkd.log, I moved the input to UDP 1514, opened the firewall, and validated ingestion — maintaining least-privilege and enterprise security best practice.**"**

❖ Why each command / config was used

Configured UDP 514 input for pfSense:

```
#bash

sudo tee /opt/splunk/etc/system/local/inputs.conf <<'EOF'
[udp://514]
index = network
sourcetype = pfsense
EOF
```

What it does: Tells Splunk to create a `UDP` input on port `514` and write events to index=network with sourcetype=pfsense.

Why used: pfSense sends syslog to UDP/514 by default; configuring Splunk for 514 is the natural first step.

[**screenshots** 👆👇]

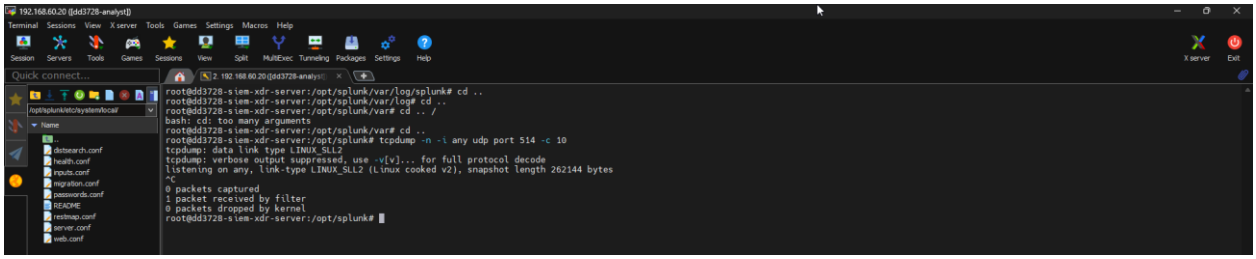
tcpdump captured 0 packets on UDP 514 (diagnostic reasoning)

```
#bash

sudo tcpdump -n -i any udp port 514 -c 5
```

and saw 0 captured. Possible, immediate causes:

- Splunk never bound `UDP 514` (most likely) — Splunk (non-root) cannot bind privileged ports → no process listening ⇒ kernel drops packets or they never targeted this host.
- Evidence to check: `ss -ulnp | grep 514` (empty → Splunk not listening).
- Check splunkd.log for bind errors.

[**screenshots** 👆👇]



Why 1514 is the right decision

"`Ports <1024` require root; running Splunk as root is unacceptable for security. Using `1514` keeps Splunk non-privileged and secure while allowing firewall/proxy rules to route or redirect standard syslog traffic as needed.**"**

1) **changed to 1514 👇🏾 (inputs.conf)**

```
sudo tee /opt/splunk/etc/system/local/inputs.conf <<'EOF'
[udp://1514] index = network sourcetype = pfsense EOF
```

What it does: Reconfigures Splunk to listen on `UDP 1514` instead of `514`.

Why used: Ports below `1024` are privileged — only root can bind them. Splunk runs as a non-root service user (**splunk-dd3728**) for security, so it cannot bind to `UDP 514`. Using `1514 ( >1024 )` avoids granting Splunk elevated privileges and follows least-privilege practice.

2) **restart Splunk (apply changes)**

```
sudo -u splunk /opt/splunk/bin/splunk restart
```

What it does: Restarts Splunk as the splunk service user so it reads the new `inputs.conf`.

Why used: Changes to inputs only take effect after Splunk restarts.

3) `allow UDP 1514` through host firewall

```
sudo ufw allow 1514/udp
```

What it does: Opens port `1514/UDP` on the Ubuntu host so incoming syslog packets are not blocked.

Why used: Even if Splunk listens, the OS firewall can still drop packets.

4) **verify listener**

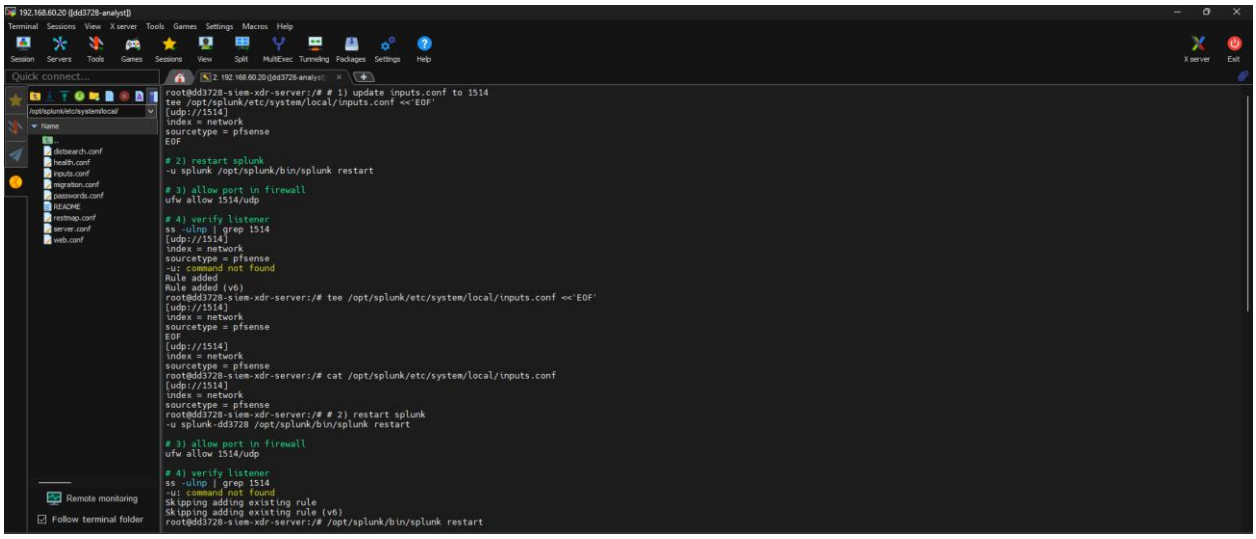
```
sudo ss -ulnp | grep 1514
```

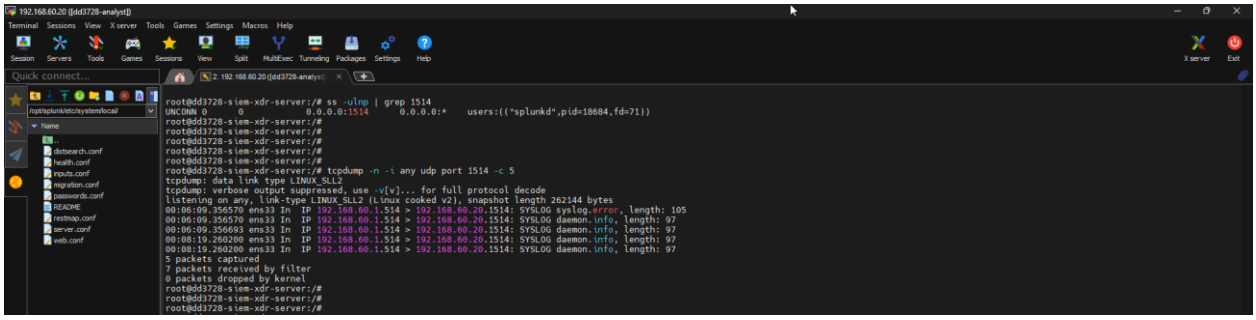
What it does: Shows UDP sockets and the processes listening on them.

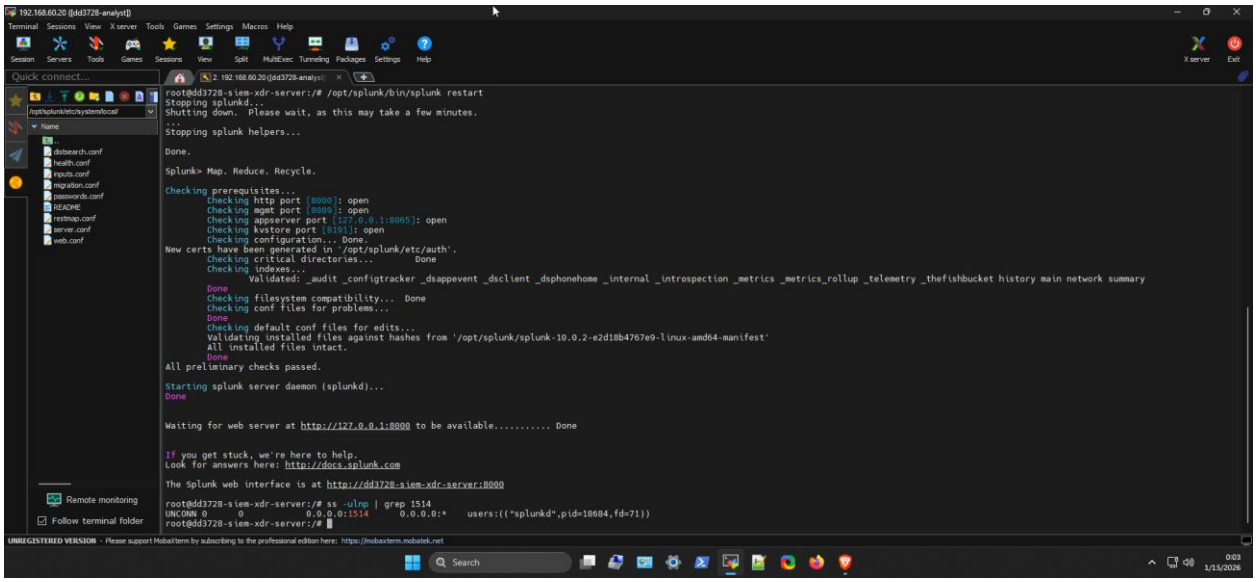
Why used: Confirms Splunk is actually bound to `1514` and ready to receive.

```
sudo -u splunk /opt/splunk/bin/splunk restart
```

[**screenshots** 👆 👇]







E. Validation tests

Jumpbox – Splunk Web connectivity test ✅ :

```
#bash
```

```
curl -I http://192.168.60.20:8000
```

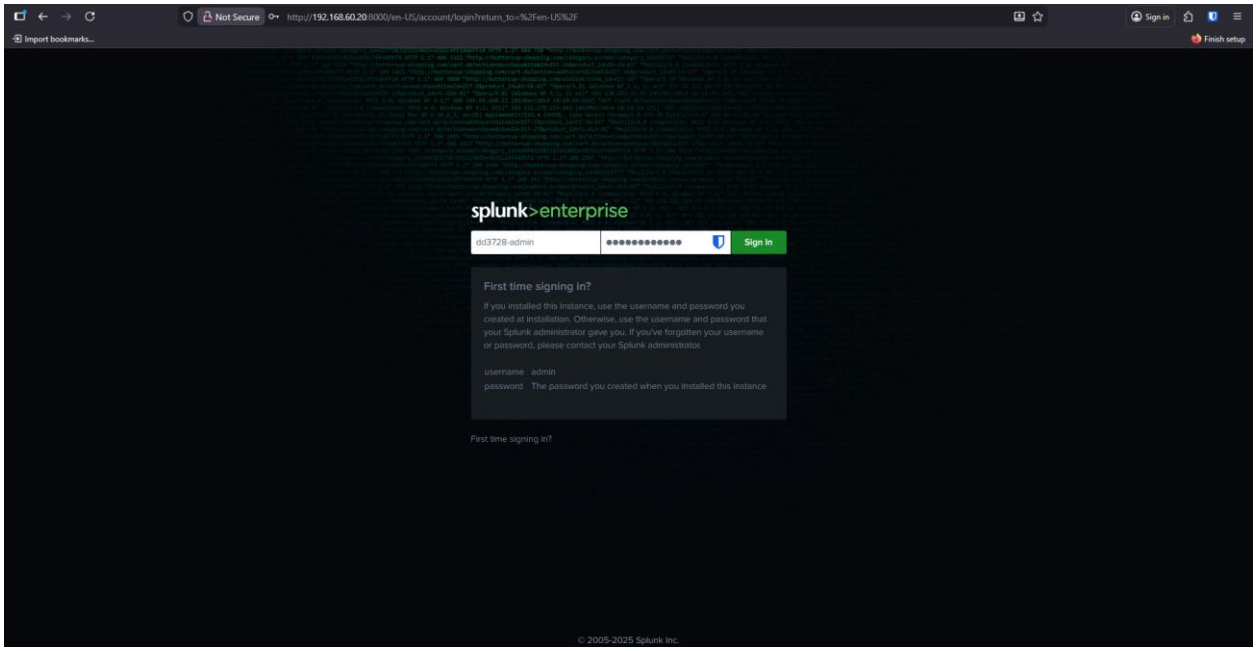
Purpose:

- Verifies that Splunk Web is reachable over the network, and that `TCP/8000` is open.

What it proves:

- Network routing is correct
- Firewall allows access
- Splunk Web service is listening

[**screenshots** 👆👇]



Splunk Server – Process validation ✅:

```
#bash
ps aux | grep splunk
```

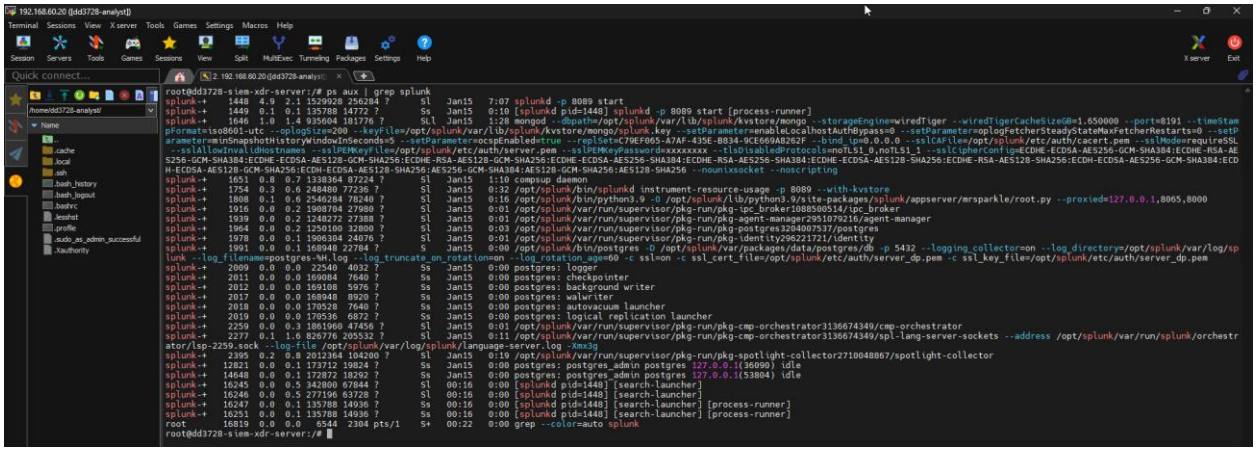
Purpose*:*

- Confirms that Splunk services are running on the server.

What it proves*:*

- Splunkd is active
- No startup or service failure
- Splunk is capable of receiving and indexing data

[**screenshots** 👆👇]



Splunk Web – Data ingestion validation ✅:

- Search:

```
#spl

index=network sourcetype=pfsense | head 20
```

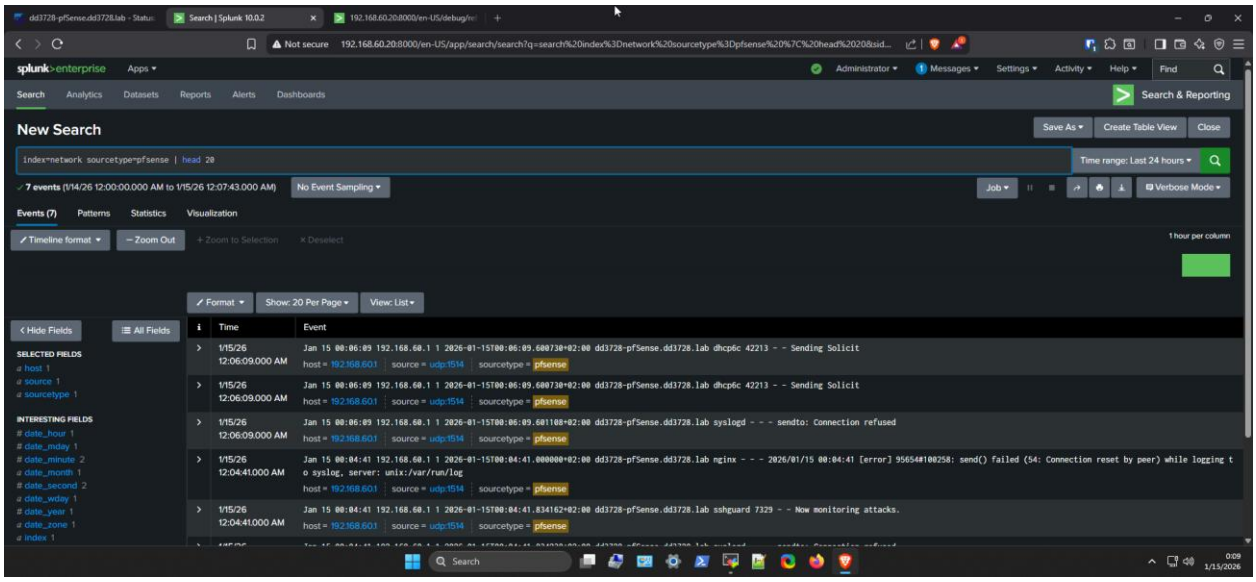
Purpose:

- Checks whether pfSense logs are being successfully indexed.

What it proves:

- UDP input is working
- Correct index assignment
- Logs are searchable in Splunk

[**screenshots** 👆👇]



9 — Windows Client: domain join, Splunk UF & Sysmon

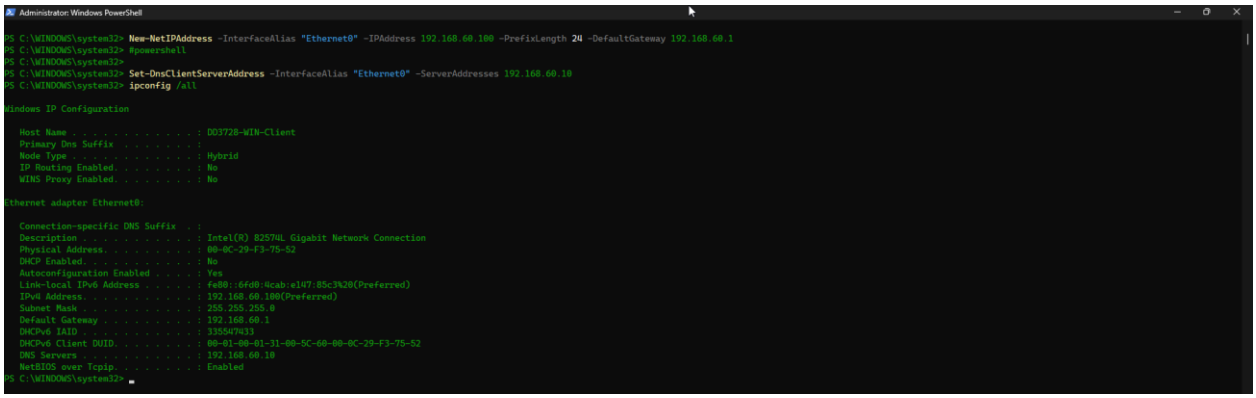
Goal: Generate realistic endpoint telemetry for Splunk.

A. Networking & DNS

#powershell

```
Set-DnsClientServerAddress -InterfaceAlias "Ethernet0" -
ServerAddresses 192.168.60.10
New-NetIPAddress -InterfaceAlias "Ethernet0" -IPAddress
192.168.60.100 -PrefixLength 24 -DefaultGateway 192.168.60.1
```

[**screenshots** 👆👇]



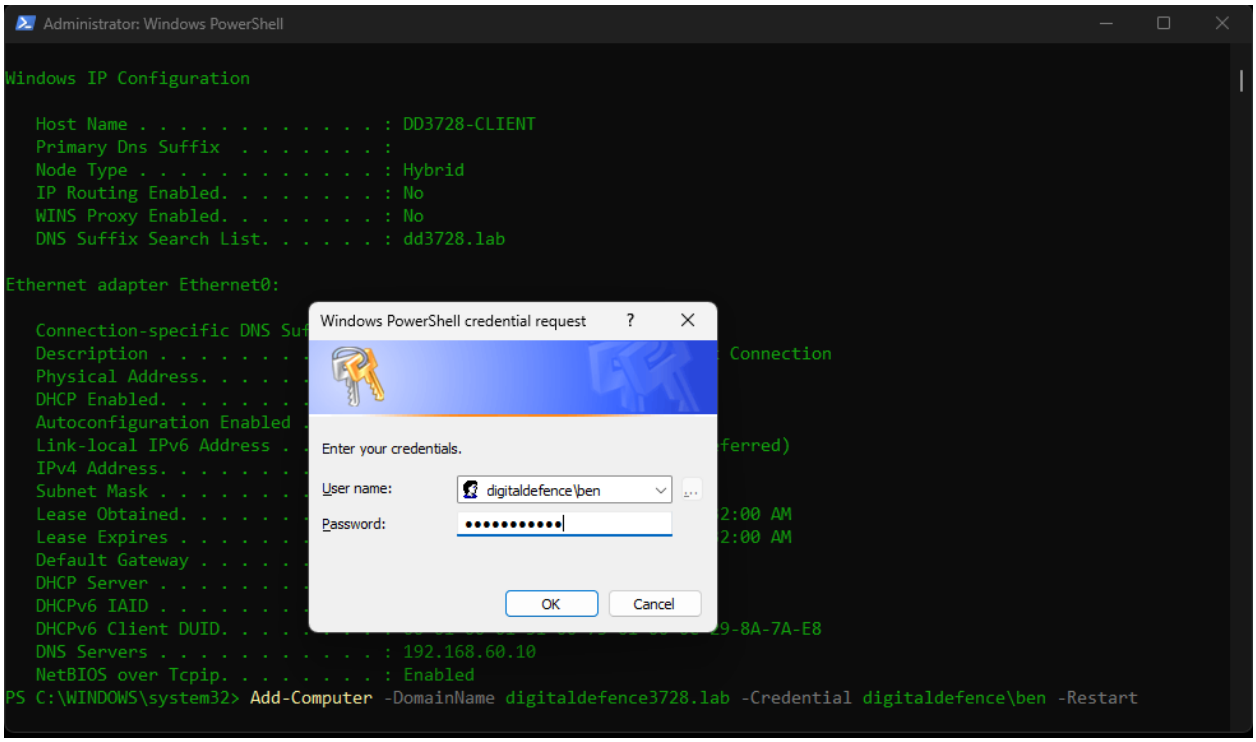
B. Domain join

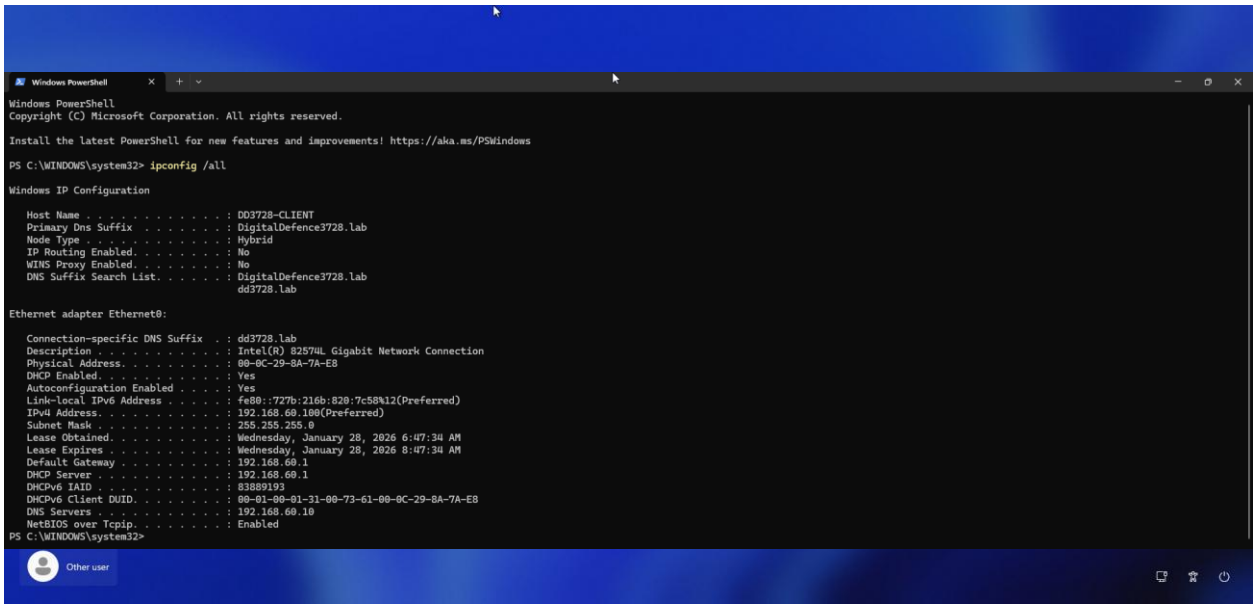
#powershell

```
Add-Computer -DomainName digitaldefence3728.lab -Credential
digitaldefence\administrator -Restart
```

Domain successfully joined ✅

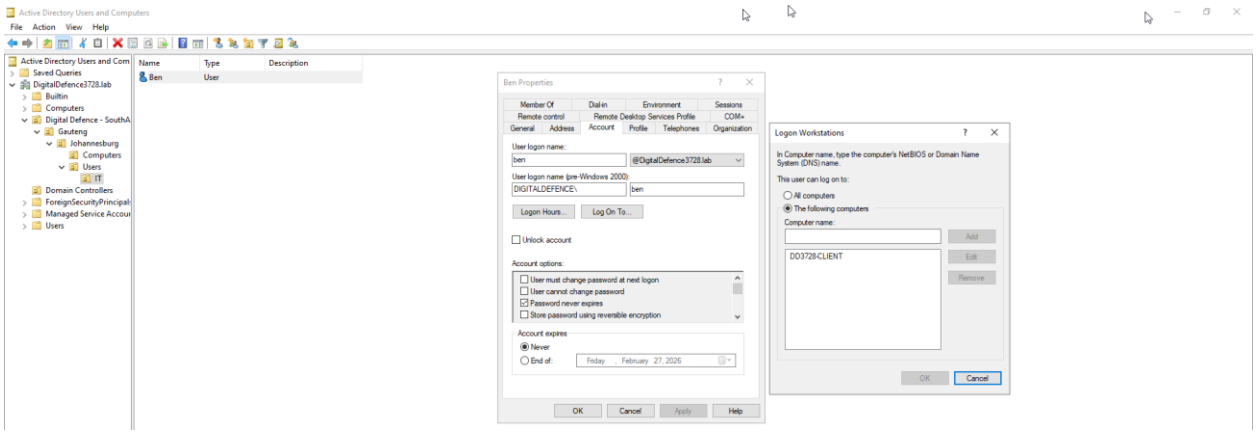
[**screenshots** 👆👇]





ℹ️ **Prior win-client DC join, I had already set up a client user named Ben and added DD3728-CLIENT VM machine under user Ben to Log-on using that machine.**

[screenshot 👆🏽👇🏽✅]



★ **Below is a brief, precise, explanation of what I configured and why. Before Windows UF installation.**

`/opt/splunk/etc/apps/admin-demo/local/indexes.conf`

`[network]`

- `homePath = $SPLUNK_DB/network/db`
- `coldPath = $SPLUNK_DB/network/colddb`
- `thawedPath = $SPLUNK_DB/network/thaweddb`

- `#sizing & lifecycle`
- `maxTotalDataSizeMB = 512000`
- `homePath.maxDataSizeMB = 300000`
- `maxDataSize = 20`
- `maxHotBuckets = 5`
- `frozenTimePeriodInSecs = 100697600`

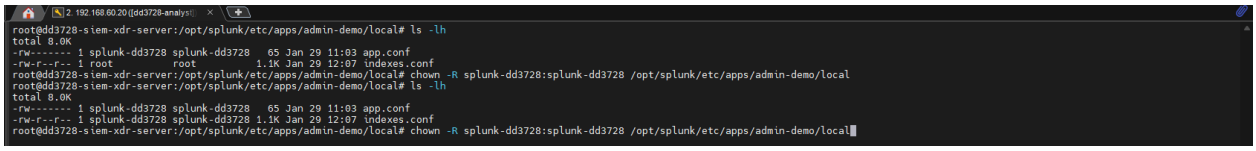
- `#behaviour`
- `enableDataIntegrityControl = true`
- `enableTsidxReduction = false`
- `compressRawdata = true`
- `coldToFrozenDir = $SPLUNK_DB/network/archive-network`
  `archiver.enableDataArchiver = false`
- `bucketMerging = false`
- `hotBucketStreaming.deleteHotsAfterRestart = false`
  `minHotIdleSecsBeforeForceRoll = 0`

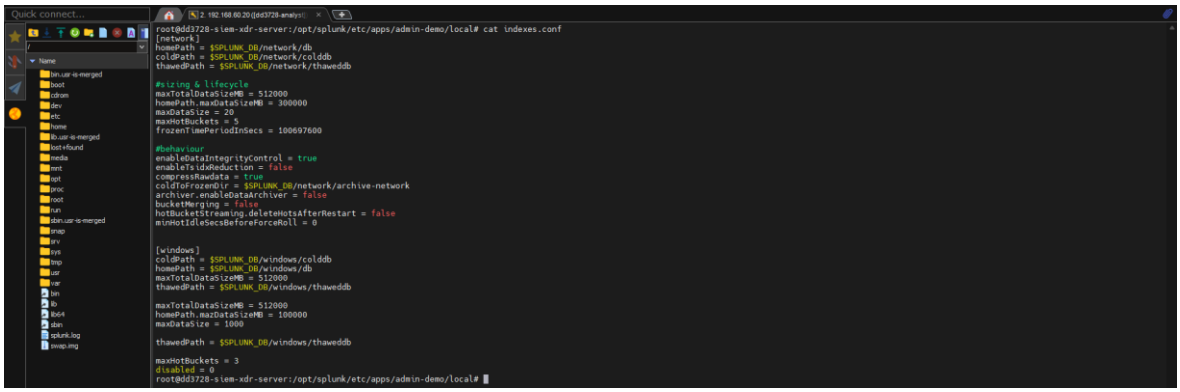
[windows]

- coldPath = $SPLUNK_DB/windows/colddb
- homePath = $SPLUNK_DB/windows/db
- maxTotalDataSizeMB = 512000
- thawedPath = $SPLUNK_DB/windows/thaweddb
- maxTotalDataSizeMB = 512000
- homePath.mazDataSizeMB = 100000
- maxDataSize = 1000

- thawedPath = $SPLUNK_DB/windows/thaweddb
- maxHotBuckets = 3
- frozenTimePeriodInSecs = 100697600
- archiver.enableDataArchiver = 0
- bucketMerging = 0

[**screenshots** 👆👇]

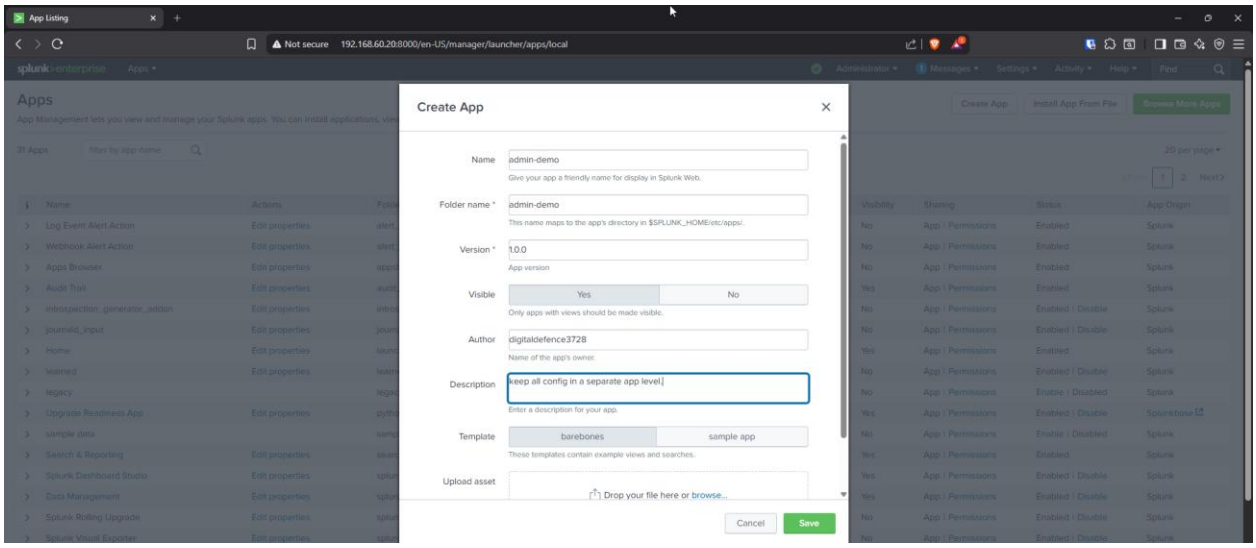


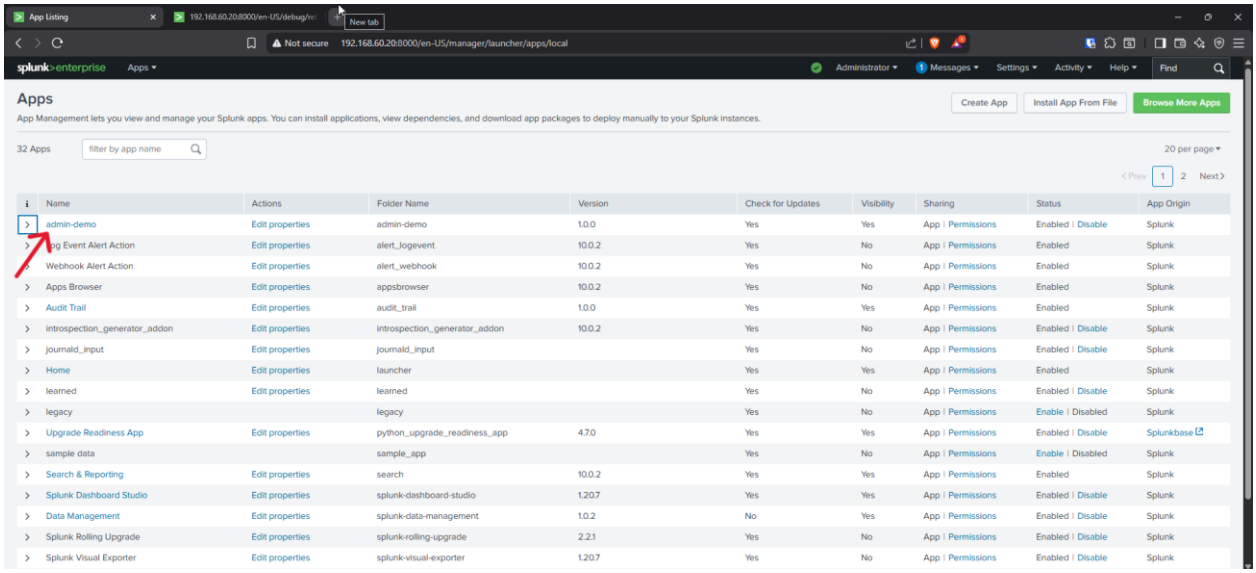


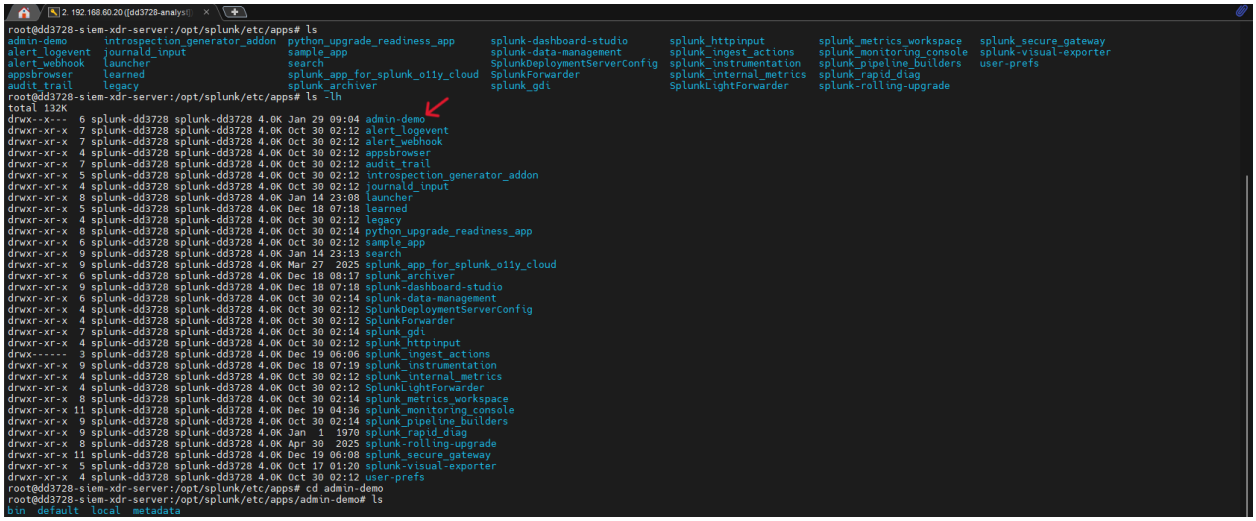
Why I created a dedicated admin app (admin-demo)

- I removed custom index definitions from the Search app and moved them into a separate admin-level app.
- This follows Splunk enterprise best practice
- Prevents upgrades from overwriting configs
- Separates platform administration from user/search content

[**screenshot** 👆👇]


App Listing
Not secure 192.168.60.20:8000/en-US/manager/launcher/apps/local
Apps
Create App
Name
admin-demo
Give your app a friendly name for display in Splunk Web.
Folder name *
admin-demo
This name maps to the app's directory in $SPLUNK_HOME/etc/apps/.
Version *
1.0.0
App version
Visible
Yes
No
Only apps with views should be made visible.
Author
digitaldefence3728
Name of the app's owner.
Description
keep all config in a separate app level.|
Enter a description for your app.
Template
barebones
sample app
These templates contain example views and searches.
Upload asset
Drop your file here or browse...
Cancel
Save


App Listing
Not secure 192.168.60.20:8000/en-US/manager/launcher/apps/local
splunk>enterprise
Apps
Administrator
Messages
Settings
Activity
Help
Find
Apps
App Management lets you view and manage your Splunk apps. You can install applications, view dependencies, and download app packages to deploy manually to your Splunk instances.
Create App
Install App From File
Browse More Apps
32 Apps
filter by app name
20 per page
Name
Actions
Folder Name
Version
Check for Updates
Visibility
Sharing
Status
App Origin
admin-demo
Edit properties
1.0.0
Yes
App | Permissions
Enabled | Disable
Splunk
Webhook Alert Action
alert_webhook
10.0.2
Apps Browser
appsbrowser
Audit Trail
audit_trail
1.0.0
introspection_generator_addon
journald_input
Home
launcher
learned
legacy
Enable | Disabled
Upgrade Readiness App
python_upgrade_readiness_app
4.7.0
Splunkbase
sample data
sample_app
Search & Reporting
search
Splunk Dashboard Studio
splunk-dashboard-studio
1.20.7
Data Management
splunk-data-management
1.0.2
No
Splunk Rolling Upgrade
splunk-rolling-upgrade
2.2.1
Splunk Visual Exporter
splunk-visual-exporter

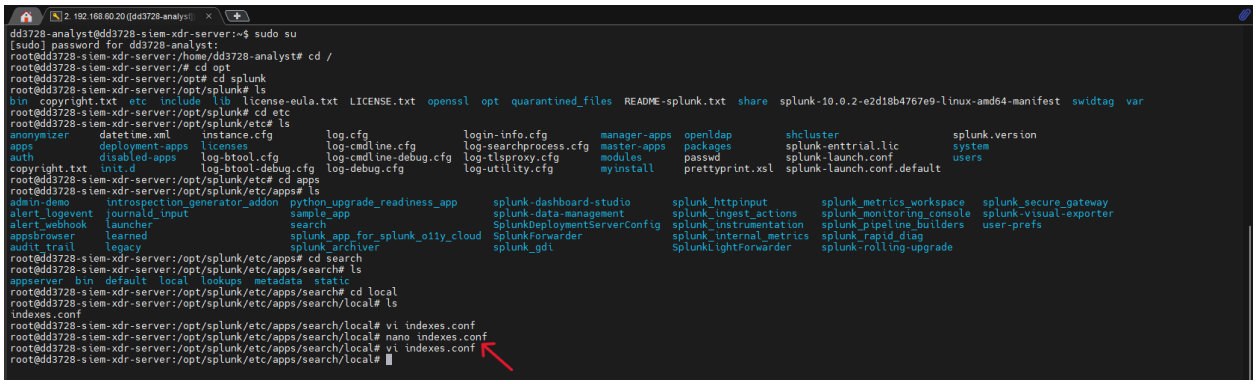

root@dd3728-siem-xdr-server:/opt/splunk/etc/apps# ls
admin-demo introspection_generator_addon python_upgrade_readiness_app splunk-dashboard-studio splunk_httpinput splunk_metrics_workspace splunk_secure_gateway
alert_logevent journald_input sample_app splunk-data-management splunk_ingest_actions splunk_monitoring_console splunk-visual-exporter
alert_webhook launcher search SplunkDeploymentServerConfig splunk_instrumentation splunk_pipeline_builders user-prefs
appsbrowser learned splunk_app_for_splunk_o11y_cloud SplunkForwarder splunk_internal_metrics splunk_rapid_diag
audit_trail legacy splunk_archiver splunk_gdi SplunkLightForwarder splunk-rolling-upgrade
root@dd3728-siem-xdr-server:/opt/splunk/etc/apps# ls -lh
total 132K
drwx--x--- 6 splunk-dd3728 splunk-dd3728 4.0K Jan 29 09:04 admin-demo
drwxr-xr-x 7 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:12 alert_logevent
drwxr-xr-x 7 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:12 alert_webhook
drwxr-xr-x 4 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:12 appsbrowser
drwxr-xr-x 7 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:12 audit_trail
drwxr-xr-x 5 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:12 introspection_generator_addon
drwxr-xr-x 4 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:12 journald_input
drwxr-xr-x 8 splunk-dd3728 splunk-dd3728 4.0K Jan 14 23:08 launcher
drwxr-xr-x 5 splunk-dd3728 splunk-dd3728 4.0K Dec 18 07:18 learned
drwxr-xr-x 4 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:12 legacy
drwxr-xr-x 8 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:14 python_upgrade_readiness_app
drwxr-xr-x 6 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:12 sample_app
drwxr-xr-x 9 splunk-dd3728 splunk-dd3728 4.0K Jan 14 23:13 search
drwxr-xr-x 9 splunk-dd3728 splunk-dd3728 4.0K Mar 27 2025 splunk_app_for_splunk_o11y_cloud
drwxr-xr-x 6 splunk-dd3728 splunk-dd3728 4.0K Dec 18 08:17 splunk_archiver
drwxr-xr-x 9 splunk-dd3728 splunk-dd3728 4.0K Dec 18 07:18 splunk-dashboard-studio
drwxr-xr-x 6 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:14 splunk-data-management
drwxr-xr-x 4 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:12 SplunkDeploymentServerConfig
drwxr-xr-x 4 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:12 SplunkForwarder
drwxr-xr-x 7 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:14 splunk_gdi
drwxr-xr-x 4 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:12 splunk_httpinput
drwx------ 3 splunk-dd3728 splunk-dd3728 4.0K Dec 19 06:06 splunk_ingest_actions
drwxr-xr-x 9 splunk-dd3728 splunk-dd3728 4.0K Dec 18 07:19 splunk_instrumentation
drwxr-xr-x 4 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:12 splunk_internal_metrics
drwxr-xr-x 4 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:12 SplunkLightForwarder
drwxr-xr-x 8 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:14 splunk_metrics_workspace
drwxr-xr-x 11 splunk-dd3728 splunk-dd3728 4.0K Dec 19 04:36 splunk_monitoring_console
drwxr-xr-x 9 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:14 splunk_pipeline_builders
drwxr-xr-x 9 splunk-dd3728 splunk-dd3728 4.0K Jan 1 1970 splunk_rapid_diag
drwxr-xr-x 8 splunk-dd3728 splunk-dd3728 4.0K Apr 30 2025 splunk-rolling-upgrade
drwxr-xr-x 11 splunk-dd3728 splunk-dd3728 4.0K Dec 19 06:08 splunk_secure_gateway
drwxr-xr-x 5 splunk-dd3728 splunk-dd3728 4.0K Oct 17 01:20 splunk-visual-exporter
drwxr-xr-x 4 splunk-dd3728 splunk-dd3728 4.0K Oct 30 02:12 user-prefs
root@dd3728-siem-xdr-server:/opt/splunk/etc/apps# cd admin-demo
root@dd3728-siem-xdr-server:/opt/splunk/etc/apps/admin-demo# ls
bin default local metadata

★ **Improves maintainability, auditing, and portability**

Indexes are now managed at:

`/opt/splunk/etc/apps/admin-demo/local/indexes.conf`

[**screenshot** 👆🏾 👇🏾]



Index design intent

- `network index` → pfSense firewall and network telemetry
- `windows index` → Windows logs forwarded by the Universal Forwarder
- This separation supports:
- Clear data ownership
- Targeted retention and sizing
- Faster troubleshooting and searches

Network index – pfSense data

- Storage paths
- `homePath` → `Hot/Warm` buckets for active firewall logs
- `coldPath` → Aged network data
- `thawedPath` → Restored data for investigations

Paths for Data Storage

- `homePath = $SPLUNK_DB/network/db`: This defines the primary storage location for the indexer's hot data (the most current data).
- `coldPath = $SPLUNK_DB/network/colddb`: This is where older hot data is moved when it's no longer actively being written to but still needs to be stored.
- `thawedPath = $SPLUNK_DB/network/thaweddb`: This path stores data that has been archived and is ready for retrieval.

Sizing & Data Lifecycle Management

- `maxTotalDataSizeMB = 512000`: The maximum limit for total data storage across all indexed data.
- `homePath.maxDataSizeMB = 300000`: Specific limit for the hot data path.
- `maxDataSize = 20`: This may refer to the maximum size of individual data buckets.
- `maxHotBuckets = 5`: Maximum number of active hot data buckets.
- `frozenTimePeriodInSecs = 100697600`: Duration (in seconds) after which data will be moved to a frozen state (not usable for queries).

Data Handling Behavior

- `enableDataIntegrityControl = true`: Ensures that data integrity checks are performed, preventing data corruption.
- `enableTsidxReduction = false`: Disables the reduction of the index size for time series data.
- `compressRawdata = true`: Enables compression for raw data to save space.
- `codToFrozenDir = $SPLUNK_DB/network/archive-network`: Specifies the directory for archiving frozen data.
- `archiver.enableDataArchiver = false`: The data archiving feature is turned off.
- `bucketMerging = false`: Disables the merging of smaller data buckets into larger ones to enhance performance.
- `hotBucketStreaming.deleteHotsAfterRestart = false`: Retains hot buckets even after a system restart.
- `minHotIdleSecsBeforeForceRoll = 0`: Indicates that there's no minimum idle time before rolling the hot buckets into cold storage.

This index is optimized for:

- Event logs
- Sysmon telemetry
- Endpoint investigations

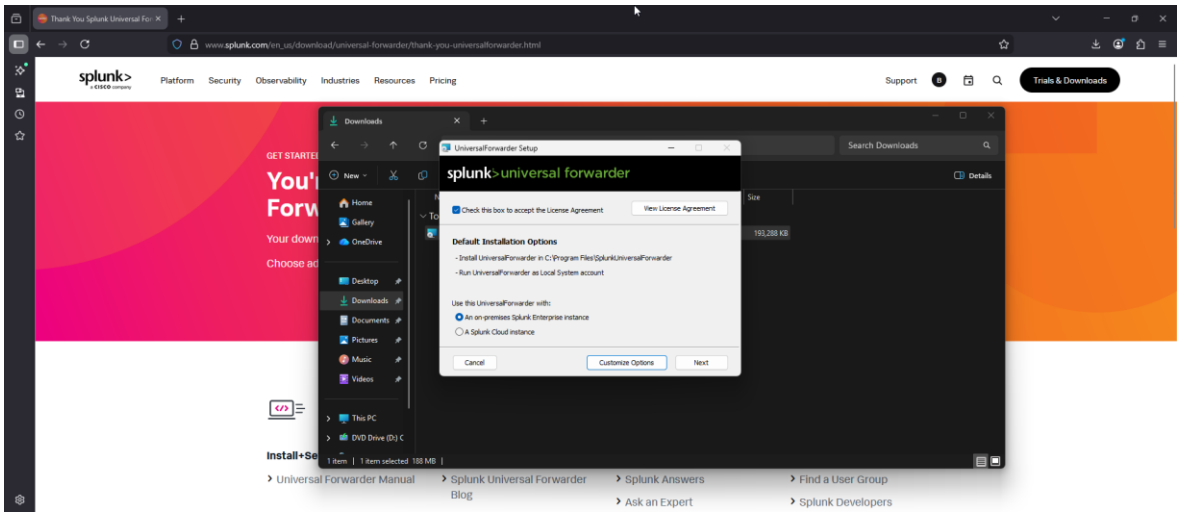
Why this matter?

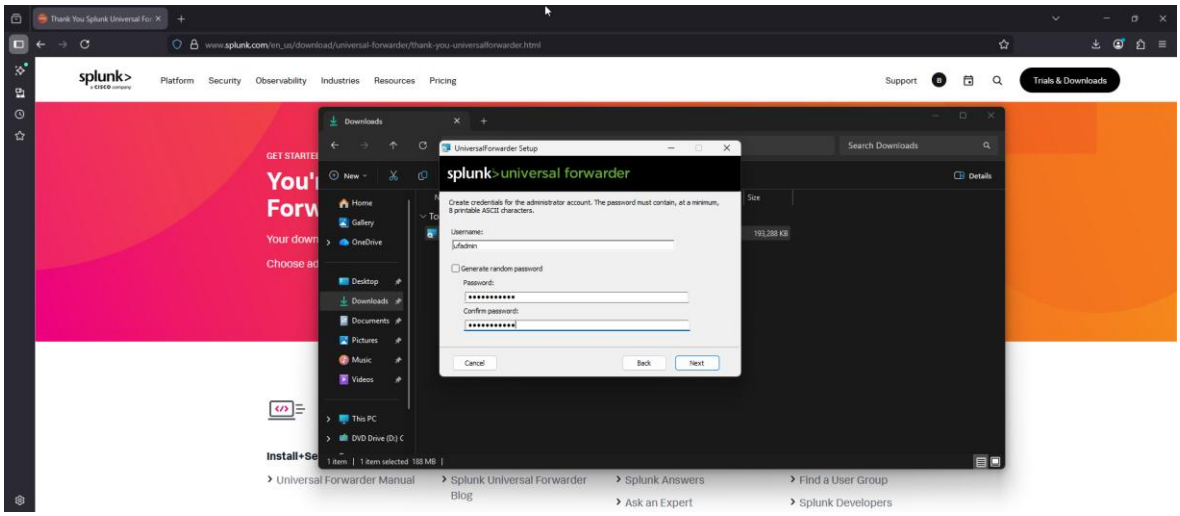
- Demonstrates real-world Splunk administration, not lab shortcuts
- Shows understanding of index lifecycle, performance, and governance
- Aligns with SOC-grade operational standards
- This is how Splunk is structured in production security environments, not demos.

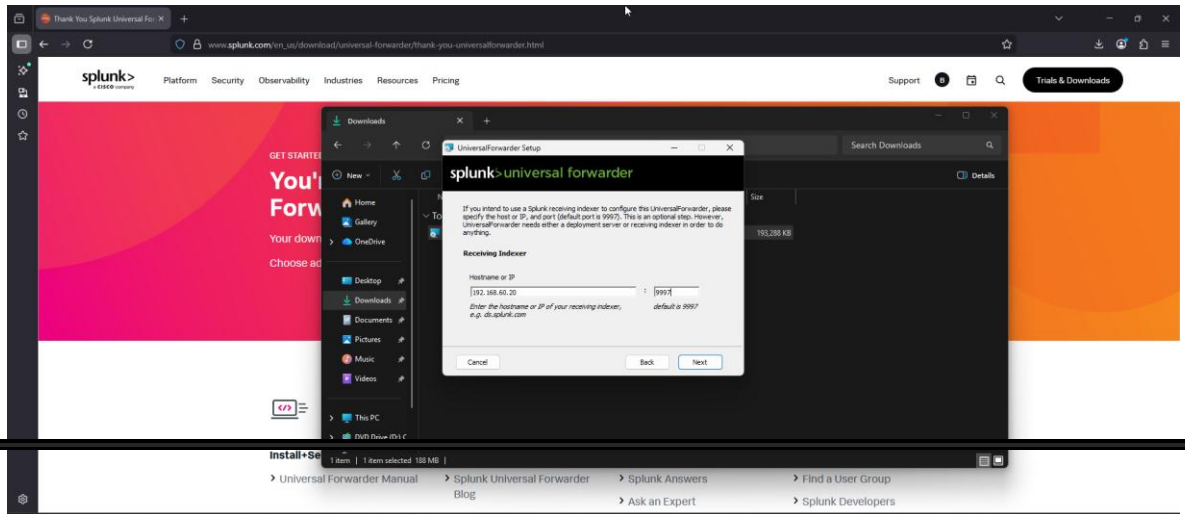
C. Splunk Universal Forwarder install

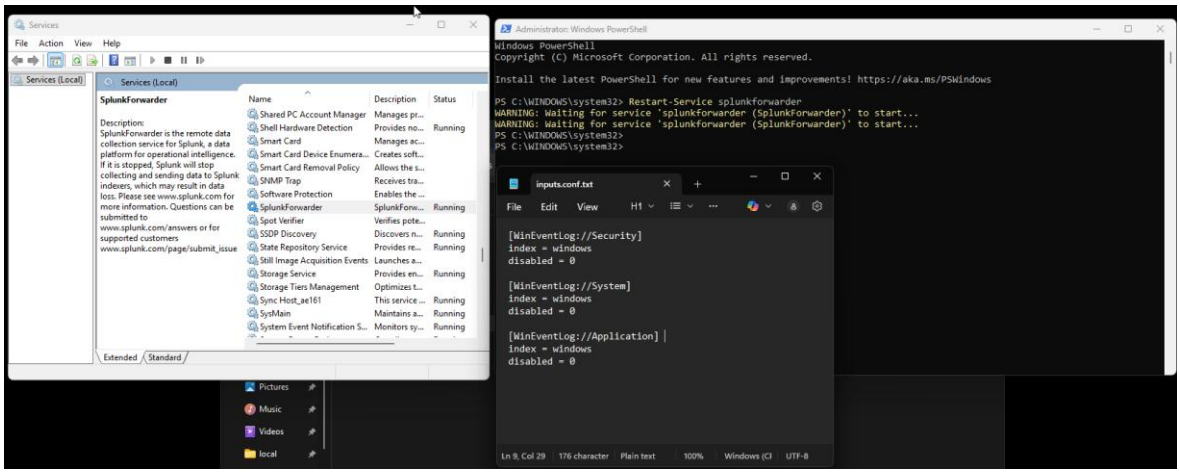
Gui installation:

[**screenshot** 👆🏾 👇🏾]









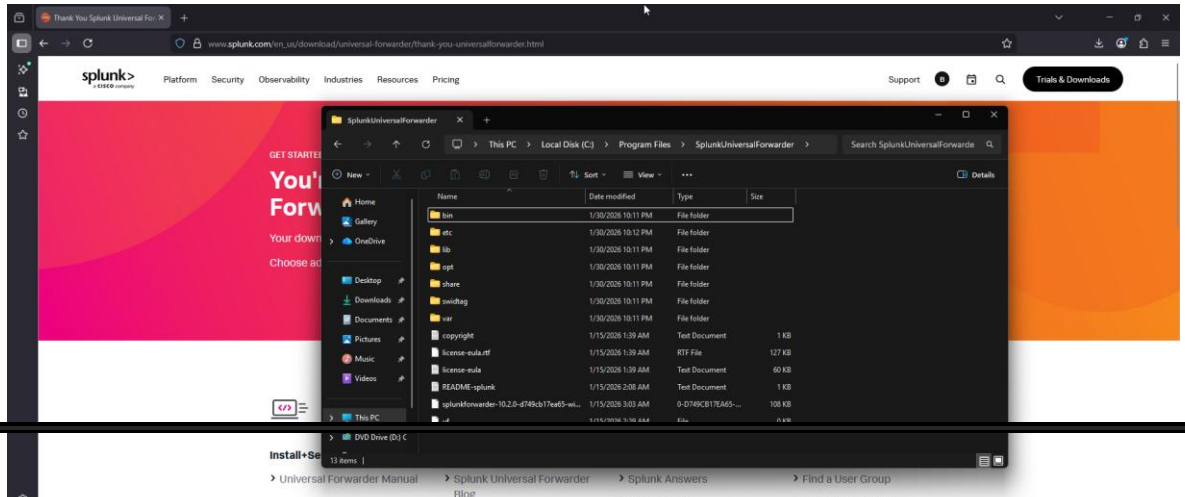
D. Configure UF inputs for Windows Event Logs

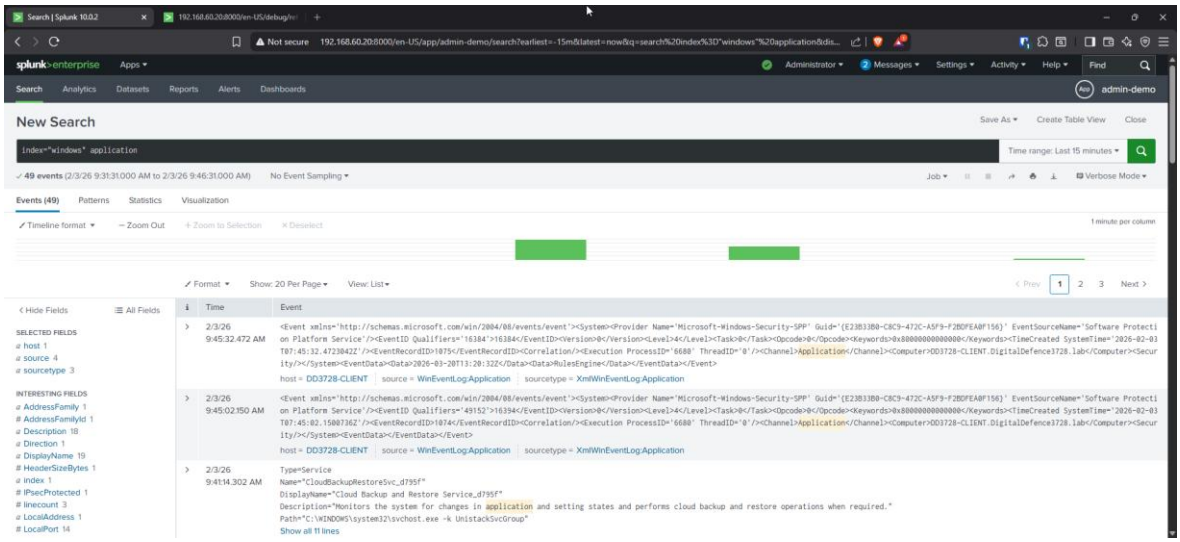
Create/edit C:\ProgramFiles\SplunkUniversalForwarder\etc\system\local\inputs.conf:

```
[WinEventLog://Security]
index = windows
disabled = 0

[WinEventLog://System]
index = windows
disabled = 0

[WinEventLog://Application]
index = windows
disabled = 0
```

Restart UF:

```
powershell

Restart-Service splunkforwarder
```





"For the Windows client, I configured Splunk Universal Forwarder telemetry using the production-grade Splunk_TA_windows add-on from Splunkbase."

Quick breakdown 👆👇

- Started with basic Event Log collection (Security, System, Application) → `index=windows`
- Upgraded to Splunk_TA_windows add-on - copied to `C:\Program Files\SplunkUniversalForwarder\etc\apps\Splunk_TA_windows\`

Outcome

- Core logs: Security, System, Application, Defender AV, ForwardedEvents
- Performance monitoring: WMI every 10min - CPU, Memory, Disk, Services, Processes, Network
- Security telemetry: Registry Run key monitoring, listening ports hourly, installed apps daily
- Infrastructure: DHCP logs, Windows Firewall logs, Windows Update status, AD replication, BIOS data
- Network: Inbound/outbound connection tracking via WinNetMon

Single restart → Restart-Service splunkforwarder → enterprise-grade endpoint telemetry flowing to Splunk.

This shows:

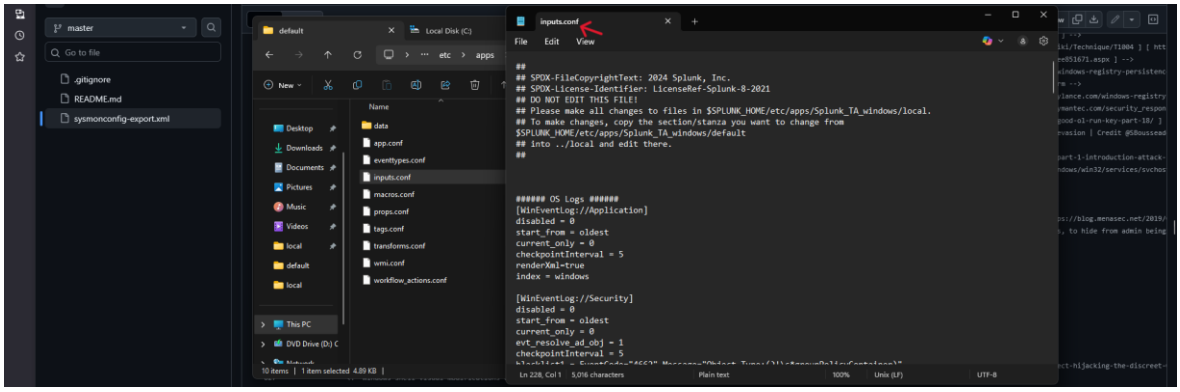
✅ UF deployment + add-on management

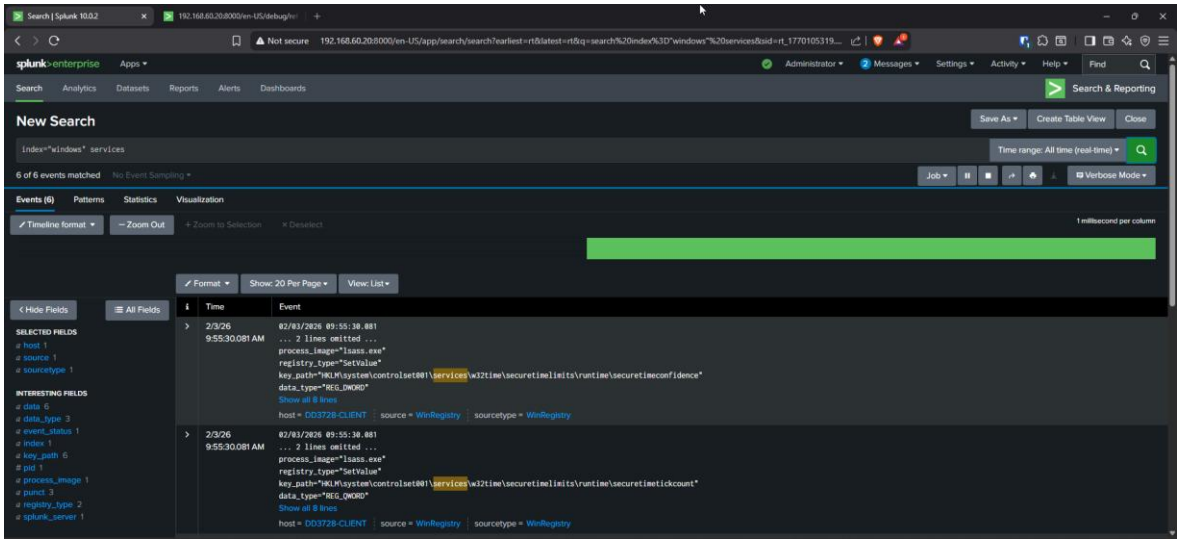
✅ Enterprise telemetry collection (not just basic logs)

✅ Production configs

✅ Security-relevant sources (Defender, Run keys, listening ports)

[**screenshot** 👆🏾👇🏾]



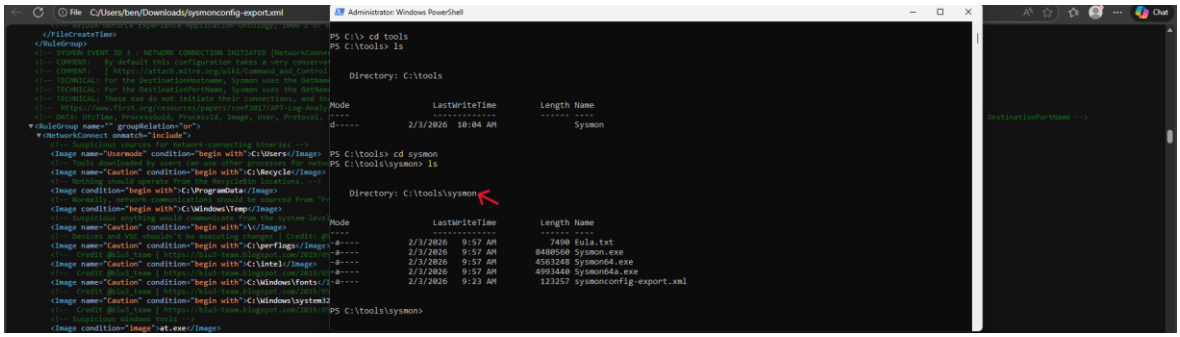


E. Install Sysmon

Downloaded Sysmon Sysmon v15.15 and a vetted configuration SwiftOnSecurity's sysmonconfig-export.xml

Extracted Sysmon zip file and created Tools folder under C:\ path then Placed in C:\Tools\Sysmon:

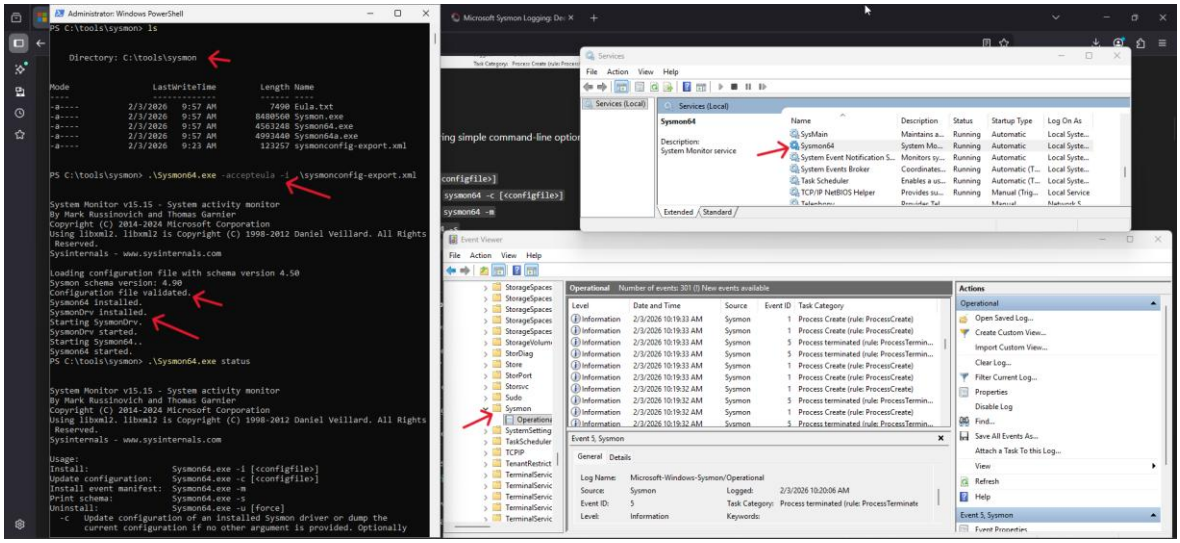
[**screenshot** 👆🏾👇🏾]



Ran:

```
#powershell

C:\Tools\Sysmon\Sysmon64.exe -accepteula -i

C:\Tools\Sysmon\sysmonconfig-export.xml
```

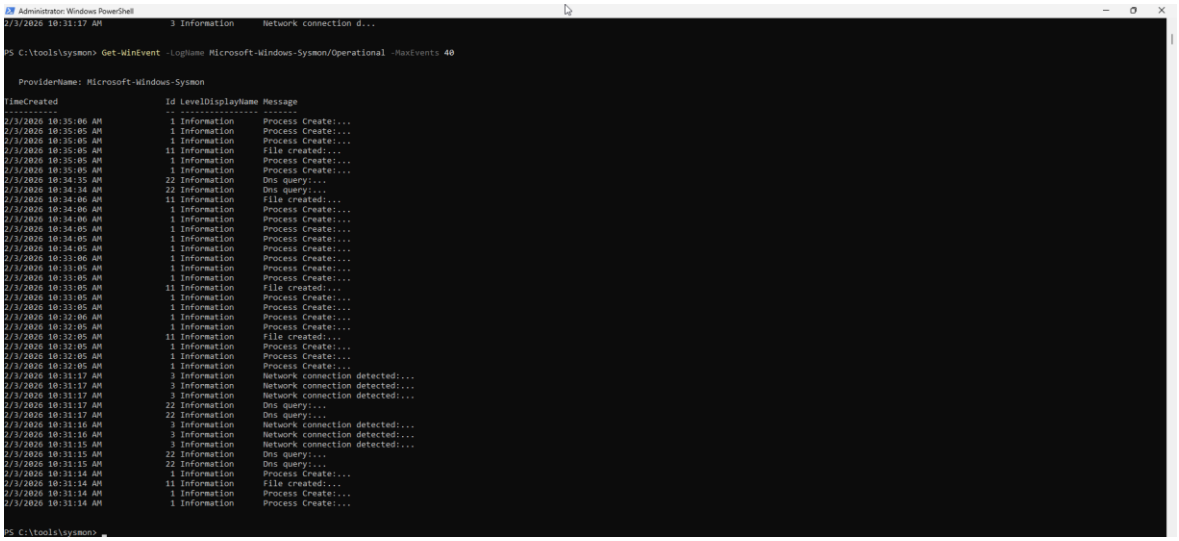
[**screenshot** 👆 👇]



Check event log:

```
#powershell

Get-WinEvent -LogName Microsoft-Windows-Sysmon/Operational -
MaxEvents 40
```

[**screenshot** 👆🏽👇🏽]

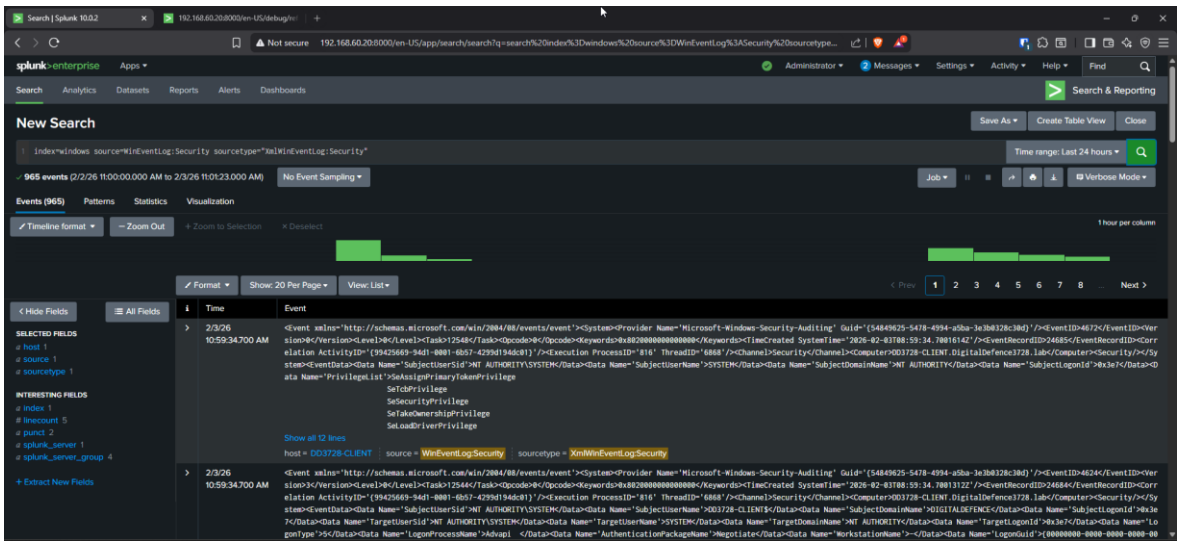


F. Validate ingestion in Splunk

In Splunk Web:

index=windows sourcetype=WinEventLog:Security

[**screenshot** 👆🏽👇🏽]



10 — Kali & Metasploitable2: controlled usage

- Kept Kali and Metasploitable2 on the 192.168.60.0/24 LAN and powered off when not testing.
- Use Kali for port scanning (nmap), web testing, and limited Metasploit modules to validate detections.

- Metasploitable2 is the intentionally vulnerable Linux target for safe exploitation exercises.

Kali static IP (Network Manager):

```
#bash

nmcli con mod "Wired connection 1" ipv4.addresses 192.168.60.40/24
ipv4.gateway 192.168.60.1 ipv4.dns 192.168.60.10 ipv4.method
manual
nmcli con up "Wired connection 1"
```

[**screenshot** 👆🏾👇🏾]

Metasploitable2 typically uses DHCP; ensured it gets a 192.168.60.x address from pfSense.

11 — Troubleshooting Log (Chronological Order)

SRV Lookup / DNS Failure on Domain Controller (DC) Symptom 👇🏾🟫

- nslookup _ldap._tcp.dc._msdcs.digitaldefence3728.lab (a command to query DNS for server records) returned "Server Unknown; Address: 127.0.0.1" with no response.

Root Cause:

- The Domain Controller (DC—a server managing Active Directory logins and authentication) was renamed after setup, leaving outdated records for the old hostname in DNS (Domain Name System—the internet's phonebook for translating names to IP addresses).

Fix:

- Old A (hostname-to-IP) and PTR (IP-to-hostname) records were deleted; correct A record for "dd3728-dc" was created; ipconfig /registerdns was run to refresh; Netlogon service was restarted; nltest /dsregdns and dcdiag /test:dns (Domain Controller diagnostic tool) were run until all passed.

Set-NetConnectionProfile Refusing Domain Authenticated 👇🏾

Symptom:

- Windows refused to set the network profile to DomainAuthenticated (a secure Windows network type for domain-joined machines).

Explanation:

- Windows only switched to DomainAuthenticated after DNS and Kerberos (a secure authentication protocol) worked properly; Private profile was used as a temporary fix while Active Directory (AD—a Microsoft directory service for user/device management) DNS issues were resolved.

Ubuntu DNS Using 127.0.0.53 and NXDOMAIN/SERVFAIL 👇🏾

Symptom:

- nslookup showed 127.0.0.53 (Ubuntu's local stub resolver—a lightweight DNS forwarder) as the resolver; internal domain names failed with NXDOMAIN (domain does not exist) or SERVFAIL (server failure) errors.

Root Causes:

- Conflicting Netplan files (Ubuntu's network config tool)—like 50-cloud-init.yaml overriding 00-installer-config.yaml; wrong file permissions; or incorrect subnet (IP network range).

Fix:

- Cloud-init networking was disabled; 50-cloud-init.yaml was deleted; correct IP 192.168.60.20/24 (subnet mask for network range) and nameserver 192.168.60.10 were set in Netplan; secure permissions were applied; netplan generate and netplan apply were run; systemd-resolved (Ubuntu's DNS resolver service) was restarted; validation was done with resolvectl and nslookup.

Wrong Subnet Symptom: 👇🏾

Symptom:

- Security Information and Event Management (SIEM—a tool for monitoring security alerts) was wrongly placed in 192.168.60.0/24 or previously set to 192.168.37.0/24, blocking reach to the DC.

Fix:

- Netplan was updated to 192.168.60.0/24 subnet and re-applied to restore network connectivity.

RDP / Firewall / Profile Mismatches 👇

Symptom:

- Remote Desktop Protocol (RDP—a remote access tool) connection failed until firewall rules and network profiles were fixed.

Fix:

- Remote Desktop firewall group was enabled; network profile was temporarily set to Private; AD DNS health was confirmed, so it switched to DomainAuthenticated.

12 — Verification & validation commands

Windows DC**:**

```
#powershell

dcdiag /v
nltest /dsregdns
ipconfig /registerdns
ipconfig /all
nslookup dd3728-dc.digitaldefence3728.lab
```

Ubuntu SIEM**:**

```
#bash

ip a
ip route
resolvectl status
nslookup dd3728-dc.digitaldefence3728.lab
sudo systemctl status splunk
tail -f /opt/splunk/var/log/splunk/splunkd.log
```

Splunk searches**:**

```
index=endpoint EventCode=1 | sort - _time | head 50
index=network sourcetype=pfsense | head 50
index=windows EventCode=4625 | stats count by Account_Name,
```

```
ComputerName | where count > 3
```

13 — Why the Domain Controller is Authoritative DNS

- In Active Directory (AD) environments, Domain Name System (DNS) is the main way clients discover services. Service (SRV) records, Address (A) records, and Pointer (PTR) records help clients locate Lightweight Directory Access Protocol (LDAP), Kerberos authentication, and Global Catalog services.
- These records must live in an AD-integrated DNS zone on Domain Controllers (DCs) to enable secure dynamic updates and proper Group Policy Object (GPO) behavior.
- The Domain Controller at 192.168.60.10 runs DNS and is authoritative for digitaldefence3728.lab.
- PfSense acts as the gateway, Network Address Translation (NAT) device, and optional Dynamic Host Configuration Protocol (DHCP) server—but it does not replace AD DNS.

14 — Kali / Metasploitable: Network Setup

```
#bash

# Kali static via NetworkManager
nmcli con mod "Wired connection 1" ipv4.addresses 192.168.60.40/24
ipv4.gateway 192.168.60.1 ipv4.dns 192.168.60.10 ipv4.method
manual
nmcli con up "Wired connection 1"
```

[screenshot 👆🏽👇🏽]

```
# Metasploitable set DNS
sudo sh -c 'echo "nameserver 192.168.60.10" > /etc/resolv.conf'
```

[screenshot 👆👇]