Internship Report
On
NATIONAL AEROSPACE LABORATORY

Information and communication technologies division

Submitted in partial fulfilment of the requirement for the V semester

Bachelor of Engineering

in

Computer Science and Engineering

at

CHENNAI INSTITUTE OF TECHNOLOGY (Autonomous)

Sarathy Nagar, Kundrathur, Chennai-600069

Submitted by:

DHANUSREE D (23CS039)

Submitted to:

SHRI. KARTHIKEYAN S (Senior Technical Officer)

Internship

Date: 05-05-2025 to 17-06-2025

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to the Information and Communication Technologies Division of National Aerospace Laboratory (NAL) for Providing me with the opportunity to undertake my internship. This experience has been invaluable in enhancing my understanding of Information Technology and its Practical applications.

I am thankful for the guidance of my guide, SHRI. KARTHIKEYAN S (Senior Technical Officer) of Information and Communication Technologies Division in National Aerospace Laboratory, for his unwavering support, guidance and mentorship throughout the internship. Their insights and expertise significantly contributed to my learning experience, which has kept us working hard.

I would be grateful to the entire team at the Information and Communication Technologies Division for sharing their knowledge and experiences with me. The collaborative and innovative environment fostered by the team has inspired me to pursue my passion for IT even further.

It was a great pleasure for me to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this internship.

First, I take this opportunity to express our sincere gratitude to Karthikeyan S sir for providing an opportunity in the Information and Communication Technologies Division, NAL Bangalore. In our college providing me with an excellent opportunity to allow for an internship.

I express my gratitude to my HOD, Principal in Chennai Institute of Technology, Chennai for providing me excellent facilities and academic ambience which have helped me in satisfactory completion of the Bachelor Degree. In debt to the support of all the teaching and non-teaching members of the Department of Information Technology, for their kind help and cooperation, throughout our graduation. Their constant support and love have made this journey memorable.

ABSTRACT

This report present my internship experience, which focused on learning and applying concepts in network security and firewall deployment. Throughout the internship, I had the chance to work hands-on with tools like OPNsense, WireGuard, VMware, PuTTY, and Wireshark. The experience was a great blend of theory and practical implementation, allowing me to build real-world skills in system configuration, secure communication, and infrastructure simulation. At the start, I worked on setting up Wazuh to monitor endpoints and learned how to install Ubuntu operating systems on virtual machines. This helped me get comfortable with the basics and set the stage for more advanced tasks. I also revisited core networking concepts by using PuTTY to configure switches a simple but essential step that helped reinforce foundational knowledge. The main part of my project involved creating a simulated network environment using two OPNsense firewalls running on VMware. Each firewall was connected to a Kali Linux instance, simulating two separate networks. I configured interfaces, assigned IP addresses, set up gateways, and created firewall rules to allow secure two-way communication between the networks. I tested everything using simple tools like ping and was excited to see the communication working as expected, just like in a real-world setup.Once the virtual setup was successful, I moved on to replicating the configuration on physical OPNsense firewalls, modifying an earlier pfSense setup. This part taught me a lot about adapting configurations between environments and maintaining network consistency. To secure the communication between the two firewalls, I set up a WireGuard VPN tunnel. Creating and assigning tunnel IPs on both sides allowed me to establish a secure connection between them. To make sure the data was truly encrypted and secure, I used Wireshark to capture and inspect the traffic. Seeing encrypted packets flow through the tunnel confirmed that the VPN was working correctly and added an extra layer of confidence to my setup. Overall, this internship gave me meaningful, hands-on experience in network security. I got to apply what I've learned in a controlled environment, troubleshoot real configurations, and see how secure networks are built and maintained. It's been a valuable step in preparing for a future career in cybersecurity and networking.

INTRODUCTION

This internship, which began on May 5, 2025, offered a valuable learning experience in network security and system configuration. I worked on a series of tasks involving open-source tools such as OPNsense, VMware, PuTTY, WireGuard and WireShark focusing on secure network simulation, firewall setup, and encrypted communication. The internship emphasized practical implementation and problem-solving in the context of cybersecurity and infrastructure design. My responsibilities began with installing Wazuh for endpoint detection and monitoring, followed by gaining proficiency in installing and configuring Ubuntu OS. I then moved on to hands-on switch configuration using PuTTY, where I learned to manage network connectivity at the hardware level. The core of my internship involved setting up a simulated network using two virtual OPNsense firewalls on VMware, each connected to a Kali Linux instance. I configured interfaces, assigned gateways, and created firewall rules to enable successful bidirectional pinging and communication between the networks. Once the simulation was successful, the project transitioned to a real-world implementation by replacing the virtual setup with two physical firewalls. I adapted configurations from a previous pfSense environment to OPNsense, ensuring compatibility and functionality. To secure communication between the firewalls, I implemented WireGuard VPN tunnels. I created unique tunnel IPs, configured routing rules, and used Wireshark to capture and analyze packet data, validating that encryption was successfully applied. Throughout the internship, I faced challenges such as debugging firewall rules, managing interface configurations, and ensuring secure, encrypted connections without compromising performance. These experiences enhanced my technical problem-solving abilities and gave me deeper insight into real-world cybersecurity practices. The internship has solidified my interest in network security and inspired me to explore further opportunities in secure infrastructure management and automation.

Week 1: Wazuh Installation

Introduction to Wazuh

Wazuh is an open-source security platform that provides threat detection, integrity monitoring, incident response, and compliance management. It uses a modular architecture consisting of three main components:

- **Wazuh Agent**: Installed on endpoints (like servers or computers), it collects system data such as logs, file integrity checks, rootkit detection, and more.
- **Wazuh Server**: The central component that receives data from agents, analyzes it based on security rules, generates alerts, and stores results.
- **Wazuh Dashboard**: A web interface that allows users to view security events, system status, and reports in a visual and user-friendly format. It's built on top of Kibana and requires Elasticsearch to function.

In this internship, the focus was on learning and setting up the **Wazuh Agent** only, without using the Wazuh server or dashboard, to understand how system-level security monitoring works.

Task 1: Installing Wazuh Agent

Objective:

To install the Wazuh Agent on a local Ubuntu machine and understand how it monitors system logs and activities independently.

Steps Followed:

1.Add the Wazuh repository and GPG key:

```
curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | sudo apt-key add -
```

```
echo "deb https://packages.wazuh.com/4.x/apt/ stable main" | sudo tee
/etc/apt/sources.list.d/wazuh.list
```

2.Update the package list:

```
sudo apt update
```

3.Install the Wazuh Agent:

```
sudo apt install wazuh-agent -y
```

4.Enable and start the agent service:

```
sudo systemctl enable wazuh-agent
```

```
sudo systemctl start wazuh-agent
```

5.Check the status of the Wazuh agent to confirm it’s running:

```
sudo systemctl status wazuh-agent
```

Outcome:

The Wazuh agent was successfully installed and configured to run as a service. Although no server was connected, the agent still performed local monitoring, which helped in understanding its basic functionality.

Task 2: Exploring Agent Logs and Configuration

Objective:

To explore how the Wazuh Agent operates after installation and learn where its data is stored and how it logs system activity.

Steps Followed:

1.Located the main configuration file of the Wazuh Agent:

```
/var/ossec/etc/ossec.conf
```

This file controls what the agent monitors, such as files, commands, and logs.

2.Checked real-time logs to see agent activity:

```
sudo tail -f /var/ossec/logs/ossec.log
```

This showed entries related to system monitoring, file changes, and process checks.

- **Observed default modules running, such as:**
  - **Syscheck** – for file integrity monitoring.
  - **Rootcheck** – for rootkit and malware detection.
  - **Logcollector** – for log file parsing.
- **Understood how Wazuh Agent behaves standalone** – even without the server and dashboard, it collects data and logs it locally for analysis.

Outcome:

Gained practical understanding of how the Wazuh Agent monitors a system, its configuration, and how it logs security events. This task laid the foundation for understanding host-based intrusion detection systems (HIDS).

Task 3: Service Management and Directory Exploration

Objective:

To manage the Wazuh agent service and understand the structure of its configuration and log directories.

Steps Followed:

1. Restarted the Wazuh agent service to ensure it runs smoothly after changes or interruptions by executing:
   ```
   sudo systemctl restart wazuh-agent
   ```
2. Verified the status of the agent service to confirm it was active and running without errors using:
   ```
   sudo systemctl status wazuh-agent
   ```
3. Explored the main Wazuh directories to understand where configuration files and logs are stored:
   - Configuration files located in /var/ossec/etc/

  - Log files located in /var/ossec/logs/
  - Additional directories for temporary data and scripts.
4. Reviewed the purpose of key modules such as Syscheck (file integrity monitoring) and Rootcheck (security auditing) that help the agent monitor system health continuously.

Conclusion:

This week, I successfully installed and verified the Wazuh agent on a standalone system, gaining a solid foundation in endpoint security monitoring. Although I did not set up the server or dashboard, working with the agent helped me understand how it independently collects and logs critical system events like file changes and process activity. This showed how the agent continuously monitors the system and maintains local security records without needing a central server. Managing the agent as a service also taught me how such tools integrate with the operating system to ensure ongoing protection even after restarts. This experience laid the groundwork for future work with centralized management and alerting.

Week 2: Ubuntu OS Installation and Switch Configuration Using PuTTY

Task 4: Install Ubuntu OS

Step 1: Downloading the Ubuntu ISO File

- Research the different Ubuntu editions (Desktop, Server, minimal) to choose the best fit for your requirements.
- Confirm system hardware compatibility with the chosen Ubuntu version (32-bit vs 64-bit).
- Download the ISO from the official Ubuntu site or a trusted mirror to ensure file integrity and security.
- Verify the ISO checksum (MD5 or SHA256) to ensure the download is not corrupted or tampered with.
- Read release notes and system requirements to prepare for installation.

- Save the ISO file in an organized directory for easy access during the next steps.

Step 2: Creating a Bootable Installation Media

- Choose the installation method based on your hardware: USB flash drive for physical machines or ISO mounting for virtual machines.
- Download and install reliable tools like Rufus (Windows), balenaEtcher (cross-platform), or UNetbootin for creating bootable USB drives.
- Format the USB drive properly before writing the ISO to avoid errors.
- For virtual machines, configure the VM's virtual CD/DVD drive to use the ISO image as the boot source.
- Verify the bootable media by safely ejecting the USB and checking it on another system or VM.
- If using USB, ensure the drive has sufficient capacity and is in good health to avoid installation failures.

Step 3: Booting From the Installation Media

- Access the BIOS/UEFI settings by pressing keys like F2, F12, DEL, or ESC during startup.
- Change the boot priority order to place the USB drive or virtual CD/DVD at the top.
- Disable secure boot if it conflicts with Ubuntu installation on certain hardware.
- Save BIOS settings and reboot the system to load the Ubuntu installer.
- Observe the boot menu for options such as "Try Ubuntu without installing" to test compatibility before installation.
- Use troubleshooting options if the boot fails, such as checking USB integrity or BIOS settings.

Step 4: Selecting Installation Preferences

- Choose the preferred language for the installation and system locale to ensure all menus and prompts display correctly.
- Select the keyboard layout that matches your physical keyboard to avoid typing errors.

- Enable accessibility options if required for easier interaction during installation.
- Review time zone settings to ensure system clock accuracy.
- Opt to install third-party software for graphics, Wi-Fi drivers, and media codecs if needed.
- Confirm installation type (normal or minimal) depending on disk space and usage requirements.

Step 5: Disk Partitioning and Setup

- Analyze existing disk partitions and data to avoid accidental loss during partitioning.
- Choose guided partitioning if you want automatic setup or manual partitioning for advanced control.
- Create a root partition with adequate size (minimum 20GB recommended) formatted as ext4.
- Set up a separate home partition for personal files to simplify backups and OS reinstalls.
- Allocate swap space based on system RAM (usually 1-2 times the RAM size) to improve performance during heavy load.
- Configure mount points and filesystem types carefully to optimize system stability and performance.

Step 6: Creating User Accounts and Security Settings

- Enter a secure username and a strong password following best security practices.
- Choose whether to require password on login or enable automatic login (less secure).
- Assign a descriptive hostname for easier identification on the network.
- Explore encryption options like full-disk encryption or encrypted home directory for data protection.
- Enable firewall settings (using tools like UFW) post-installation to protect the system from unauthorized access.
- Set up SSH server if remote access will be required in your network environment.

Step 7: Installing and Completing Setup

- Monitor the installation progress to ensure files are copied without errors.
- If connected to the internet, allow the installer to download updates and drivers to keep the system current from the start.
- Choose to install third-party drivers if prompted, especially for proprietary hardware components.
- Once installation finishes, remove the installation media to avoid boot loops.
- Reboot into the fresh Ubuntu environment and perform initial login with the user credentials created.
- Check system logs if installation issues arise to troubleshoot effectively.

Step 8: Post-Installation Exploration

- Familiarize yourself with the Ubuntu desktop interface, including menus, settings, and software center.
- Open the terminal and practice basic Linux commands such as ls, cd, sudo apt update, and sudo apt install to manage packages.
- Customize the desktop environment to suit your workflow, like installing preferred text editors or network tools.
- Explore Ubuntu's security features such as AppArmor profiles and firewall configuration.
- Set up essential network tools for future projects, like OpenSSH, Wireshark, or network monitoring utilities.
- Schedule regular system updates and backups to maintain stability and data safety.

Task 5: Configuring Switches Using PuTTY

Step 1: Understanding Network Switches and PuTTY

- Research the different types of switches (managed vs unmanaged) and their roles.
- Learn about the OSI model layers switches operate on (Layer 2 and sometimes Layer 3).

- Understand why remote management is important for scalability and security.
- Explore PuTTY's features, such as session logging, saved profiles, and key-based SSH authentication.
- Test PuTTY connection to a dummy device or virtual switch to get familiar before the actual configuration.

Step 2: Connecting to the Switch

- Check physical connectivity: ensure Ethernet cable is plugged into the correct port on the switch and computer.
- Verify the IP address or hostname of the switch before connection.
- Choose the right protocol (SSH or Telnet) based on device support and security policies.
- Adjust PuTTY settings like connection timeout, keep-alive packets, and terminal type for better stability.
- Save the session profile in PuTTY for quick future access.

Step 3: Logging Into the Switch CLI

- Enter username and password carefully, respecting case sensitivity.
- Observe any login banners or messages about security policies.
- After login, check the current mode (user EXEC, privileged EXEC, or global configuration) and learn how to switch between them.
- Use the enable command to access privileged EXEC mode if needed.

Step 4: Setting the Switch IP Address

- Identify the correct management VLAN interface (e.g., VLAN 1 or another).
- Enter global configuration mode using the configure terminal command.
- Assign the static IP address and subnet mask to the management interface using commands like interface vlan 1 and ip address x.x.x.x y.y.y.y.
- Enable the interface with the no shutdown command.
- Test IP connectivity with ping from the switch to other devices.
- Check routing or gateway settings if remote access is required beyond the local subnet.

Step 5: Configuring VLANs (Virtual Local Area Networks)

- Determine the number of VLANs needed based on organizational or network design requirements.
- Create VLANs using the command vlan [ID] and assign meaningful names using name [VLAN-name].
- Assign ports to VLANs by entering interface configuration mode for each port and using switchport mode access and switchport access vlan [ID].
- Configure trunk ports if the switch connects to other switches, allowing multiple VLANs over one link with switchport mode trunk.
- Verify VLAN configuration with show vlan brief and show interfaces switchport.
- Understand VLAN tagging and untagging for proper traffic segregation.

Step 6: Enabling and Managing Interfaces

- Review all switch interfaces and their current status using show interfaces status.
- Enable or disable ports based on network topology and device connectivity needs using shutdown and no shutdown commands.
- Configure speed and duplex settings explicitly with speed [10|100|1000] and duplex [half | full] to avoid auto-negotiation mismatches.
- Apply security features on interfaces like port security to restrict device access by MAC address.
- Monitor interface errors and bandwidth utilization to detect potential hardware or configuration problems.

Step 7: Saving Configuration and Verifying Connectivity

- Use write memory or copy running-config startup-config commands to save configurations permanently.
- Confirm saved configuration by checking the startup-config file with show startup-config.
- Use ping and traceroute commands to test network connectivity and troubleshoot reachability issues.

- Check interface statuses, VLAN memberships, and routing tables to ensure settings are correctly applied.
- Perform configuration backup by exporting the config file for disaster recovery.

Step 8: Troubleshooting Common Issues

- When connectivity fails, verify IP addressing and subnet masks are correct and non-conflicting.
- Check VLAN assignments to ports and confirm devices are on correct VLANs.
- Use commands like show interfaces, show vlan, and show running-config to diagnose misconfigurations.
- Review logs with show logging to identify errors or security alerts.
- Test cable and physical connections to rule out hardware failures.
- Reset misbehaving interfaces or reload the switch if necessary, after saving configuration.
- Document issues and resolutions for future reference and knowledge sharing.

Conclusion

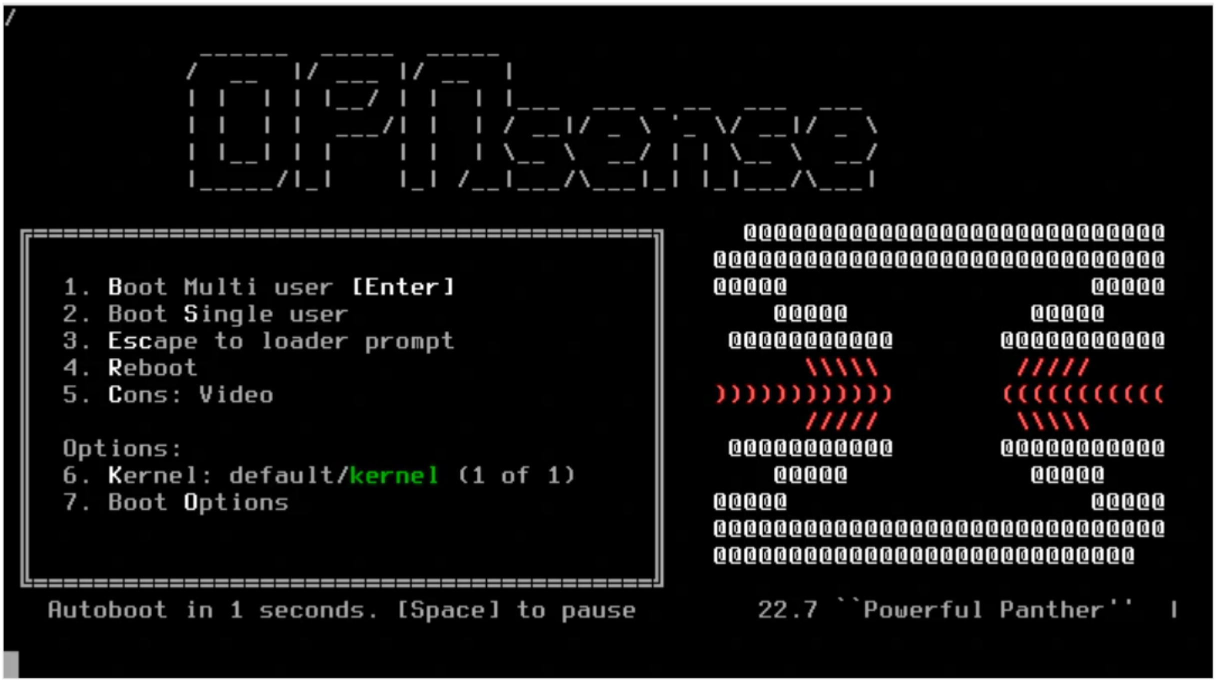
During Week 2, I gained essential hands-on experience by learning to install the Ubuntu operating system and configuring network switches using PuTTY. Installing Ubuntu provided me with a solid understanding of setting up a reliable and secure OS environment, which is fundamental for deploying network and security tools. Configuring switches through PuTTY enhanced my knowledge of remote network management, allowing me to work with VLANs, interfaces, and basic switch commands to control and segment network traffic effectively. Together, these tasks improved my technical skills and troubleshooting abilities, building a strong foundation in both system administration and network configuration that will support my future work in network security.

Week 3: Setting Up Network Security: Installing OPNsense Firewall and Integrating Kali Linux

Task 6: OPNSense Installation

Step 1: Download and Extract OPNsense ISO

- Visit the official OPNsense website and download the latest ISO image.
- The downloaded file usually has a .bz2 extension, which is a compressed file format.
- Extract the .bz2 file using extraction software or command-line tools to get the .iso file needed for installation.



```
 1. Boot Multi user [Enter]
 2. Boot Single user
 3. Escape to loader prompt
 4. Reboot
 5. Cons: Video

 Options:
 6. Kernel: default/kernel (1 of 1)
 7. Boot Options

Autoboot in 1 seconds. [Space] to pause        22.7 ``Powerful Panther''
```

Diagram:1

Step 2: Create a New Virtual Machine in VMware

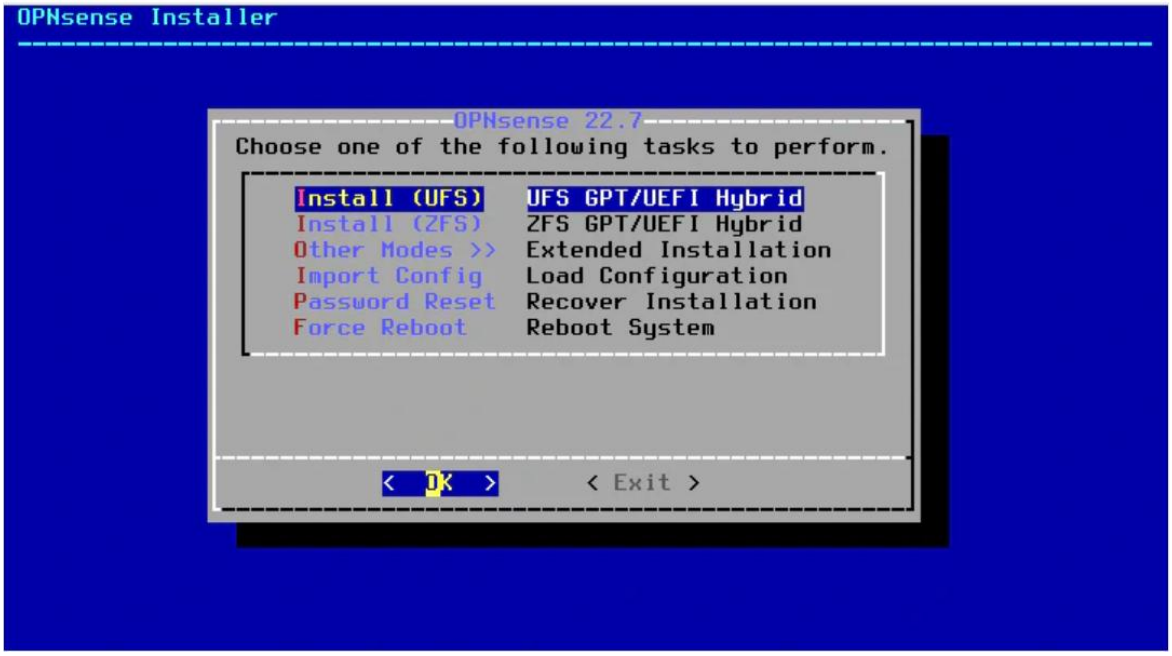
- Open VMware Workstation and start creating a new VM.
- Choose “Custom” installation to manually configure settings.
- When prompted for the guest OS, select **FreeBSD** because OPNsense is based on FreeBSD.
- Allocate at least **2 GB RAM**, **1 CPU core**, and **20 GB storage** for the VM.
- Use the extracted OPNsense ISO as the installation media for this VM.

Step 3: Configure Network Adapters

- Add **two network adapters** to the VM:
  - The first adapter is set to **Bridged** mode to act as the WAN (internet-facing) interface.
  - The second adapter is set to a **Private LAN segment** to simulate the internal network (LAN).
- This setup allows OPNsense to route traffic between the external network (WAN) and internal network (LAN).

Step 4: Start the VM and Begin Installation

- Power on the VM and boot from the OPNsense ISO.
- The OPNsense installer will load, displaying the installation menu.
- Select the **default keyboard layout** and press Enter.
- Choose to **Install OPNsense with UFS (Unix File System)** for a stable filesystem.



```
OPNsense Installer
--------------------------------------------------------------------------------

              ----------------OPNsense 22.7----------------
              Choose one of the following tasks to perform.

                   Install (UFS)     UFS GPT/UEFI Hybrid
                   Install (ZFS)     ZFS GPT/UEFI Hybrid
                   Other Modes >>    Extended Installation
                   Import Config     Load Configuration
                   Password Reset    Recover Installation
                   Force Reboot      Reboot System

                       <  OK  >          < Exit >
```

Diagram: 2

Step 5: Select Installation Target Disk

- The installer will prompt you to select the disk to install OPNsense on.
- Select the virtual disk you created earlier (usually named something like da0).
- Confirm the installation and allow the installer to copy files and configure the system.

Step 6: Set the Root Password

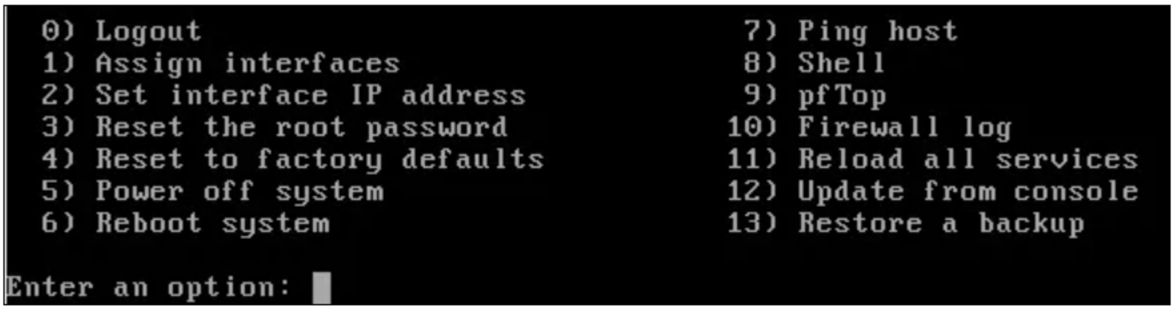
- After the installation completes, you will be prompted to set a **root password** for the system.
- Choose a strong password to secure administrative access.

Step 7: Reboot and Remove Installation Media

- Once installation finishes, reboot the VM.
- Before rebooting, **unmount the ISO** from the virtual CD/DVD drive to avoid booting into the installer again.

Step 8: Assign Network Interfaces

- After reboot, log in at the console with the username root and the password you set.
- The console menu will appear; select the option to **assign network interfaces**.
- When asked about advanced configurations like LAGGs or VLANs, select **no** for a basic setup.
- Assign the WAN interface to the bridged adapter (em0) and the LAN interface to the private network adapter (em1).
- Confirm MAC addresses to ensure correct mapping.



```
  0) Logout                              7) Ping host
  1) Assign interfaces                   8) Shell
  2) Set interface IP address            9) pfTop
  3) Reset the root password            10) Firewall log
  4) Reset to factory defaults          11) Reload all services
  5) Power off system                   12) Update from console
  6) Reboot system                      13) Restore a backup

Enter an option:
```

Diagram: 3

Step 9: Configure IP Addresses

- Configure the WAN interface to use **DHCP** (usually automatic) to get an IP from your network.
- Set a **static IP address** and subnet mask for the LAN interface to define your internal network.
- When asked, decide whether to enable a **DHCP server on LAN** (for the lab, it can be disabled).

Step 10: Access OPNsense Web Interface

- Open a web browser on a machine connected to the LAN network.
- Enter the LAN IP address you set during interface configuration to access the OPNsense dashboard.
- Use the root username and password to log in.
- The web interface uses HTTPS with a self-signed certificate, which is suitable for lab use.

Step 11: Start Configuring Your Firewall

- Once logged in, you can explore firewall rules, VPN setup, and other features through the GUI.
- This interface simplifies managing complex network security settings without command-line use.

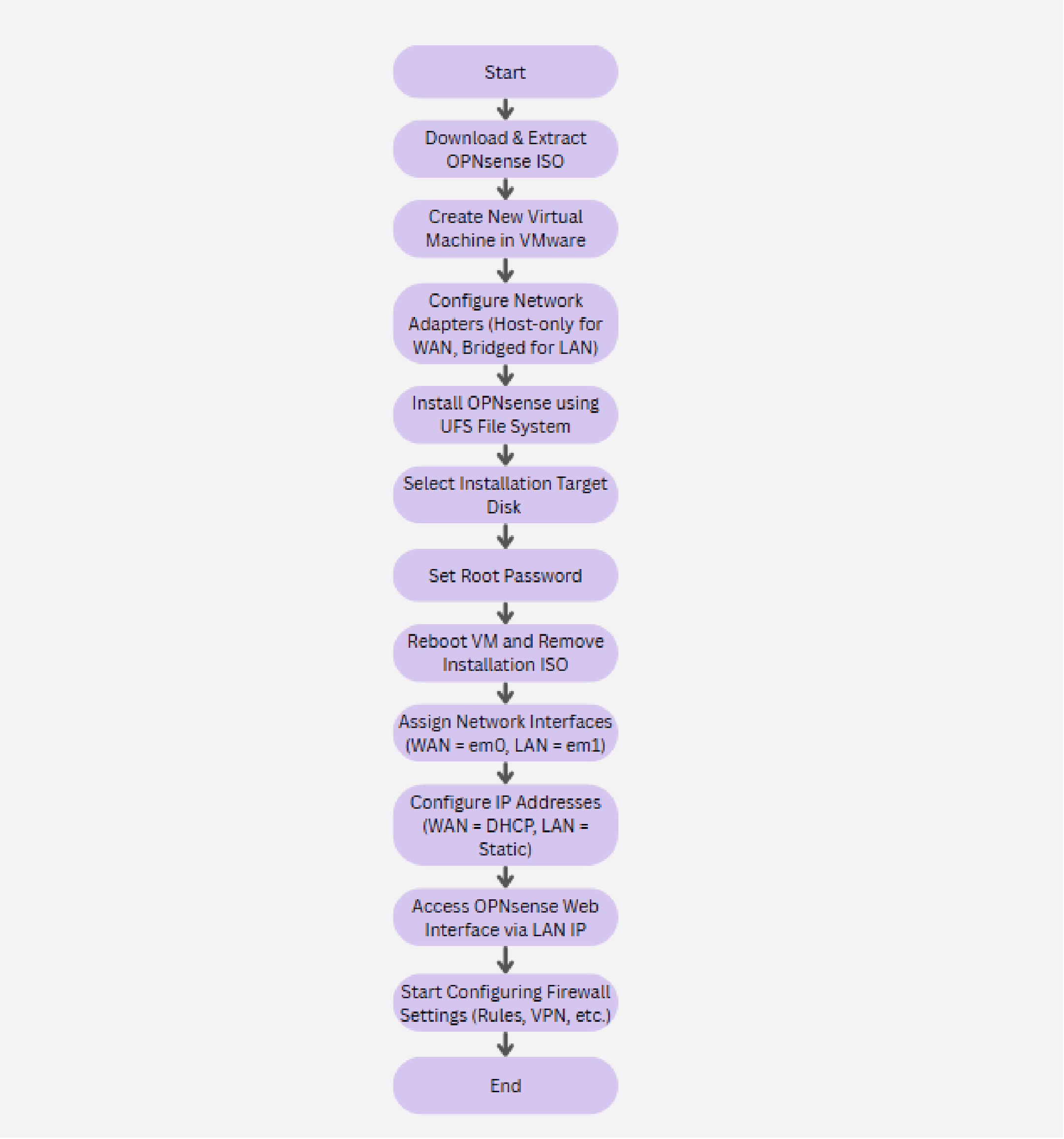

Start
Download & Extract OPNsense ISO
Create New Virtual Machine in VMware
Configure Network Adapters (Host-only for WAN, Bridged for LAN)
Install OPNsense using UFS File System
Select Installation Target Disk
Set Root Password
Reboot VM and Remove Installation ISO
Assign Network Interfaces (WAN = em0, LAN = em1)
Configure IP Addresses (WAN = DHCP, LAN = Static)
Access OPNsense Web Interface via LAN IP
Start Configuring Firewall Settings (Rules, VPN, etc.)
End

Diagram : 4

Task 7: Kali Linux Installation and Integration

Step 1: Create a New Virtual Machine for Kali Linux

Open VMware Workstation and start creating a new virtual machine. Choose “Custom” setup for manual configuration. When prompted to select the operating system, choose “Linux” and select “Debian 64-bit” if Kali Linux is not listed directly. Assign at least 2 GB of RAM, 2 processor cores, and around 20–30 GB of disk space to the VM. Use the Kali Linux ISO as the installation media.

Step 2: Configure Kali’s Network Adapter

In the VM’s network settings, remove the default NAT adapter and instead **add a network adapter connected to the same LAN segment** used in your OPNsense setup (the one you assigned as LAN in OPNsense). This ensures Kali Linux is placed inside the internal network protected by the firewall. This way, Kali can access the internet through OPNsense and interact with devices inside the LAN.

Step 3: Install Kali Linux

Boot the VM using the Kali ISO. Follow the standard installation steps:

- Select your language, location, and keyboard layout.
- Set hostname and domain (optional).
- Configure a user and password.
- Partition the disk (use guided option unless you prefer manual).
- Wait for the installation to complete and reboot the system.

Step 4: Assign a Static IP Address to Kali (optional)

After installation, log in to Kali Linux. You can either set a static IP (e.g., 192.168.1.10) or let Kali get an IP from OPNsense if DHCP is enabled on the LAN interface. If you opted not to use DHCP:

```
sudo nano /etc/network/interfaces
```

And add:

```
auto eth0
iface eth0 inet static
  address 192.168.1.10
  netmask 255.255.255.0
  gateway 192.168.1.1
  dns-nameservers 8.8.8.8
```

Then run:

```
sudo systemctl restart networking
```

Step 5: Test the Integration

Open a terminal in Kali Linux and try pinging:

- 192.168.1.1 (LAN IP of OPNsense) to verify local connectivity.
- 8.8.8.8 or google.com to check internet access (only works if firewall rules allow WAN access from LAN).
- Access OPNsense's web UI at https://192.168.1.1 from Kali’s browser to confirm full connectivity.

Step 6: Start Penetration Testing or Traffic Monitoring

With Kali Linux behind the OPNsense firewall, you can now test:

- Firewall rules (e.g., blocking/allowing traffic from Kali).
- IDS/IPS features in OPNsense like Suricata.
- Network scanning tools in Kali like nmap to see how the firewall reacts.

Conclusion:

This OPNsense setup is implemented entirely for simulation and educational purposes within a virtualized lab environment. The following points highlight the intent and benefits of this configuration:

- It allows for safe experimentation with firewall policies, NAT rules, and security features without impacting a live network.
- The use of Kali Linux behind the OPNsense firewall enables testing of penetration tools, network scans, and intrusion detection in a controlled manner.
- The virtual lab provides an ideal environment for students and cybersecurity enthusiasts to understand how traffic flows between WAN and LAN.
- Simulating real-world network architecture with virtual machines prepares users for practical scenarios they may encounter in production environments.
- This approach supports repeated testing and learning by resetting or modifying network parameters as needed, promoting hands-on learning.
- The configuration helps in understanding the interaction between client machines and firewalls, as well as monitoring and logging network activities.

- The web interface of OPNsense offers a convenient way to manage firewall rules and visualize network security concepts.

Week 4: Configuring Default Gateways on OPNsense Firewalls

Task 8: Assigning Default Gateways on OPNsense Endpoints

Step 1: Access the OPNsense Web Interface

- Open a web browser on a device connected to the OPNsense LAN or transit network.
- In the address bar, type the LAN IP address of the first OPNsense firewall, for example, https://192.168.1.1.
- Since OPNsense uses a self-signed certificate, your browser may warn about security; choose the option to proceed anyway.
- When the login page appears, enter the username root and the password you set during installation.
- Successful login will take you to the OPNsense dashboard, which provides a user-friendly interface to manage all system settings.

Step 2: Navigate to Routing Settings

- From the dashboard, look for the menu at the top.
- Click **System**, then choose **Routing**, and finally select the **Configuration** tab.
- This page lists all gateways currently configured on this firewall and allows you to add new gateways.
- Gateways are IP addresses of next-hop routers through which traffic is forwarded. Setting gateways correctly is crucial for directing outbound traffic.

Step 3: Add a New Gateway

- Click the **+ Add** button to open the gateway creation form.
- In the **Interface** dropdown, select the interface that connects to the other OPNsense firewall. This might be your LAN or a transit network interface (often labeled em1, em2, or similar).
- In the **Name** field, enter a clear, descriptive name like Gateway-to-OPNsense2 to help identify this gateway later.

- In the **Gateway IP** field, input the IP address assigned to the second OPNsense firewall's interface on this shared network. This IP is where outgoing traffic should be sent to reach the other firewall.
- Optionally, add a description to document the purpose of this gateway, such as "Default gateway for transit network to OPNsense VM 2."
- Double-check that the IP address matches the other firewall's interface exactly—errors here will cause routing failures.
- Click **Save**, and then **Apply Changes** to activate the new gateway configuration.

Step 4: Set the Default Gateway

- On the same **Gateways** page or under **System** > **Gateway** > **Configuration**, look for a setting labeled **Default Gateway** or **Gateway Group**.
- From the dropdown, select the newly created gateway (Gateway-to-OPNsense2).
- This action tells the firewall to send all traffic destined for unknown networks through the other OPNsense firewall.
- Click **Save** and then **Apply Changes** to commit this configuration.

Step 5: Repeat on the Second OPNsense VM

- Open a web browser and access the second OPNsense firewall's LAN IP address (for example, https://192.168.2.1).
- Log in as root
- Repeat Steps 2 through 4, but this time add a gateway pointing back to the first OPNsense firewall's IP address on the shared network.
- Set that gateway as the default gateway for this second firewall.
- This reciprocal configuration ensures that traffic from either firewall destined to unknown networks passes through the other firewall.

Step 6: Verify Gateway Status

- Return to **System** > **Routing** > **Configuration,** on both OPNsense devices.
- Look for the status indicators next to each configured gateway.
- A green or “Online” status confirms that the OPNsense firewall can reach the gateway IP via the assigned interface.
- If the status shows “Offline” or “Down,” troubleshoot network connectivity (check cables, VMware network settings, firewall rules, etc.).

Step 7: Test Basic Connectivity

- Access the console of the first OPNsense firewall via VMware or SSH.
- Use the ping command to send ICMP requests to the IP address of the second OPNsense firewall’s interface (the one you set as the default gateway).
- Example: ping 192.168.2.1
- Expect successful replies indicating that the routing path between the two firewalls is functional.
- Repeat this ping test from the second firewall to the first.
- If ping fails, verify firewall rules, interface assignments, and gateway IP addresses to find misconfigurations.

Task 9: Verifying Connectivity Between OPNsense Firewalls

Step 1: Access the First OPNsense Firewall

Start by logging into the first OPNsense firewall to perform the connectivity test. You can access it in one of several ways:

- **Using the Web GUI:** Open a browser on a device connected to the LAN or management network, and enter the OPNsense firewall’s LAN IP address (e.g., https://192.168.1.1). Log in with your admin credentials.
- **Using SSH:** If SSH access is enabled, use an SSH client like PuTTY or Terminal and connect to the firewall’s IP address.
- **Using Console Access:** If the firewall is running in a VM or physical device, you can use the direct console interface.

Step 2: Identify the IP Address of the Second Firewall

Determine the IP address assigned to the second OPNsense firewall's interface that you want to test connectivity with. Typically, this is the IP address configured on the transit network interface or the LAN segment used for inter-firewall communication. For example, if the first firewall has 192.168.10.1 on its transit interface, the second might have 192.168.10.2.

Make sure you note the exact IP address, as this will be the target for the ping test.

Step 3: Navigate to the Ping Utility

If you are using the web interface:

- Log in and go to **Interface** > **Diagnostics** > **Ping** from the main menu. This tool allows you to send ICMP echo requests (pings) directly from the firewall's interface.

If using command line:

- You will be using the standard Unix-like ping command.

Step 4: Initiate the Ping Command

- **Web GUI:** Enter the target IP address of the second firewall in the ping utility input box. Select the interface from which to send the ping (usually the interface connected to the transit network). Click the **Ping** button to start sending ICMP echo requests.
- **Command Line:** Type the command:

```
ping -c 4 <target-IP-address>
```

The -c 4 option limits the ping to 4 packets, which is enough to verify connectivity.

Example:

cmd

```
ping -c 4 192.168.10.2
```

This sends four ICMP packets to the target IP.

Step 5: Analyze Ping Responses

Watch the responses from the target firewall:

- If you receive replies, the output will look like:

```
64 bytes from 192.168.10.2: icmp_seq=1 ttl=64 time=2.3 ms

64 bytes from 192.168.10.2: icmp_seq=2 ttl=64 time=2.1 ms

64 bytes from 192.168.10.2: icmp_seq=3 ttl=64 time=2.0 ms

64 bytes from 192.168.10.2: icmp_seq=4 ttl=64 time=2.2 ms

--- 192.168.10.2 ping statistics ---

4 packets transmitted, 4 packets received, 0% packet loss

round-trip min/avg/max/stddev = 2.0/2.15/2.3/0.12 ms
```

This confirms the first firewall can reach the second over the network.

- If you do **not** receive replies, the output may show:

```
Request timeout for icmp_seq 1

Request timeout for icmp_seq 2
```

or

cmd

```
Destination Host Unreachable
```

This indicates connectivity problems that must be investigated.

Step 6: Repeat Ping from the Second Firewall

Log in to the second OPNsense firewall, and repeat the ping test targeting the first firewall’s transit interface IP. This confirms bi-directional communication:

cmd

```
ping -c 4 192.168.10.1
```

Successful replies here show the route is correctly established in both directions.

Step 7: Troubleshooting Failed Pings

If the pings fail from either firewall:

- **Check IP Address Configuration:** Ensure both firewalls have correct IP addresses configured on their transit or LAN interfaces, and that they are on the same subnet.
- **Verify Default Gateway:** Confirm the default gateway settings on each firewall point to the other firewall’s IP on the transit network.
- **Check Firewall Rules:** Review the firewall rules on both devices to confirm ICMP traffic is allowed. By default, firewalls may block ping requests.
- **Network Interface Status:** Check if the interfaces are up and not administratively disabled.
- **VLAN or Network Segment:** If VLAN tagging or special LAN segments are used, confirm they are configured correctly and traffic is allowed.
- **Ping from Host Machine:** If the firewalls still cannot ping each other, try pinging each firewall from the host machine or other VMs on the same network to isolate where the problem is.

Step 8: Verify Routing Tables

On both firewalls, verify that routing tables include routes to the other firewall's subnet via the correct interfaces and gateways. In OPNsense, you can view routing info under **System> Routes>Configuration**.

Step 9: Confirm Successful Communication

Once both firewalls successfully ping each other without packet loss, you have validated that:

- The physical and virtual network setup is correct.
- IP addressing and default gateways are properly configured.
- Firewall rules permit the required ICMP traffic.
- Routing between the two OPNsense devices is functioning.

Step 10: Document and Log Results

Note the IP addresses used, test results, any changes made, and observations during troubleshooting. Proper documentation helps maintain clarity and serves as a reference for future network management or audits.

Conclusion:

Verifying connectivity between the two OPNsense firewalls through ICMP ping tests confirmed that the default gateway and routing configurations were correctly implemented. Successful bidirectional communication demonstrated that the firewalls can effectively exchange traffic across the transit network. This validation is crucial to ensure that the network topology is properly established, enabling further firewall configuration and security policies to be applied in the simulated environment.

Week 5: Physical Deployment of OPNsense Firewall

Task 10: OPNsense Installation on Physical Firewall

1. **Prepare the Installation Media**

   Download the latest OPNsense ISO file from the official OPNsense website. This will be the installation media you use to install OPNsense on the physical firewall device using USB Drive.

2. **Connect the USB Drive and Boot**

   Insert the bootable USB drive into the physical firewall hardware. Power on the device and enter the UEFI menu (usually by pressing a key like F2, F12, DEL, or ESC during startup).

   Set the boot priority to boot from the USB drive first. Save the changes and exit. The device should boot from the USB installer.

3. **Start the OPNsense Installation**

   Once booted from the USB, the OPNsense installer menu will appear. Follow the on-screen prompts: select your keyboard layout and proceed to the installation steps.

4. **Install OPNsense on Internal Storage**

   Select the target disk for installation, usually the internal hard drive or SSD of the firewall device. Confirm that you want to erase and format this disk for the OPNsense installation. The installer will copy files and configure the system; this process may take several minutes.

5. **Set the Root Password**

   After the installation files are copied, you will be prompted to set a root password. Choose a strong, secure password to protect administrative access.

6. **Assign Network Interfaces**

   After the installation completes, OPNsense will ask you to assign network interfaces. The physical firewall likely has multiple network ports, including Intel network adapters.

   - Assign the **WAN interface** to the Intel network port that connects to your external network or internet (this corresponds to the bridged adapter in your virtual setup).

  - Assign the **LAN interface** to the port connected to your internal or trusted network.

    This assignment is crucial as it allows OPNsense to properly route traffic between the external and internal networks.

7. **Use Bridged Adapter Concept on Physical Intel Network**

   In your virtual environment, a bridged adapter connects the VM's network card directly to the host's physical network interface (Intel network in this case), allowing it to act like a separate device on the same network. On the physical firewall, the Intel NIC assigned as the WAN interface performs this function naturally by being connected directly to the external network (modem or switch), effectively bridging the firewall to that network. This setup allows OPNsense to receive real external IP addresses or communicate directly with the WAN network.

8. **Finalize Installation and Reboot**

   Once interfaces are assigned and the root password is set, reboot the firewall device. Remove the USB installation media to prevent booting into the installer again. OPNsense will boot into the installed system, ready for further configuration.

Task 11: Configuring and Verifying OPNsense Network Settings

Step 1: Log Into OPNsense Console

- **Boot the Physical Firewall:**

  After installing OPNsense on your physical firewall hardware, power on the device.

- **Access Console:**
  - Connect a monitor and keyboard directly to the physical firewall.
  - Alternatively, if SSH is enabled on OPNsense, you can remotely log in using an SSH client.

- **Login Credentials:**

  Use the username root and enter the password you set during the OPNsense installation. This gives you access to the system's console interface.

Step 2: Assign and Verify Network Interfaces

- **Check Interface Assignments:**
  Within the OPNsense console menu, you will find options related to network interfaces.
  - Confirm that the WAN and LAN interfaces are assigned to the correct physical Intel Network on the firewall. Typically, interfaces will be named like em0, em1, etc.
- **Reassign Interfaces if Needed:**
  If the assignments are incorrect or need adjustment, select the option to assign or reassign interfaces.
  - For example, ensure that the WAN interface corresponds to the port connected to your external network (internet or upstream network).
  - The LAN interface should be connected to your internal network or transit segment.
- **Note Interface Names:**
  Write down the interface names (e.g., em0 for WAN and em1 for LAN) for future reference and configuration.

Step 3: Configure Default Gateways

- **Access the OPNsense Web Interface:**
  From a computer connected to the LAN network of each OPNsense firewall, open a web browser.
  Enter the LAN IP address assigned to the OPNsense device (e.g., https://192.168.1.1).
  Log in using root credentials.
- **Navigate to Routing Settings:**
  In the web interface, go to **System → Routing → Configuration**.
- **Set Default Gateway:**
  On each firewall, configure the default gateway to be the IP address of the other OPNsense firewall's LAN or transit interface.
  - This setting directs all traffic destined for external networks to route through the other device.

    - For example, Firewall A’s default gateway is Firewall B’s LAN IP, and vice versa.
- **Save and Apply Changes:**
  Ensure to save your configurations and apply the settings so the routes become active.

Step 4: Configure IP Addresses on Interfaces

- **Assign Static IPs on LAN Interfaces:**
  To enable direct communication, assign static IP addresses to the LAN interfaces of both firewalls.
    - These IPs should be in the same subnet (e.g., Firewall A LAN: 192.168.10.1/24, Firewall B LAN: 192.168.10.2/24).
    - This allows them to “see” each other directly on the LAN side.
- **Configure WAN Interfaces:**
  Depending on your setup, the WAN interface can be configured to use DHCP (automatically obtain an IP) or a static IP if you have a fixed network configuration.
- **Configure DHCP on LAN as Needed:**
    - Optionally enable DHCP on the LAN if you want connected devices (like Kali Linux VMs) to receive IP addresses automatically.
    - Alternatively, disable DHCP and assign static IPs manually for better control.

Step 5: Test Connectivity Using Ping

- **Use Ping Tool in OPNsense Web GUI:**
  Navigate to **Diagnostics → Ping** on each firewall’s web interface.
    - Ping the IP address of the other firewall’s LAN or transit interface.
- **Use Console Ping:**
  Alternatively, from the OPNsense console command line, run:

  ```
  ping <other-firewall-LAN-IP>
  ```

- **Interpret Results:**
  Successful replies indicate that the default gateway and interface

- configurations are correct and the firewalls can communicate.

  If the ping fails, proceed to troubleshooting.

Step 6: Troubleshoot If Needed

- **Firewall Rules:**
  Check firewall rules on both devices to ensure ICMP (ping) traffic is allowed between the LAN and WAN zones.
    - Temporarily disable restrictive rules or explicitly allow ICMP to test.
- **Physical Connections:**
  Verify that all Ethernet cables are properly connected and that the Intel NIC ports show link status (usually LEDs on the device).
- **Verify IP Settings:**
  Double-check that IP addresses, subnet masks, and gateways are configured correctly and consistently on both firewalls.
- **Review Routing Tables:**
  Ensure the routing tables on both OPNsense devices include the correct default gateways pointing to each other.

Step 7: Document Configuration

- Maintain a record of:
    - IP addresses assigned to each interface.
    - Interface names and their physical ports.
    - Default gateway IPs configured on each firewall.
    - Any firewall rules related to routing or ICMP allowed.
- This documentation will be useful for future troubleshooting and network management.

Week 6: Implementing and Verifying WireGuard VPN for Secure Communication Between OPNsense Firewalls

Task 12: WireGuard VPN Setup Between OPNsense Firewalls

Step 1: Access the WireGuard Configuration on OPNsense

- Log into OPNsense Web GUI.
- Navigate to **VPN > WireGuard**.
- You will find tabs like **Instance**, **Peers**, **Peer Generator**, and **Status**.

Step 2: Create a WireGuard Instance

- **Instance** means your own firewall's WireGuard interface.
- Click **Add** under the **Instance** tab to create your WireGuard interface.
- **Generate private key:** This key stays secret on your firewall and is used to encrypt outgoing traffic.
- **Public key:** Automatically generated from the private key, this will be shared with the remote peer.
- **Tunnel address:** Assign an IP address from a private subnet, e.g., 10.0.0.1/24.
  - This IP is used within the VPN tunnel for routing.
  - The /24 is the subnet mask (255.255.255.0) — it defines the address range in the VPN.
- Save and apply.

Step 3: Configure Peer on Firewall A

- Now you add Firewall B as a **peer** on Firewall A.
- Go to the **Peers** tab on Firewall A, click **Add Peer**.
- **Public key:** Paste Firewall B's public key (from its Local WireGuard instance). This identifies Firewall B.
- **Endpoint:** Enter Firewall B's **physical IP address** and WireGuard port (UDP 51820, or whichever port you chose).
  - Example: 192.168.1.2:51820
- **Allowed IPs:** Define the IP range Firewall A will route through this peer.

    - Put Firewall B’s tunnel IP with a /32 mask for a single IP — e.g., 10.0.0.2/32.
- **Persistent keepalive:** Set to 25 seconds to ensure the connection stays alive, especially if NAT or firewalls exist between the peers.
- Save and apply.

Step 4: Mirror Configuration on Firewall B

- Log into Firewall B and repeat:
- Create the local WireGuard instance:
    - Generate private key, get public key.
    - Assign its own tunnel IP, e.g., 10.0.0.2/24.
- Add Firewall A as a peer:
    - Paste Firewall A’s public key.
    - Enter Firewall A’s physical IP and port as endpoint, e.g., 192.168.1.1:51820.
    - Allowed IPs set to Firewall A’s tunnel IP, e.g., 10.0.0.1/32.
    - Persistent keepalive 25 seconds.
- Save and apply.

Step 5: Enable and Assign WireGuard Interfaces

- Go to **Interfaces > Assignments**.
- Assign the newly created WireGuard interface (wg0) to an OPNsense interface slot.
- Enable the interface, give it a meaningful name like WG-OPNsense.
- Set it as **enabled** and configure any firewall rules on this interface to allow traffic.

Step 6: Configure Firewall Rules for WireGuard Traffic

- Make sure both firewalls allow incoming UDP traffic on the WireGuard port (default: 51820) on their WAN interfaces.
- Add rules allowing WireGuard interface traffic to pass through.
- This is crucial to let encrypted traffic enter and leave the firewalls.

Step 7: Verify Tunnel and Connectivity

- Once configured, the WireGuard tunnel is active.
- Test by pinging the remote WireGuard tunnel IP from one firewall to the other (e.g., from 10.0.0.1 to 10.0.0.2).
- If ping replies are received, encrypted traffic is successfully flowing through the WireGuard VPN.

Physical IPs as Endpoints and Tunnel IPs

- The **physical IPs** are the real-world addresses of the devices (their WAN or LAN IPs).
- These IPs are used as **endpoints** to establish the WireGuard connection over the physical network.
- The **tunnel IPs** are virtual IPs that exist only inside the VPN tunnel and are used for secure, encrypted communication.
- This separation ensures that traffic inside the tunnel is encapsulated and encrypted, while the underlying transport uses the actual physical network.

Task 13: Encrypted Traffic Capture and Verification

Step 1: Access the OPNsense Web Interface

- Open a web browser on a device connected to the OPNsense LAN network.
- Log in to the OPNsense web GUI using your admin/root credentials.

Step 2: Navigate to Packet Capture Tool

- In the OPNsense dashboard, go to **Interfaces** > **Diagnostics** > **Live Packet**.
- This tool allows you to capture and analyze live traffic on specific network interfaces.

Step 3: Select the WireGuard Interface

- From the interface dropdown menu, select the WireGuard tunnel interface you configured earlier (usually named something like wg0).
- This interface handles the encrypted VPN traffic.

Step 4: Configure Capture Settings

- Set a capture length limit if desired (for example, 1000 packets).
- Choose to capture **all traffic** or filter by protocol/port if needed (for

  example, UDP port 51820, WireGuard's default port).

- Enable packet capture and start the process.

Step 5: Generate Network Traffic Over the VPN

- From a client behind one OPNsense firewall, generate traffic intended for the other firewall’s LAN (for example, ping, or a test VoIP call if applicable).
- This ensures packets flow through the WireGuard VPN tunnel.

Step 6: Observe Captured Packets

- Monitor the captured packets in real-time or download the capture file for offline analysis.
- Use the OPNsense interface or export to tools like Wireshark for detailed inspection.

Step 7: Verify Encryption

- Confirm that the payload data in the captured packets appears encrypted that is, the contents should not be readable or recognizable (no plain text IP, no recognizable protocols like SIP or RTP directly visible).
- WireGuard encrypts packets end-to-end, so only the WireGuard headers are visible, and the actual data is encapsulated and encrypted.

Step 8: Repeat on the Second OPNsense Firewall

- Repeat the same packet capture steps on the other OPNsense device’s WireGuard interface to verify encryption on both ends of the VPN tunnel.

Step 9: Troubleshoot if Needed

- If packets appear unencrypted, verify that the WireGuard tunnel is active and properly configured.
- Check firewall rules to ensure packet capture traffic is allowed and that the tunnel interfaces are up.

Conclusion

The successful packet capture and verification process confirmed that the WireGuard VPN tunnel between the two OPNsense firewalls is effectively encrypting all traffic passing through it. This ensures secure and private communication by preventing any unauthorized interception or tampering, thereby safeguarding sensitive data such as telephone call traffic from man-in-the-middle attacks. The implementation validates the robustness of WireGuard as a lightweight, high-performance VPN solution for enhancing network security in this setup.