Network Security Monitoring and Analysis with ELK, pfSense, and Snort

Table of Contents

Section 1: Project Overview and Introduction

1.1 : Project Objectives and Significance

The main goal of this project is to build a basic **Security Information and Event Management (SIEM) system**. A SIEM is like a central control room for cybersecurity. It collects and analyzes security logs from different places to help you detect threats and respond to them quickly.

The project's significance is that it shows you can set up a system to:

-> Ingest Logs: Collect data from various network devices in real time.

-> Analyze and Visualize Data: Use a centralized platform (Kibana) to look at and understand all the security events happening on a network.

-> Detect Threats: Spot suspicious activity like port scans or other attacks using a powerful tool like Snort.

-> Respond to Incidents: When you see an alert, you can take action to fix the problem.

1.2 : Tools and Technologies Used

This project uses three key open-source technologies that work together to form the SIEM system.

ELK Stack: This is a powerful suite of tools for handling log data.

Elasticsearch: This is the database that stores all the logs from your network. It makes the data easy to search and analyze.

Logstash: This tool collects logs from different sources, processes them, and then sends them to Elasticsearch. Think of it as the data pipeline.

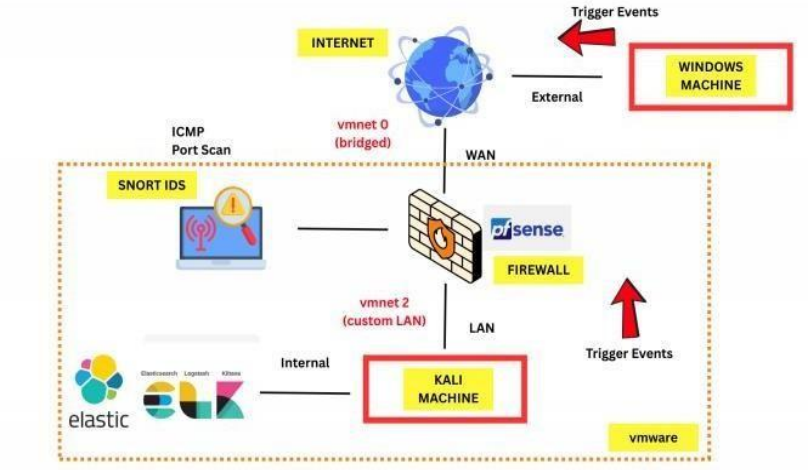
Kibana: This is the web interface where you can view dashboards, create charts, and see all your logs in a clear, easy-to-understand way.

pfSense: This is a powerful firewall and network gateway. It's the front line of your network, filtering traffic and creating logs of everything that happens. It sends these logs to the ELK Stack for analysis.

Snort IDS: Snort is an Intrusion Detection System that works with pfSense. It's specifically designed to watch for suspicious network activity, such as port scans, and generate alerts. These alerts are also sent to the ELK Stack.

Section 2: Network Infrastructure Setup

2.1 : Detailed Network Diagram



The Central Firewall:
At the very center is the pfSense Firewall, which acts as the main gateway for your network. It has two sides: a **WAN** side that connects to the internet and an external machine (the Windows machine), and a **LAN** side that connects to your internal network. This firewall is the gatekeeper that controls all traffic coming in and going out.

The Network and Machines:
The **Kali Machine** is on the **internal** network, safely behind the firewall. Any traffic from this machine that goes to the internet must first pass through pfSense. The diagram shows that both the Windows machine and the Kali machine can be used to **trigger events** to test the network.

The Security Tools:
The diagram also shows two key security tools. **Snort IDS** is a detective that watches all the network traffic for suspicious activity, like an **ICMP Port Scan**. When Snort finds something, it sends the information as a log. The **ELK stack** (Elasticsearch, Logstash, Kibana) is a logging and analysis system that collects all these logs, stores them, and lets you view them in a professional dashboard.

Section 3: ELK Stack Deployment

3.1 : Kali Linux System Preparation

Virtual Box Installation:

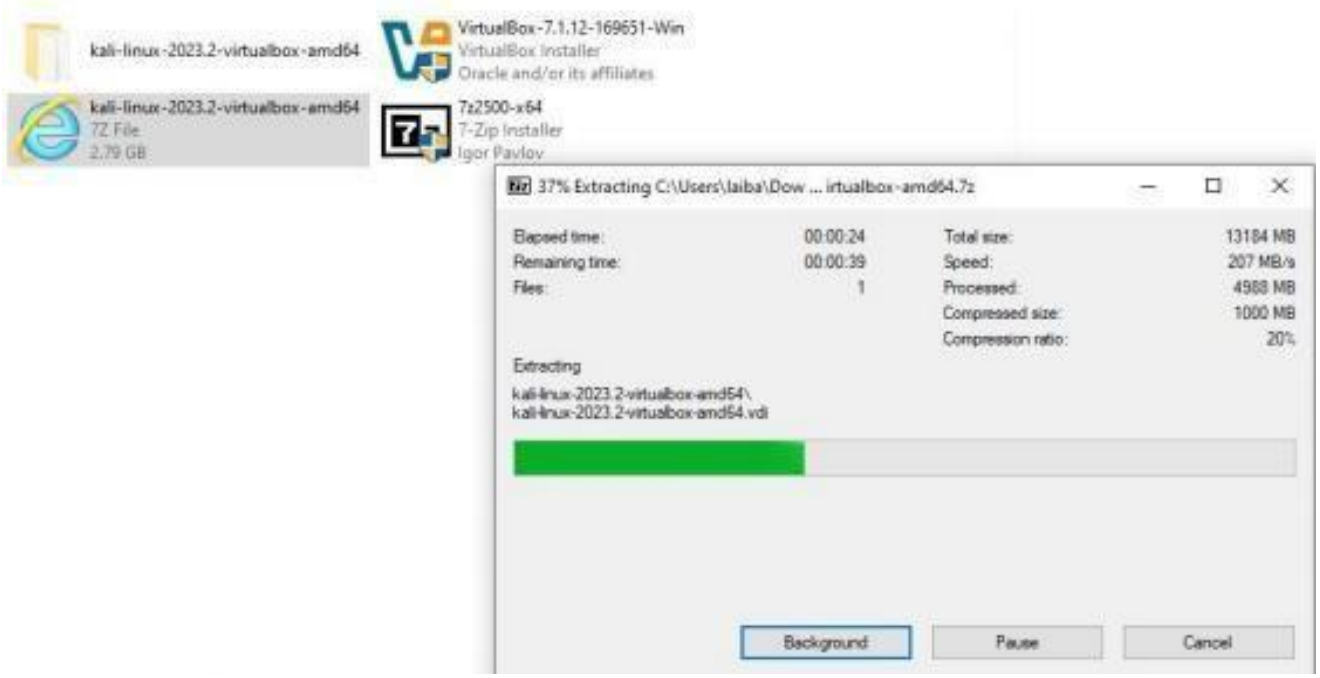
1)- First, download the latest version of Oracle Virtual Box from its official website: https://www.virtualbox.org
2)- Install it using the on-screen instructions. No need to change default settings during setup.

Downloading the Kali Linux Image File:
3)- Pre-configured Kali Linux image in .7z format through Google Drive.
4)- File name: kali-linux-2023.2-virtualbox-amd64.7z
5)- Download it from the provided link.

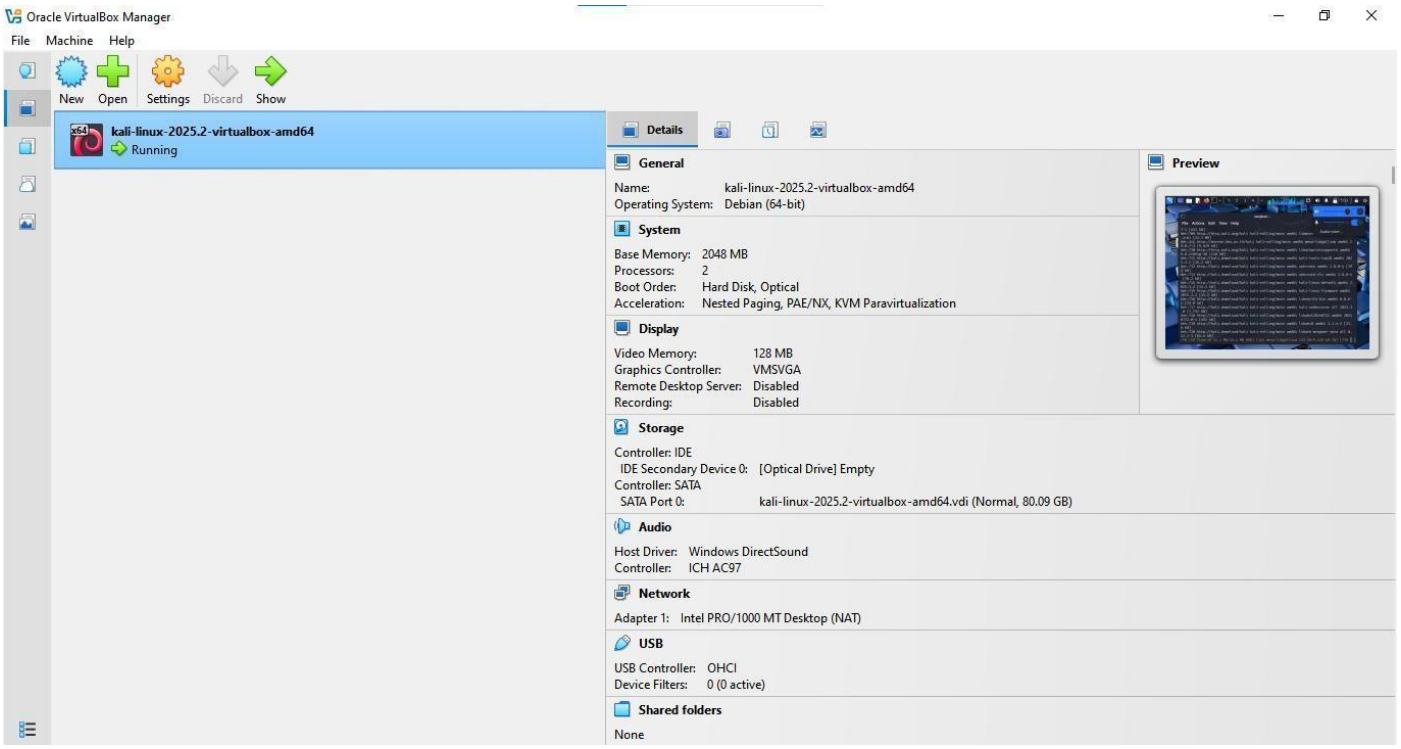
Extracting the Image File:
6)- After download, use **7-Zip** or **WinRAR** to extract the .7z file.
7)- The extracted file will be a **.vbox** file and a **.vdi** file.

Importing the Kali Linux VM into Virtual Box:
8)- Open Virtual Box.
9)- Click on **"Machine" > "Add"**, then navigate to the folder where you extracted the .vbox file.
10)- Select the .vbox file and click **Open**.
11)- The Kali Linux VM will be added to Virtual Box.

Starting the Virtual Machine:
12)- Click on the newly added Kali Linux VM.
13)- Click **Start**.
14)- Kali Linux will boot up without needing manual installation.



Power On kali-linux-2023.2-virtualbox-amd64 on your virtual box. So we can proceed further. My K**ali Linux** virtual machine was already running as shown in the snapshot below.



Finally the setup kali-linux-2023.2-virtualbox-amd64 is completed on the virtual box as shown below in the snapshpot.

After Kali Linux setup, open the **terminal** and run the command shown below as it first checks for the **newest software** on your computer, then installs all the latest security features and bug fixes for you. The **-y** part automatically agrees to install everything without asking.

Command:

sudo apt upgrade && sudo apt upgrade -y

```
┌──(kali㉿kali)-[~]
└─$ sudo apt update && sudo apt upgrade -y
[sudo] password for kali:
Hit:1 http://http.kali.org/kali kali-rolling InRelease
944 packages can be upgraded. Run 'apt list --upgradable' to see them.
The following packages were automatically installed and are no longer required:
  libgdal36          libhdf4-0-alt       libsigsegv2          libvpx9
  libgdata-common    libogdi4.1          libsoup-2.4-1        python3-packaging-whl
  libgdata22         libqt5ct-common1.8  libsoup2.4-common    python3-wheel-whl
  libgeos3.13.1      libsframe1          libtheora0
```

3.2 : Docker and Docker Compose Installation

This command shown below installs **Docker** on the Linux system using the **apt package manager**. Docker is a platform that allows applications to run inside isolated environments called **containers**. It helps in deploying, managing, and running applications more efficiently.

Command:

install docker.io -y

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install docker.io -y
[sudo] password for kali:
The following packages were automatically installed and are no longer required:
  libgdal36        libhdf4-0-alt       libsigsegv2        libvpx9
  libgdata-common  libogdi4.1          libsoup-2.4-1      python3-packaging-whl
  libgdata22       libqt5ct-common1.8  libsoup2.4-common  python3-wheel-whl
  libgeos3.13.1    libsframe1          libtheora0
Use 'sudo apt autoremove' to remove them.

Installing:
  docker.io

Installing dependencies:
  containerd     libcompel1           libproc-processtable-perl  runc
  criu           libintl-perl         libsort-naturally-perl     tini
  docker-buildx  libintl-xs-perl      needrestart
  docker-cli     libmodule-find-perl  python3-pycriu

Suggested packages:
  containernetworking-plugins  btrfs-progs     rinse        zfs-fuse
  docker-doc                   cgroupfs-mount  rootlesskit  | zfsutils-linux
  aufs-tools                   debootstrap     xfsprogs
```

This command puts a new tool on your computer called **Docker Compose**. We need this tool because our project has many different parts that need to work together. **Docker Compose** lets us start all those parts at the same time with just one easy command. If we didn't use this, we would have to start each part separately, which is confusing and takes a lot of time.

Command:

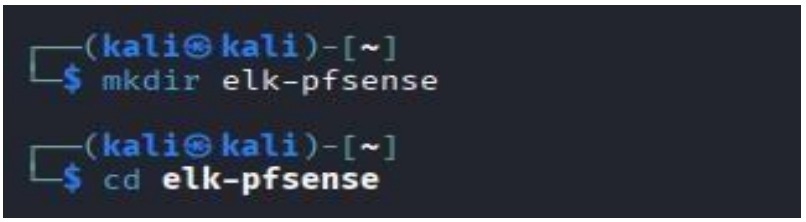
sudo apt install docker-compose -y

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install docker-compose -y
[sudo] password for kali:
docker-compose is already the newest version (2.26.1-4).
The following packages were automatically installed and are no longer required:
  libgdal36        libhdf4-0-alt       libsigsegv2        libvpx9
  libgdata-common  libogdi4.1          libsoup-2.4-1      python3-packaging-whl
  libgdata22       libqt5ct-common1.8  libsoup2.4-common  python3-wheel-whl
  libgeos3.13.1    libsframe1          libtheora0
Use 'sudo apt autoremove' to remove them.

Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 3
```

Moving further, the command shown below was used to create a new directory named **elk-pfsense** in the system.It is used to store project-related files in one organized location.Creating a separate folder makes **management** and navigation easier.
The second command **(cd - change directory)** was used to move inside the elk-pfsense directory.It allows you to work from that folder and run commands in it.
Command:

mkdir elk-pfsense

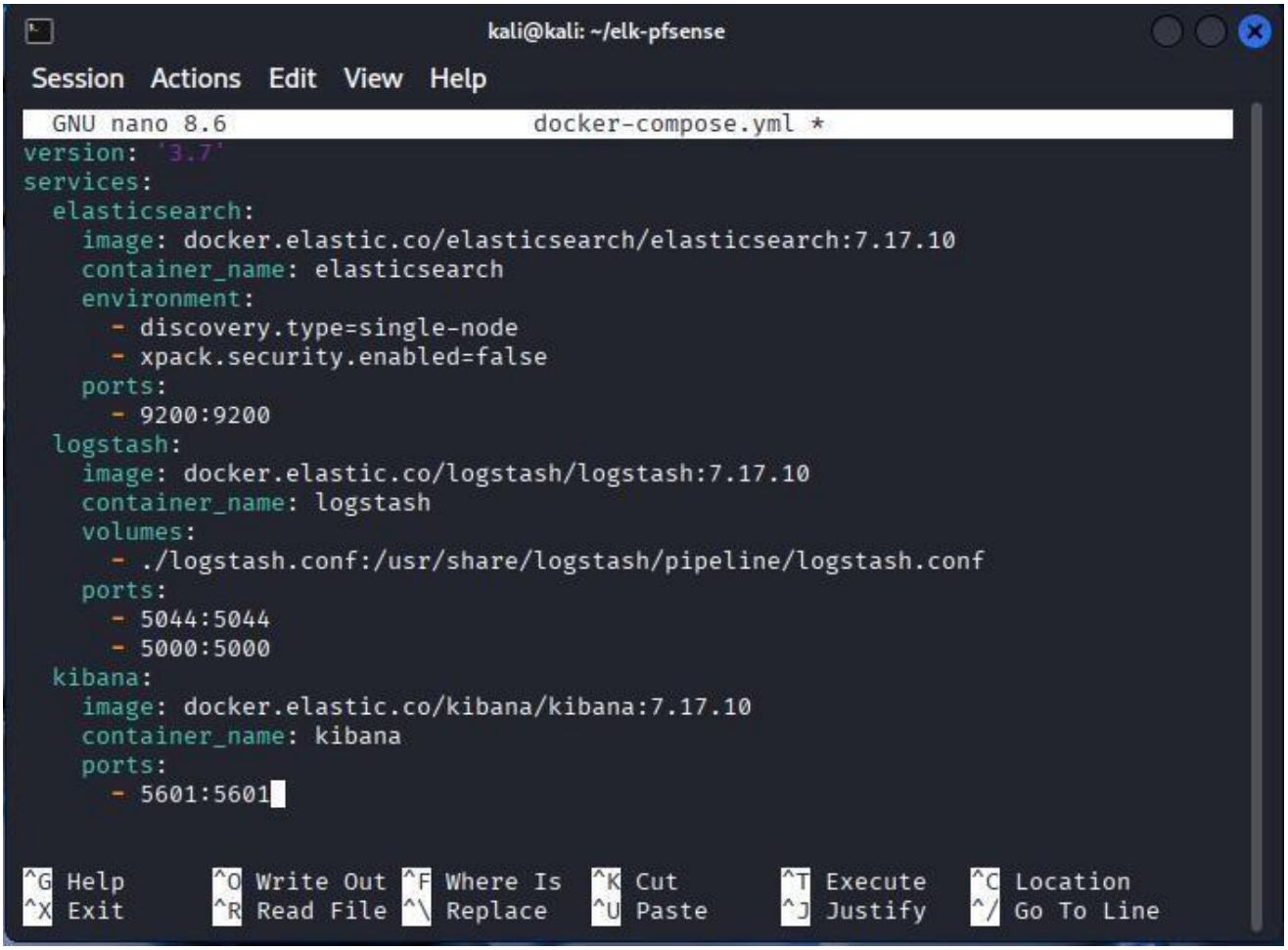


The command you entered, **nano docker-compose.yml**, opens a text editor called **nano** to create or edit a file named **docker-compose.yml**. This file is a **YAML** configuration file that serves as a blueprint for your application, telling Docker Compose which containers to build, how they should connect, and what settings to use. In short, this command allows you to define and manage your entire multi-container Docker application in a single file.
Command:

nano docker-compose.yml

Configuration:

```
version: '3.7'
services:
elasticsearch:
image: docker.elastic.co/elasticsearch/elasticsearch:7.17.10
container_name: elasticsearch
environment:
- discovery.type=single-node
- xpack.security.enabled=false
ports:
- 9200:9200
logstash:
image: docker.elastic.co/logstash/logstash:7.17.10
container_name: logstash
volumes:
- ./logstash.conf:/usr/share/logstash/pipeline/logstash.conf
ports:
- 5044:5044
- 5000:5000
kibana:
image: docker.elastic.co/kibana/kibana:7.17.10n
container_name: kibanaDeployment of SIEM Infrastructure (ELK + Pfsense + Snort)
ports:
- 5601:5601
```



3.3 : Logstash Configuration

Then create another logstash configuration by using the command shown below. This configuration tells Logstash to receive data from a TCP **input** on port 5000. It then uses a **grok filter** to parse and structure the log messages. Finally, the output sends the processed data to the **Elasticsearch** server.

Command:

nano logstash.conf

Point to be Noted:

· Ctrl + O → **Write Out (Save)**

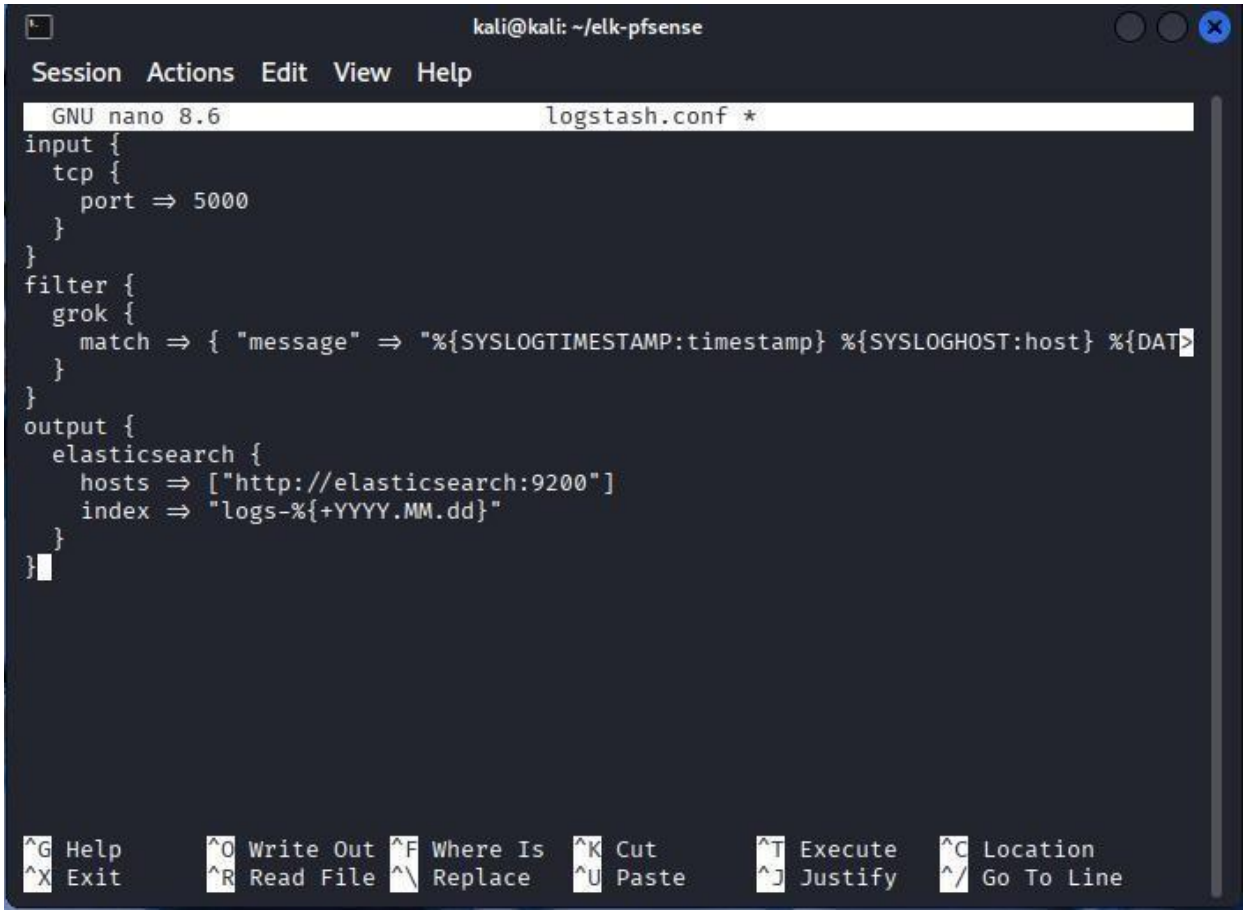
This saves the file in **nano** editor. When you press it, nano asks for the "File Name to Write". If you just press Enter, it saves with the same name.

· Ctrl + X → **Exit**

This closes the **nano** editor. If you already saved the file, it exits directly.

Configuration:

```
input {
tcp {
port => 5000
}
}
filter {
grok {Deployment of SIEM Infrastructure (ELK + Pfsense + Snort)
match => { "message" =>
"%{SYSLOGTIMESTAMP:timestamp} %{SYSLOGHOST:host}
%{DATA:program}(?:\[%{POSINT:pid}\])?: %{GREEDYDATA:message}" }
}
}
output {
elasticsearch {
hosts => ["http://elasticsearch:9200"]
index => "logs-%{+YYYY.MM.dd}"
}
}
```

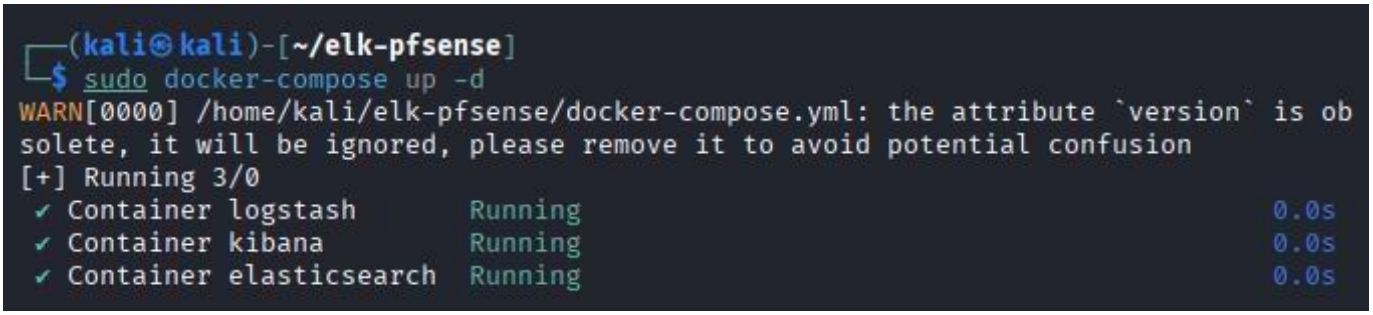


3.4 Deployment of ELK stack Containers

The command used shows that sudo docker-compose up -d successfully started all three containers: **Logstash, Kibana,** and **Elasticsearch**. A minor warning about the version attribute can be safely ignored because all three containers are confirmed as **Running**. This means your ELK stack is now **active** and ready to use.

Command:

sudo docker-compose up -d



The command **sudo fuser -k 9200/tcp** is a troubleshooting step. It is used to find and stop any other processes that are already using port **9200** on your machine. This command is necessary to avoid port conflicts and ensure that your Elasticsearch container can start properly.

Command:

sudo fuser -k 9200/tcp

The **sudo docker ps** command confirms that all three containers are actively running in the background. The output shows each container's ID and name, along with its status as **"Up"** and the correct port mappings. This proves that all components of the ELK stack are working as intended.

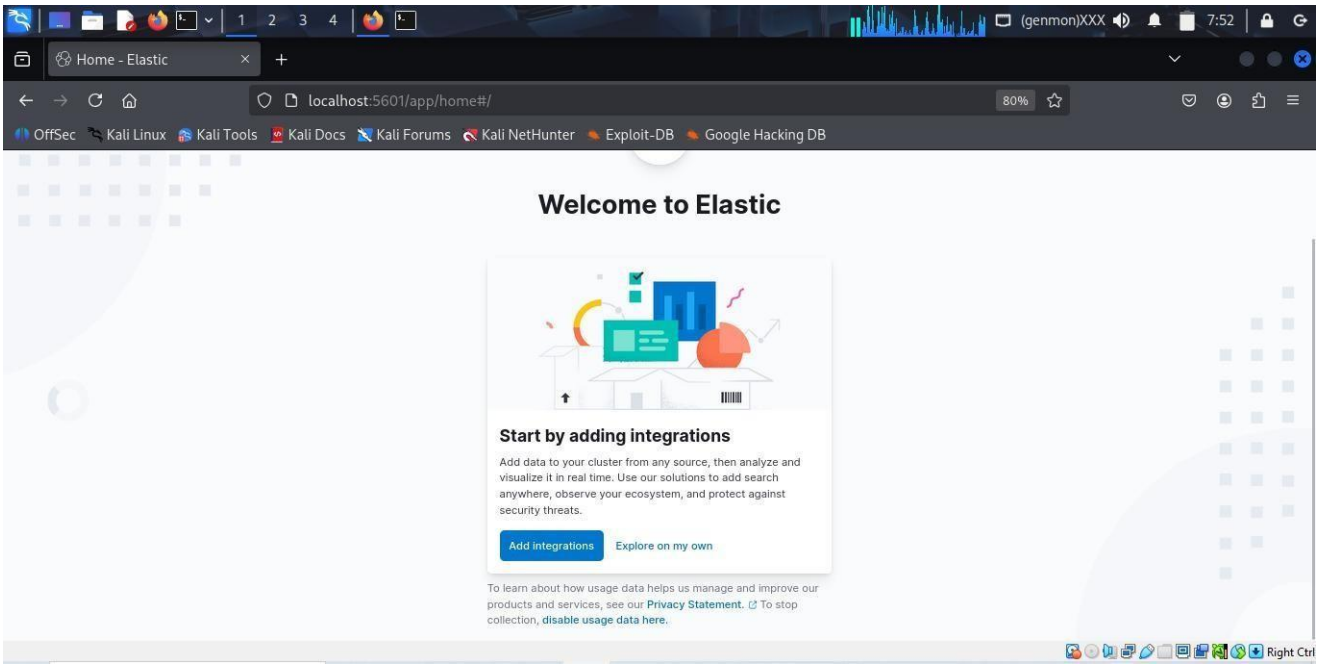
Command:

sudo docker ps

```
┌──(kali㉿kali)-[~/elk-pfsense]
└─$ sudo docker ps
CONTAINER ID   IMAGE                                                  COMMAND
            CREATED       STATUS          PORTS
                                                  NAMES
952bd2264f77   docker.elastic.co/elasticsearch/elasticsearch:7.17.10   "/bin/tini -
- /usr/l…"    5 hours ago   Up 15 minutes   0.0.0.0:9200→9200/tcp, :::9200→9200/tc
p, 9300/tcp                                       elasticsearch
b17bdd9e949a   docker.elastic.co/logstash/logstash:7.17.10            "/usr/local/
bin/dock…"    5 hours ago   Up 15 minutes   0.0.0.0:5000→5000/tcp, :::5000→5000/tc
p, 0.0.0.0:5044→5044/tcp, :::5044→5044/tcp, 9600/tcp   logstash
f8aa7aafb76d   docker.elastic.co/kibana/kibana:7.17.10                "/bin/tini -
- /usr/l…"    5 hours ago   Up 15 minutes   0.0.0.0:5601→5601/tcp, :::5601→5601/tc
p                                                 kibana
```

This screenshot shows the **Kibana web interface**, which you successfully accessed via your web browser at **local host:5601**. This proves that the Kibana container is running correctly and is accessible as shown below.

Link: http://localhost:5601

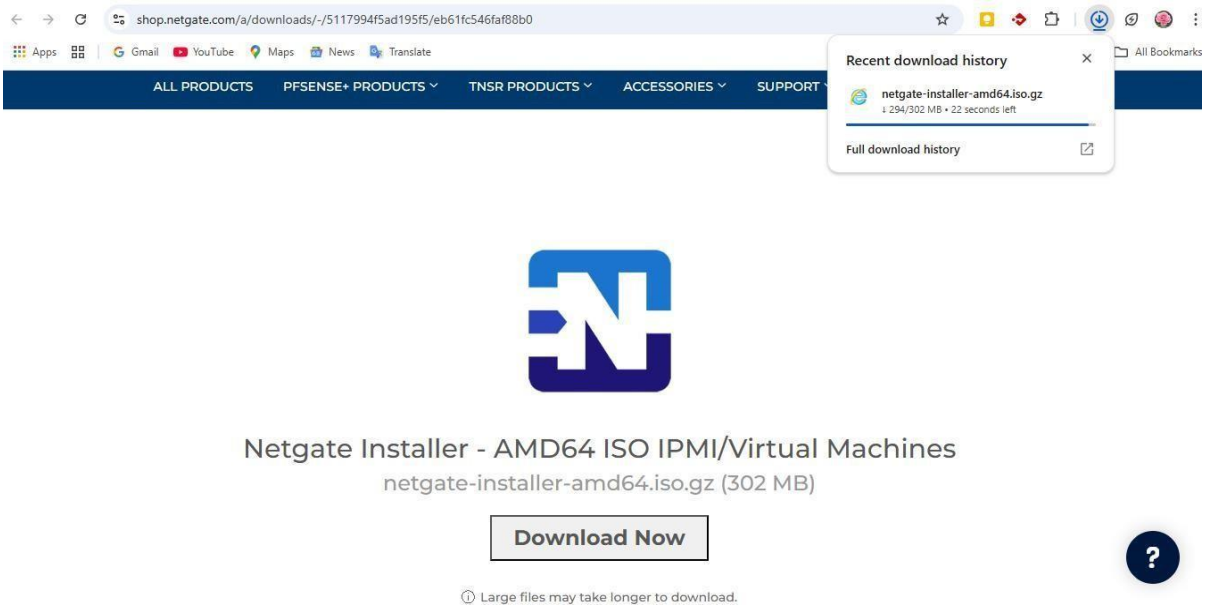


Section 4: pfSense Firewall Deployment

4.1 : pfSense Installation and Configuration

The **Net-gate Installer** is the main software needed to install the pfSense firewall on your virtual machine. It is being downloaded so you can boot your virtual machine from this file and begin the **installation process**. After the download is complete, you will need to decompress the file to get the **.iso** and then load it into your virtual machine's settings.

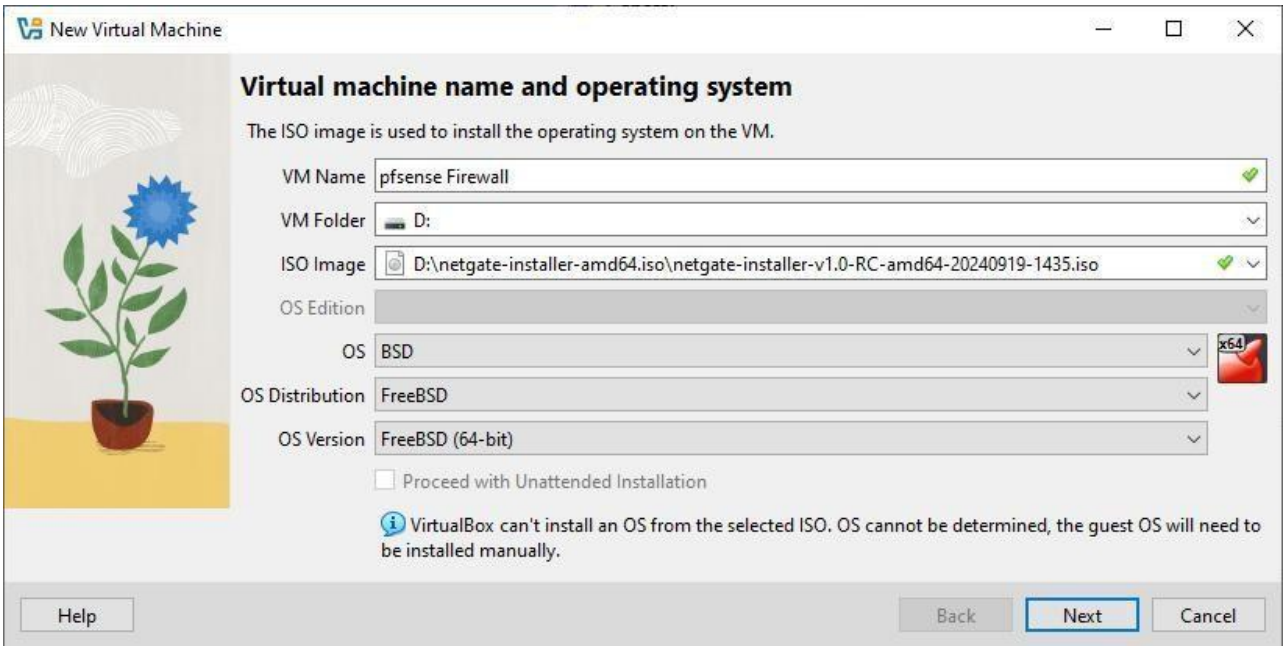
Link:https://www.pfsense.org/download/



After the installation is complete, we will go back to our virtual box and create a **new** virtual machine named as **pfsense Firewall** by clicking the new option from the above menu of the virtual box as shown in the snapshot.

The screenshot shows that you are beginning the process of creating a new virtual machine for pfSense. You have to correctly name the VM **"pfSense Firewall"** and loaded the **Netgate Installer .iso file** that you previously downloaded. It is important to select **FreeBSD (64-bit)** as the operating system, as pfSense is built on the FreeBSD platform.

BSD, or **Berkeley Software Distribution**, is a type of computer operating system that is free and open to everyone. It is known for being very stable and secure, which makes it a good choice for important network machines. Because of this, it is used as the base for many powerful tools like your **pfSense** firewall. Even well-known operating systems like Apple's macOS are built on BSD.

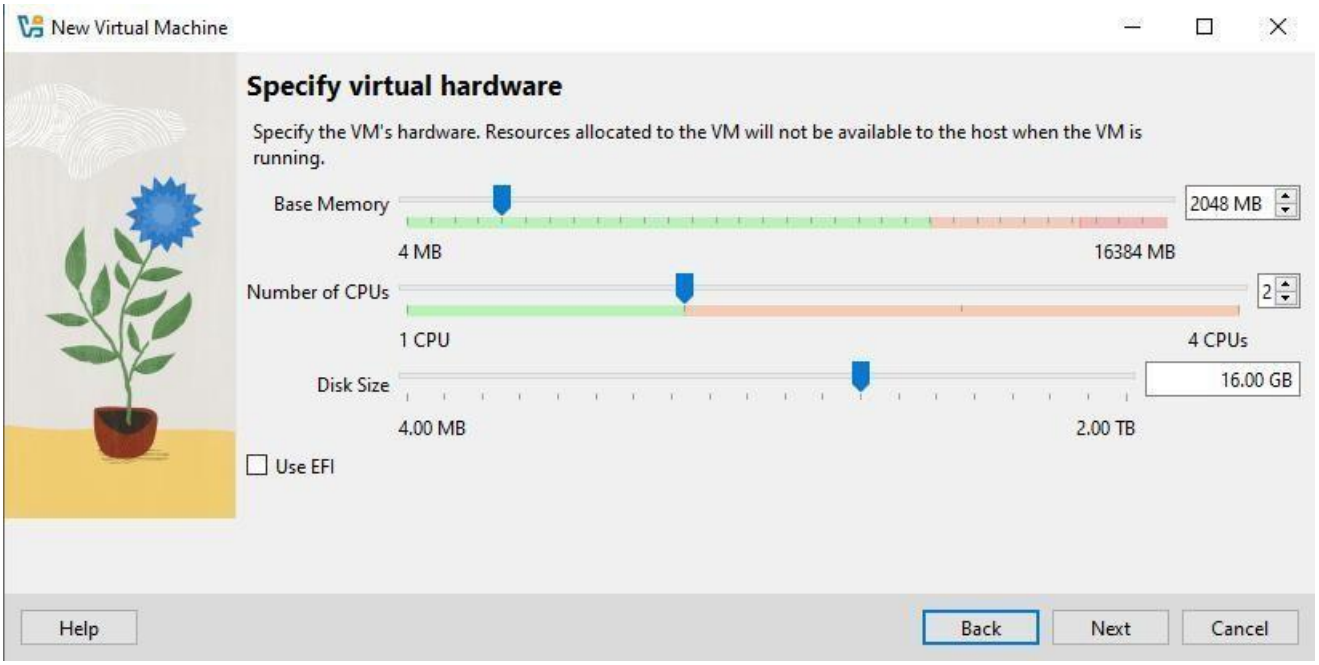


Here in the screenshot shown below shows the **allocation** of virtual hardware resources for our pfsense firewall virtual machine so it can run smoothly. After allocation click on the next button to proceed further with the working.

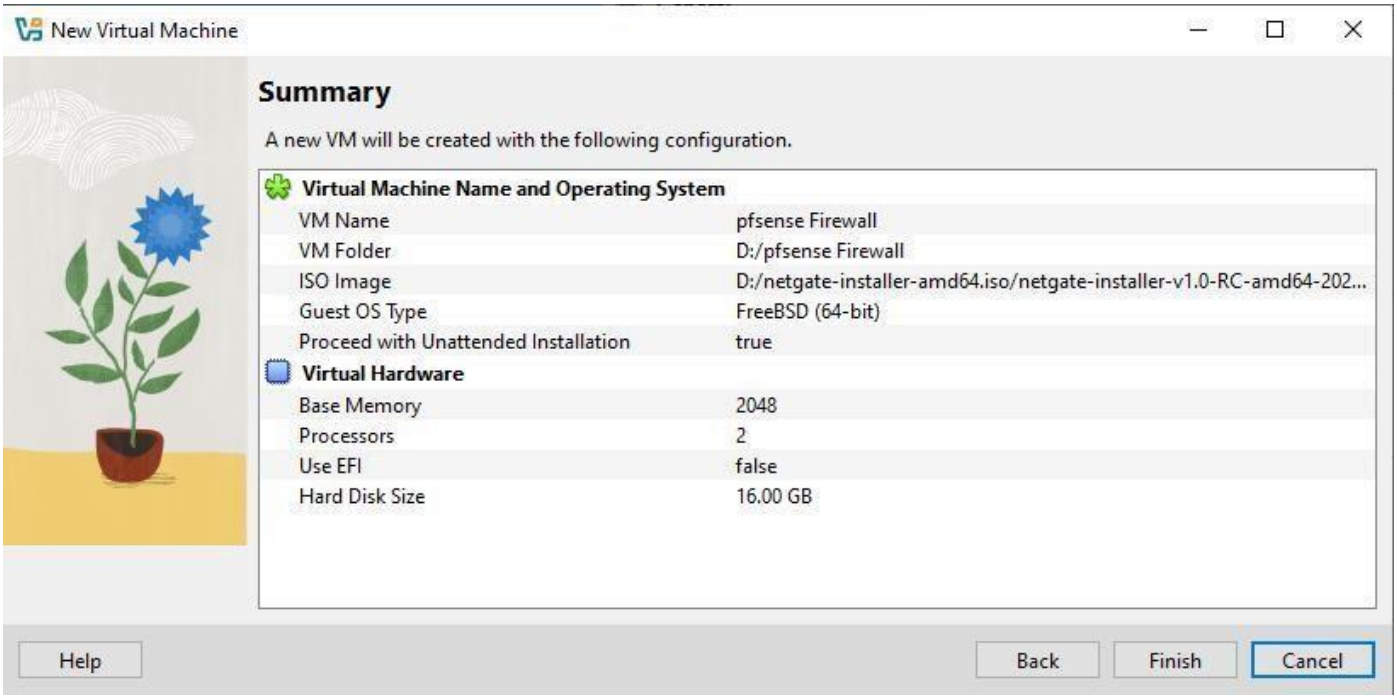
Base Memory: 2048 MG (2 GB)

Number of CPU(s): 2

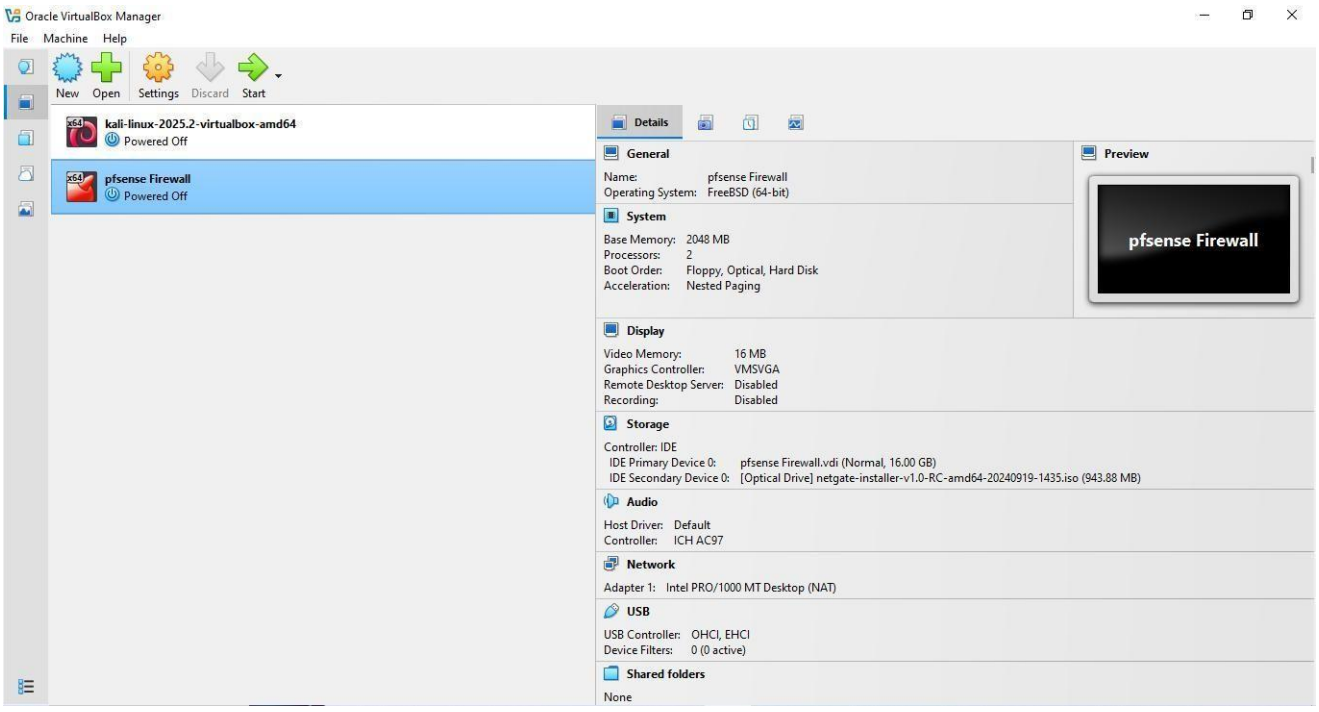
Disk Size: 16 GB



This screenshot shows the **Summary** of all the virtual hardware and operating system settings you have selected for your new virtual machine. This screen serves as a final review to confirm that all your choices are correct, including the **2048 MB of RAM**, **2 CPUs**, and **16 GB of hard disk space**. Once you click **"Finish,"** the virtual machine will be created with these exact settings, and you will be ready to begin the pfSense installation.



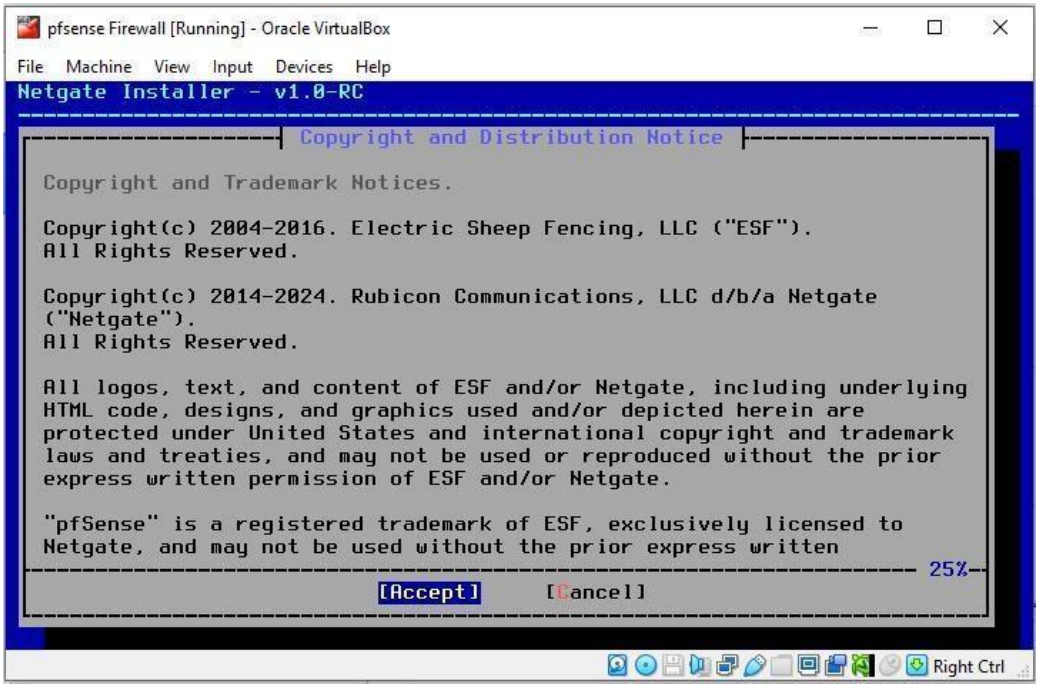
After finishing the above process, now you can see that pfsense firewall virtual machine is finally created. Now click to **start** it so we can began with our pfsense firewall **configuration**.



4.2 : WAN and LAN Interface Configuration

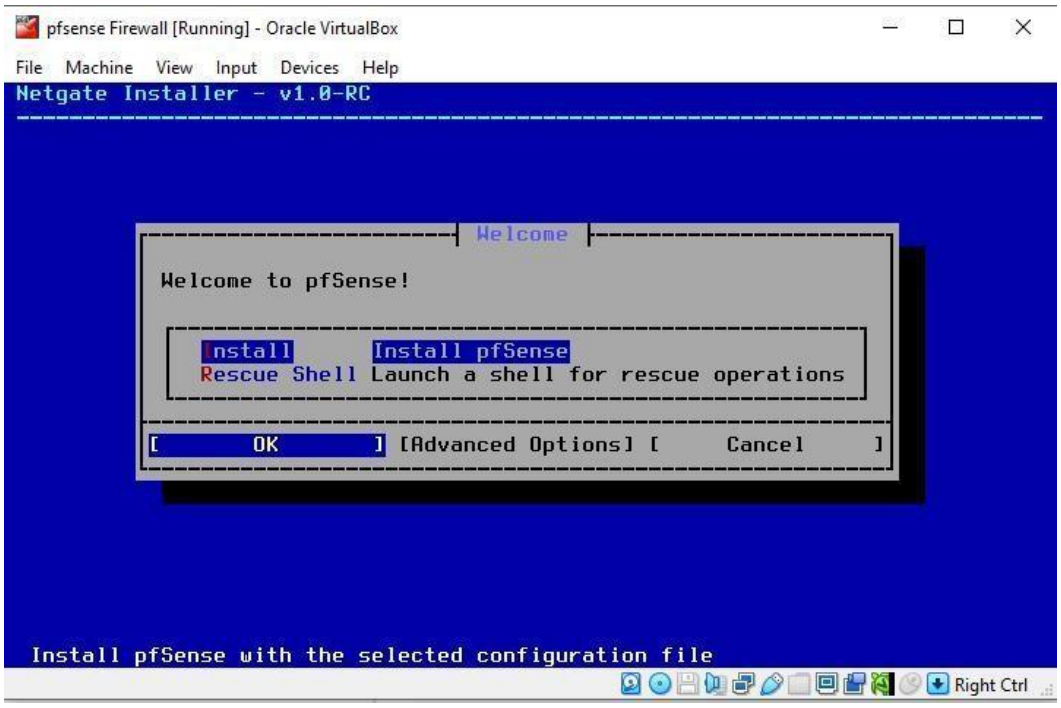
Copyright and Distribution Notice:

This screen shows the copyright and trademark information for pfSense. We selected Accept to agree to the terms and continue with the installation. If we had chosen not to **accept**, the installation would have stopped right there.

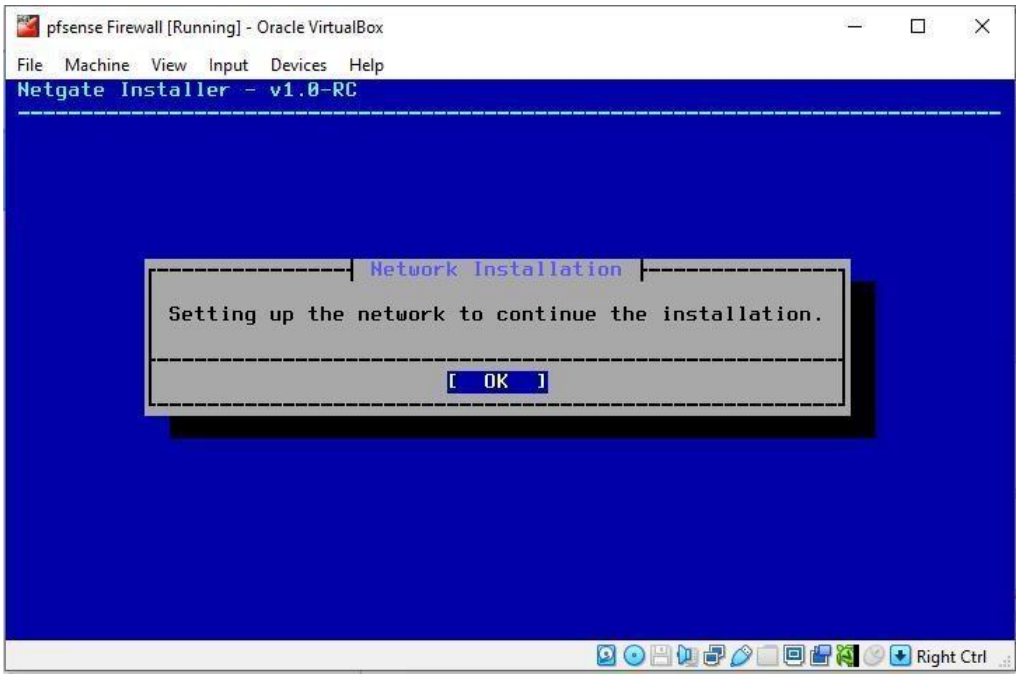


Welcome to pfSense:

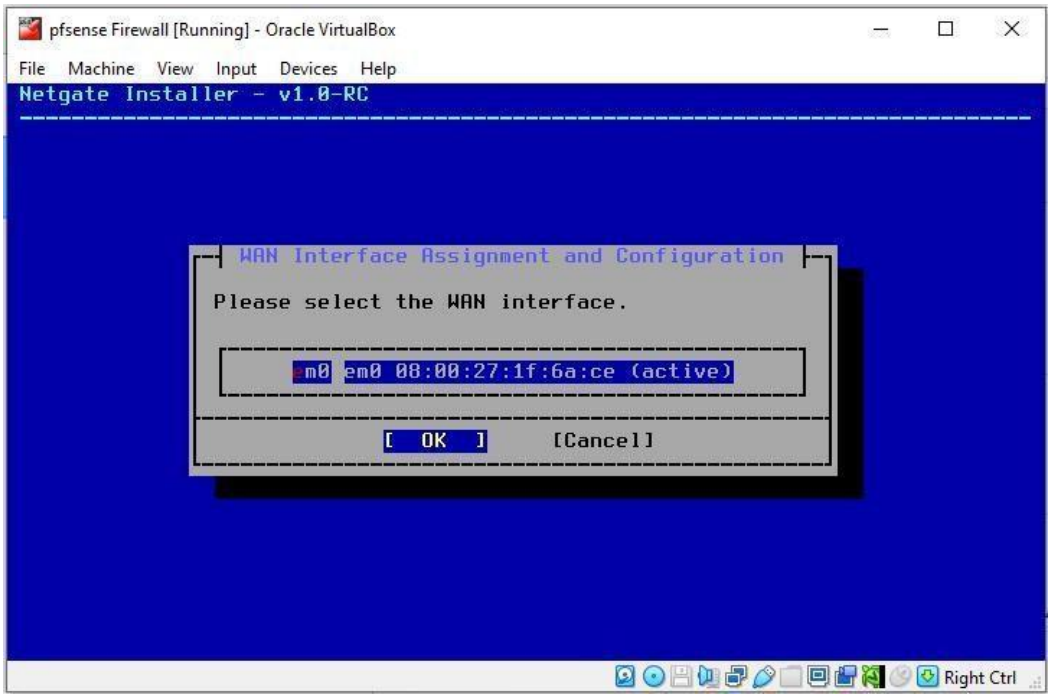
On this screen, we chose Install to begin the installation process. The other option, Rescue Shell, is used for advanced tasks like troubleshooting or fixing a broken installation. We didn't need that, so we chose the main installation option to proceed.



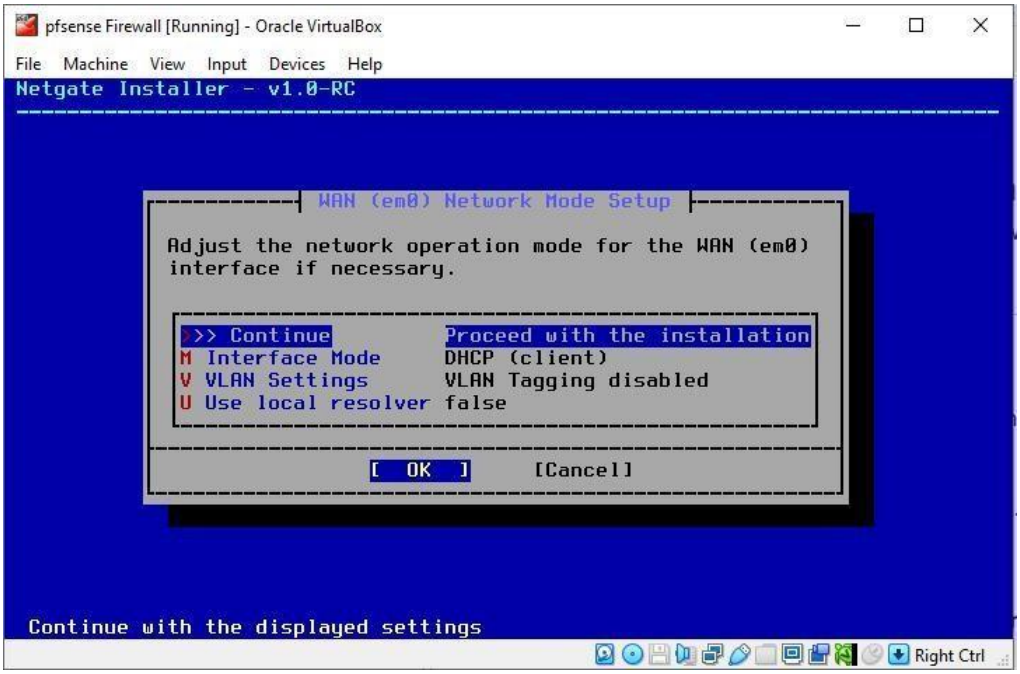
Network Installation:
This is a simple confirmation screen. We just selected **OK** to move on to the next step. This screen is simply telling us that the installer is preparing the network settings before moving forward.



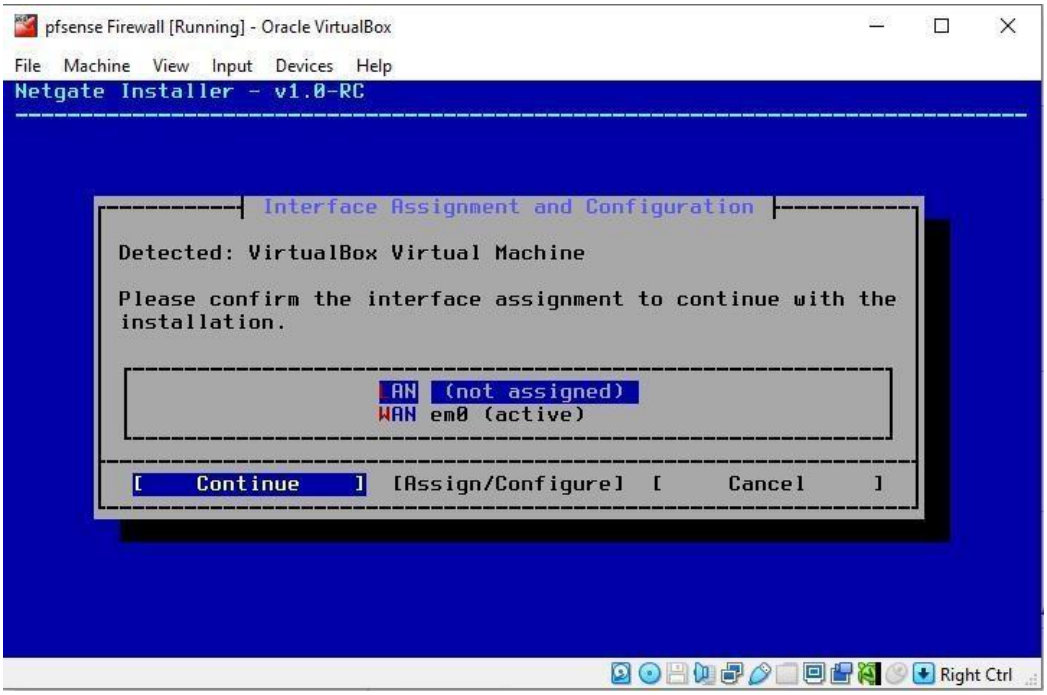
WAN Interface Assignment:
This step is where we assign our **WAN (Wide Area Network)** interface, which is the network adapter that connects our firewall to the internet. We chose the em0 interface because it was the only active option that was configured as a **Bridged Adapter** in Virtual Box. If you don't assign the correct interface here, your firewall won't be able to connect to the internet.



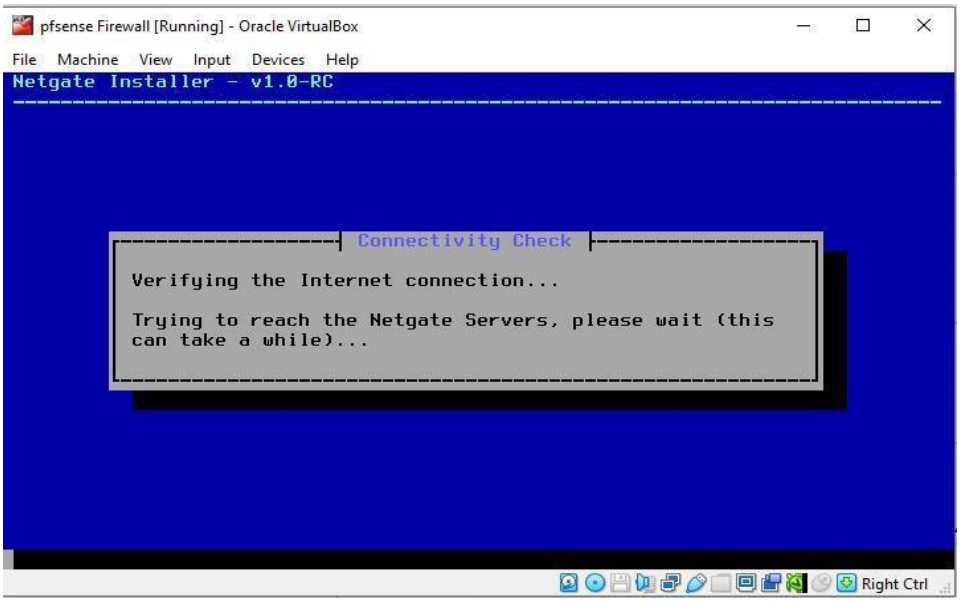
WAN Network Mode Setup:
On this screen, we chose Continue to proceed with the **default** settings. The DHCP (client) mode is the right choice because it tells the firewall to automatically get an IP address from our home network. If we had chosen a different mode, like a static IP, we would have had to manually enter an IP address, which is not what our setup required.



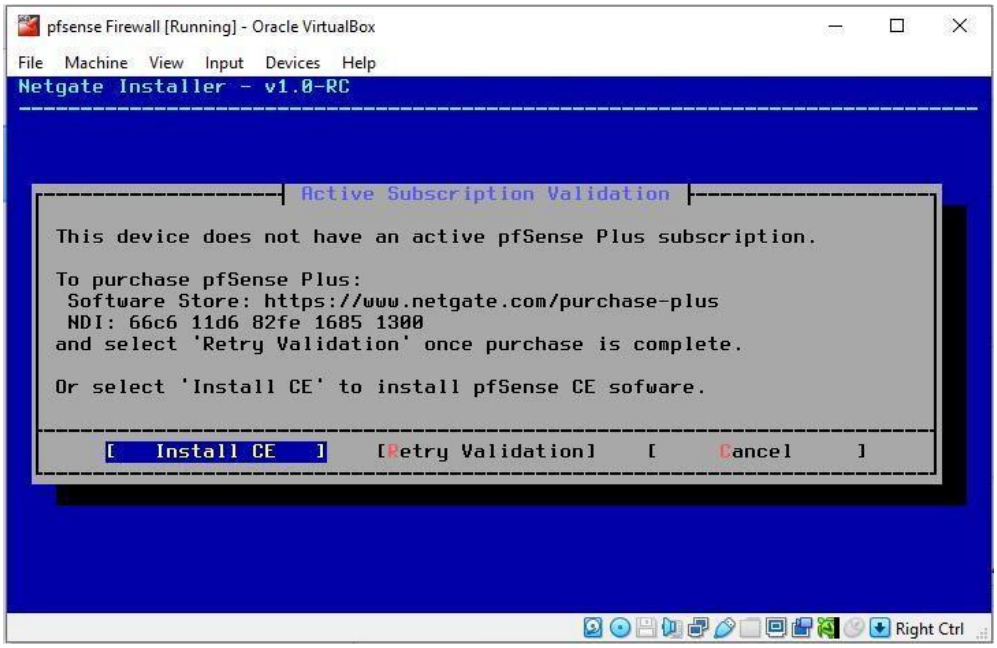
This screen is confirming the network interfaces that the installer detected for our virtual machine. It shows that the **WAN** interface (em0) is active, while the **LAN** interface is still **unassigned**. We chose **Continue** to move on to the next step, where we would manually assign the LAN interface ourselves.



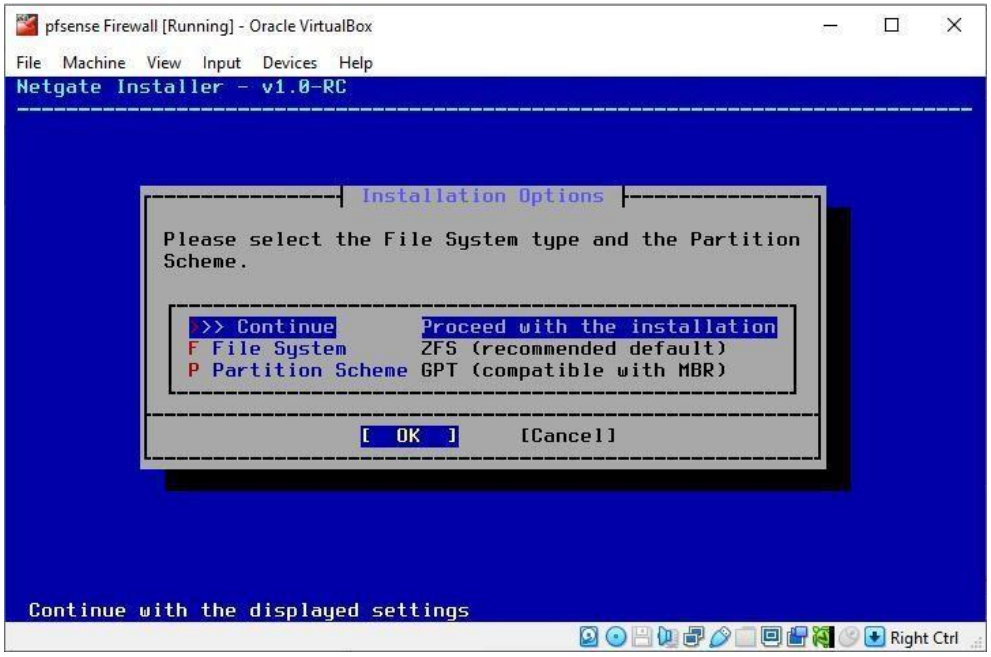
Connectivity Check:
This screen shows the installer verifying the **Internet connection**. The system is trying to reach the **Netgate servers** to ensure it can download any necessary files for the installation. If the connectivity check had failed, it would have meant there was a problem with our network settings, like a firewall blocking the connection or an incorrect adapter configuration.



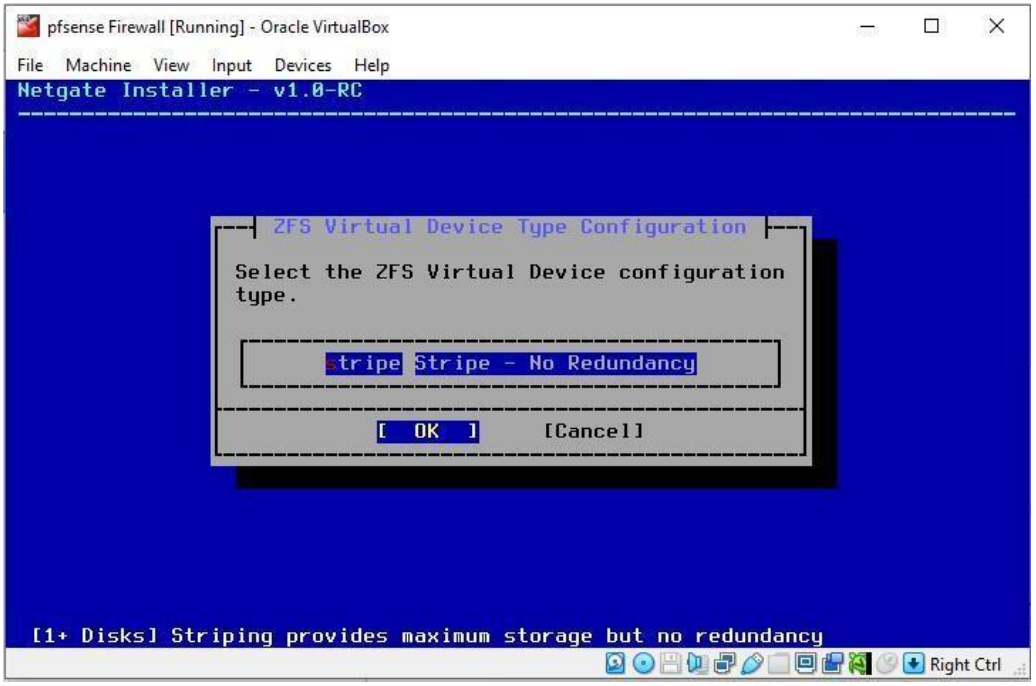
Active Subscription Validation:
This is a crucial screen where the installer checks for a paid pfSense Plus subscription. Since we don't have one, we chose **Install CE** to proceed with the free Community Edition of pfSense. If we had tried to continue with pfSense Plus, the installation would have failed. This option allows us to use the **free version**, which has all the features we need.



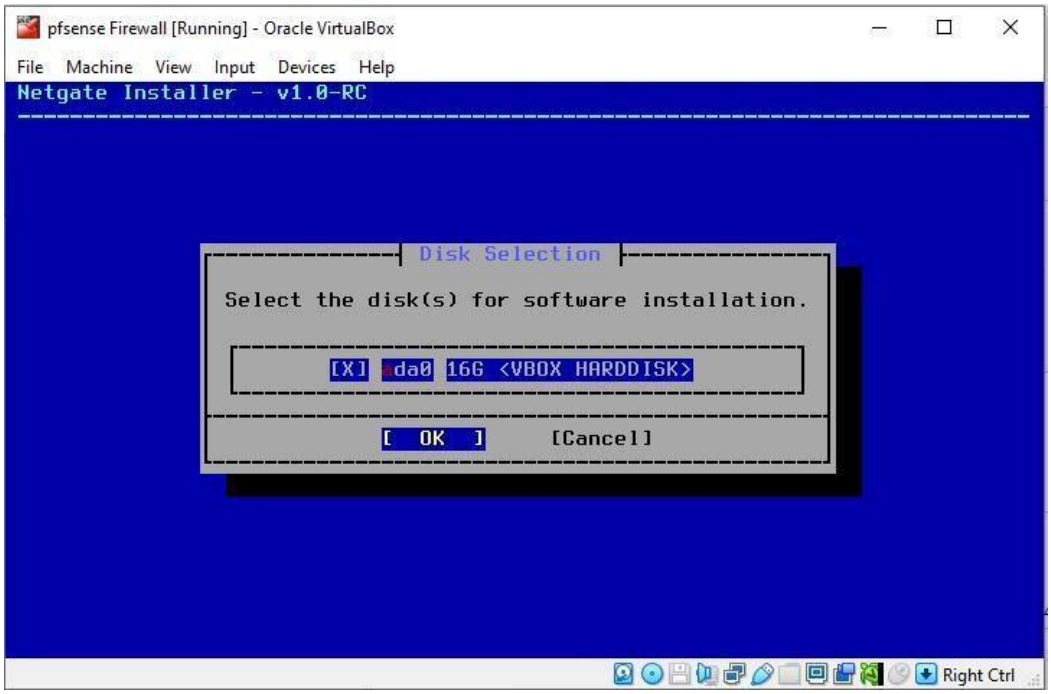
Installation Options:
The **Continue** option was selected to proceed with the installation using the recommended settings. The new things here are the **ZFS File System** and the **GPT Partition Scheme**, which are modern, stable options for managing disk space. If you had chosen a different file system, the system might not be as reliable or have the same advanced features.



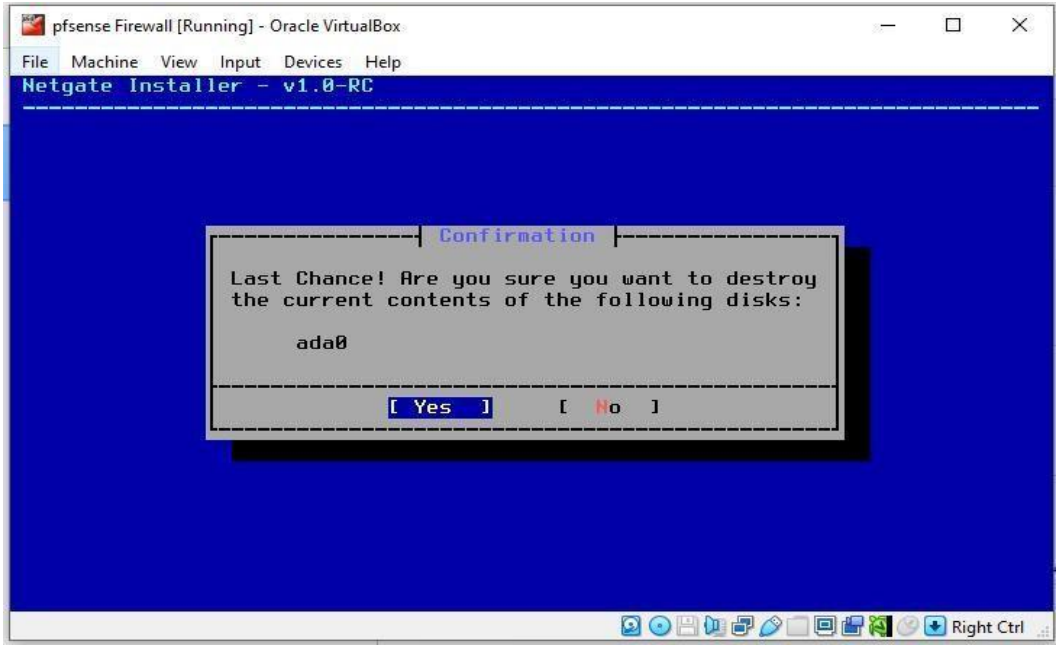
ZFS Virtual Device Type:
The **Stripe - No Redundancy** option was chosen to get the maximum storage space possible from the disks. A new thing here is **Striping**, which is a method that combines multiple disks to create one large volume, but it does not protect your data from disk failure. If you chose an option with **redundancy**, like a mirror, you would have a backup of your data but less total storage space.



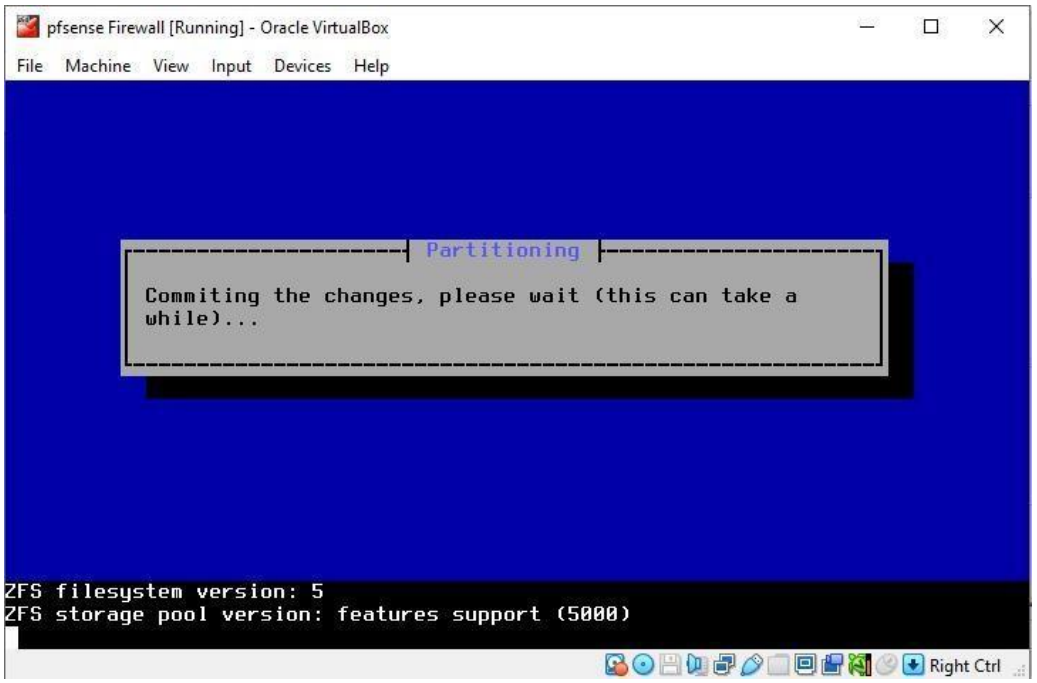
Disk Selection:
The **da0** disk was chosen because it's the only available hard drive for the installation. The new thing to note is that **da0** is a specific name given to the first disk in this type of system, and "VBOX HARDDISK" means it's a virtual disk in a program called VirtualBox. If you didn't choose this disk, the installation could not continue because there is no other disk available.



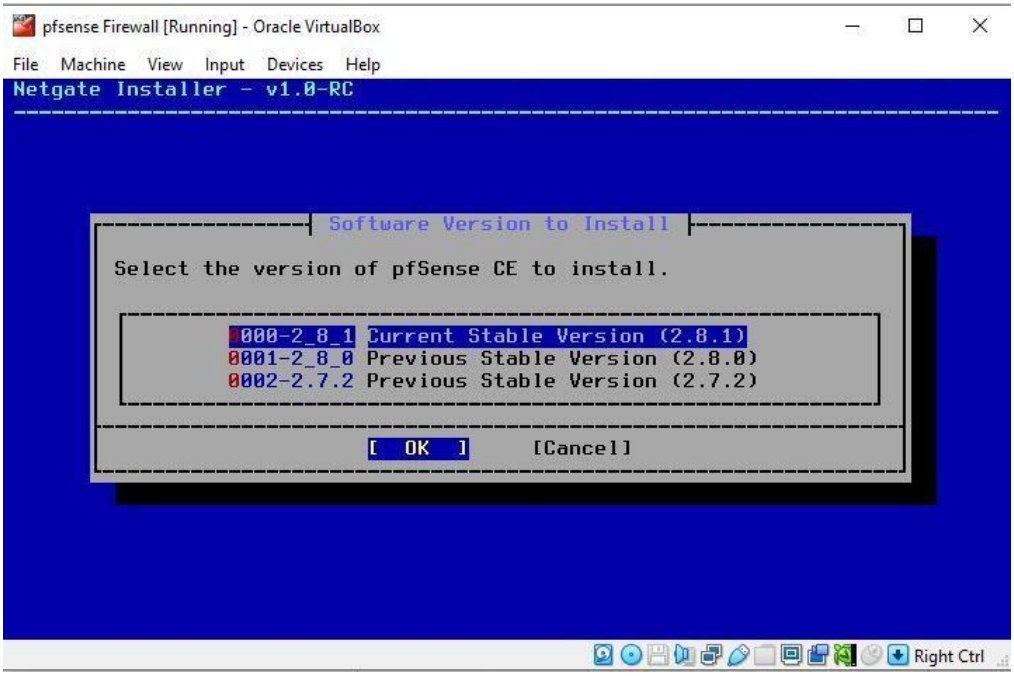
Confirmation:
The **Yes** option was chosen to confirm that you want to completely erase the selected disk and install the new software. This screen is a final warning that all data on the **da0** disk will be destroyed permanently. If you had chosen **No**, the installation would have been canceled, and your data would have remained on the disk.



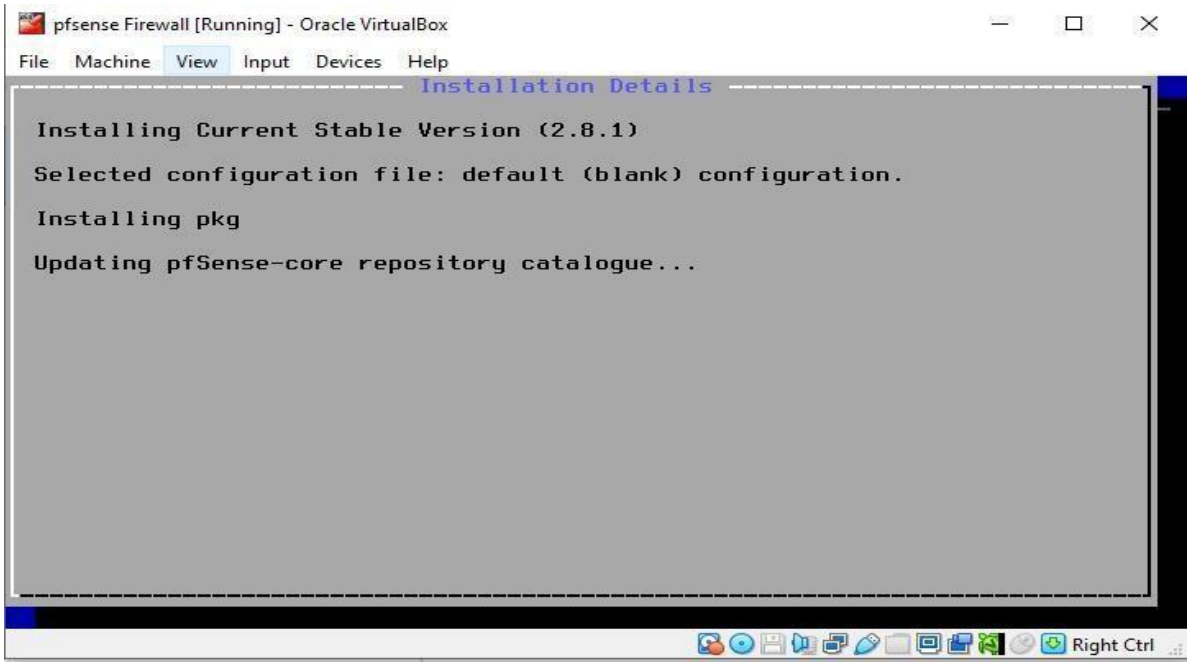
Partitioning:
This screen is not a choice but an automatic process where the system is **committing the changes** to the disk. This means it is writing all the previous settings, like the file system and partition layout, to the hard drive. You must wait for this process to finish to continue. There is no other option here.

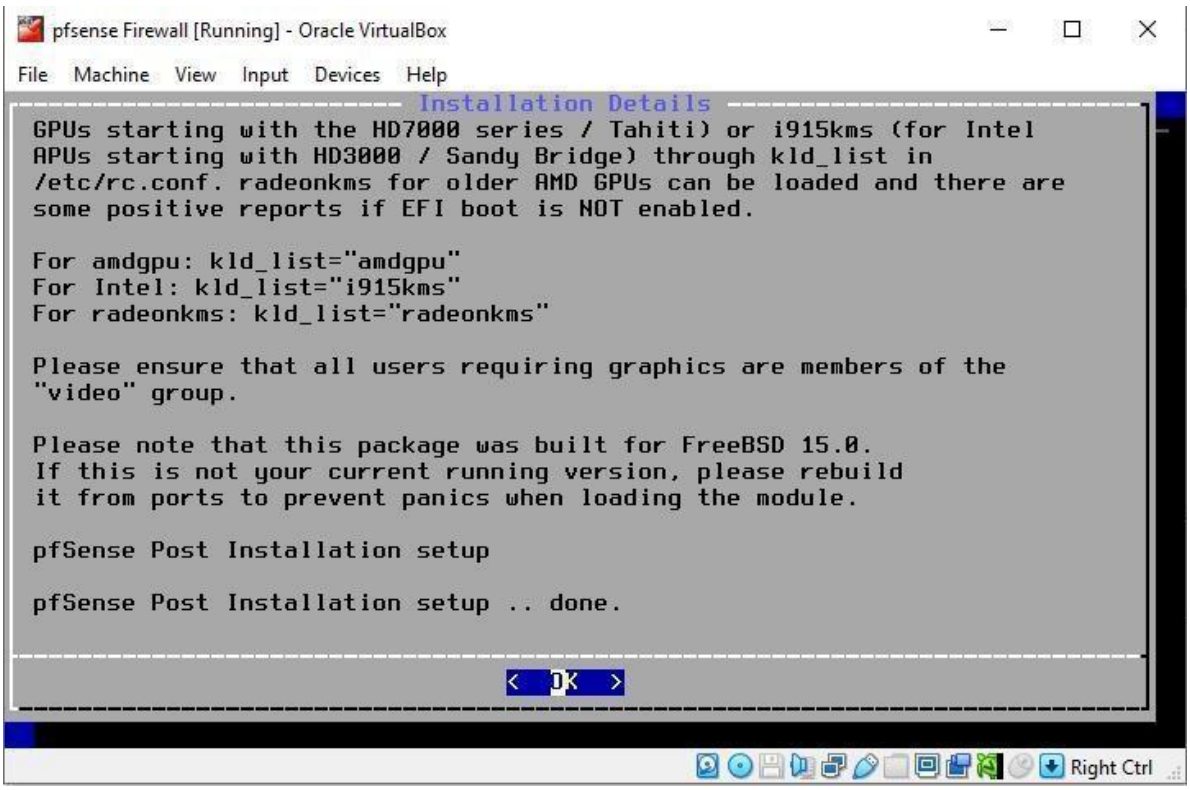


Software Version to Install:
The **Current Stable Version (2.8.1)** was selected because it's the most up-to-date and reliable version, containing the latest features and security updates. A **stable version** means it has been fully tested and is ready for use. If you had chosen a previous version, you would miss out on these important updates and bug fixes.



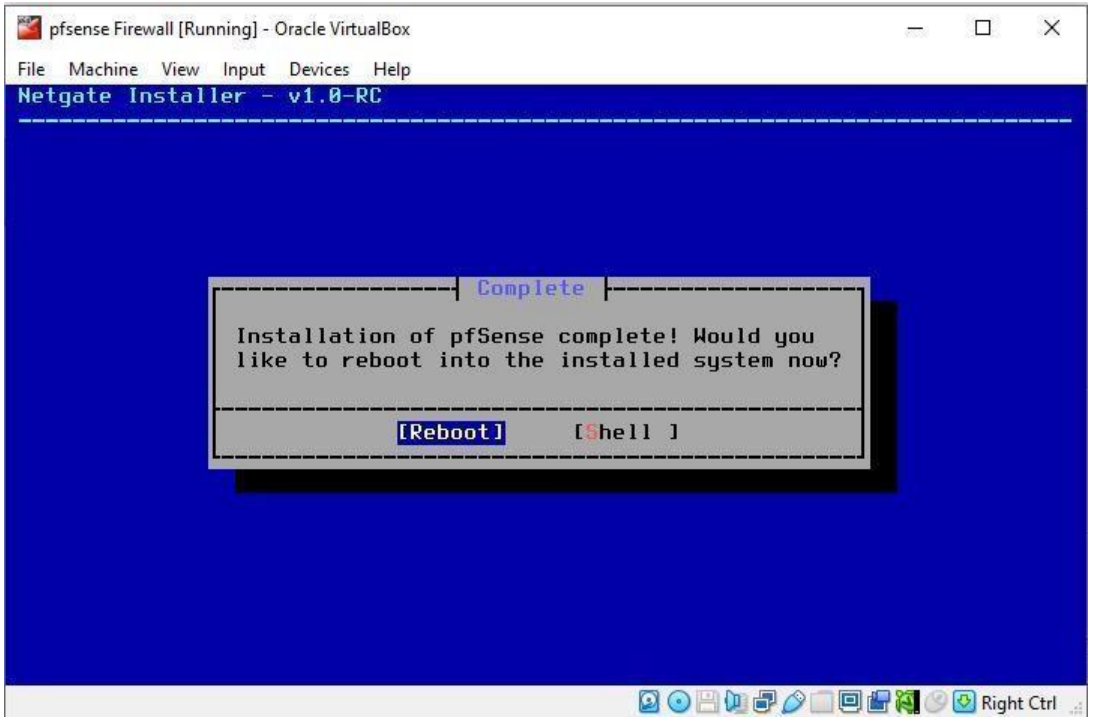
Installation Details:
This screen shows that the installation is already in progress, based on your previous choices. The system is automatically installing the software. A new thing to understand is that **pkg** is a tool that handles the download and installation of different software parts. You must wait for this process to complete; if you stop it, the installation will be corrupted.

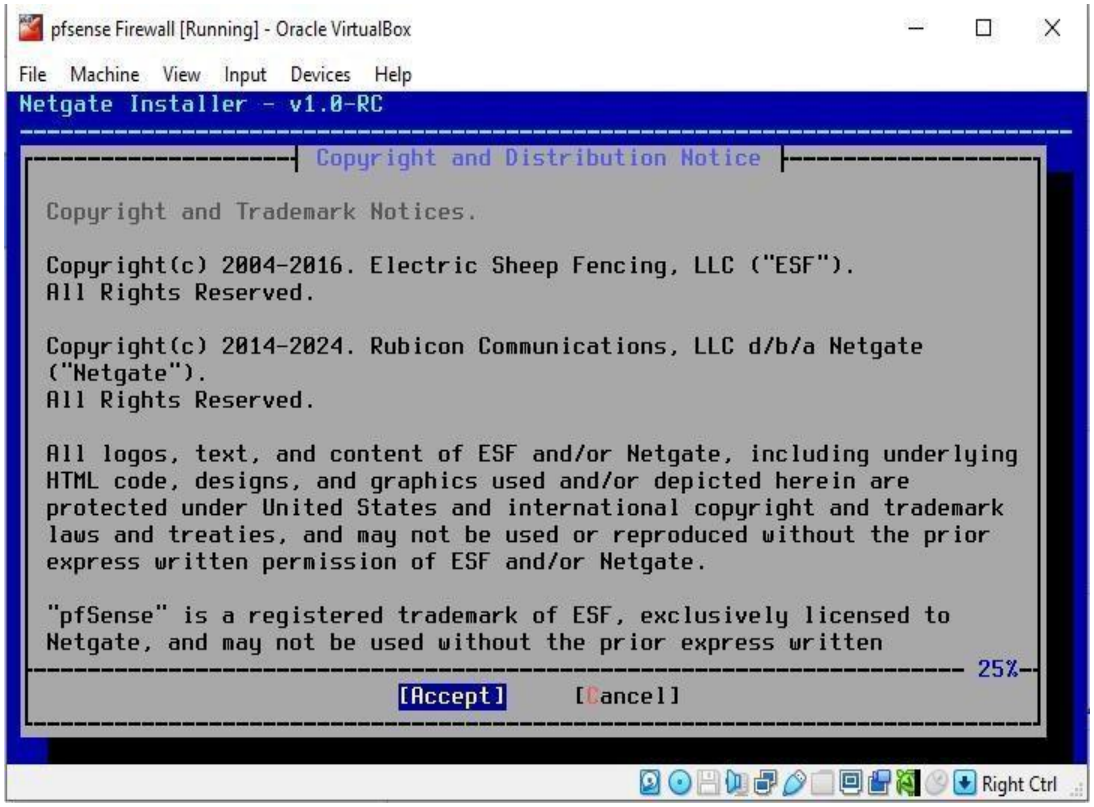




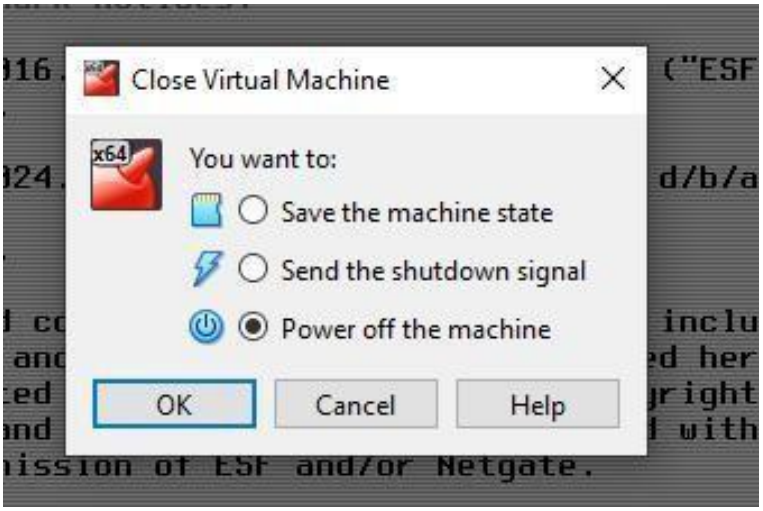
Complete:

The **Reboot** option was chosen to restart the computer and boot into the newly installed pfSense firewall. This is the final step to begin using the system. A new thing is the **Shell** option, which gives you a text-based command screen to do advanced configuration before you finally reboot the system. If you choose the shell, the system won't restart automatically, and you would have to do it yourself.



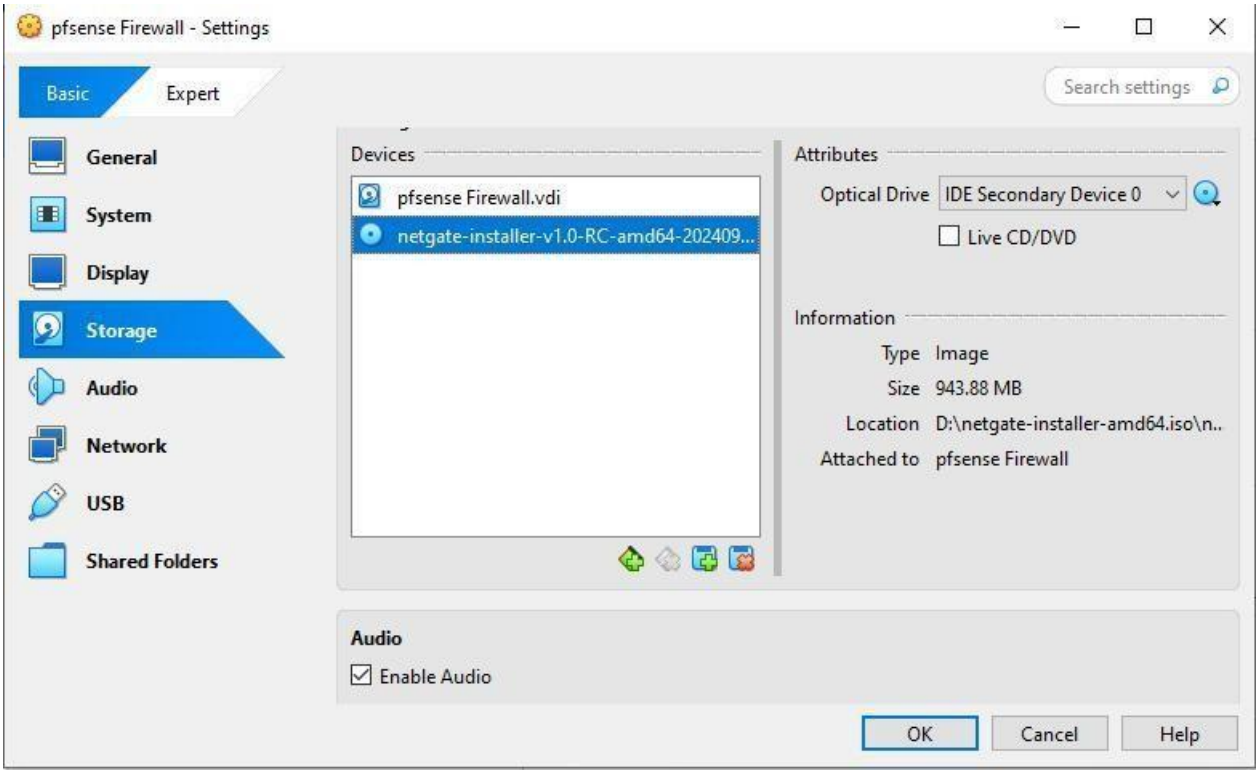


Now you have to power off your pfsense firewall virtual machine because we would now have to assign the **LAN interface.**

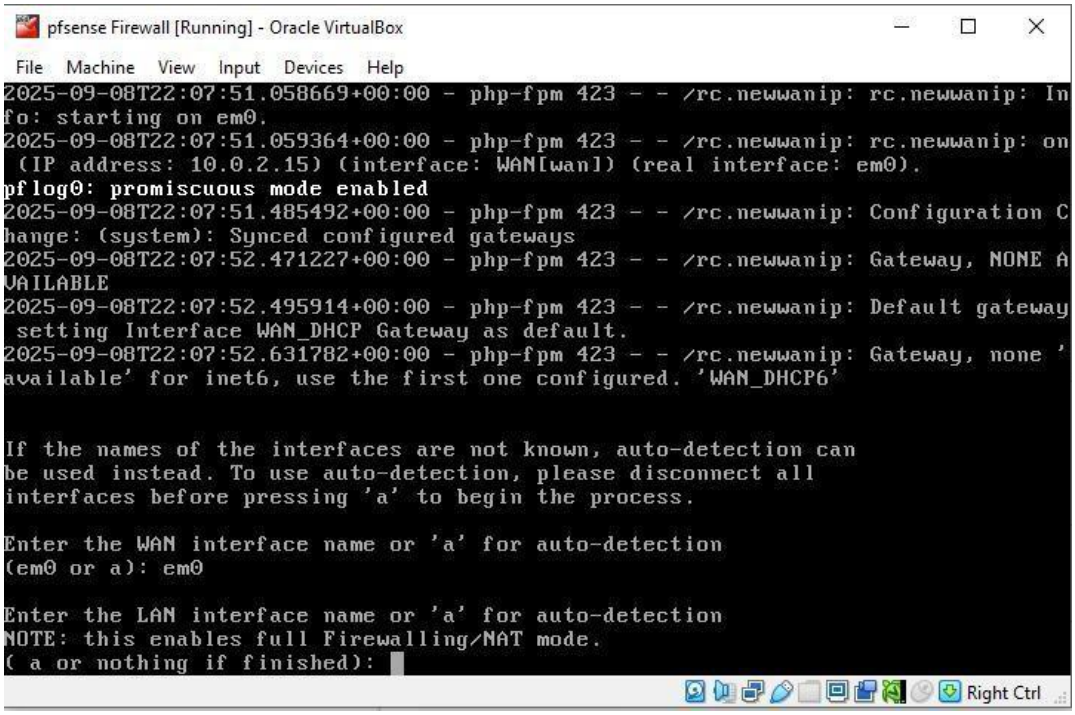


Now that the installation is complete, you need to remove the installation disc, which is the **.iso file**. To do this, go to the **Virtual Machine settings**, select the **Storage tab**, and click on the .iso file to remove it. You do this so that when you reboot the virtual machine, it doesn't start the installation process again. Instead, it will boot from the virtual hard drive where pfSense is now installed.

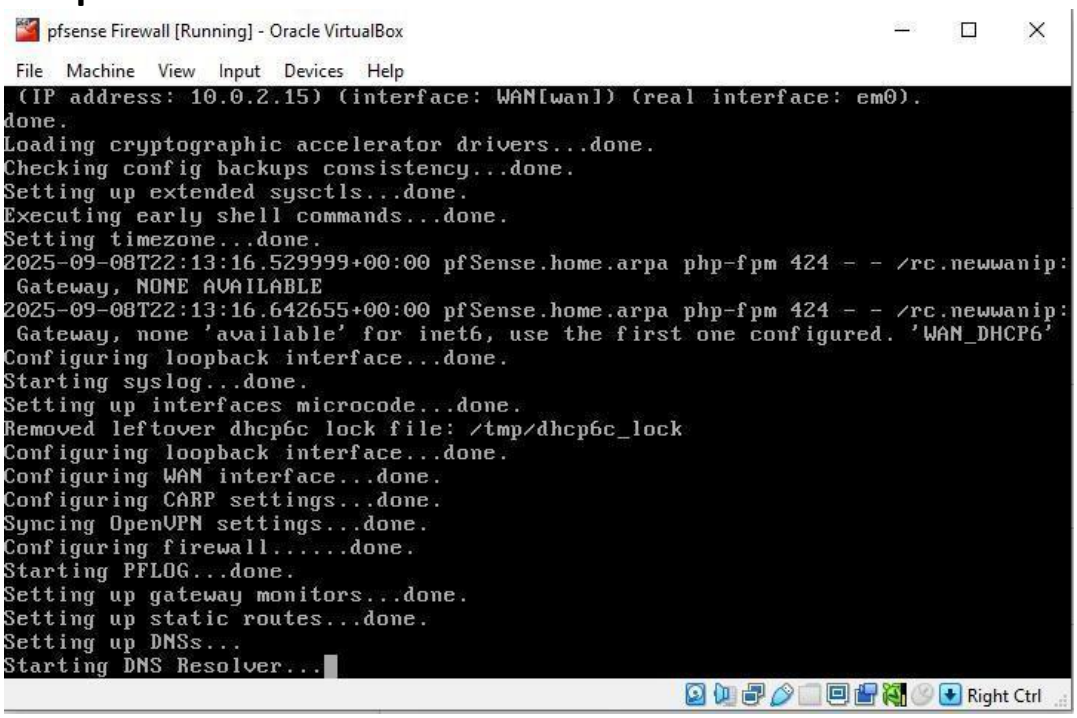
Click on the **Okay** Button to proceed further.



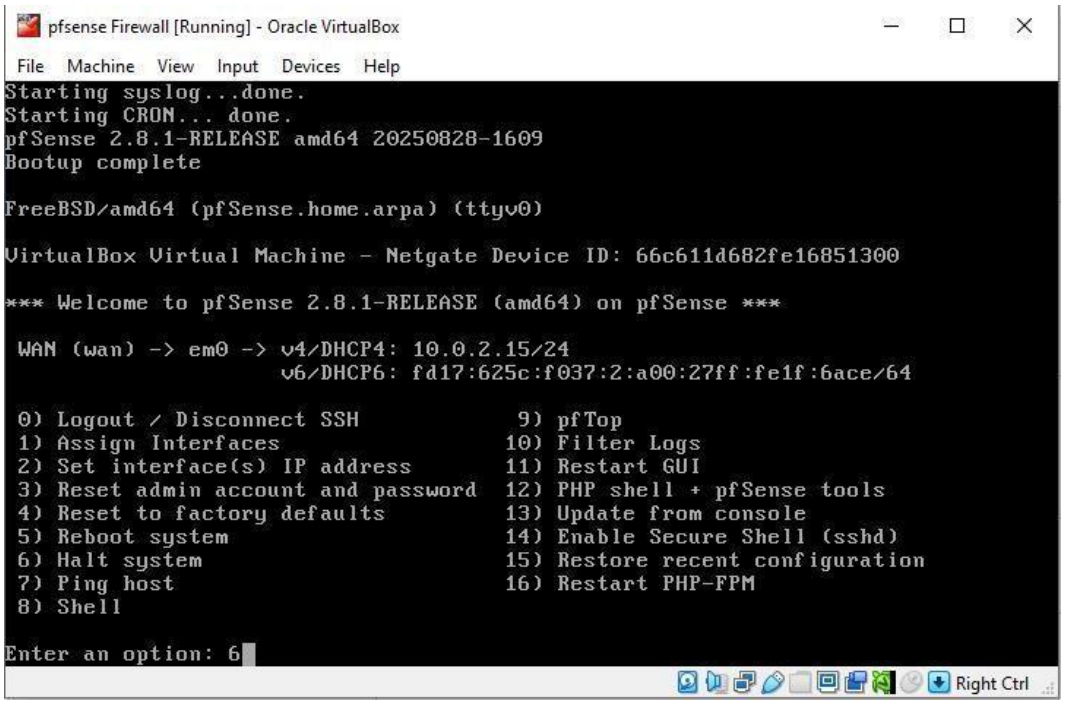
After you've **removed** the installation file, restart the pfSense virtual machine. This is the final step to make sure the system boots from the hard disk where the firewall is now installed. When it reboots, you will see the **pfSense login screen** for the first time.



Once the system asks you to enter the **LAN interface name**, just press **Enter**. This will accept the default settings and automatically assign the network interfaces for you. This is the simplest way to proceed and helps avoid making any mistakes during the initial configuration. The system will then continue with the **final setup.**



The output confirms that your **WAN** (Wide Area Network) interface is properly configured and connected to the internet. It shows that your firewall has been automatically given an **IPv4** address (10.0.2.15) and an **IPv6** address using **DHCP**. The name **em0** refers to the specific physical network card that is being used for this connection. This is a good sign that your firewall is now ready to receive an internet connection.



After completing the installation and initial configuration, you'll need to **halt** or turn off the virtual machine to finish the setup process. This is done to safely **shut down** the pfSense firewall from within the **operating system**, just as you would with a physical computer.

Adding a LAN Interface in Virtual-Box:
Now that your pfSense virtual machine is shut down, you need to add a second network card to act as the LAN (Local Area Network) interface. This will allow other virtual machines to connect to your pfSense firewall and get internet access.

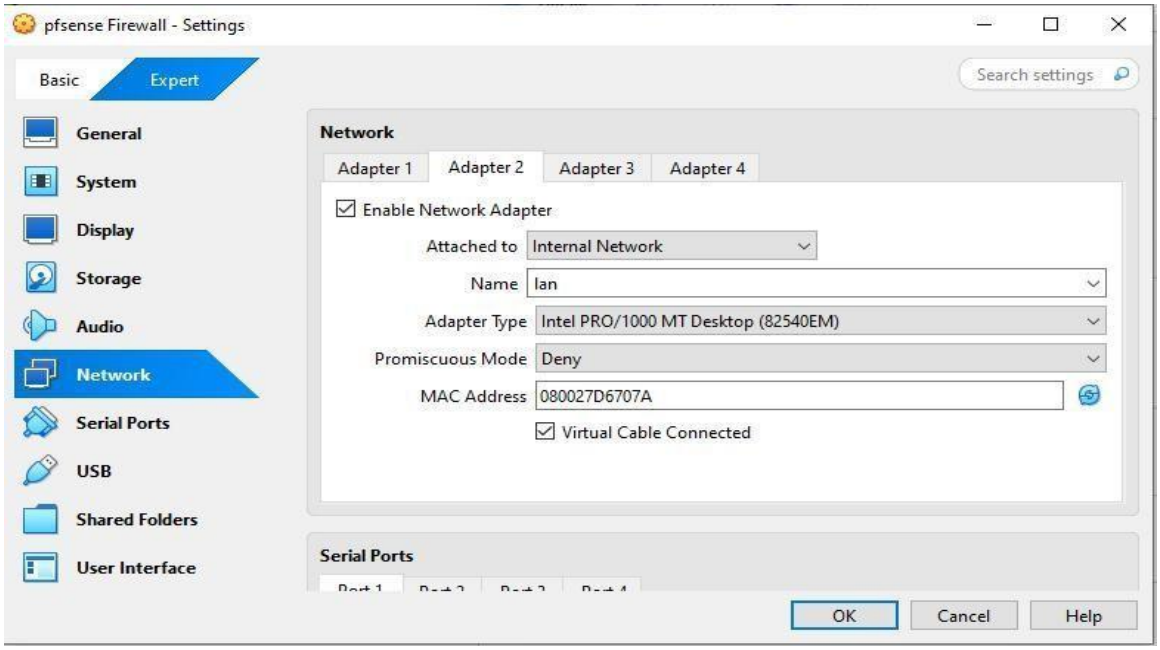
Step-by-Step Instructions:
Open Settings: On the main Virtual Box screen, select your "pfSense Firewall" machine and click the Settings button at the top.

Go to Network Tab: In the settings window, click on the **Network** tab from the menu on the left. You should see "Adapter 1," which is already set up as your WAN connection.

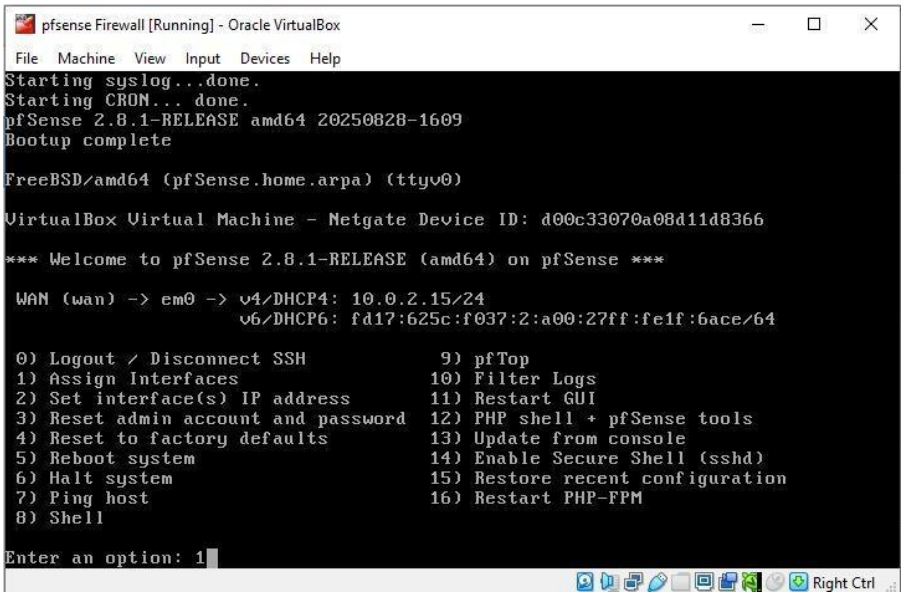
Add a New Adapter: Click on the **Adapter 2** tab. Check the box that says **Enable** Network Adapter. This turns on the second network card for your virtual machine.

Set the Network Type: From the "Attached to" drop-down menu, select Internal Network. This is a very important step because it creates a private network that is only accessible to other virtual machines you connect to it. This isolates your internal network from the main network on your computer.

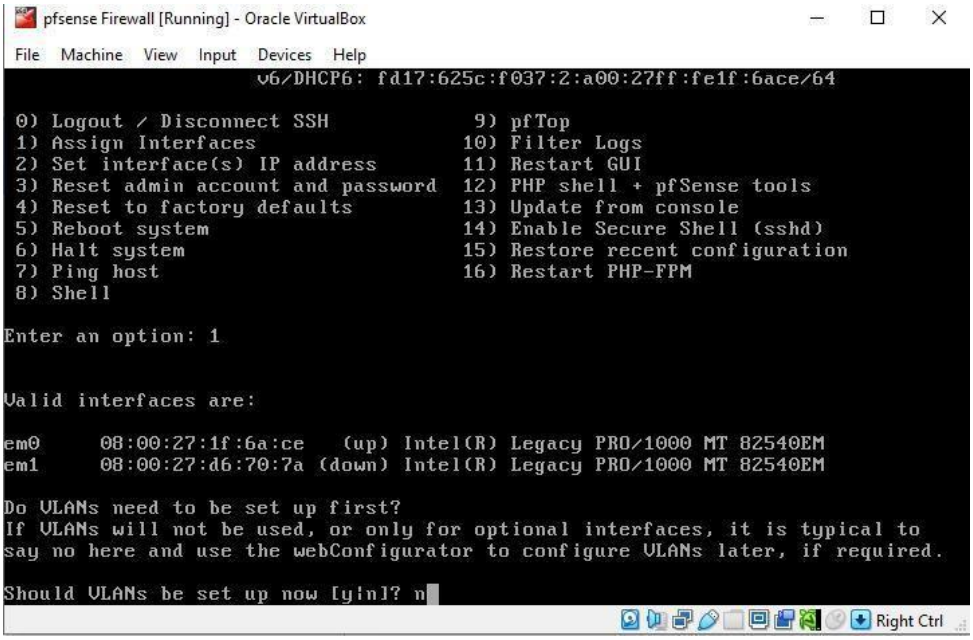
Name the Network: In the **Name** field, type a simple name like **lan**. This helps you identify this network easily. Once you are done, click **OK** to save the changes.



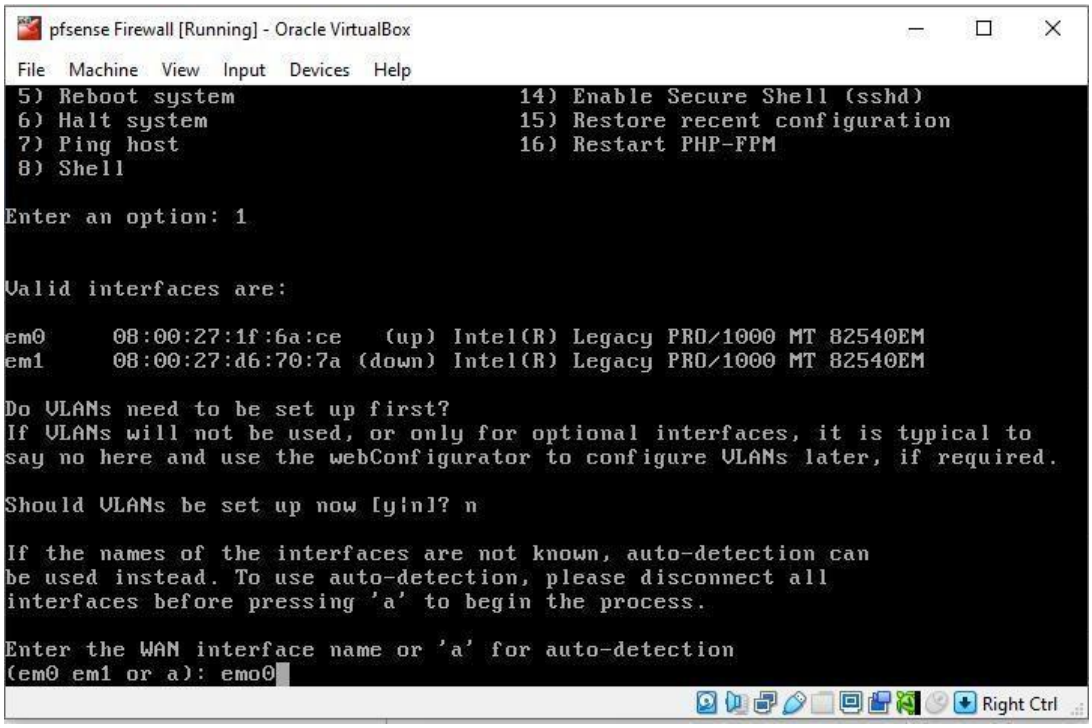
Navigating the pfSense Console:
After rebooting, the pfSense firewall starts up and presents you with its main console menu. By entering **Option 1 (Assign Interfaces)**, you can manually configure which network adapters will be used for the WAN and LAN connections. This is necessary because you just added a new network card in Virtual Box for your LAN.



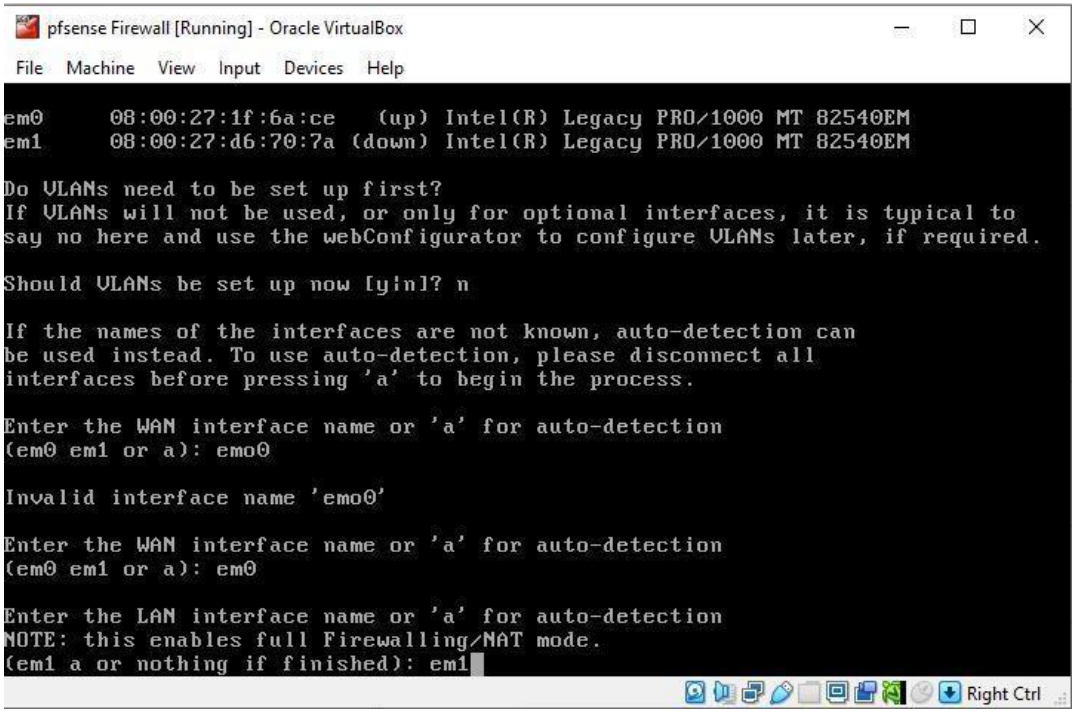
Assigning Network Interfaces:
Next, the system shows you the valid network interfaces it has found, which are **em0** and **em1**. It then asks if you want to set up **VLANs**. A VLAN (Virtual Local Area Network) is a way to create separate virtual networks on a single physical network adapter. For a basic setup like this, you don't need VLANs, so you correctly choose **'n'** for no.

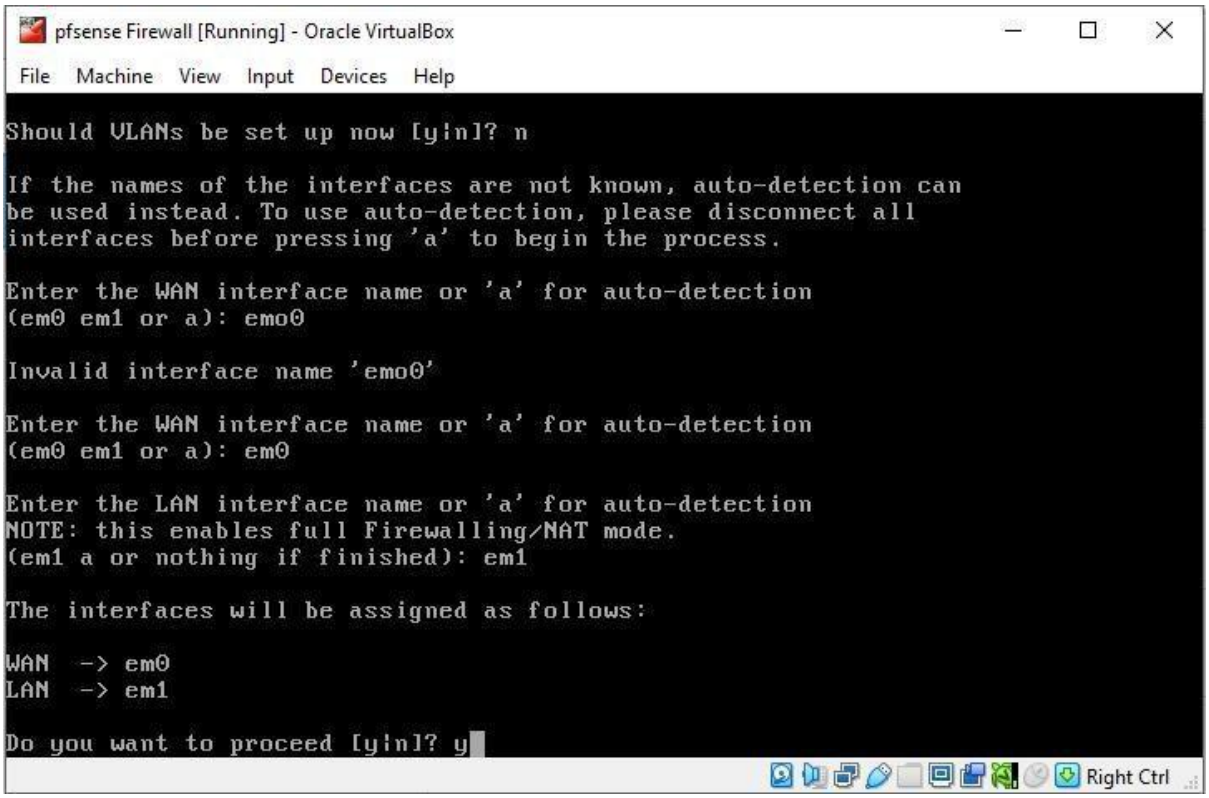


After this, the system asks you to identify which interface should be the **WAN** (your internet connection). You correctly identify and enter **em0** because it is already active ("up"). You could have also used **'a'** for auto-detection, but manually selecting the interface is a more precise method to ensure the correct connection is used.

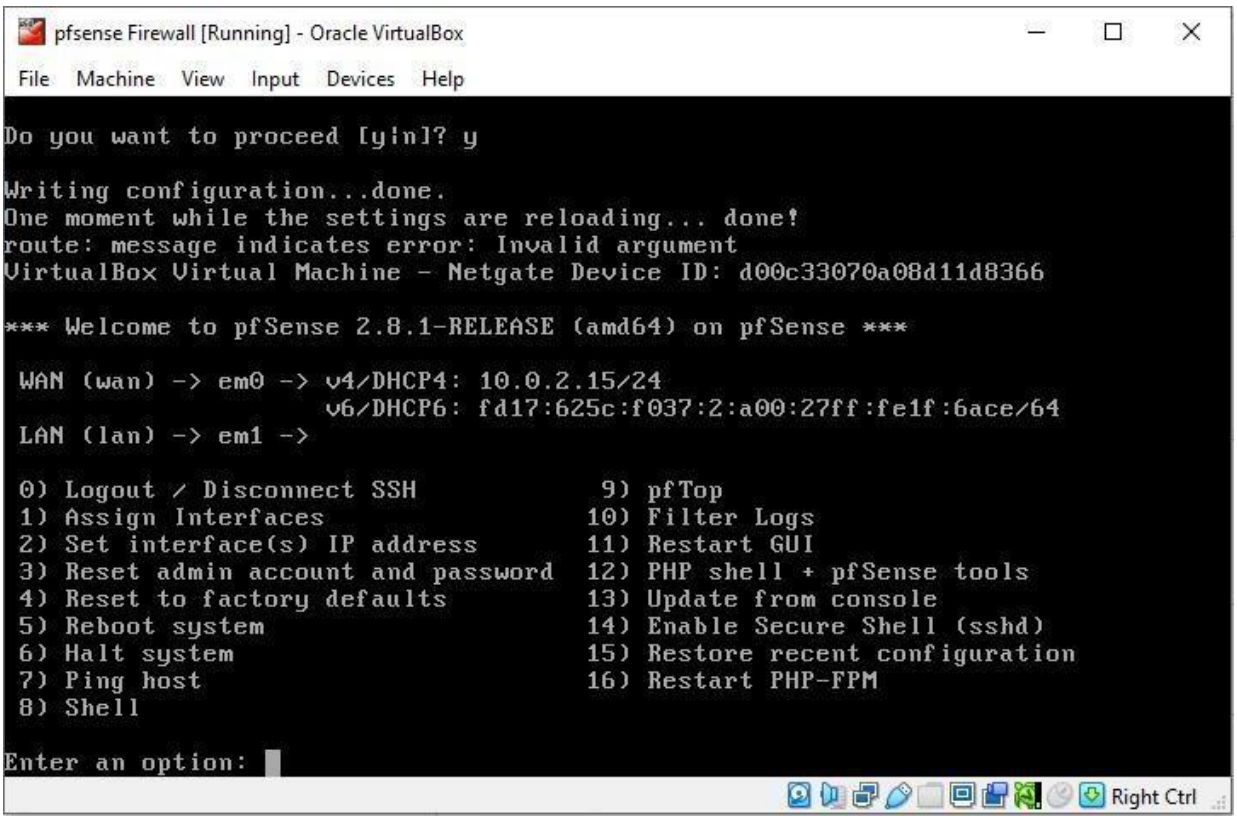


The screenshot confirms that you have successfully assigned both the **WAN** and **LAN** interfaces. By entering **'y'** to proceed, the system wrote the new configuration and reloaded the network settings.





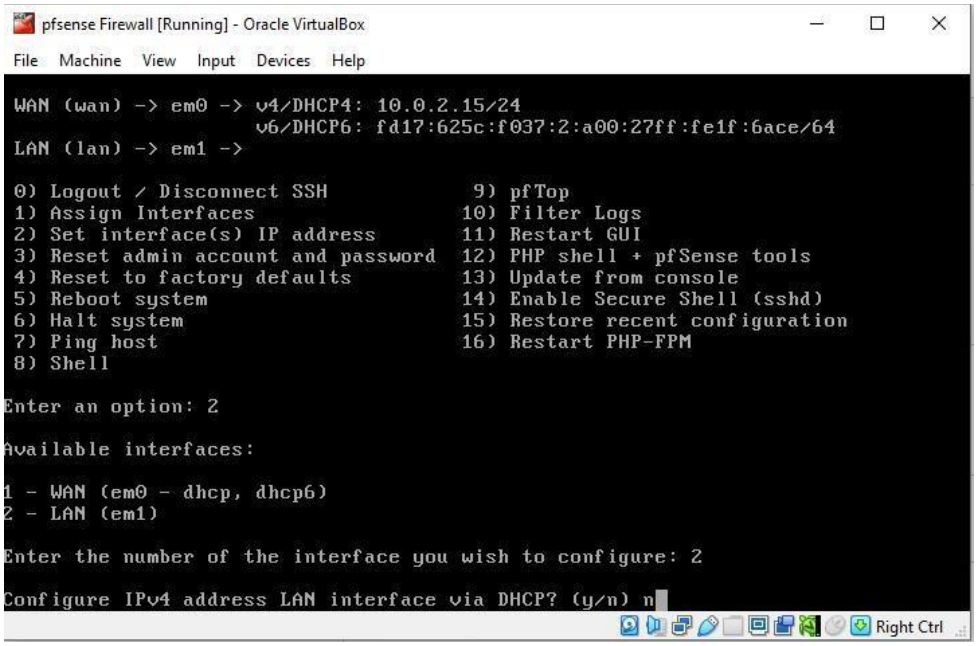
You can see that **em0** is now correctly assigned as your **WAN** interface, and the new **em1** is assigned as the **LAN** interface.The system is now ready for the next step, which will involve setting up the IP address for your LAN.



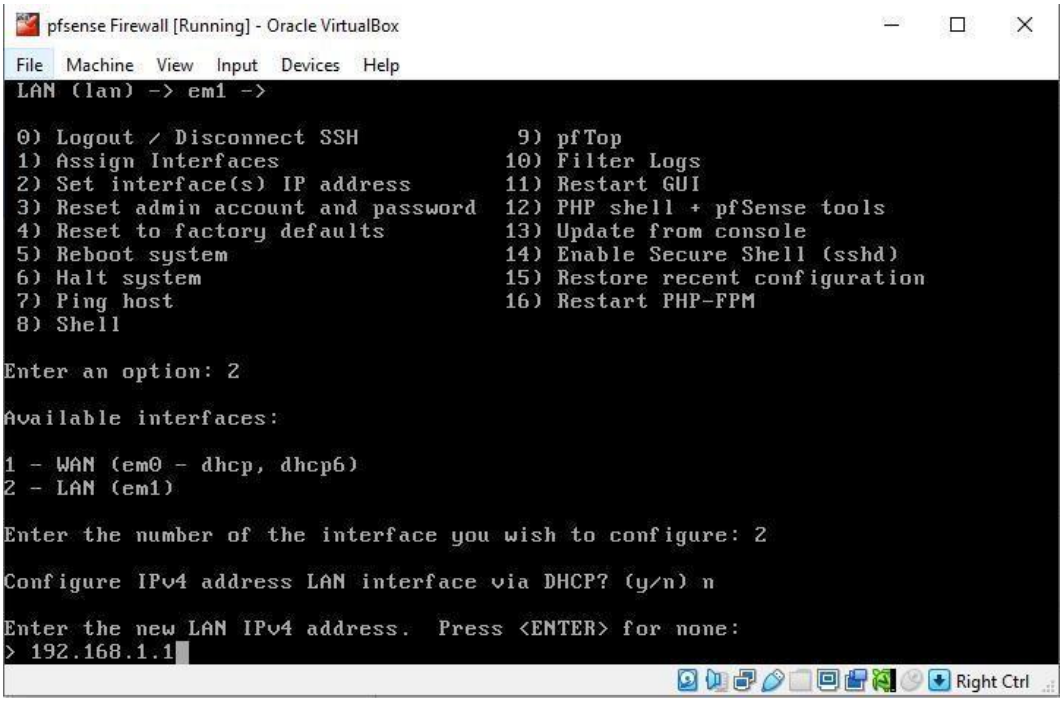
For assigning an **IP address to LAN** (Local Area Network) you would have to choose the **option No. 2.**

When you're prompted to configure the LAN interface with DHCP, you must decline by choosing **No (n)**. The **WAN** (internet) interface of a firewall gets its IP address from a router or an ISP using DHCP, which is a common way to connect to the outside world.

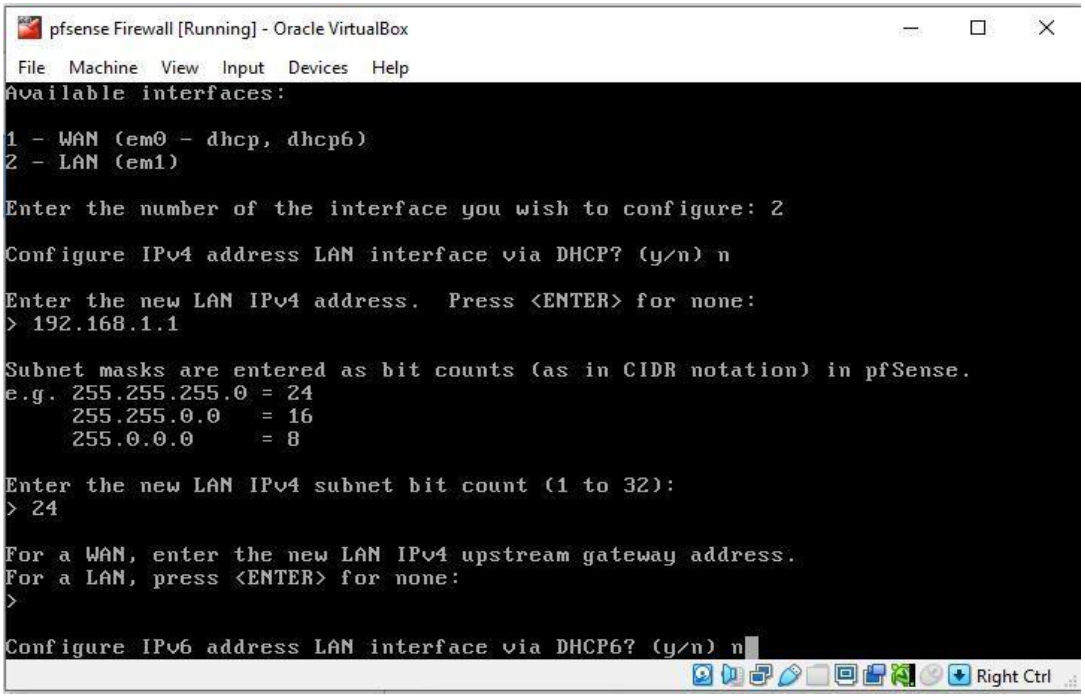
The **LAN** (internal) interface, however, is meant to provide IP addresses to devices on your local network. It needs to have a **static IP address** so that it can act as the main gateway and DHCP server for all other devices on the network. If you were to set the LAN interface to get an IP address via DHCP, you would create a **conflict** in the network, and your internal network would not function correctly.



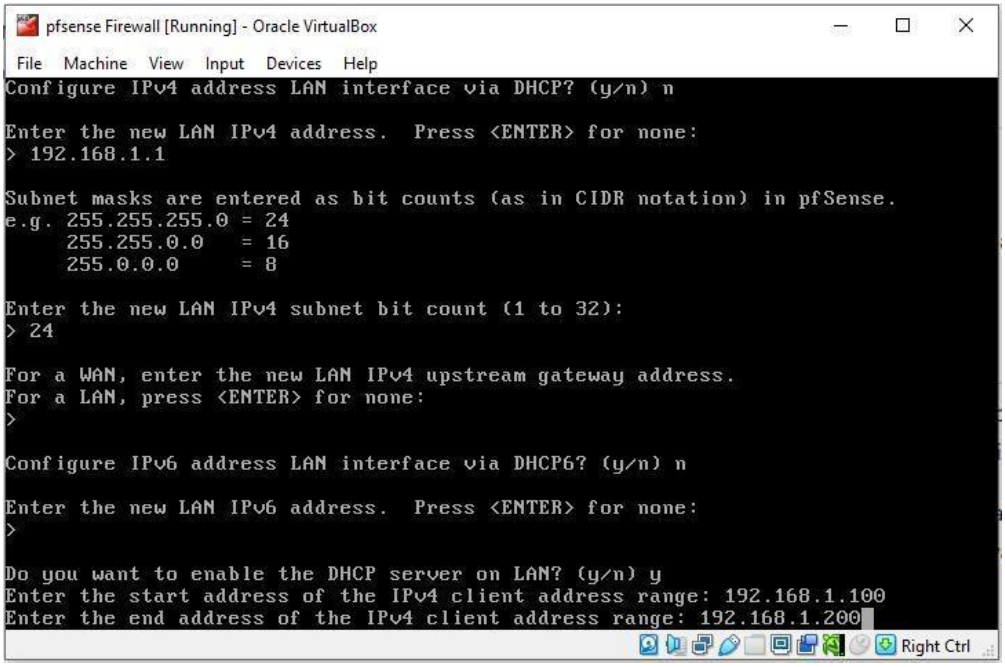
Now that you have declined DHCP for the LAN, you can manually set the **static IP address**. By entering **192.168.1.1**, you are assigning a fixed IP address to your LAN interface. This is a very common address for a router's internal network. This IP address will serve as the **gateway** for all the devices on your internal network.



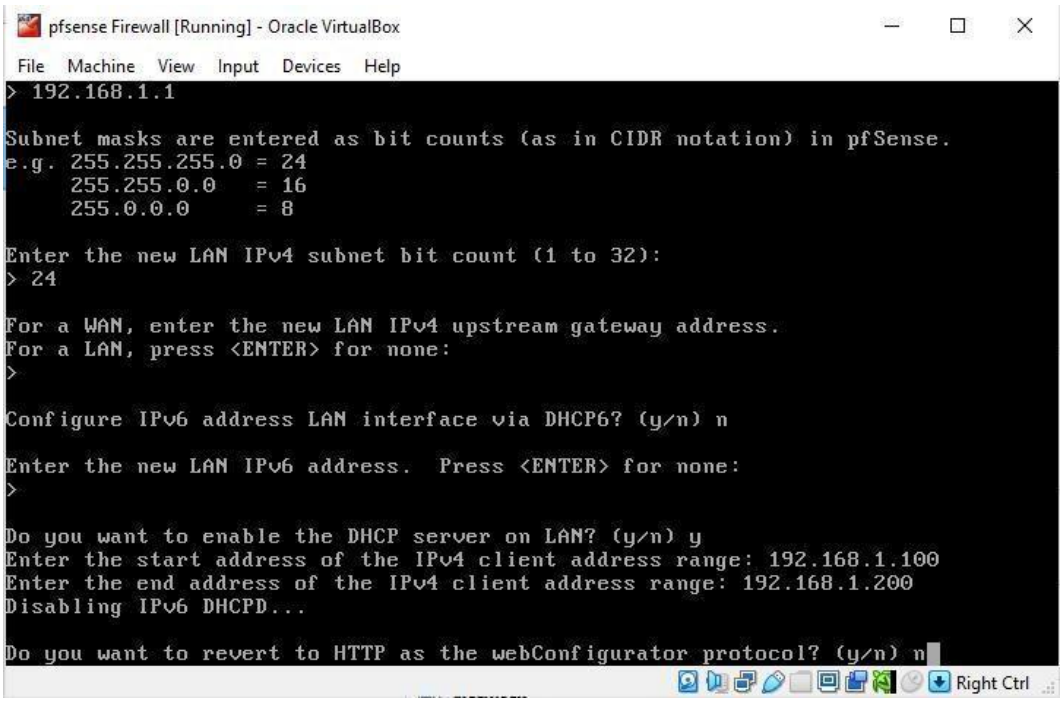
Now you will be asked about configuring IPv6 for the LAN interface. You should decline this as well by typing **'n'** for **'no'** and pressing Enter. Just like with IPv4, the **LAN interface needs a static IP address** to function as a reliable gateway for your internal network. Since you've already set up the IPv4 address, you don't need to configure IPv6 at this stage. You can always set up IPv6 later if you need to.



By typing **'y'** to "enable the **DHCP server** on **LAN**," you are turning on a critical feature of your new firewall. This means the pfSense firewall will now **automatically** give out **IP addresses** to any devices that connect to the LAN. This allows other virtual machines on your internal network to get an IP address automatically from pfSense without needing to be configured manually.

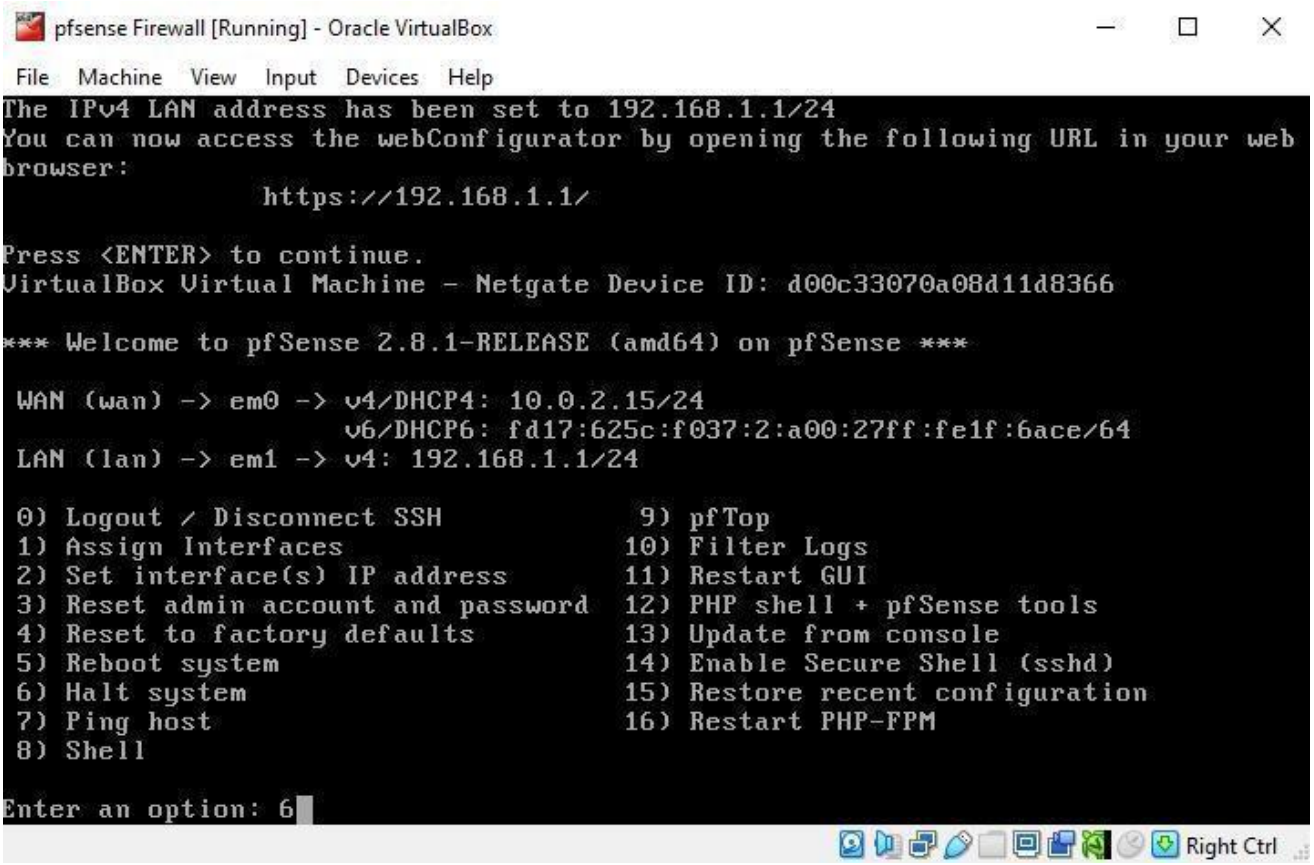


This screenshot shows the final steps of configuring your pfSense firewall's LAN interface. You have set a static IP address of **192.168.1.1** with a **/24** subnet, which is standard for a small network. You also enabled the **DHCP server** to automatically give out IP addresses to other devices within the range of **192.168.1.100 to 192.168.1.200**.You have to chose **No** to keep the connection secure by using the **HTTPS** protocol instead of the less secure **HTTP.**



This screenshot below confirms that the initial setup of your pfSense firewall is complete. The console shows that the **LAN IP address** has been successfully set to **192.168.1.1**, which is the gateway for your internal network.
WAN (wan) **- > em0 - >** v4/DHCP4: 10.0.2.15/24
v6/DHCP6: fd17:625c:f037:2:a00:27ff:fe1f:6ace/64
LAN (lan) **- > em1 - >** v4: 192.168.1.1/24

Choose the **Option 6** from the console menu to power off your pfsense firewall virtual machine so we can further manually **configure LAN** on our Kali Linux virtual machine.



After configuring the LAN interface in pfSense, you need to go to your Kali Linux virtual machine's settings and enable its network adapter. You will then select **Internal Network** and name it **lan** to connect it to the same network as your firewall. This ensures all of Kali's internet traffic is routed through pfSense for security and management.

Set the settings like this:

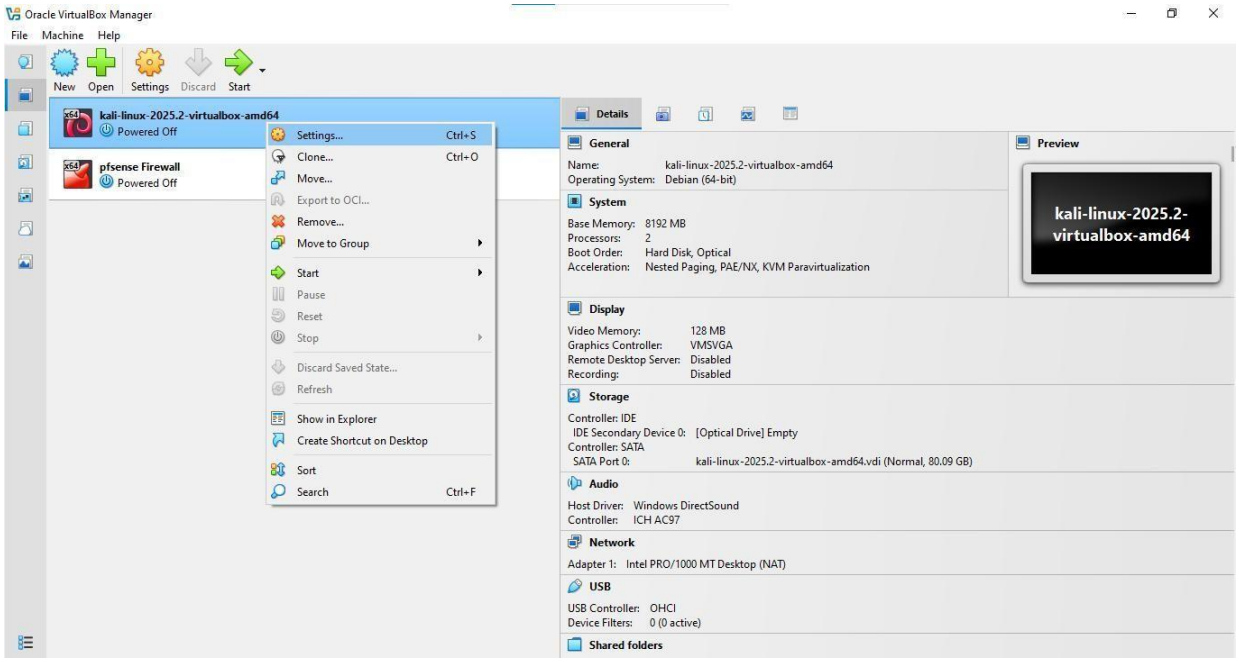
Attached to: Internal Network

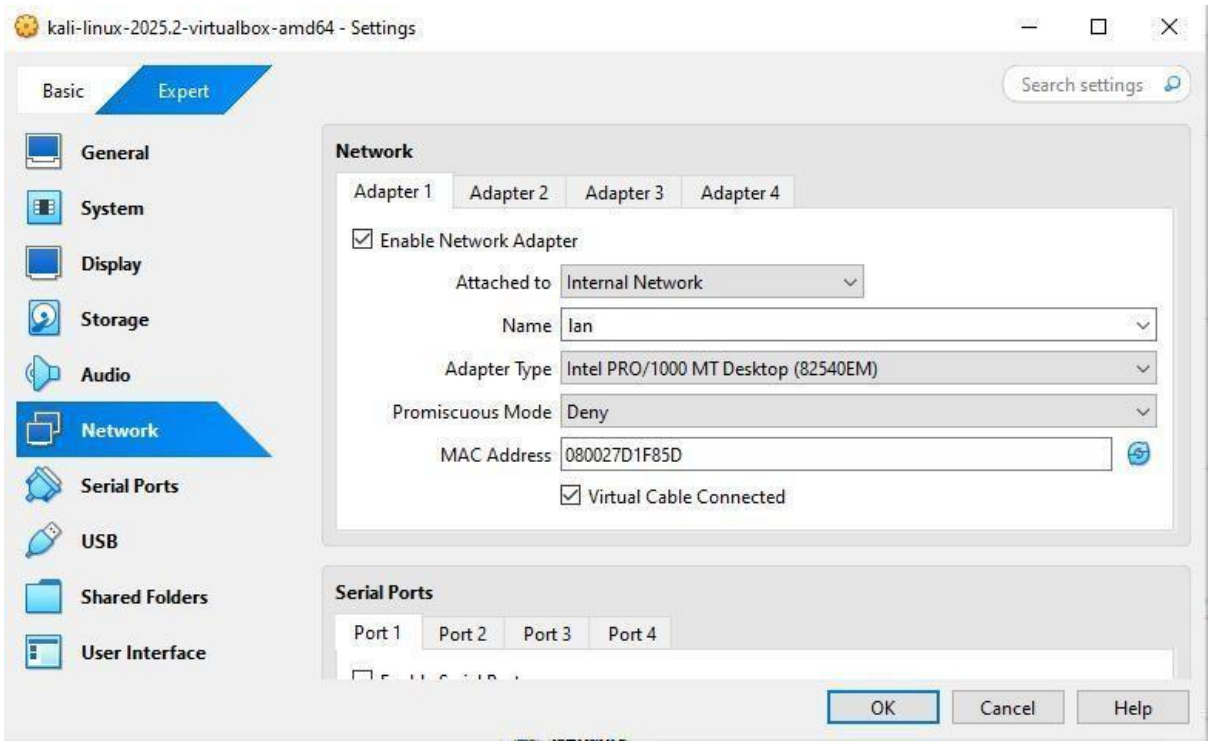
Name: LAN

Adapter Type: Intel PRO /1000 MT Desktop (82540 EM)

Promiscuous Mode: Deny

MAC Address: 080027D1F852

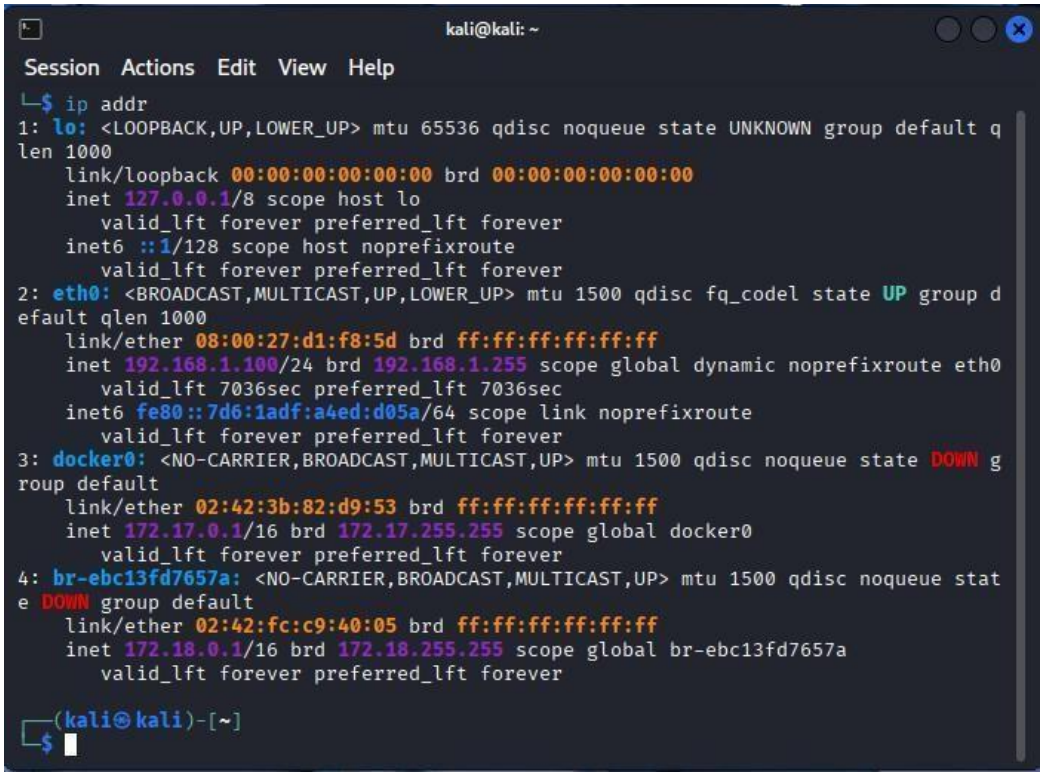




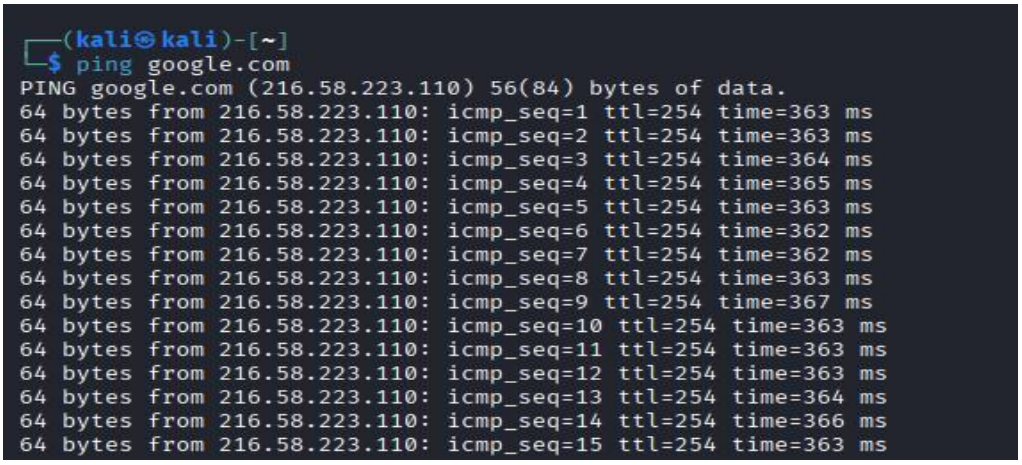
After that **Power on** both the virtual machines **(Kali Linux - PFsense Firewall)**. This screenshot shows the output of the **ip addr** command, which displays the network details of your Kali Linux machine. The most important line is the **eth0** interface, which now has an **IPv4** address of **192.168.1.100**. This confirms that Kali Linux has successfully connected to the **LAN** interface of your pfSense firewall and was automatically assigned an IP address from the **DHCP server** you configured earlier meaning the Kali Linux and pfsense firewall have successfully connected.

Command:

ip addr

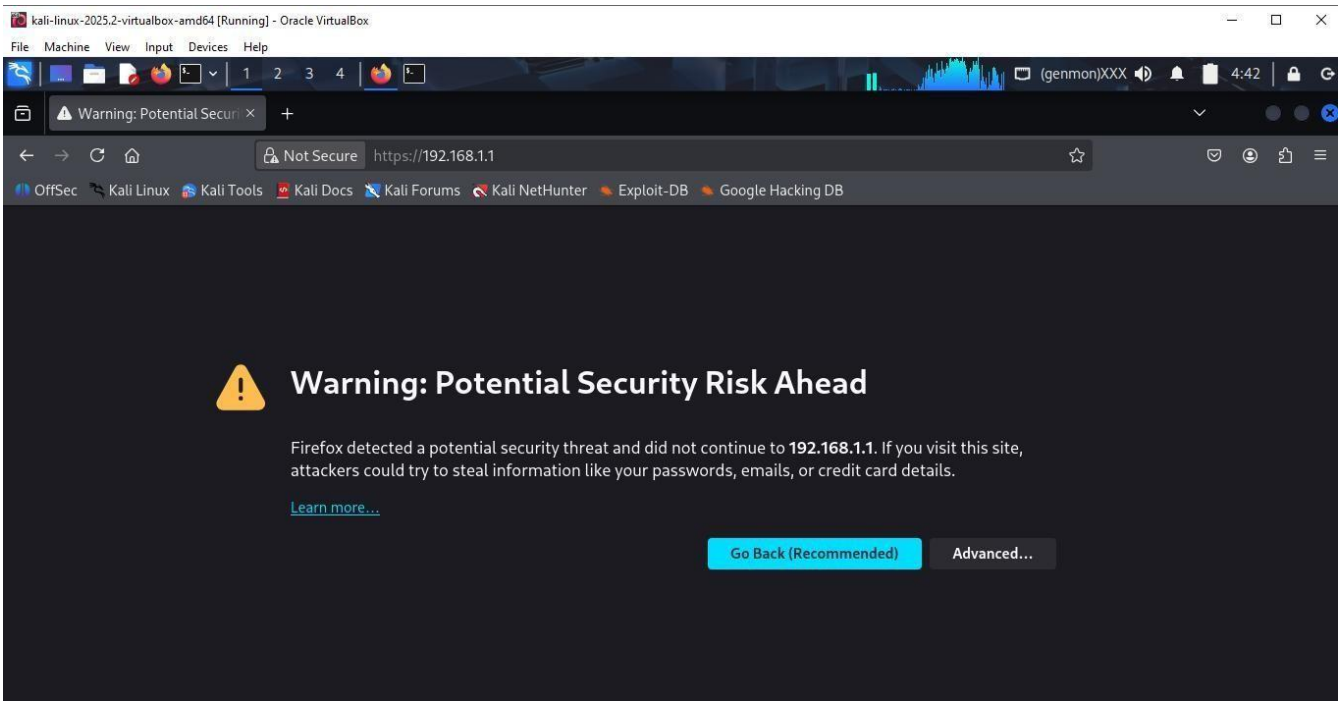


The **ping google.com** command was used as a final test to confirm that your Kali Linux machine has an **internet connection**. A successful ping proves that your entire network setup is functioning correctly.

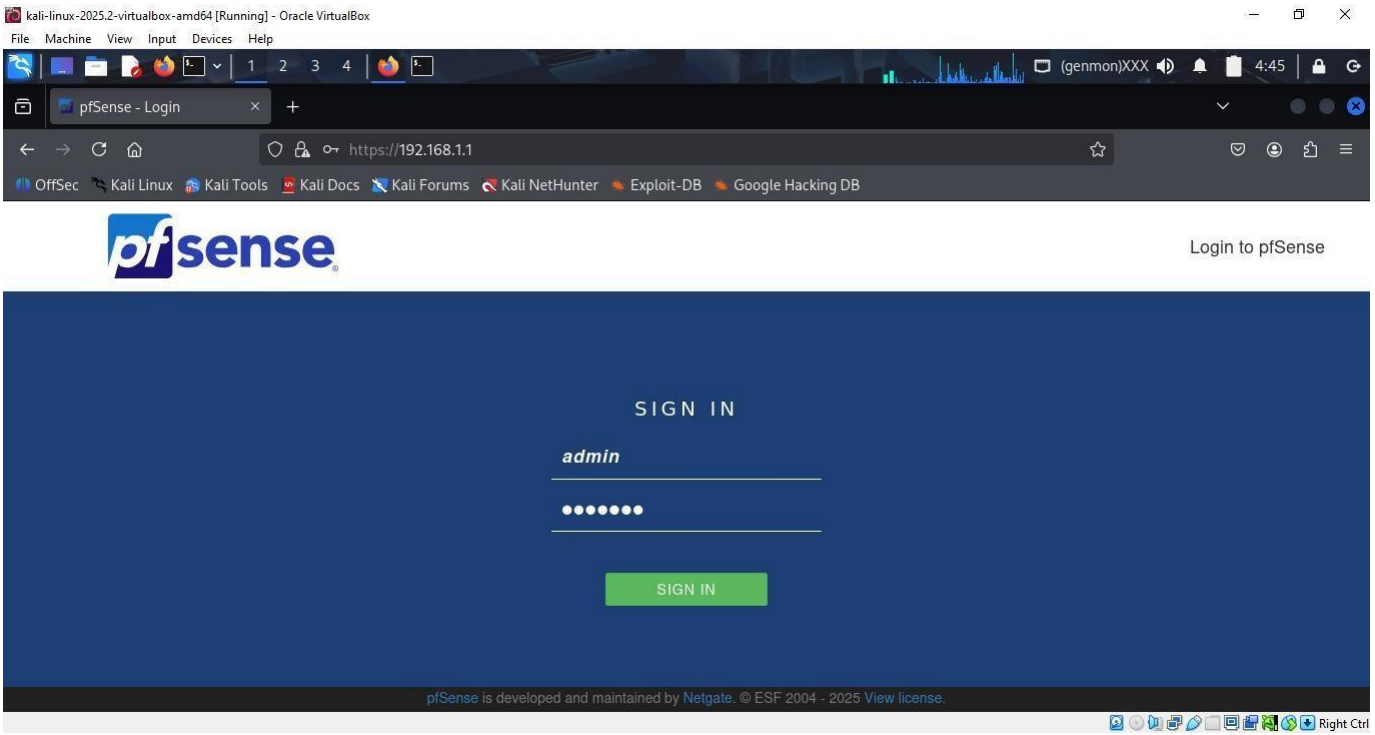


4.3 : Remote Logging Enablement

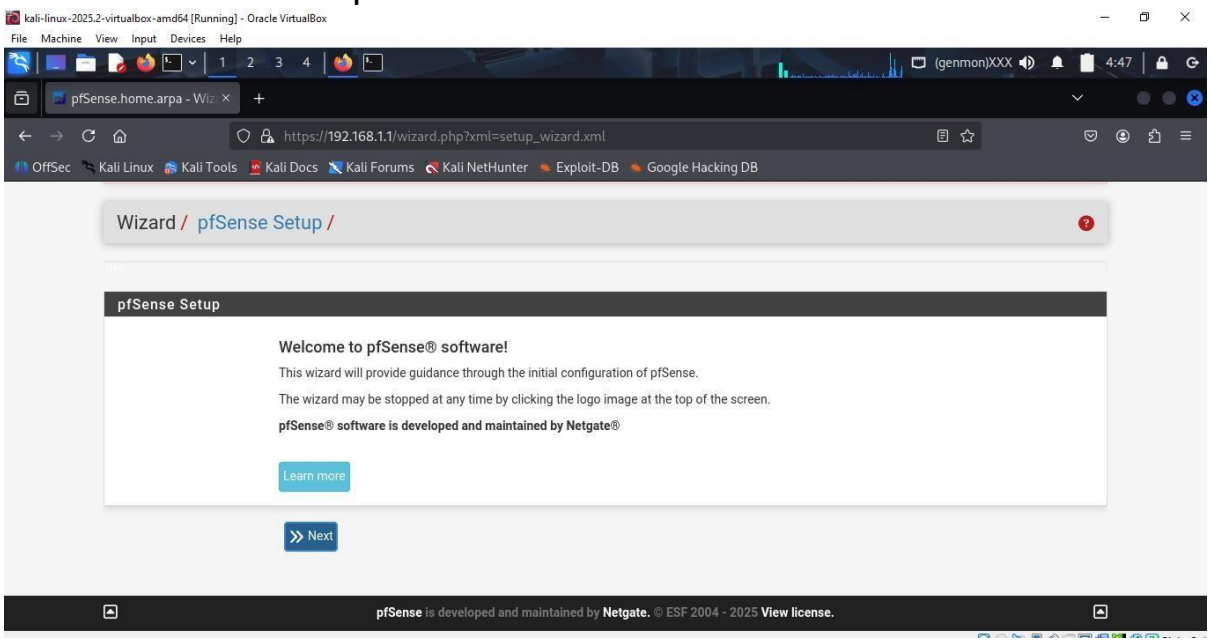
Navigating Security Warnings:
When you first try to access the pfSense web interface, your browser displays a security warning. This is a normal and expected step because pfSense uses a **self-signed certificate**. Since this certificate wasn't issued by a recognized third-party authority, your browser's security features flag it as an untrusted site. You can safely proceed by clicking **"Accept the Risk and Continue"**.



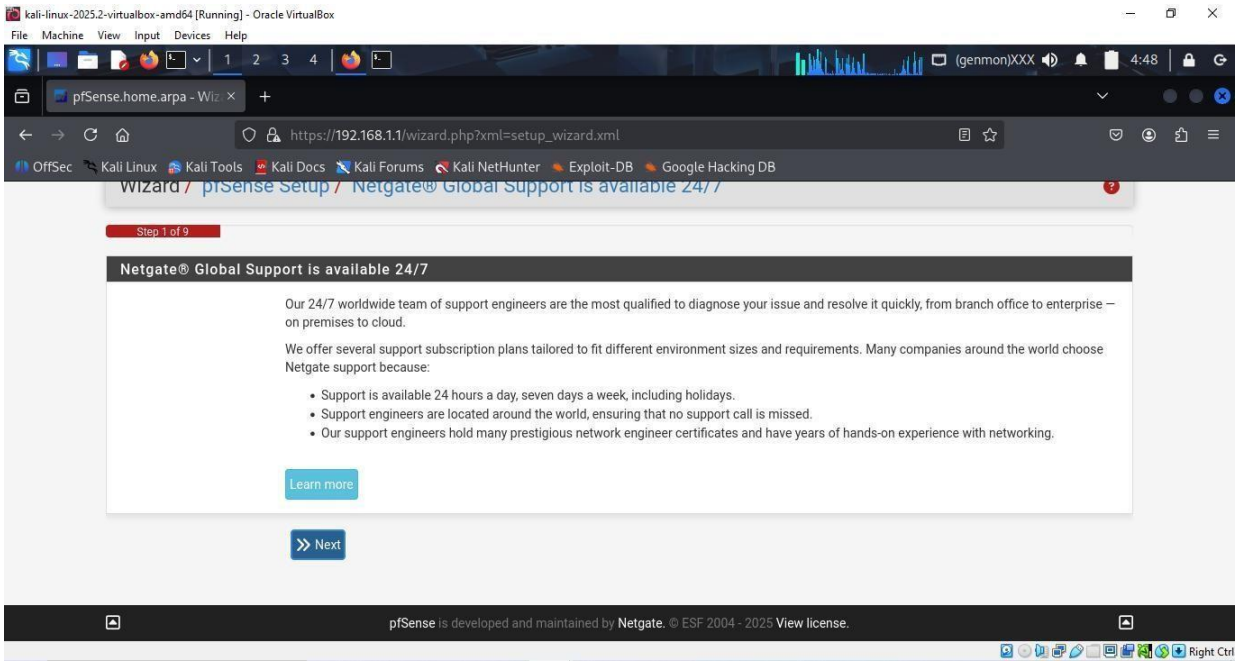
Logging into pfSense:
After bypassing the security warning, you'll reach the main login page for the pfSense web interface. The default username is **admin**, and the default password is also **pfSense**. After you log in for the first time, you'll be prompted to change this password for security.



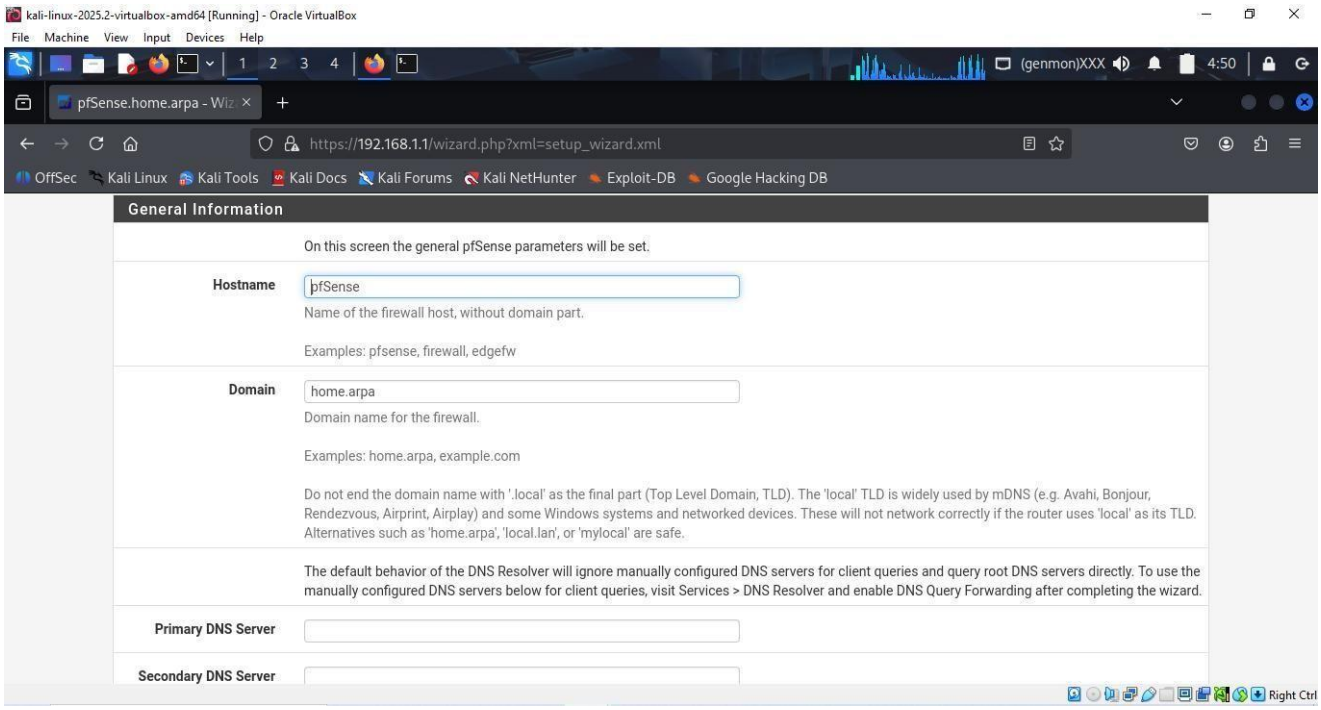
Welcome to the Setup Wizard:
This final screenshot confirms that you have successfully logged into the **pfSense web interface**. You are now on the welcome page of the Setup Wizard, which will guide you through the final configuration steps. You can click **"Next"** to continue with the setup.



Netgate Support:
This screen shows a page with information about Net-gate Global Support. This is the first step of the setup wizard and is just for information. You can simply click on the Next button to move to the next screen.

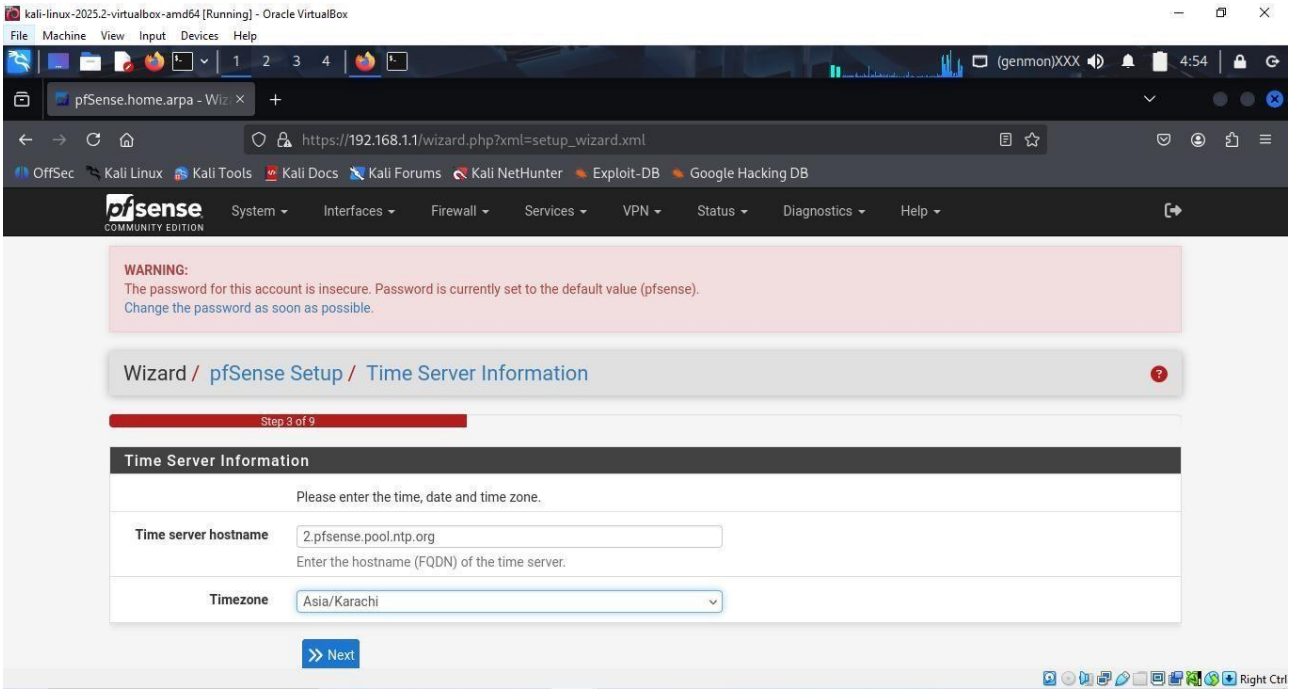


General Information:
Here, you can set the basic information for your firewall. It's a good idea to leave the Host-name as **pfSense** and the Domain as **home.arpa** for this simple setup. Keeping the defaults here makes it easier to manage later on.



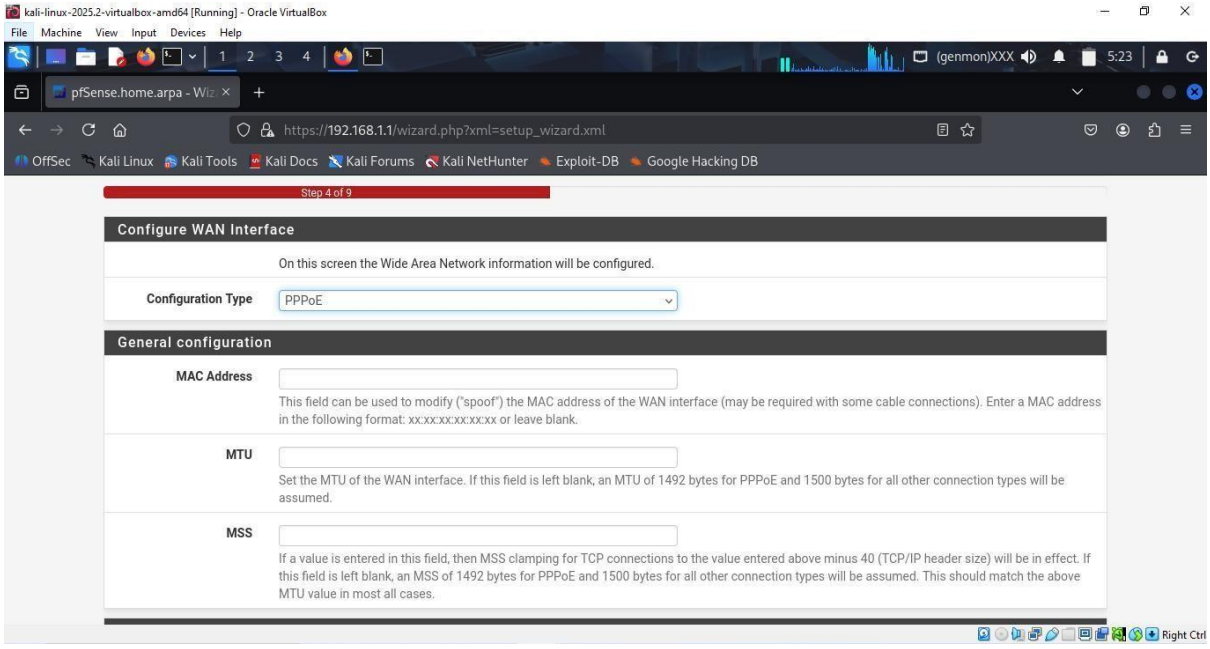
Time Server Information:

This screen is where you configure the time settings for your firewall. It's important to have the correct **Timezone** for **accurate logs** and **security** certificates. The default time server is usually fine, so you just need to select your location to ensure the firewall's clock is correct.



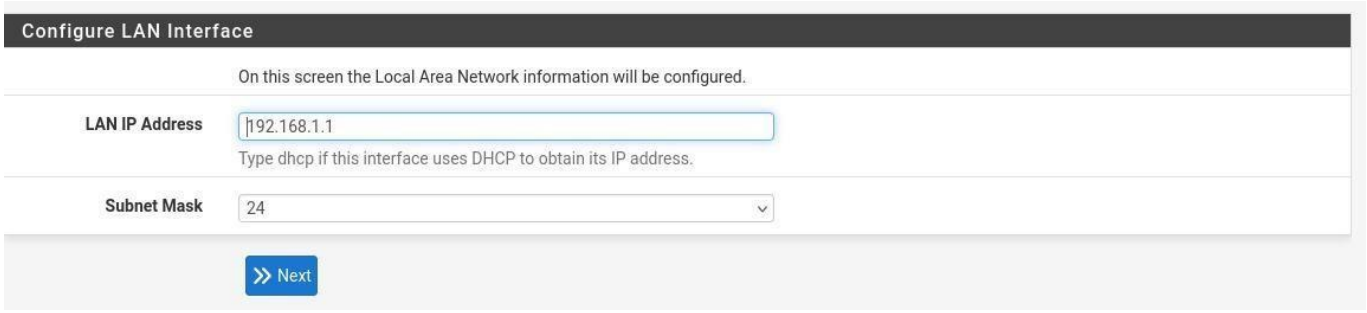
Configure WAN Interface

This page allows you to configure your **WAN (internet) interface**. Since your virtual machine gets its internet connection from your host computer using DHCP, you should **not** change the configuration type to **PPPoE**. The default settings will work just fine for this setup.

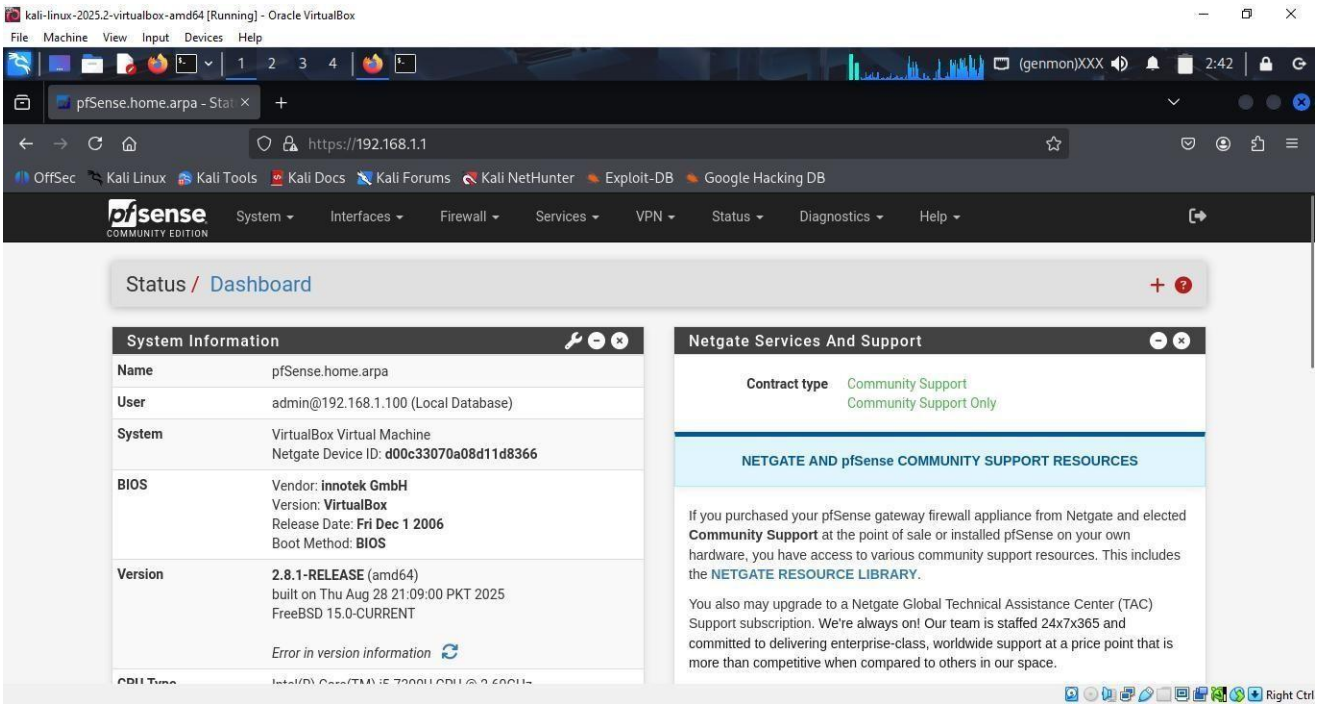


Configure LAN Interface:
This last screenshot confirms your LAN configuration. You can see that the LAN IP Address is already set to **192.168.1.1** with a **24 subnet mask**. This confirms the settings you previously configured from the command line interface are correct. You just need to click Next to **continue** with the final steps of the wizard.

Same process would be carried on step by step.

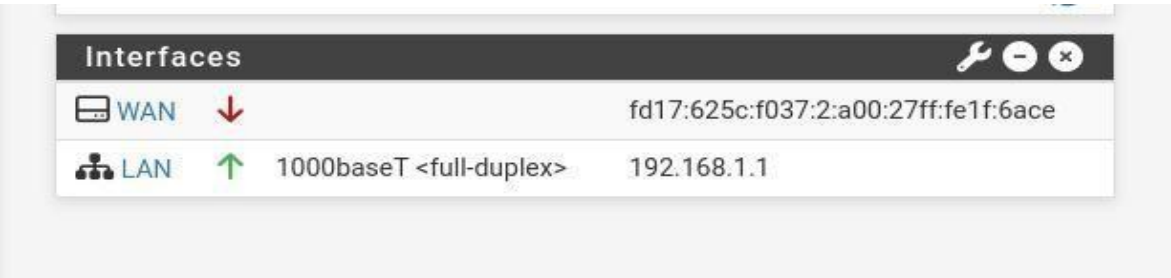


pfSense Dashboard Overview:
This is the main pfSense dashboard, which provides a complete overview of your firewall's status. The **"System Information"** section on the left confirms that pfSense is running on a Virtual Box Virtual Machine with the correct host-name. The navigation menu at the top allows you to access all the advanced configuration options for your firewall.

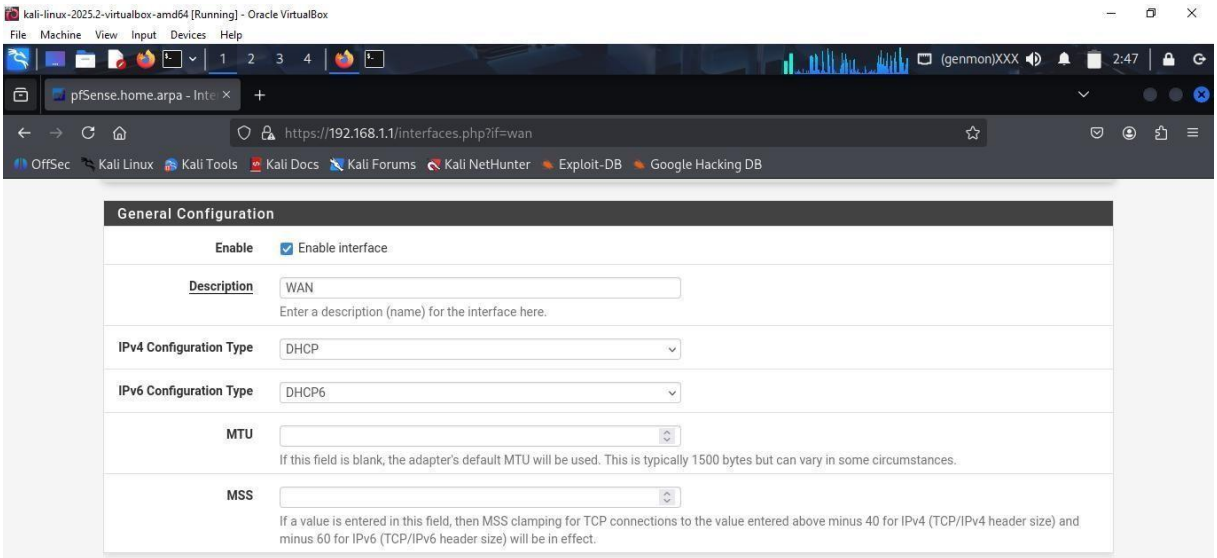


Interfaces Status:

The Interfaces widget gives you a quick look at your network connections. The LAN interface is active and has the correct static IP address of **192.168.1.1**. However, the WAN interface has a red arrow pointing down, which indicates an issue with the connection. It seems that the WAN is only receiving an IPv6 address and is not getting an IPv4 address, which is necessary for most internet access.



To fix your **WAN interface**, go to the Interfaces menu and change the IPv4 Configuration Type to DHCP. This tells pfSense to automatically get a valid internet IP address from your host machine. After saving and applying the changes, the WAN interface will be **active** and your firewall will be connected to the **internet.**

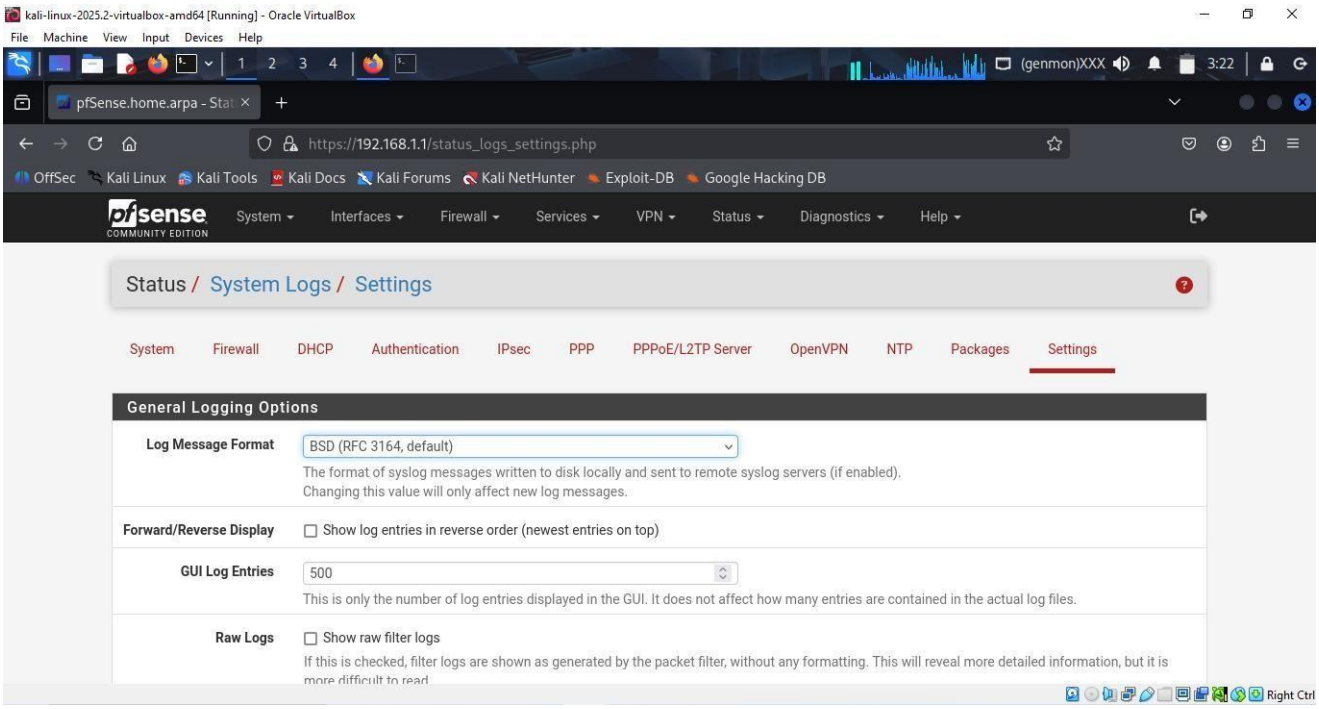


WAN UP - > Your Internet Connection:
When your WAN interface shows "UP," it means that your pfSense firewall is successfully connected to the internet and has a valid IP address. This is the first step in getting your network online.

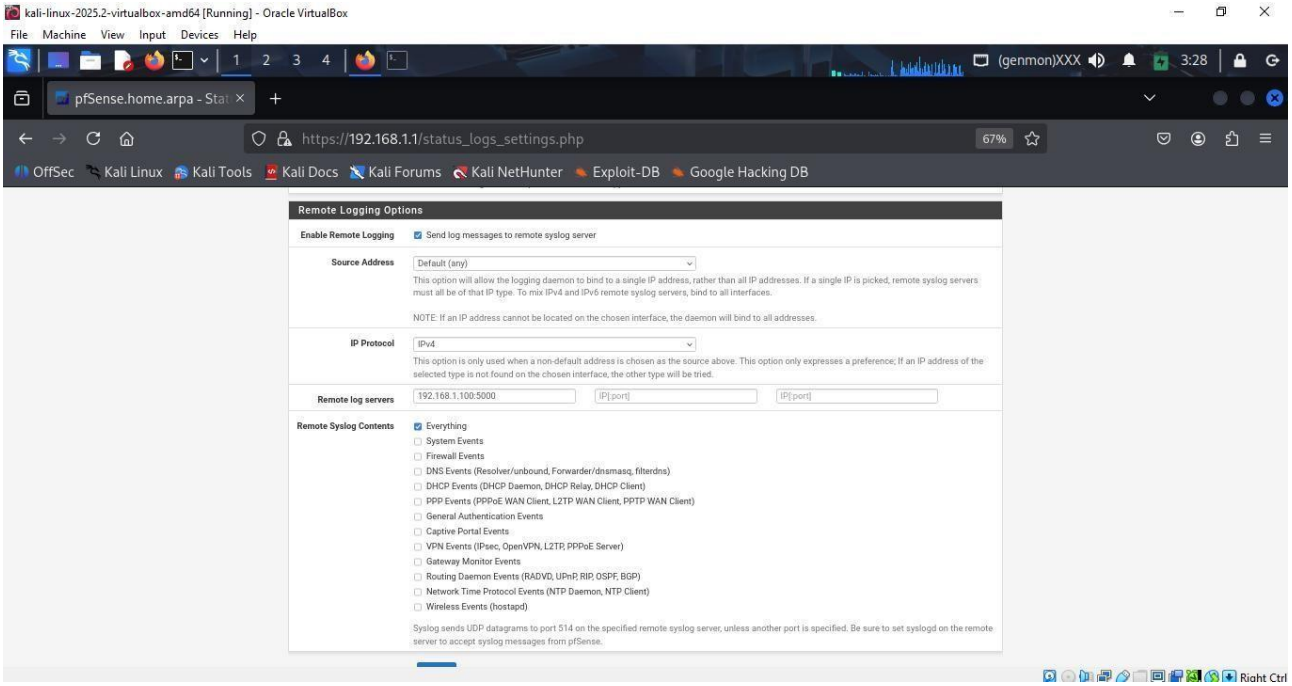
LAN UP - > Your Internal Network:
When your LAN interface shows "UP," it means that your pfSense firewall is properly managing your internal network, where your Kali Linux machine is located. The firewall is correctly controlling the network, giving it access to the internet through the WAN interface.

Final Result:
When both interfaces are "UP," your pfSense firewall is fully functional as a router and a security gateway, allowing your Kali Linux machine to connect to the internet through it.

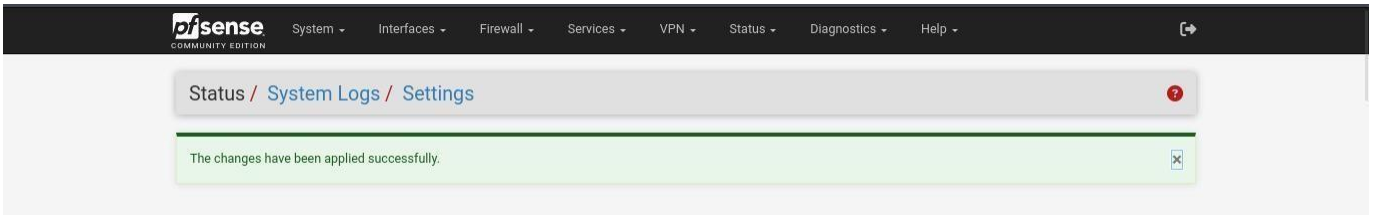
Settings >> System Logs >> Settings
We are on this page to set up how your firewall records all its activity in a log file, just like a diary. We do this to keep a record of all network events and to monitor for any security threats. This helps you troubleshoot problems and keep your network safe.



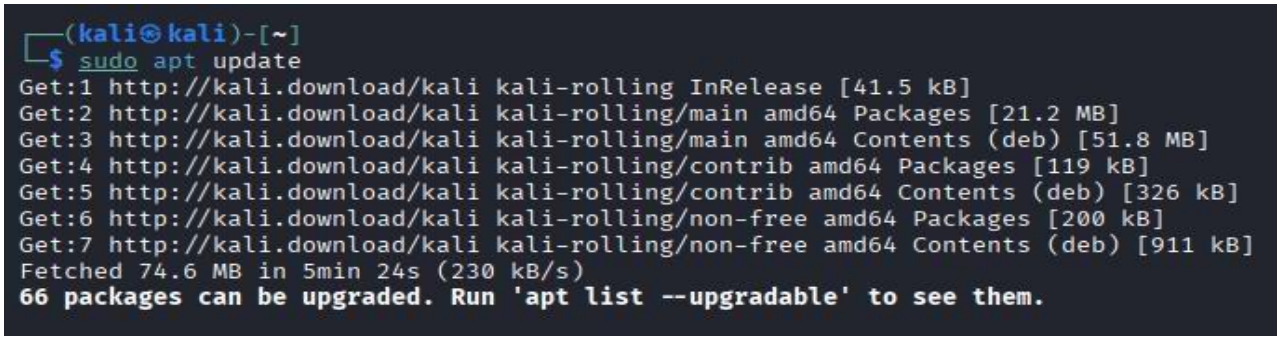
Scroll down and go on the Remote Logging Options.
This screenshot shows the **Remote Logging** settings in pfSense. This feature is used to send your firewall's log files to another computer. in this case, our Kali Linux machine.We do this for **analysis**, the ELK stack on your Kali machine can now easily read and organize these logs.
I first selected Everything from **remote syslog server** but later then I checked **system events** and **firewall events** as we only need those for our project.



Click on the Save Button Below and apply the changes. The changes have been applied successfully as shown in the snapshot.



Update the list of all available software to prevent any sort of errors during installation process.

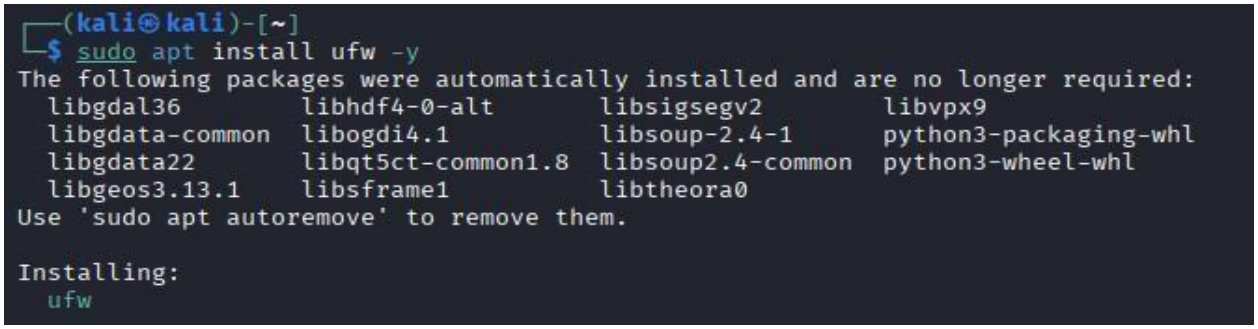


Installing the Firewall:

The command sudo apt install ufw -y installs the **Uncomplicated Firewall (ufw)** on your Kali Linux machine. This is a simple but powerful firewall that will be used to control what traffic is allowed to enter your system. Installing it is a necessary step before you can configure it.

Command:

sudo apt install ufw -y



Enabling the Firewall:

The final command, sudo ufw enable, turns on the firewall. The output confirms that it is now active and will automatically start up every time you turn on the machine. This ensures that the firewall rules you created (like allowing logs on port 5000) are working and protecting your system.

Command:

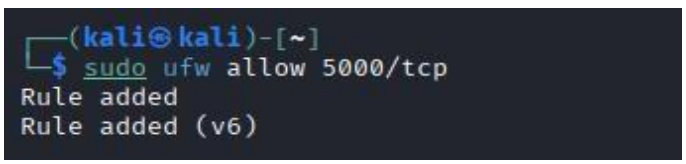
sudo ufw enable

Allowing Log Traffic:

The command sudo ufw allow 5000/tcp creates a rule to allow incoming traffic on TCP port 5000. This is a very important step because it opens the door for the pfSense firewall to send its log data to your Logstash container. Without this rule, the data would be blocked by your firewall.

Command:

sudo ufw allow 5000/tcp



Confirming connections before defining rules:

- Internal system to external network

Command:

ping -4 google.com

```
┌──(kali㉿kali)-[~]
└─$ ping -4 google.com
PING google.com (172.217.21.14) 56(84) bytes of data.
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=1 ttl=254 time=152 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=2 ttl=254 time=314 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=3 ttl=254 time=203 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=4 ttl=254 time=225 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=5 ttl=254 time=250 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=6 ttl=254 time=193 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=7 ttl=254 time=273 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=8 ttl=254 time=259 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=9 ttl=254 time=279 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=10 ttl=254 time=195 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=11 ttl=254 time=146 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=12 ttl=254 time=151 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=13 ttl=254 time=163 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=14 ttl=254 time=233 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=15 ttl=254 time=153 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=16 ttl=254 time=141 ms
^C64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=17 ttl=254 time=274 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=18 ttl=254 time=184 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=19 ttl=254 time=184 ms
^C
── google.com ping statistics ──
19 packets transmitted, 19 received, 0% packet loss, time 24111ms
rtt min/avg/max/mdev = 141.385/209.074/314.079/51.934 ms
```

- Internal network to pfsense (192.168.1.1)

Command:

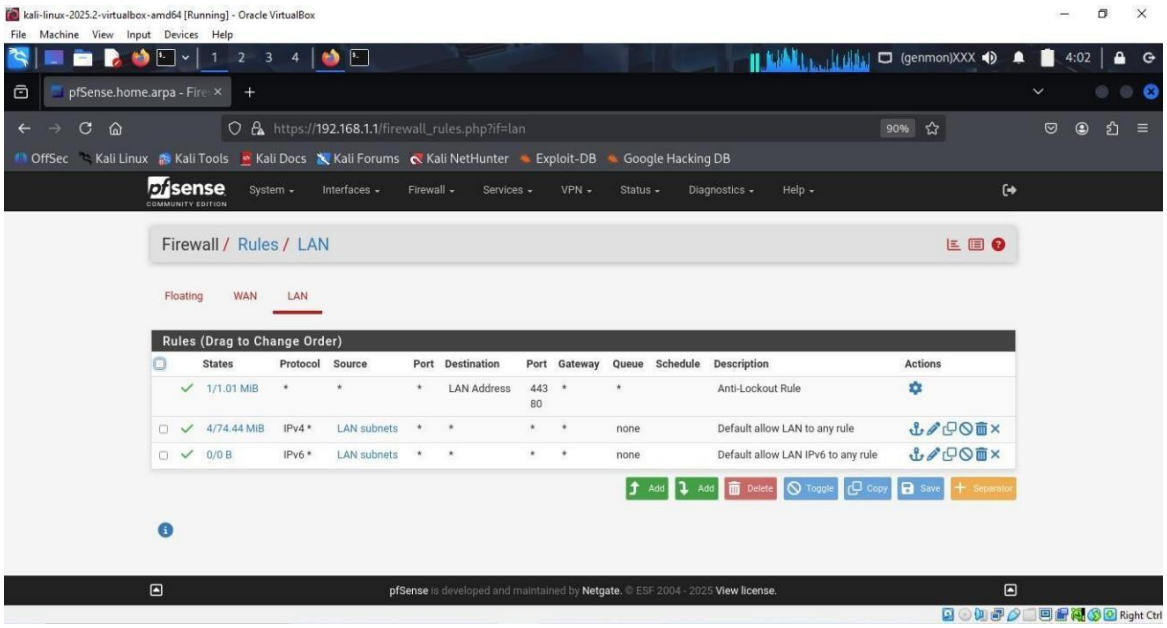
ping 192.168.1.1

```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=2.14 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=7.67 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=40.7 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=24.1 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=11.8 ms
64 bytes from 192.168.1.1: icmp_seq=6 ttl=64 time=5.95 ms
64 bytes from 192.168.1.1: icmp_seq=7 ttl=64 time=10.4 ms
64 bytes from 192.168.1.1: icmp_seq=8 ttl=64 time=11.2 ms
64 bytes from 192.168.1.1: icmp_seq=9 ttl=64 time=9.09 ms
64 bytes from 192.168.1.1: icmp_seq=10 ttl=64 time=30.3 ms
64 bytes from 192.168.1.1: icmp_seq=11 ttl=64 time=48.9 ms
64 bytes from 192.168.1.1: icmp_seq=12 ttl=64 time=19.6 ms
64 bytes from 192.168.1.1: icmp_seq=13 ttl=64 time=13.4 ms
64 bytes from 192.168.1.1: icmp_seq=14 ttl=64 time=16.3 ms
64 bytes from 192.168.1.1: icmp_seq=15 ttl=64 time=7.77 ms
64 bytes from 192.168.1.1: icmp_seq=16 ttl=64 time=12.5 ms
^C
── 192.168.1.1 ping statistics ──
16 packets transmitted, 16 received, 0% packet loss, time 15923ms
rtt min/avg/max/mdev = 2.138/16.989/48.875/12.564 ms
```

Firewall >> Rules >> LAN

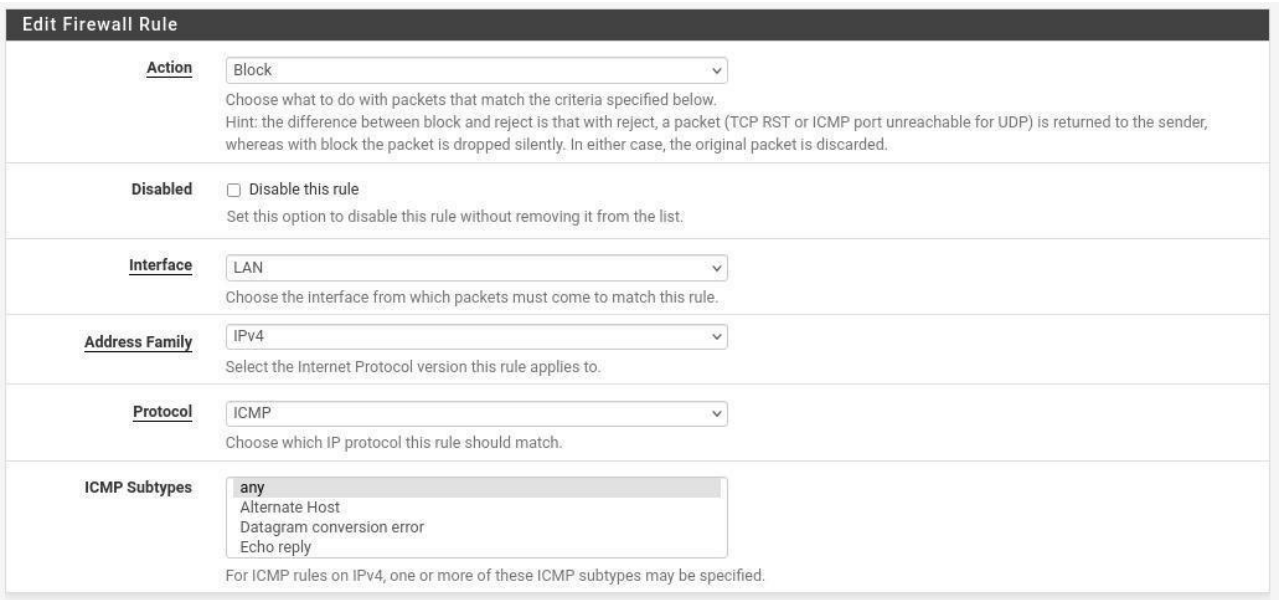
Creating a New Firewall Rule:

The first screenshot shows the main Firewall Rules page for the LAN interface. This is where you can manage all the rules that control what traffic is allowed or blocked on your internal network. By clicking on the **"Add"** button, you begin the process of creating a **new rule.**

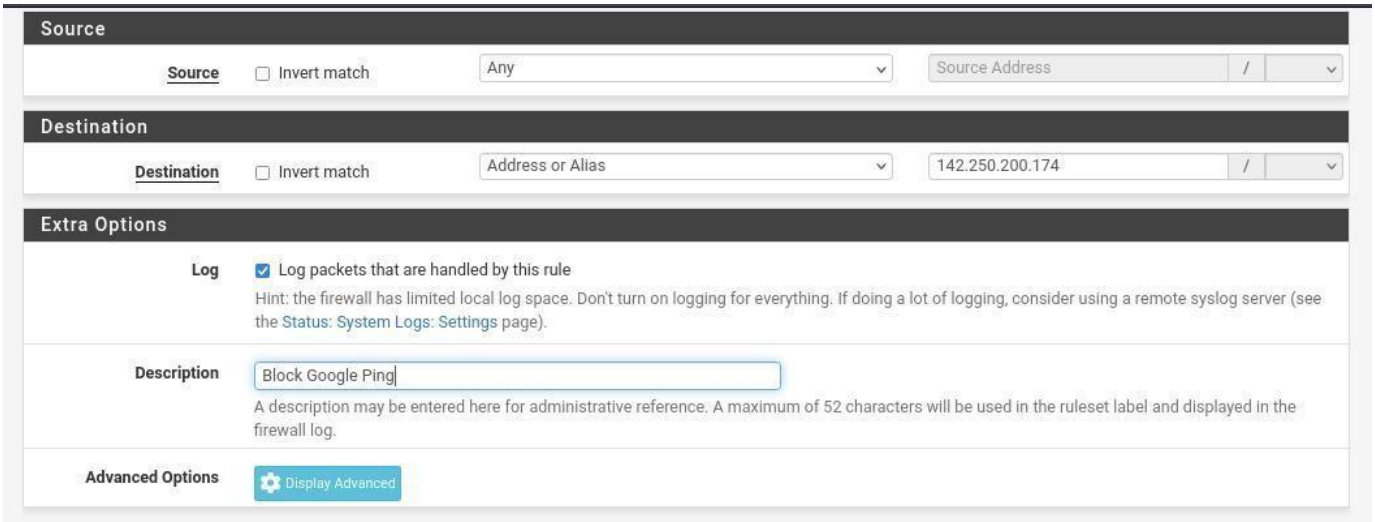


Defining the Rule Action and Protocol:

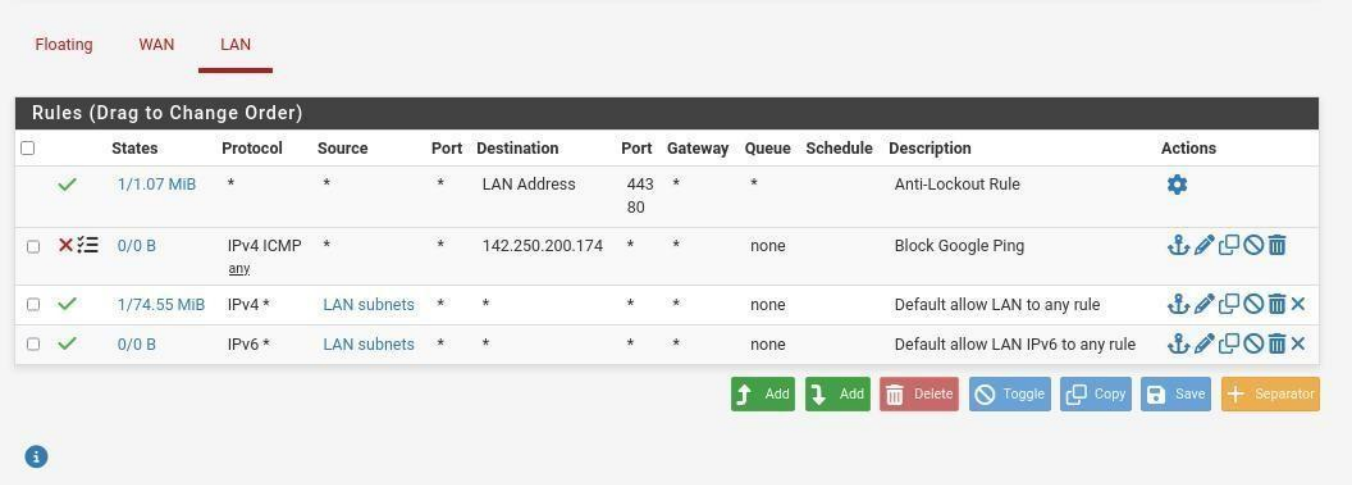
On the next screen, you define the new rule. The Action is set to **Block** because the goal is to stop certain traffic. The Protocol is set to **ICMP**, which is the specific protocol used by the **ping command**. This tells the firewall to look for and block any ping requests.



Setting the Rule's Source and Destination:
Here you define the Source and Destination of the traffic you want to block. The rule's Source is set to Any, meaning it will block traffic from any device on your network. The Destination is set to a specific IP address from Google. You also added a clear description, **"Block Google Ping"** so you know what the rule does.



Activating the New Rule:
The final screenshot shows that your new rule has been successfully added to the rule list. It is placed at the top of the list and has a red icon, confirming it is a Block rule. The placement is important because the firewall processes rules from top to bottom. It will see the "Block Google Ping" rule first and apply it, preventing any ping requests from reaching Google before it sees the default "allow all" rule.

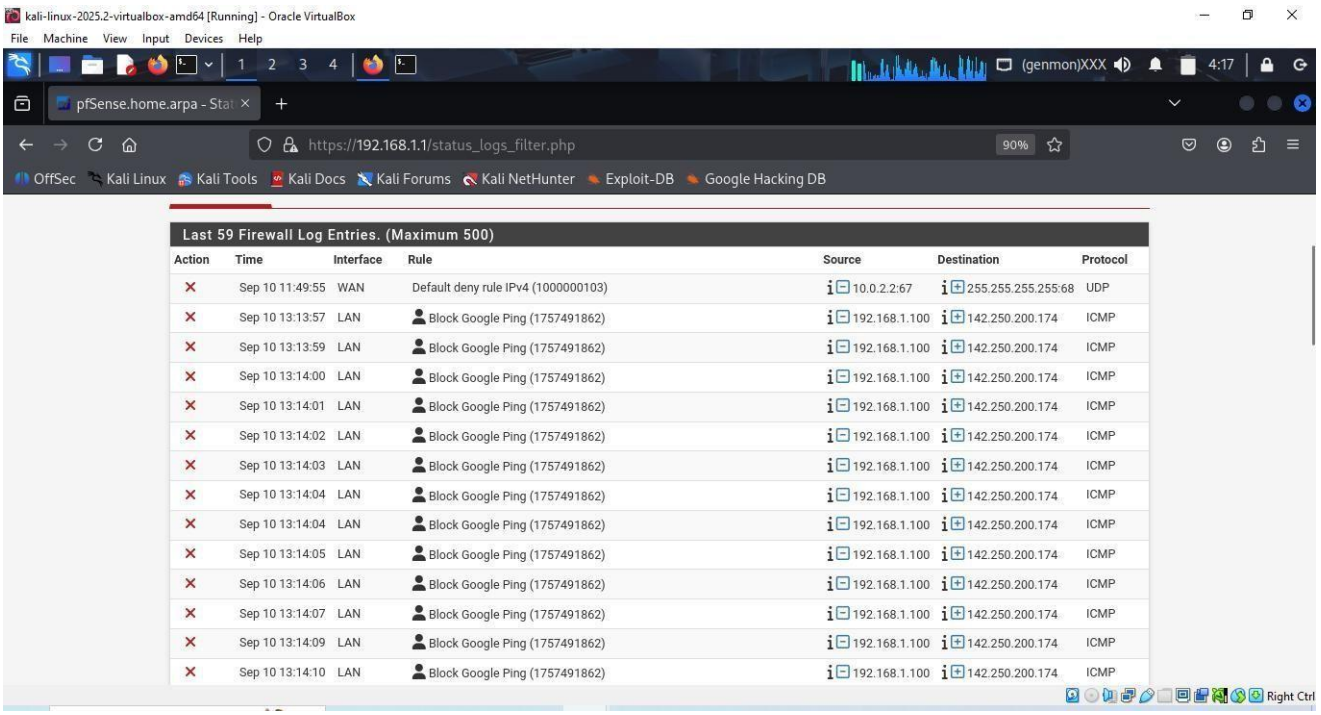


Result of the Ping Test:
The screenshot shows that your ping google.com command failed completely, with **100% packet loss**. This is the expected result and proves that the firewall rule you created is working **correctly.** The packets sent from your Kali Linux machine were blocked and did not reach Google's servers.

```
┌──(kali㉿kali)-[~]
└─$ ping google.com
PING google.com (142.250.200.174) 56(84) bytes of data.

^C
── google.com ping statistics ──
58 packets transmitted, 0 received, 100% packet loss, time 59705ms
```

Status>> system logs >>Firewalls tab
The screenshot confirms this from the firewall's perspective. The pfSense firewall logs show multiple entries with the description **"Block Google Ping"**. The logs confirm that the ICMP packets sent from your Kali Linux machine (**192.168.1.100**) were blocked from reaching Google's IP address. This provides solid evidence that your firewall rule is successfully enforcing your network policy.

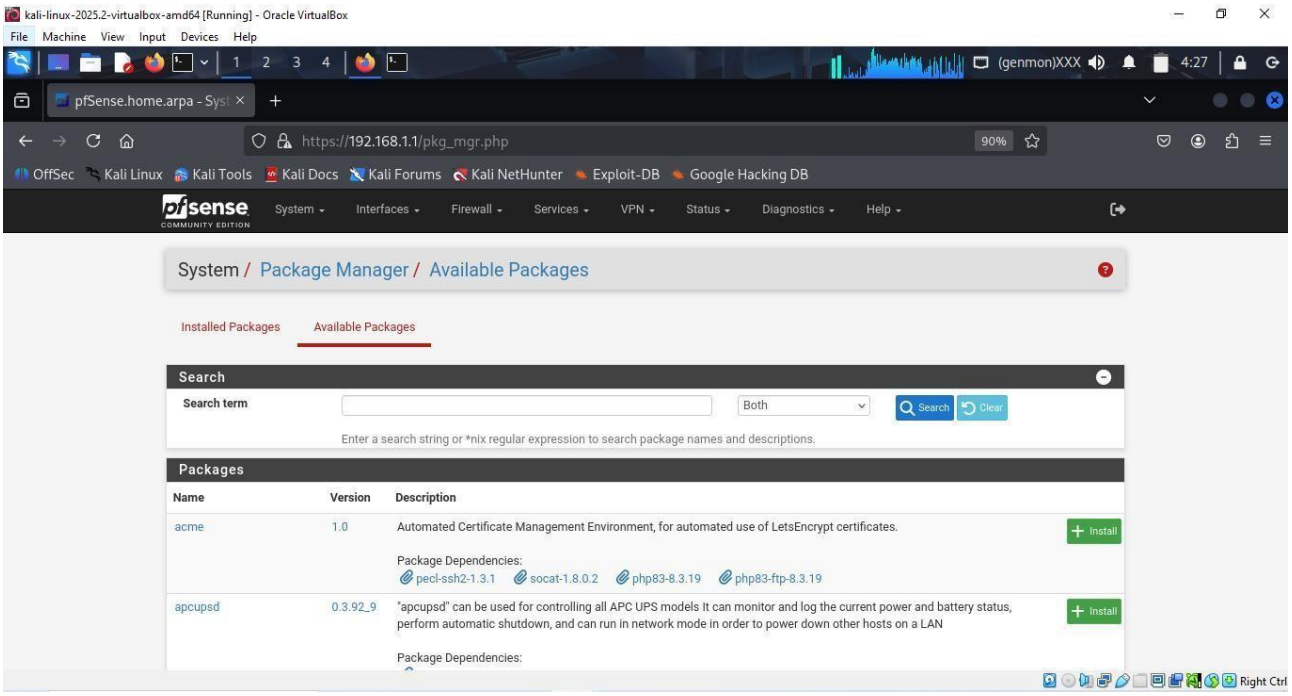


Section 5: Intrusion Detection System (Snort) Integration

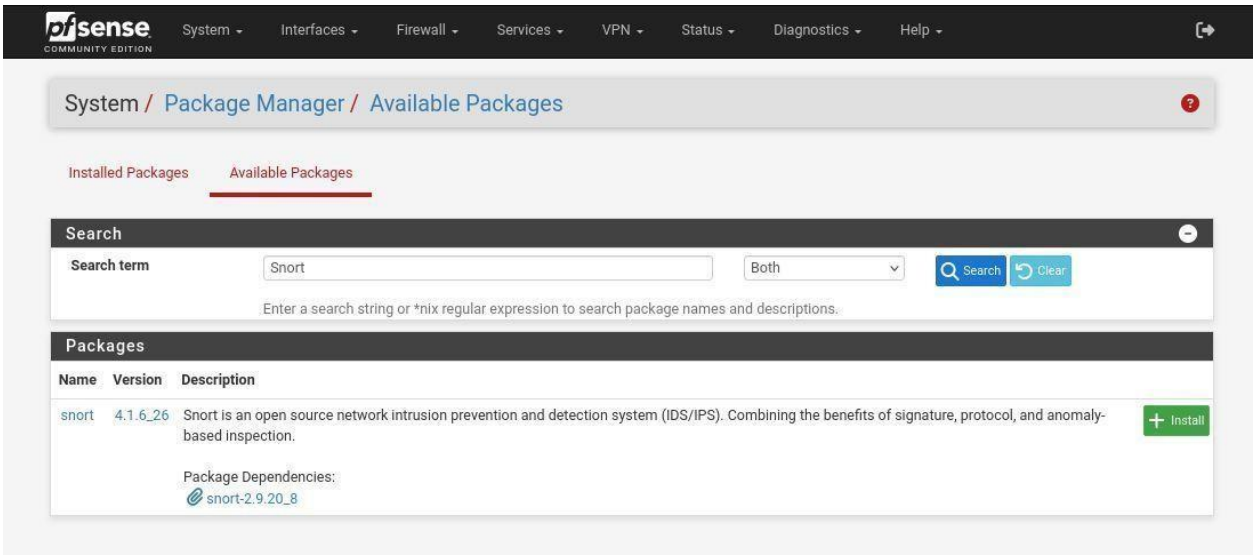
5.1 : Snort Installation

Your first step is to go to the pfSense **Package Manager** and search for **Snort** in the **"Available Packages"** section. Snort is an open-source network intrusion prevention and detection system (IDS/IPS) that will add a powerful layer of security to your firewall. Once you find it, you can click on the **+ Install** button to begin the installation process.

System >> Packet Manager>> Available Packages

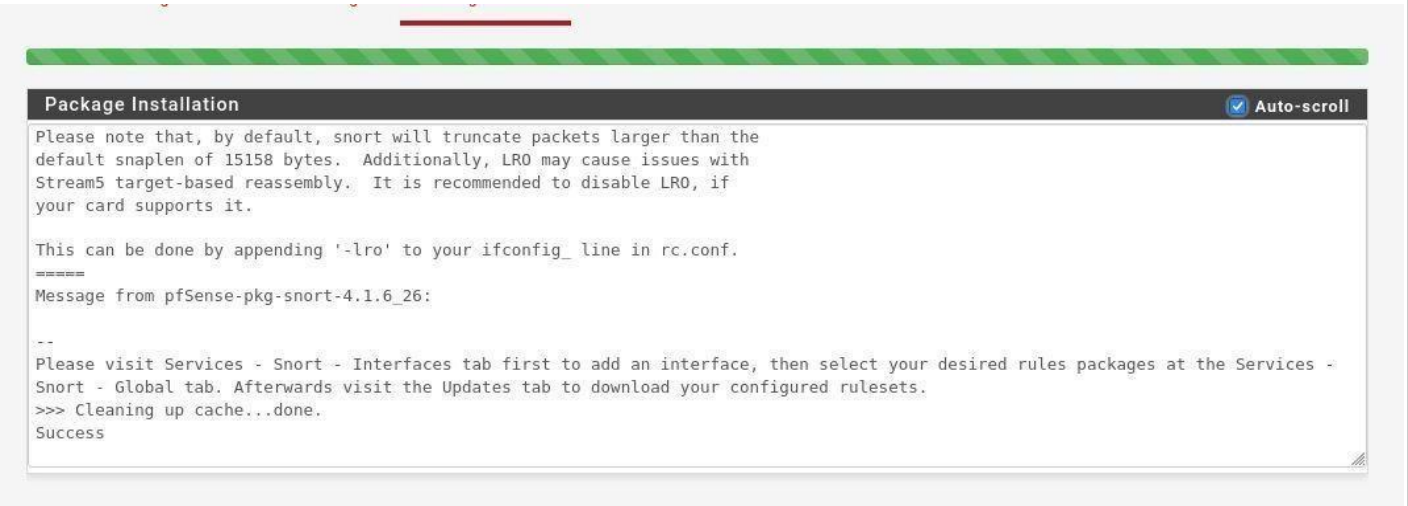


Search Snort and Install it.



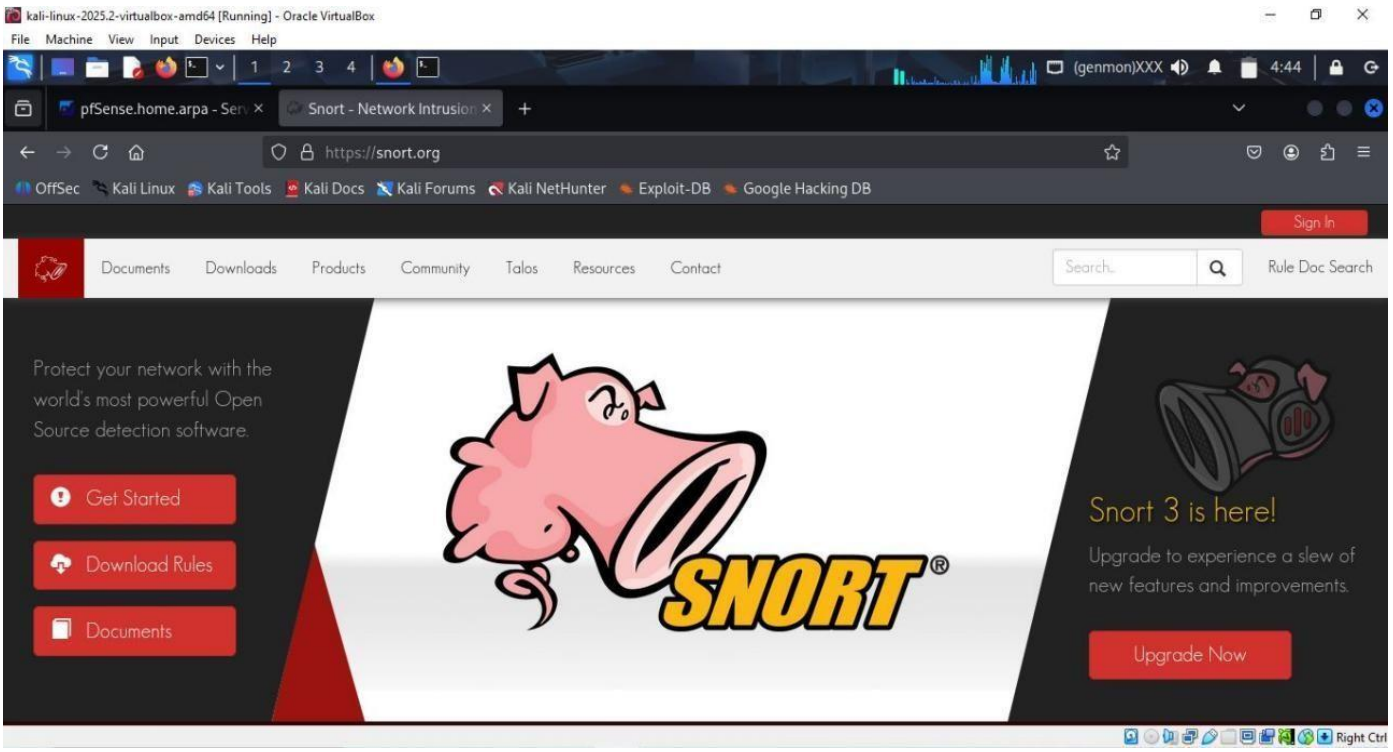
Confirm a successful installation:

After the installation is complete, a screen will show a **"Success"** message. This confirms that Snort has been correctly installed on your pfSense firewall. The message will also provide instructions for the next steps, which involve configuring Snort to start protecting your network.

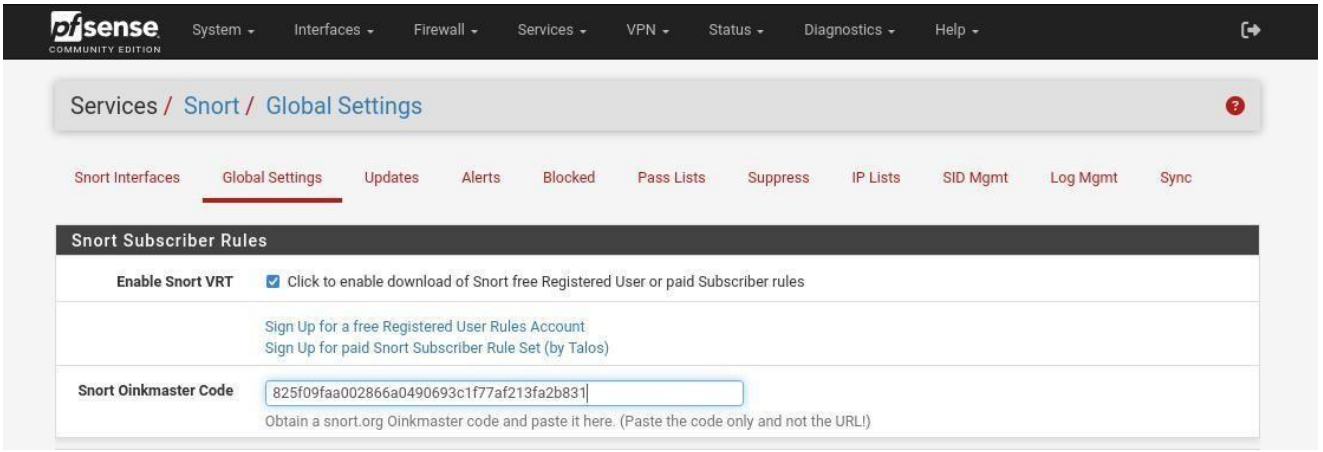


5.2 : Rule Download and Configuration

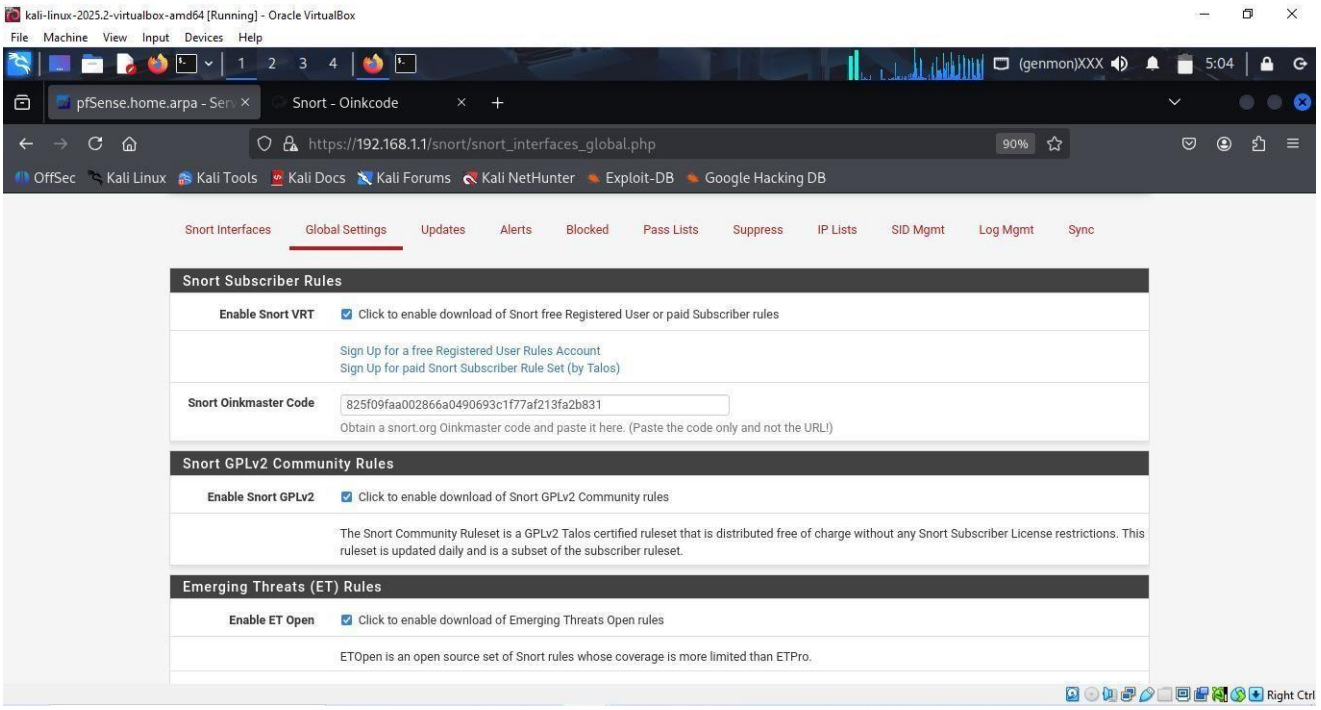
Your next step is to create a free account on **Snort.org**. The reason for this is to get a special code called an **Oinkcode**. This code is essential because it allows Snort to download the latest security rules and updates. Without these rules, Snort cannot effectively detect new threats and protect your network.



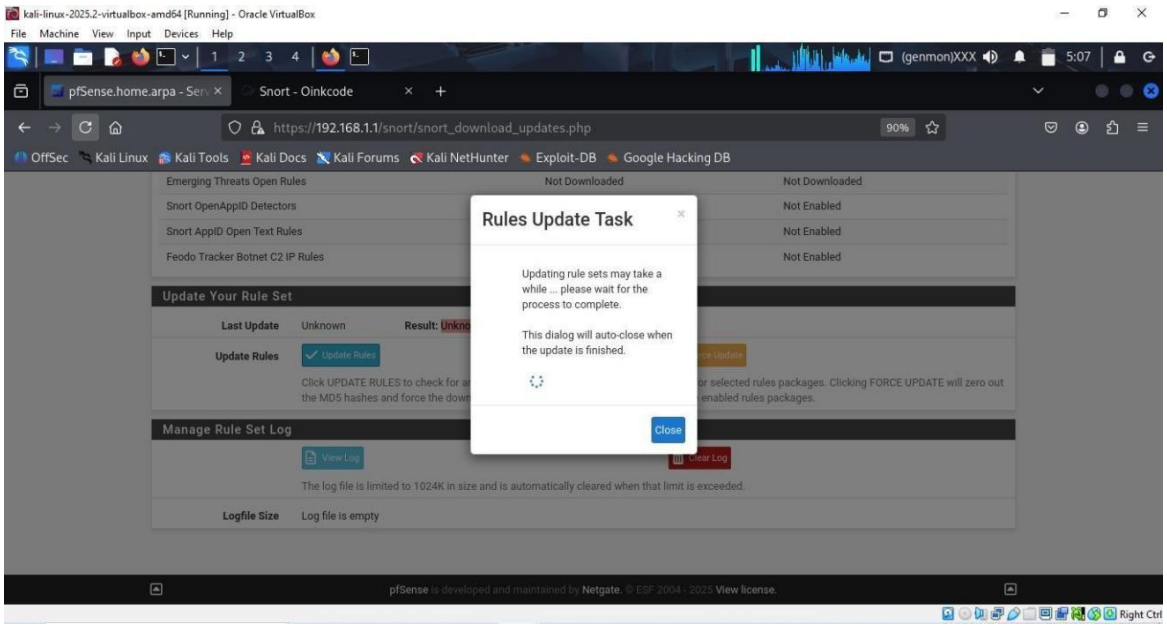
Services >> Snort >> Global settings



This screenshot shows the **Snort Global Settings** page, where you configure what rules Snort will use to protect your network. By enabling the different rule sets, you are telling Snort where to get its threat definitions from. The **Oinkmaster Code** lets you download subscriber rules, and by enabling the **Community Rules**, you can ensure Snort stays up to date on new threats, making your firewall more secure.

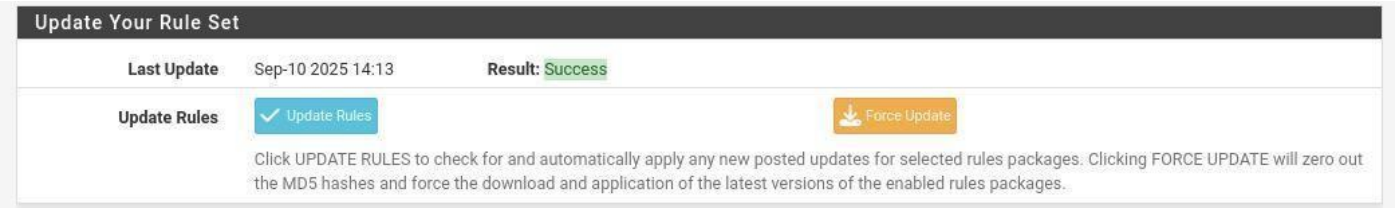


Updating Rulesets:
After enabling the rule sets, a dialog box will appear to show you that the **update process** is in progress. The firewall is now connecting to the Snort servers to download the latest threat rules using your Oinkmaster Code. This may take a few moments to complete, depending on your internet speed.



Confirming a Successful Update:

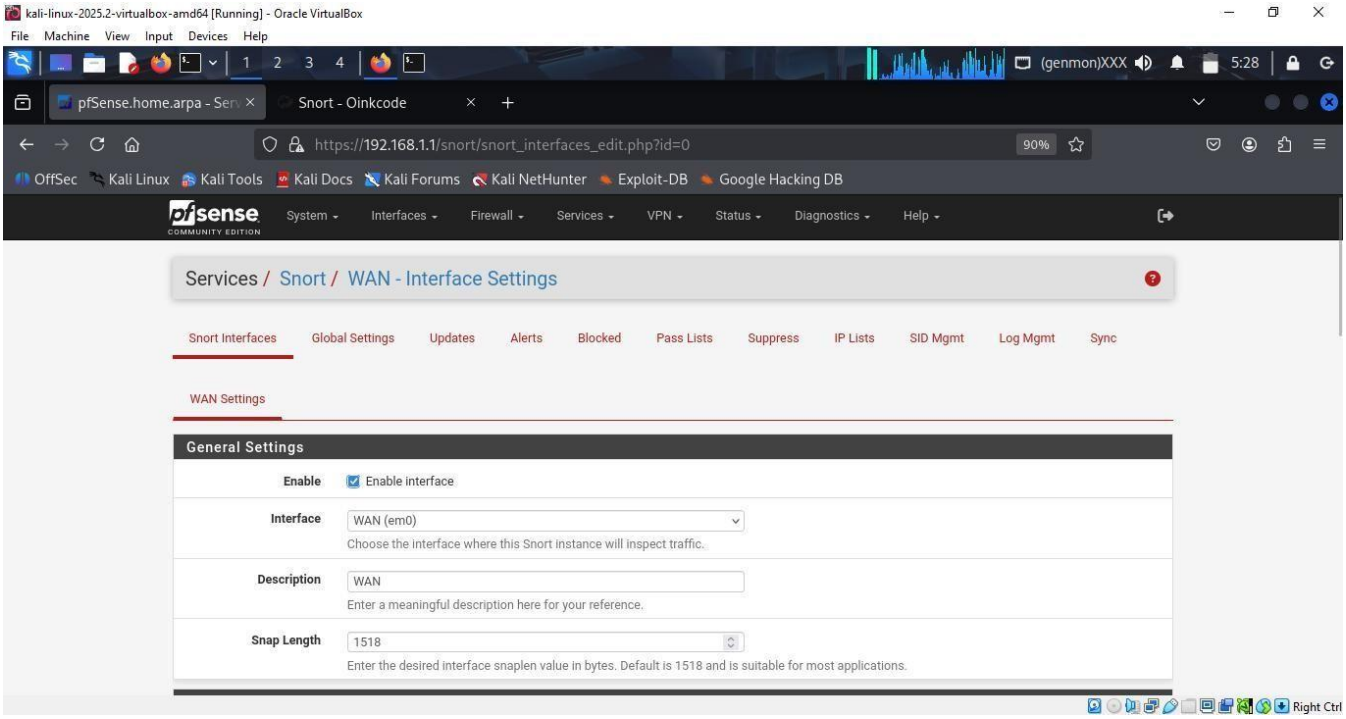
The final screenshot confirms that the rules update was a success. You can see the date of the last update and the **"Result: Success"** message. This means Snort has successfully downloaded and applied the latest threat rules, and it is now ready to detect security threats on your network using the most current information.



5.3 Enabling Snort on WAN and LAN Interfaces

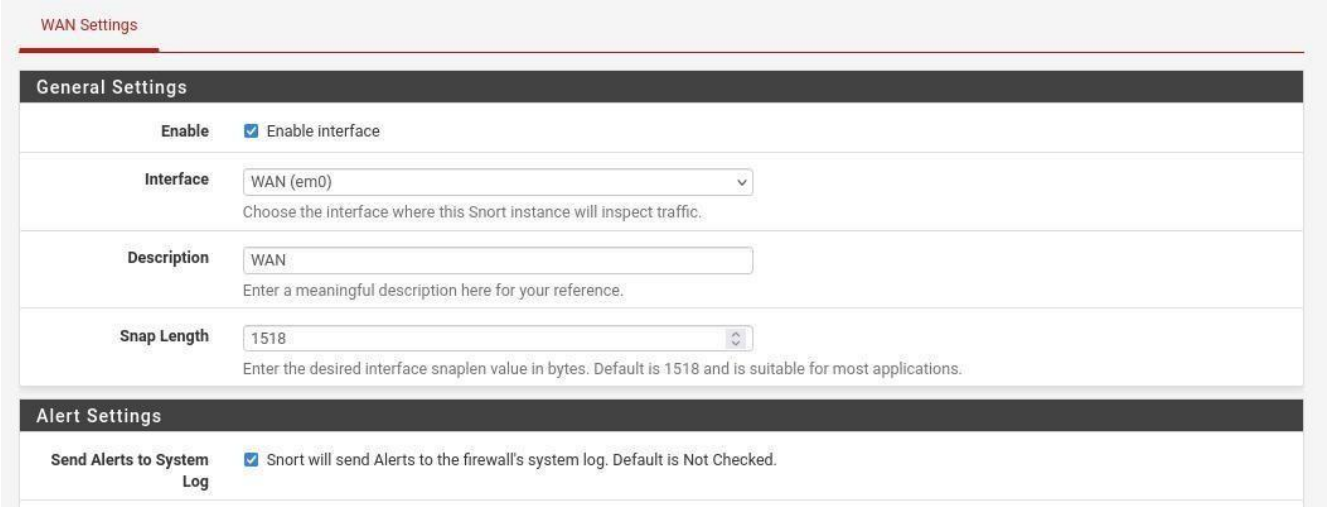
Services >> Snort >> Interface settings

Enable the interface for WAN (wide area network) as WAN (em0). Save the settings so you can proceed further with the LAN Interface for Snort.



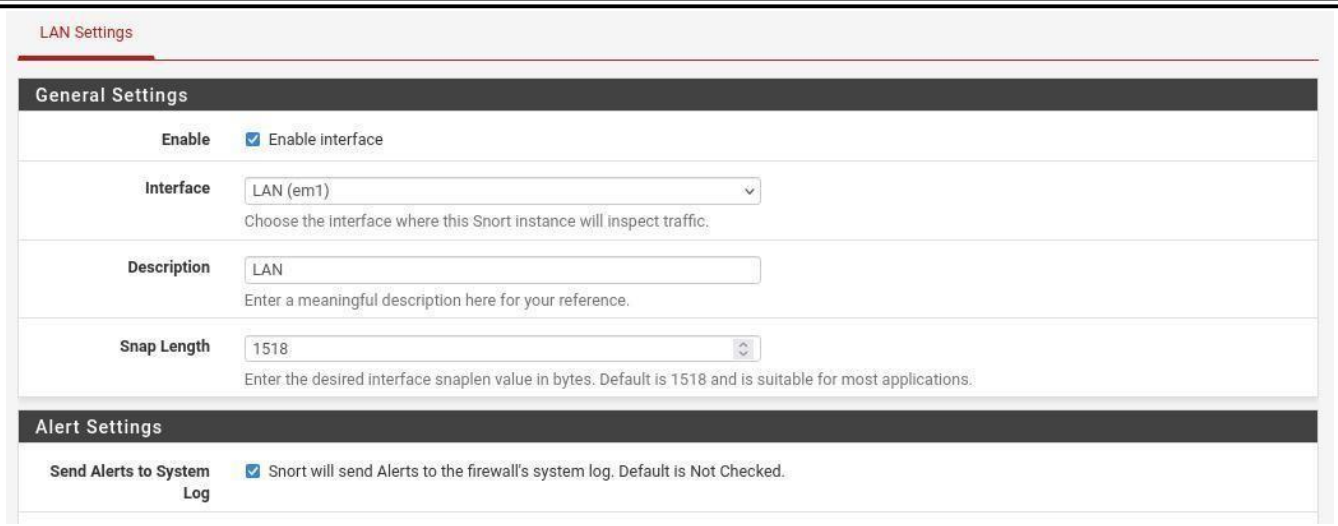
Enabling Snort on the WAN Interface:

You are enabling Snort on the WAN (internet) interface. This is very important because it allows Snort to check for attacks coming from the internet. If you do not enable Snort on your WAN, your firewall will not be able to detect or stop any attacks before they enter your network.



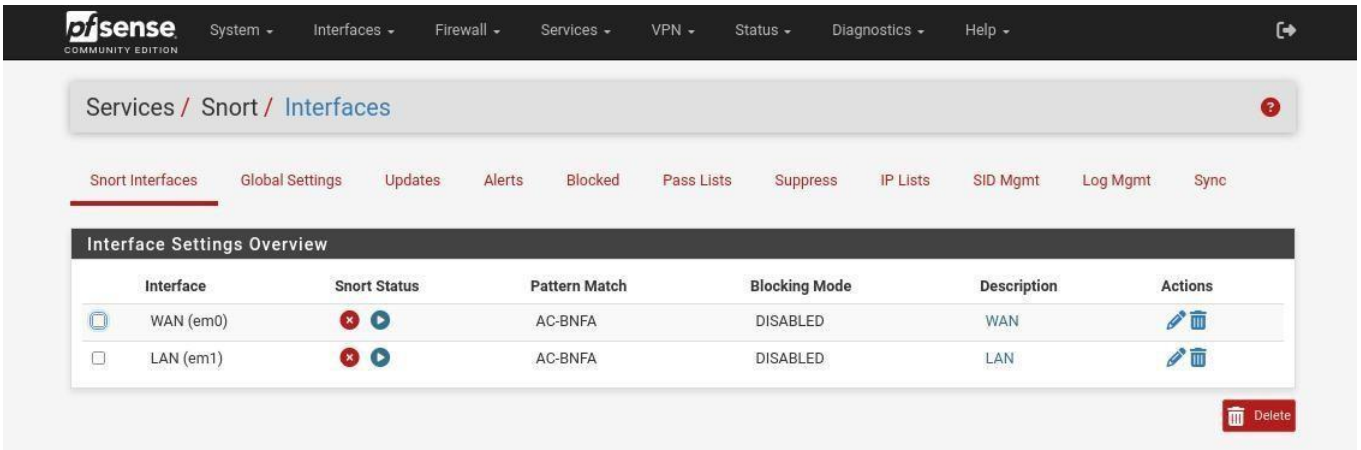
Enabling Snort on the LAN Interface:

Here, you are enabling Snort on your LAN (internal) interface. We do this to look for threats that might already be inside your network. If you do not enable Snort on the LAN, any suspicious activity originating from within your network will go undetected.

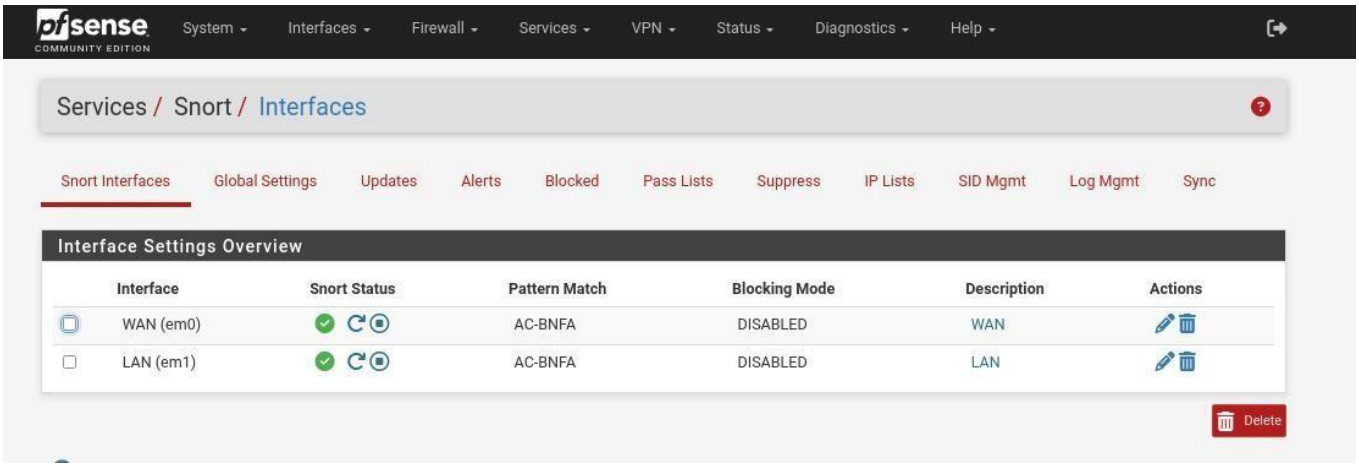


Snort Interfaces Overview:

This is the Snort Interfaces page, which shows that Snort is currently not active on your WAN or LAN connections. The reason you need to go here is to tell Snort where to look for network traffic. If you don't enable it on these interfaces, Snort will not inspect any traffic, and it will not be able to detect any threats.



By clicking the enable option (x) snort status would be enable.



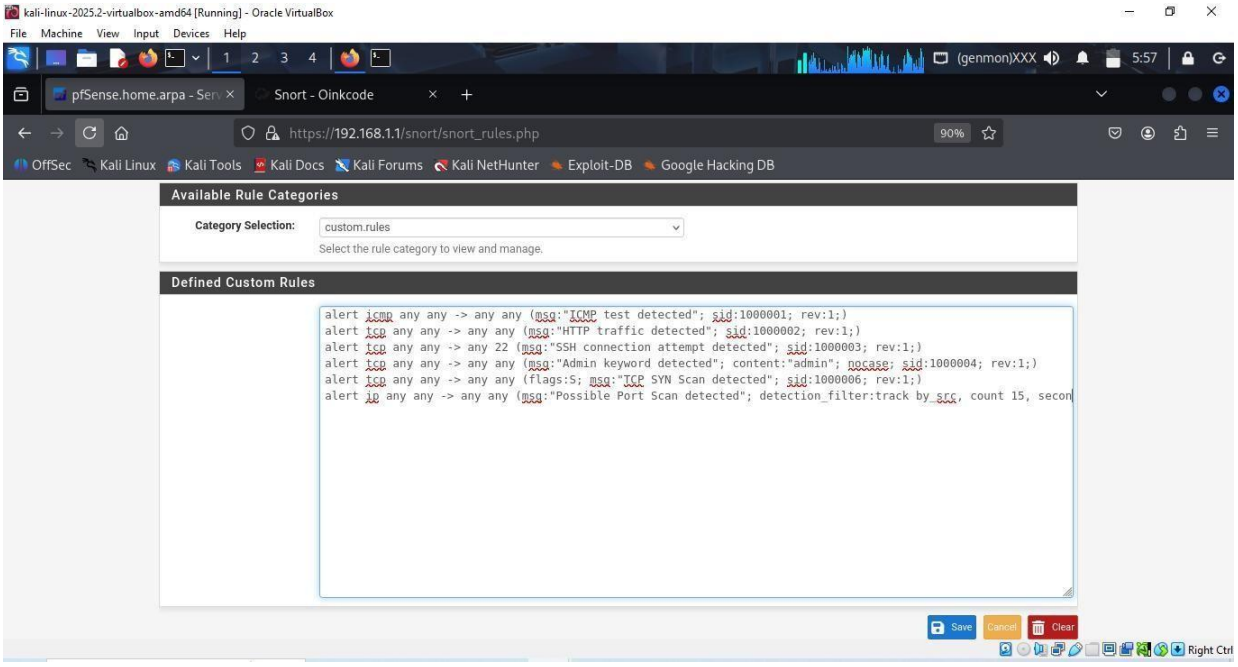
5.4 : Alert Generation and Verification

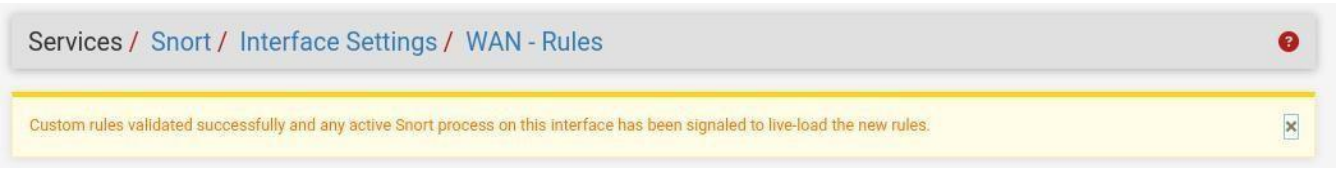
Services >> Snort >> Interface Settings >> WAN Rules

Snort Rules Configuration:

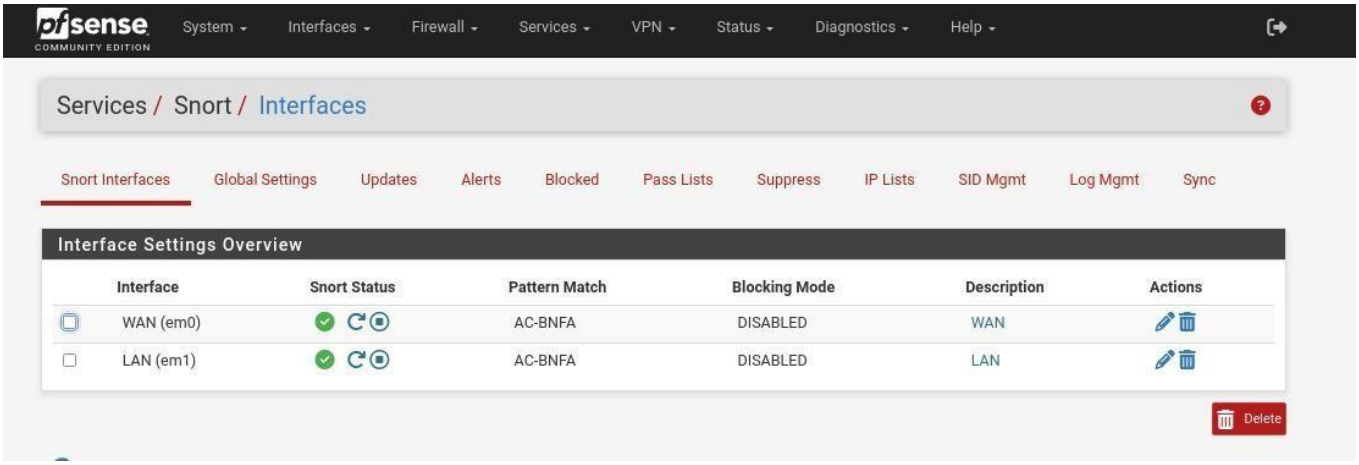
alert icmp any any -> any any (msg:"ICMP test detected"; sid:1000001; rev:1;)
alert tcp any any -> any any (msg:"HTTP traffic detected"; sid:1000002; rev:1;)
alert tcp any any -> any 22 (msg:"SSH connection attempt detected"; sid:1000003; rev:1;)
alert tcp any any -> any any (msg:"Admin keyword detected"; content:"admin"; nocase; sid:1000004; rev:1;)
alert tcp any any -> any any (flags:S; msg:"TCP SYN Scan detected"; sid:1000006; rev:1;)
alert ip any any -> any any (msg:"Possible Port Scan detected"; detection_filter:track by_src, count 15, seconds 60; sid:1000005; rev:1;)

This screenshot shows the **Custom Rules** section of your **Snort** configuration. On this page, you can create your own specific security rules to tailor your firewall's threat detection. The examples shown are custom rules designed to **detect and alert on specific events**, such as an **ICMP ping**, an **SSH connection attempt**, or a possible **port scan**. This is a very important feature because it allows you to create rules to detect threats or activities that are specific to your own network, giving you more control over your security.





To make your new custom rules active, you need to **restart** Snort on both the WAN and LAN interfaces. Go to **Services / Snort / Interfaces** and click on the restart icon for both the **WAN** and **LAN** interfaces. This will force Snort to reload with your new rules, preparing you to test if they are working.



Test Snort Configuration bt viewing alerts

Scanning for Open Ports:

The nmap command is used here to scan your pfSense firewall for open ports. The scan was successful and found three open ports: **port 53 (DNS)**, **port 80 (HTTP)**, and **port 443 (HTTPS)**. This is a normal and expected result, as these services are needed for the firewall to handle web-based management and network traffic.

Command:

nmap -Pn -p 1-1000 192.168.1.1

```
┌──(kali㉿kali)-[~]
└─$ nmap -Pn -p 1-1000 192.168.1.1
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-10 06:04 EDT
Nmap scan report for pfSense.home.arpa (192.168.1.1)
Host is up (0.0072s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE
53/tcp  open  domain
80/tcp  open  http
443/tcp open  https
MAC Address: 08:00:27:D6:70:7A (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 7.47 seconds
```

Connectivity Test:

Snort will detect and alert on this activity because you created a specific **custom rule** for **ICMP** traffic. This rule tells Snort to look for and log any **ping** attempts, which allows you to monitor that specific type of network traffic.

Command:

ping google.com

```
┌──(kali㉿kali)-[~]
└─$ ping google.com
PING google.com (172.217.21.14) 56(84) bytes of data.
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=1 ttl=254 time=258 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=2 ttl=254 time=292 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=3 ttl=254 time=234 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=4 ttl=254 time=242 ms
64 bytes from fra07s29-in-f14.1e100.net (172.217.21.14): icmp_seq=5 ttl=254 time=150 ms
^C
── google.com ping statistics ──
5 packets transmitted, 5 received, 0% packet loss, time 4099ms
rtt min/avg/max/mdev = 150.475/235.432/292.225/46.894 ms
```

SSH Connection Attempt:

The command ssh -p 22 192.168.1.1 is an attempt to connect to your pfSense firewall using SSH (Secure Shell) on port 22. This connection would allow you to manage the firewall's console from the Kali Linux terminal. The attempt was canceled, so no connection was established.

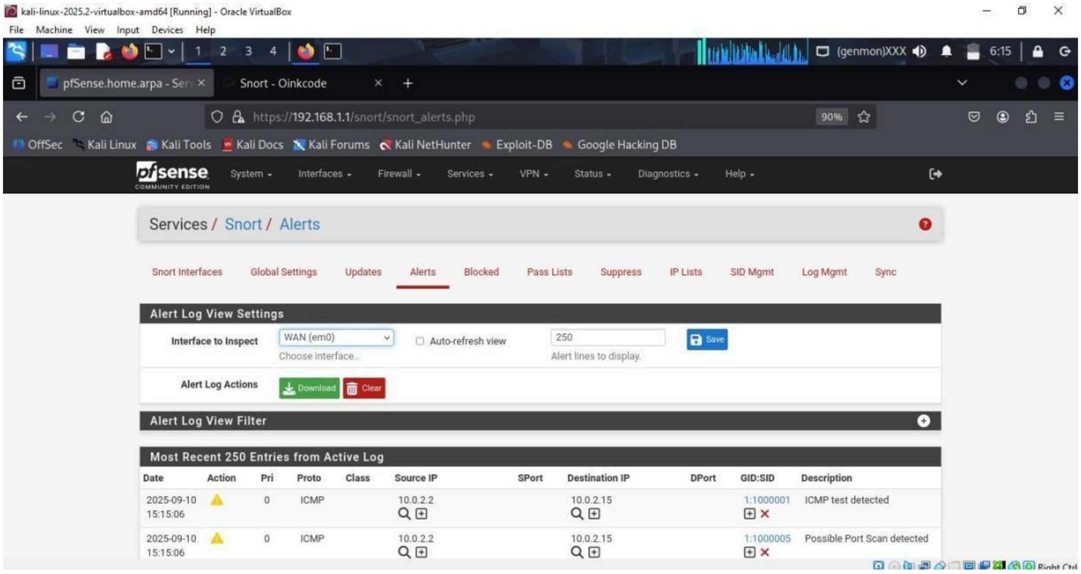
Command:

ssh -p 22 192.168.1.1

```
┌──(kali㉿kali)-[~]
└─$ ssh -p 22 192.168.1.1

^C
```

These screenshots show the **Snort Alerts** page, which is the most important part of the Snort configuration. This page confirms that your firewall is now actively detecting threats on your network. The logs show that your custom **"ICMP test detected"** rule is working as intended. More importantly, Snort's built-in rules are also active, as you can see alerts for a **"Possible Port Scan detected."** This proves that your Snort installation is a success and your firewall is now intelligently monitoring your network for threats.



Alert Log View Filter

Most Recent 250 Entries from Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2025-09-10 15:15:06	⚠	0	ICMP		10.0.2.2		10.0.2.15		1:1000001	ICMP test detected
2025-09-10 15:15:06	⚠	0	ICMP		10.0.2.2		10.0.2.15		1:1000005	Possible Port Scan detected
2025-09-10 15:15:06	⚠	0	ICMP		10.0.2.15		10.0.2.2		1:1000001	ICMP test detected
2025-09-10 15:15:06	⚠	0	ICMP		10.0.2.15		10.0.2.2		1:1000005	Possible Port Scan detected
2025-09-10 15:15:06	⚠	0			fe80::2		fe80::a00:27ff:fe1f:6ace		1:1000001	ICMP test detected
2025-09-10 15:15:06	⚠	0			fe80::2		fe80::a00:27ff:fe1f:6ace		1:1000005	Possible Port Scan detected
2025-09-10 15:15:06	⚠	0			fe80::a00:27ff:fe1f:6ace		fe80::2		1:1000001	ICMP test detected
2025-09-10 15:15:06	⚠	0			fe80::a00:27ff:fe1f:6ace		fe80::2		1:1000005	Possible Port Scan detected

Last 500 General Log Entries. (Maximum 500)

Time	Process	PID	Message
Sep 12 14:41:44	snort	72985	[1:1000005:1] Possible Port Scan detected {TCP} 142.250.201.59:443 -> 10.0.2.15:1230
Sep 12 14:41:44	snort	72985	[1:1000002:1] HTTP traffic detected {TCP} 142.250.201.59:443 -> 10.0.2.15:1230
Sep 12 14:41:44	snort	72985	[1:1000005:1] Possible Port Scan detected {TCP} 142.250.201.59:443 -> 10.0.2.15:1230
Sep 12 14:41:44	snort	72985	[1:1000002:1] HTTP traffic detected {TCP} 142.250.201.59:443 -> 10.0.2.15:1230
Sep 12 14:41:44	snort	72985	[1:1000005:1] Possible Port Scan detected {TCP} 142.250.201.59:443 -> 10.0.2.15:1230
Sep 12 14:41:44	snort	72985	[1:1000002:1] HTTP traffic detected {TCP} 142.250.201.59:443 -> 10.0.2.15:1230
Sep 12 14:41:44	snort	72985	[1:1000005:1] Possible Port Scan detected {TCP} 142.250.201.59:443 -> 10.0.2.15:1230
Sep 12 14:41:44	snort	72985	[1:1000002:1] HTTP traffic detected {TCP} 142.250.201.59:443 -> 10.0.2.15:1230
Sep 12 14:41:44	snort	72985	[1:1000005:1] Possible Port Scan detected {TCP} 142.250.201.59:443 -> 10.0.2.15:1230
Sep 12 14:41:44	snort	72985	[1:1000002:1] HTTP traffic detected {TCP} 142.250.201.59:443 -> 10.0.2.15:1230
Sep 12 14:41:44	snort	72985	[1:1000005:1] Possible Port Scan detected {TCP} 142.250.201.59:443 -> 10.0.2.15:1230
Sep 12 14:41:44	snort	72985	[1:1000002:1] HTTP traffic detected {TCP} 142.250.201.59:443 -> 10.0.2.15:1230
Sep 12 14:41:44	snort	72985	[1:1000005:1] Possible Port Scan detected {TCP} 142.250.201.59:443 -> 10.0.2.15:1230

Section 6: Data Analysis and Visualization

6.1 : Accessing the Kibana Dashboard

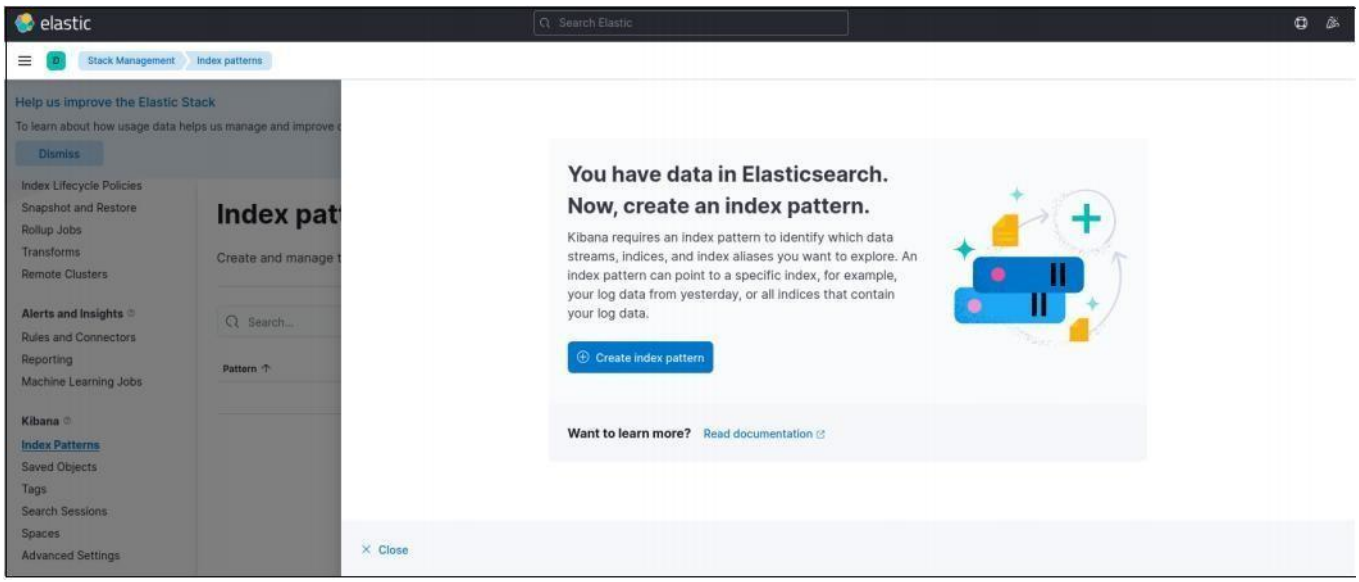
Kibana is a user interface that lets you visualize your Elasticsearch data and navigate the Elastic Stack. We are doing this to access and view logs.First we go to access the kibana dashboard: http://localhost:5601.
As told earlier as well while setting up the containers.

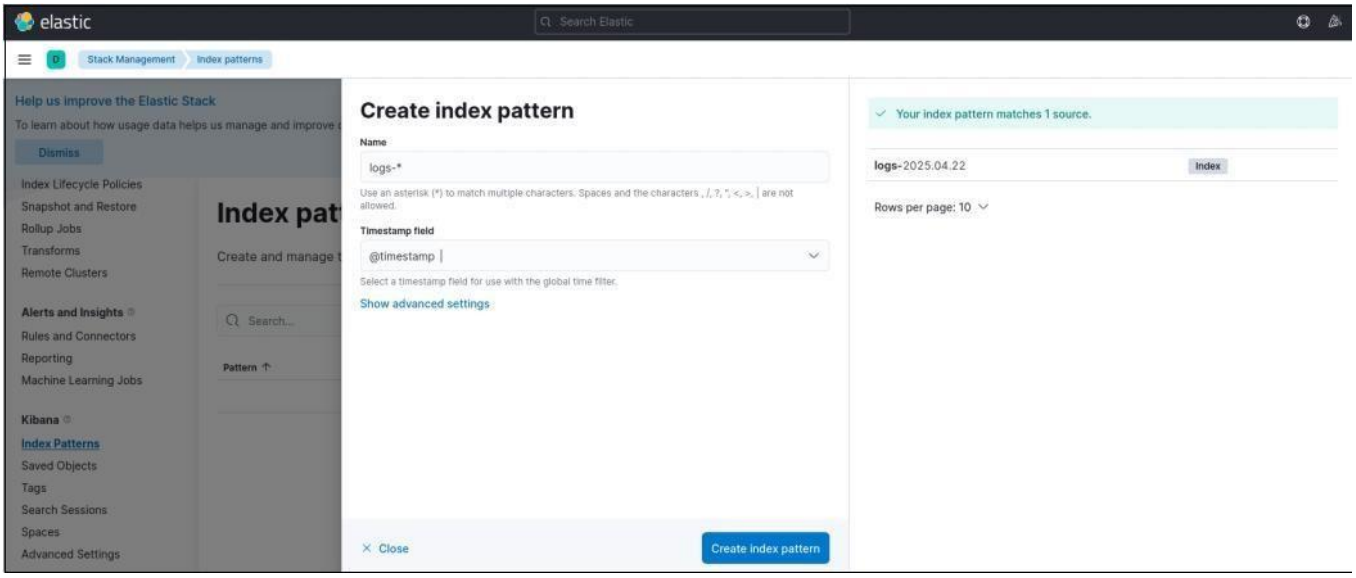
6.2 : Index Pattern Creation and Log Viewing

An **index pattern** is like a blueprint or a filter that tells Kibana which data to look at inside Elasticsearch. It's not the data itself, but rather a way to organize and group your data so that it can be easily accessed and analyzed.

"Stack Management" → "Index Patterns"

This screen shows that data has successfully arrived in **Elasticsearch** from your Logstash container. Now, you need to create an **index pattern** in **Kibana** to make that data visible and ready for analysis. Think of an index pattern as a way to tell Kibana which **"folders"** of data it should look at. For example, if all your logs are saved in folders named like **logs-2025.09.13**, an index pattern like logs-* will allow Kibana to see all of them together.





Now go to the discover tab and see that our logs will appear.

Breakdown of the Log Entry:

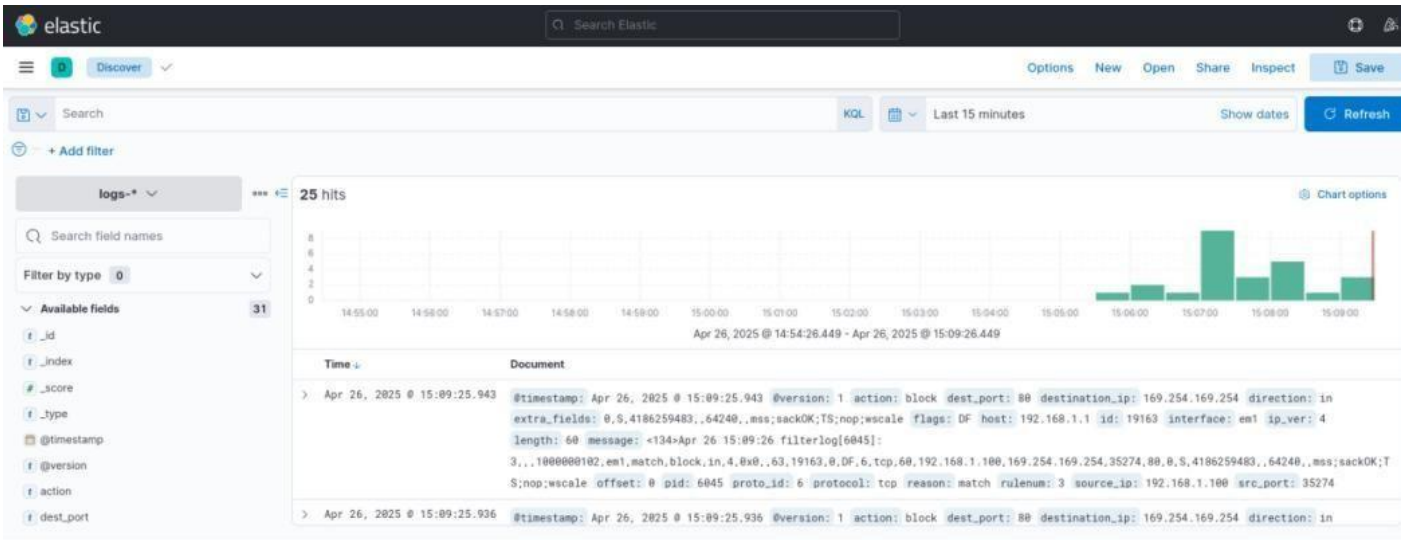
The information in each log entry provides a clear picture of a network event. For the example shown in the screenshot, we can identify the following:

Action: The action: block confirms that the firewall successfully blocked this network activity.

Source IP: The source: 192.168.1.100 tells you that the traffic came from your Kali Linux machine.

Destination IP: The destination: 169.254.169.254 shows the specific IP address that the Kali machine was trying to reach.

Protocol: The protocol: tcp indicates the type of network traffic that was blocked by your firewall.



Section 7: Project Conclusion / Outcomes

7.1 : Project Outcomes

1)- I successfully installed a **pfSense firewall** in a virtual machine and configured it to work as a router and security gateway for my network.

2)- I established a functional virtual network, allowing my **Kali Linux machine** to connect to the internet through the new firewall.

3)- I successfully created and tested a custom **firewall rule** to block a specific type of network traffic, proving my ability to manage network access.

4)- I installed and configured an **Intrusion Detection System (IDS)** named Snort to intelligently detect and log network threats.

5)- I set up the **ELK stack** to collect, store, and analyze all the log data from my firewall, giving me a complete overview of my network's activity.

6)- I successfully **troubleshot and fixed** a network connectivity issue to ensure my pfSense firewall had a working internet connection.

7)- I successfully **built and configured** a complete, end-to-end network security solution, including a firewall, an intrusion detection system (Snort), and a centralized logging platform (ELK).

7.2 : Future Enhancements

Following Future enhancements can be made:

Advanced Rule-making: Practice creating more complex firewall rules to manage different types of traffic and secure specific ports.

Automated Threat Blocking: Configure Snort to not just detect threats but also to automatically block them, turning your IDS into an Intrusion Prevention System (IPS).

Additional Services: Add services like a VPN to your pfSense firewall to secure remote access to your network.

User Management: Create and manage different user accounts for your firewall to practice access control and user authentication.

Real-time Dash-boarding: Use Kibana to create a custom dashboard with graphs and charts that give you a real-time, visual overview of your network's security status.

----------THE END----------