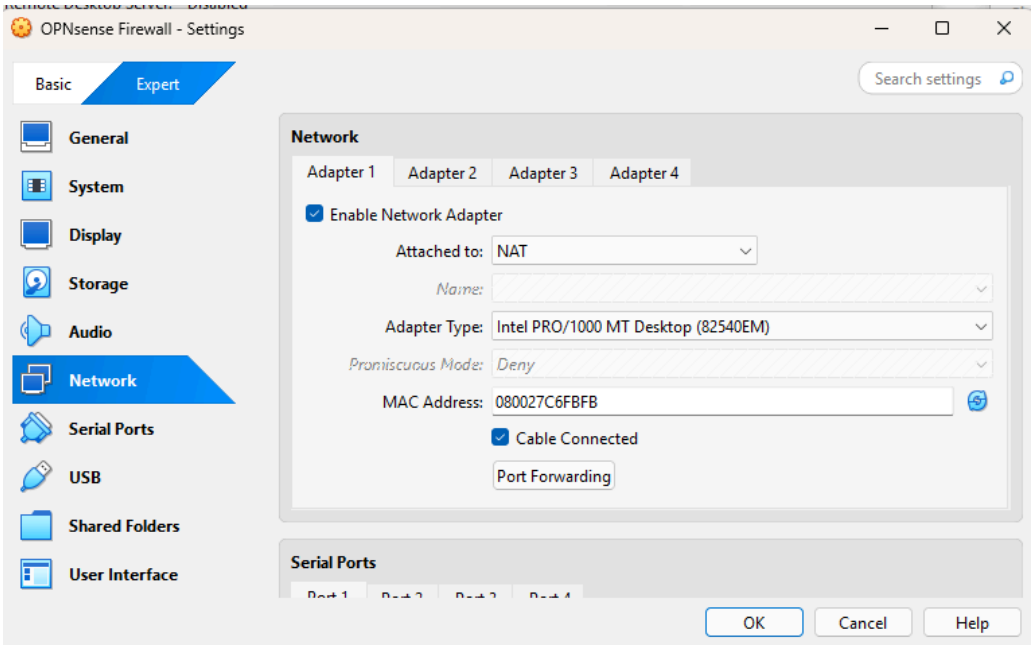
Firewall Setup

OPNSense Firewall Installation & Configuration

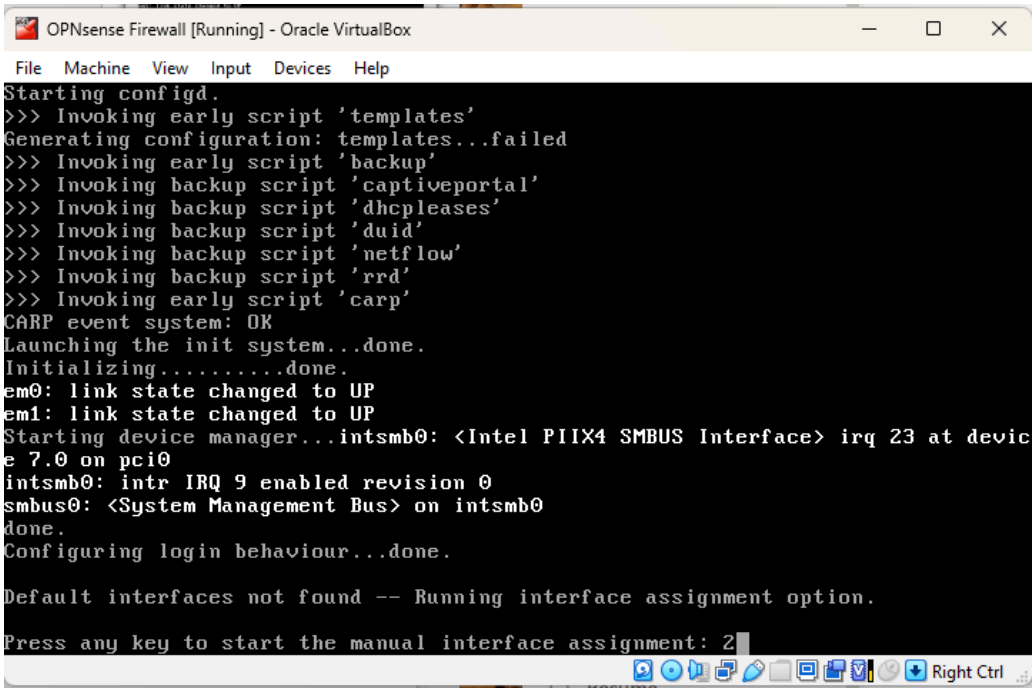
- To install and configurate this firewall we are using applications OPNsense and bzip. These programs allow us to download a mirror iso file, decompress it using powershell, and import the disk image file it into VirtualBox to set up our firewall

- For OPNsense to work correcty we need atleast two interfaces on my firewall, one for my WAN which is connected to my internet connection, and one connected to my LAN.
- So I will configure my Network adapters as such:
- Adapter 1 will share our internet connection making it the WAN in our lab, so I will keep the network at NAT, to maintain that internet connection



- Adapter 2 will be the internal network, as mentioned before the firewall needs to be connected to a WAN (internet) and LAN,
- We will give the internal network a name, intnet
- So now anything with the intnet name will connect to a virtual switch

At this point we will start our VM and add our iso file into virtual box and OPN sense will begin to install.



- It will soon prompt use to log in. We will then login with the username installer and password OPNsense ,to go through with the installation. Our important disk will get erased and OPNsense will replace it with its necessary programs.
- I will then create a root password: 123
- Lastly, the system will install and reboot. We can remove then go to devices > remove virtual disk, so opnsense can boot on the virtual hard drive we

installed it on.

- This shows the configuration of the LAN and WAN ip addresses by opn sense which we will possibly change

- At this point we will choose the option to assign interfaces
- **LAGG (Link Aggregation)** is a method of combining multiple network interfaces into a single logical interface to increase bandwidth and provide redundancy.

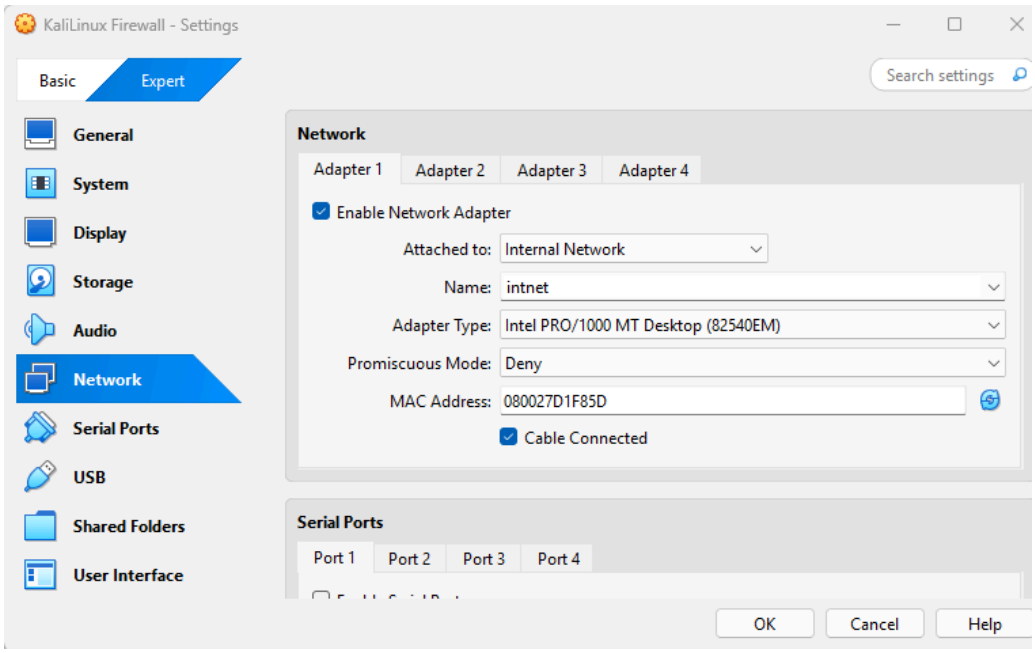
- In this screens shot we have linked the WAN and LAN interface names and will enter a new IP address for both later on in the installation

- In the image above you can see that the we have entered a LAN IP address and the WAN has now picked up an ip adress from our virtualBox network intnet, allowing us to have internet connection from our host computer.

Kali Linux & Firewall Configuration

Now that the OPNsense Firewall has been installed and configured. I will open my Kali Linux machine and make sure the network configuration is set on the same internal network I have created earlier (intnet).

-

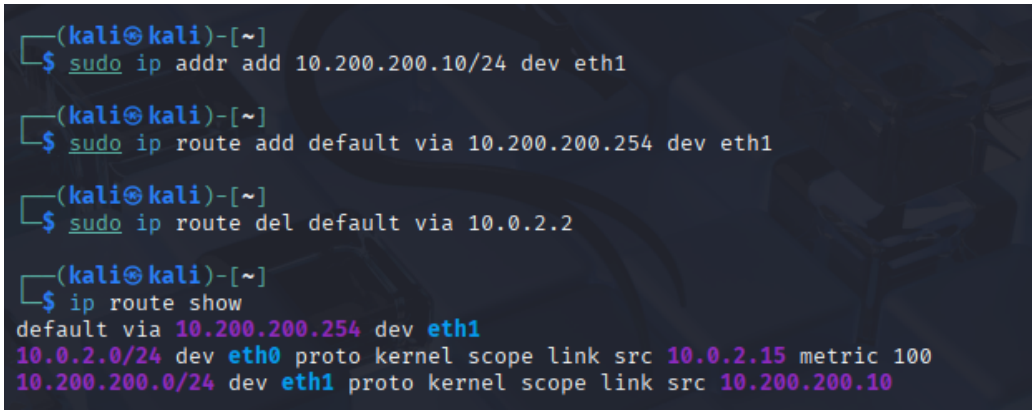


After a few hours of trouble configuring I was finally able to get the GUI to work, I also realized I must keep the firewall VM running while using its features.

The image above takes a look at my firewall dashboard. Furthemore, I make sure the firmware is updated and install extra plugins for virtual box.

Trouble I had during installation

- This installation process had a lot of configuration issues
- The IP address on my firewall and Linux machine were not matching up even though I configured them to be connected through the same internal network (intnet)
- To troubleshoot this issue I researched and experimented. I knew that I had created two adapters for the firewall, to implement the WAN (intnet) and the LAN. However, with my Kali machine I included one adapter to the LAN, and I had no connection to my WAN.



- After adding NAT connection (WAN) to my linux machine, I needed to add the ip address to sudo add and route a ip address for my Kali VM on my internal network. While also deleting the default ip address
- Alternatively, I could have enable DHCP and used "sudo dhclient eth1", which tells my Kali machine to ask the DHCP server on my network for an IP address, subnet mask, gateway and DNS servers.
- I did not use this method to allow me to view how to properly connect the network without the automatic config.
- **This has helped me get a better understanding of how tools and devices connect on a network**

- I also had to remove the iso disk so the firewall can boot from the harddrive so now the iso disk says empty

Learned concepts through this process:

DHCP Meaning

- **DHCP (Dynamic Host Configuration Protocol)** = the "handshake" where a device asks the network for an IP address.
- Without DHCP, your machine has no address → can't talk on the network.

Think of dhclient like your laptop saying:

- "Hey Wi-Fi router, can you give me an IP?"
- And OPNsense (if DHCP is turned on) says:
- "Sure, here's 10.200.200.101. Use me (10.200.200.254) as your gateway to the internet."

A **VLAN** is a way of splitting one physical network into **multiple logical networks**.

Remember IP Address rules

ip address example: 10.200.200.254

IP Address Rules in Subnetting

- In your LAN subnet `10.200.200.0/24`:
  - **Network address** = `10.200.200.0` (reserved, not usable by hosts)
  - **Broadcast address** = `10.200.200.255` (reserved for broadcasts)
  - **Usable host range** = `10.200.200.1 → 10.200.200.254`

Each host (firewall, Kali, Windows VM, etc.) must have a **unique** IP within that range.

Example

- OPNsense LAN (gateway): `10.200.200.254`
- Kali: `10.200.200.10`
- Windows VM: `10.200.200.11`

- Another VM: `10.200.200.12`

👉 No two devices can share the same IP at the same time — otherwise you'll get an **IP conflict** (both machines fight for responses, and networking breaks).

Important Bash Commands (note to self)

Checklist for Kali to talk to OPN Sense

Assign static IP on LAN

- sudo ip addr add 10.200.200.10/24 dev eth1

Add route to LAN subnet (optional if /24 is already there)

- sudo ip route add 10.200.200.0/24 dev eth1

Setdefault gateway = OPNsense LAN IP

- sudo ip route add default via 10.200.200.254 dev eth1

Set DNS

- echo "nameserver 8.8.8.8" | sudo tee /etc/resolv.co

Show the IP

- ip route show

Erase the default ip

- sudo ip addr flush dev eth1

Using Network manager allows the ip to stick

This command will allow eliminate the process of adding a new ip address for each VM start up

See what profiles Exist

- nmcli connection show

In this lab I am renaming wireless connection 2

- sudo nmcli connection modify "Wired connection 2" connection.id "eth1"

Next add the ip address and gateway

- sudo nmcli connection modify eth1 ipv4.addresses "192.168.1.10/24"
  sudo nmcli connection modify eth1 ipv4.gateway "192.168.1.1"
  sudo nmcli connection modify eth1 ipv4.method manual
- (If you want to use the `10.200.200.0/24` LAN later, just replace the addresses with `10.200.200.10/24` and gateway `10.200.200.254`.)
- **LAN IP = your own unique address in the subnet.**
- **Gateway IP = the router/firewall's address in the same subnet.**

Bring the profile down and up to restart it

- sudo nmcli connection down eth1
  sudo nmcli connection up eth1

Verify

- ip addr show eth1
- ip route show

```
┌──(kali㉿kali)-[~]
└─$ ip route show
default via 10.200.200.254 dev eth1 proto static metric 103
10.0.2.0/24 dev eth0 proto kernel scope link src 10.0.2.15 metric 102
10.200.200.0/24 dev eth1 proto kernel scope link src 10.200.200.10 metric 103
```

Ping the connection to show conntectivity

- ping -c 3 10.200.200.254