NCY-2 Assignment 4&5

PFsense

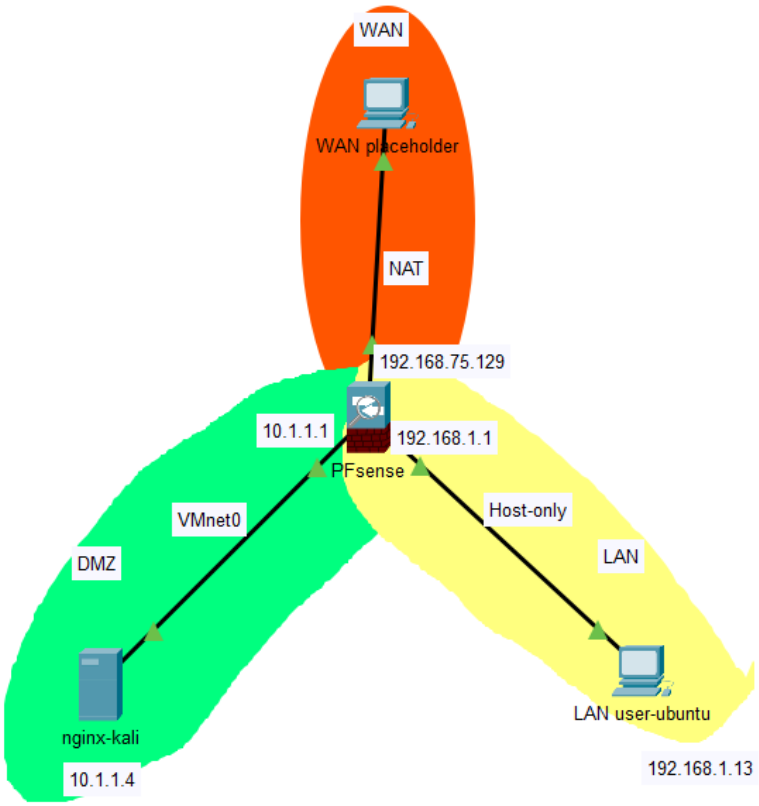
Submitted by
Abdul Sami Qasim (22i-1725)
Ahmad Abdullah (22i-1609)

Submitted to
Prof. Abdullah Abid

Date of Submission
November 12, 2024

Table of Contents

Network Topology



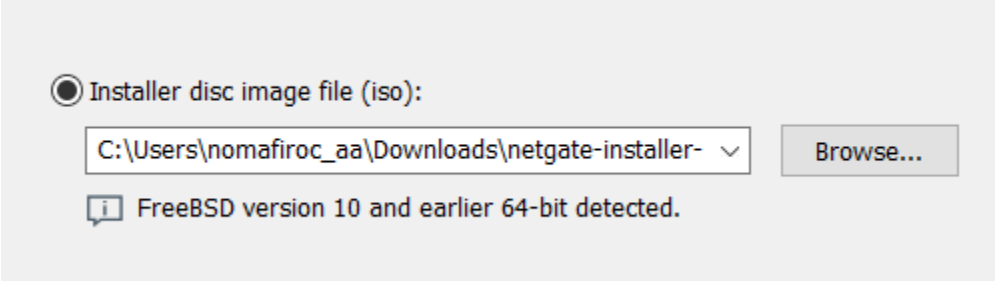
This is the network topology that we made, it consists of

1. 3 Virtual machines on vmware
    a. FreeBSD for pfsense
    b. Kali for nginx web server
    c. Ubuntu as the LAN client
2. 3 interfaces
    a. VMnet0 to connect DMZ to the firewall
    b. Host-only to connect LAN to the firewall
    c. NAT to connect WAN to the firewall
3. The networks we used are
    a. 10.1.1.0 for DMZ config
    b. 192.168.1.0 for LAN config
    c. NAT is on 192.168.75.0 (as of now)

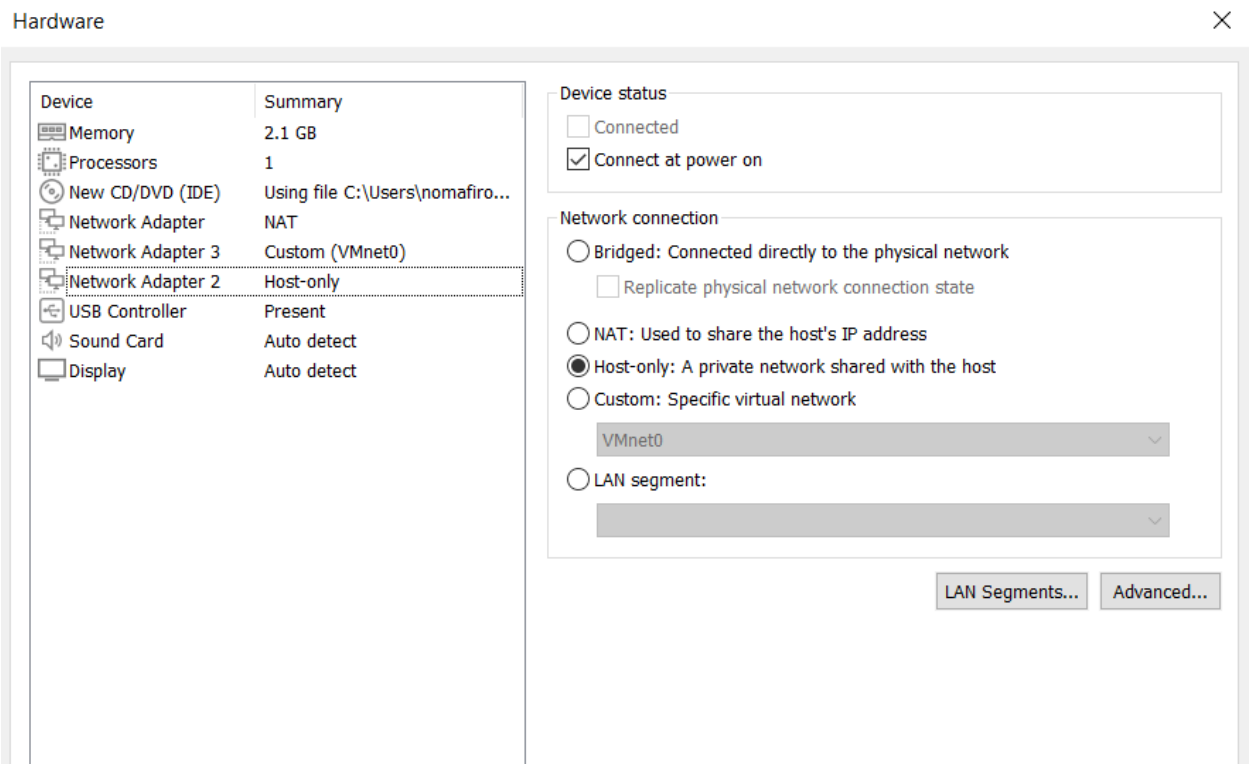
PFsense configuration:

1. Installing the iso and machine hardware configurations

First of all, we have to install the iso file from netgate and make a virtual machine with it.



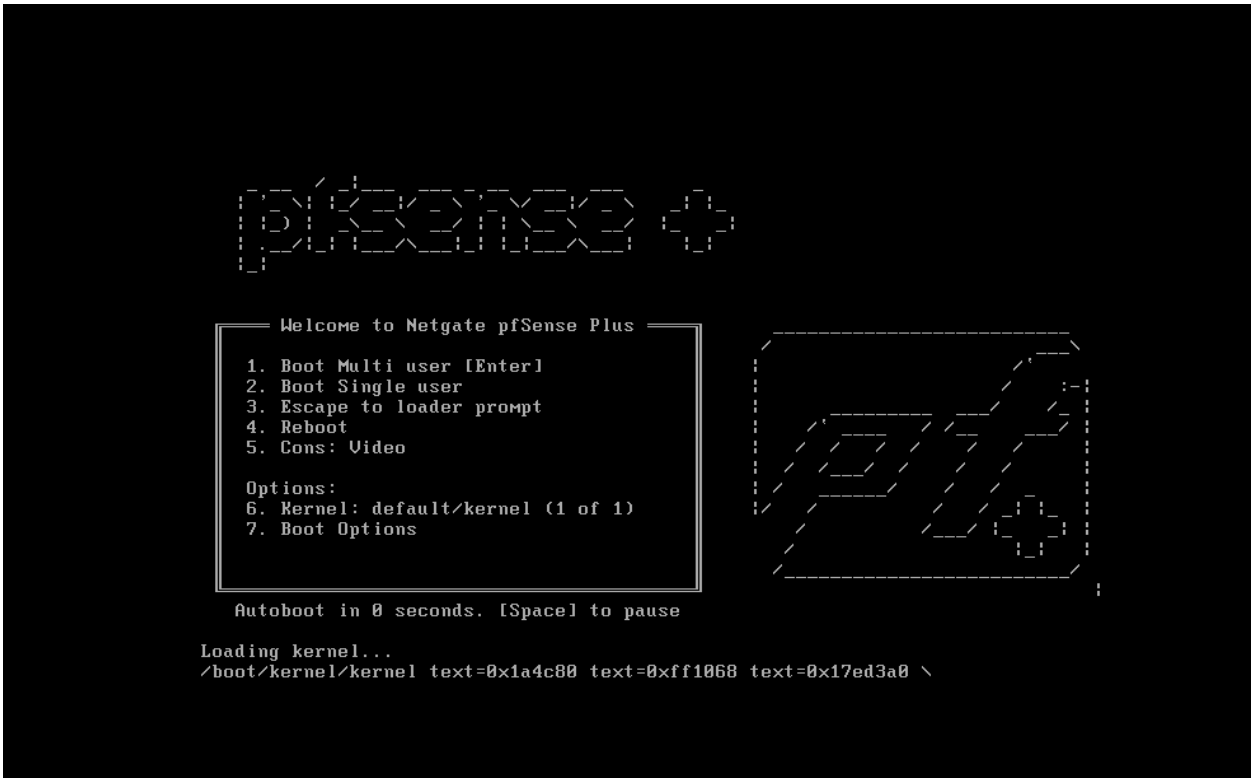
These are the hardware configurations I used for this machine:



The 3 Network Adapters are for the three connections were supposed to make to connect LAN, WAN and DMZ.

2. Machine setup

Once you start the machine, you will get a screen like this:



After this you have to configure your LAN, WAN interfaces and then once you have installed the CE version of the firewall and the installation goes well, you'll get a screen like this:

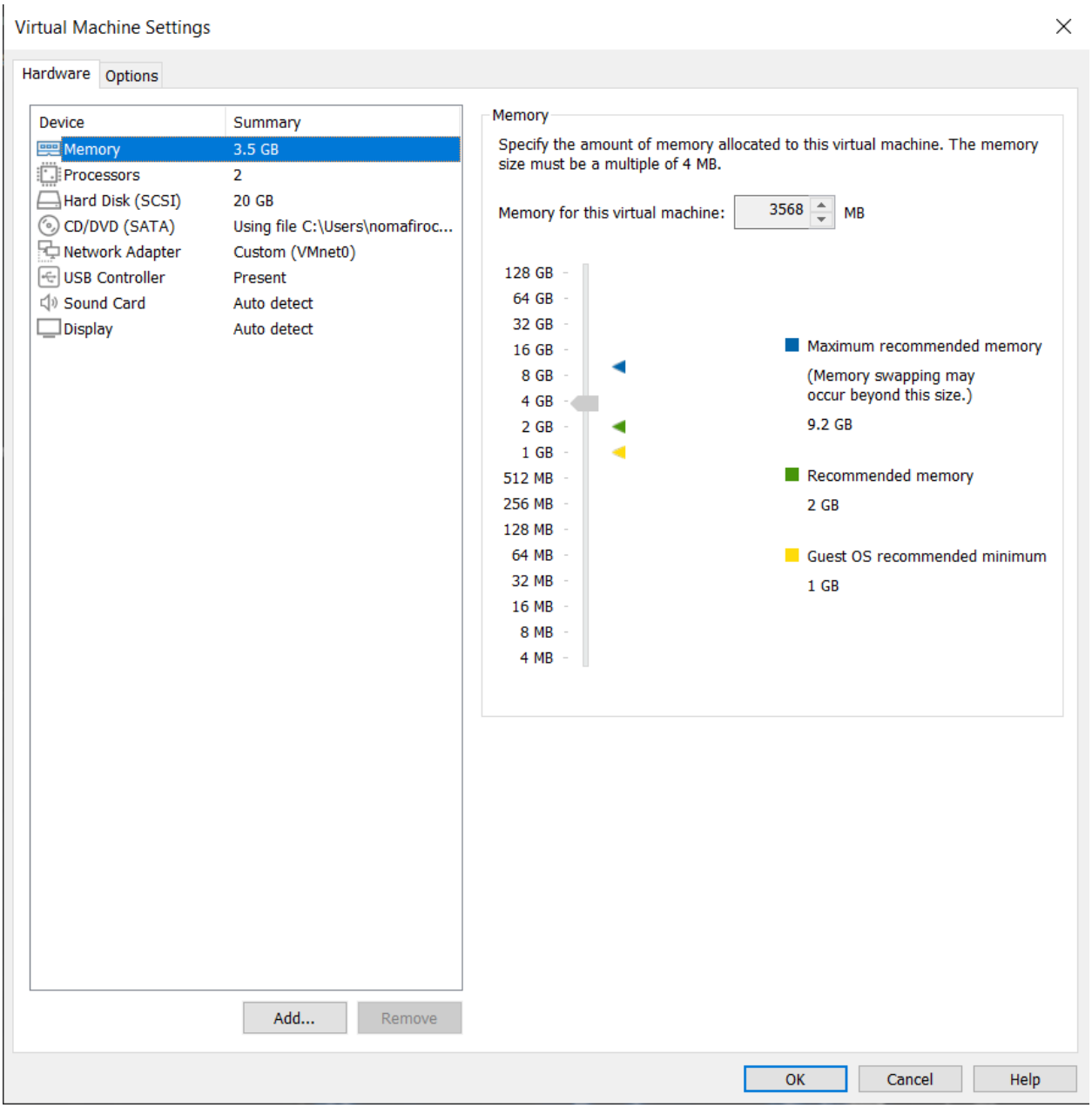
```
 (Local Database)


FreeBSD/amd64 (pfSense.home.arpa) (ttyv0)

VMware Virtual Machine - Netgate Device ID: 43e9307723c563c561ad

*** Welcome to pfSense 2.7.2-RELEASE (amd64) on pfSense ***

 WAN (wan)       -> em0        -> v4/DHCP4: 192.168.75.129/24
 LAN (lan)       -> em1        -> v4: 192.168.1.1/24
 DMZ (opt1)      -> em2        -> v4: 10.1.1.1/24

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell

Enter an option:
```

Here, I've configured the interfaces and logged into the GUI due to which I get the welcome message along with the IPs assigned to the interfaces.

Nginx web server configuration

1. Kali hardware configurations

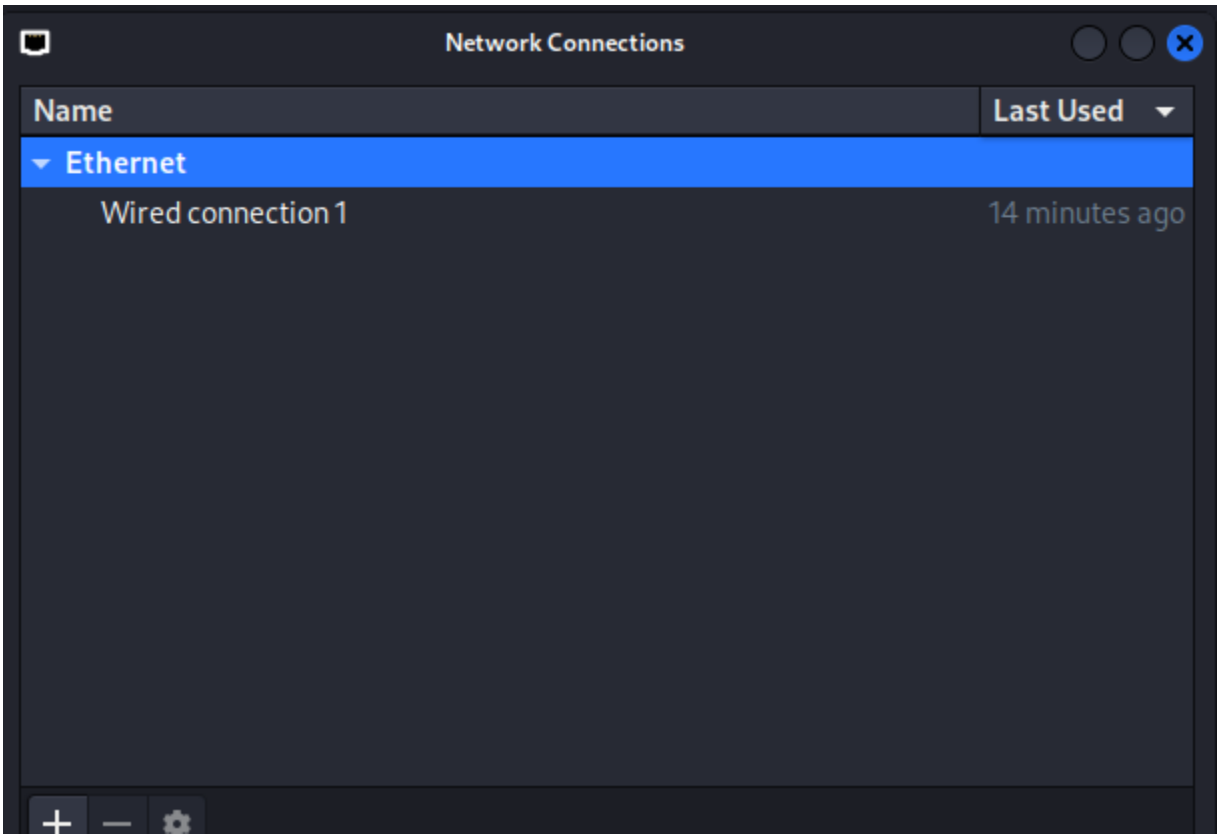
For nginx, we used Kali Linux and while adding the pre-made machine, we changed the network adapter to this:



We connected the DMZ on the adapter VMnet0 in the pfsense iso setup, so to bring the two machines together, we put this one on VMnet0 aswell.

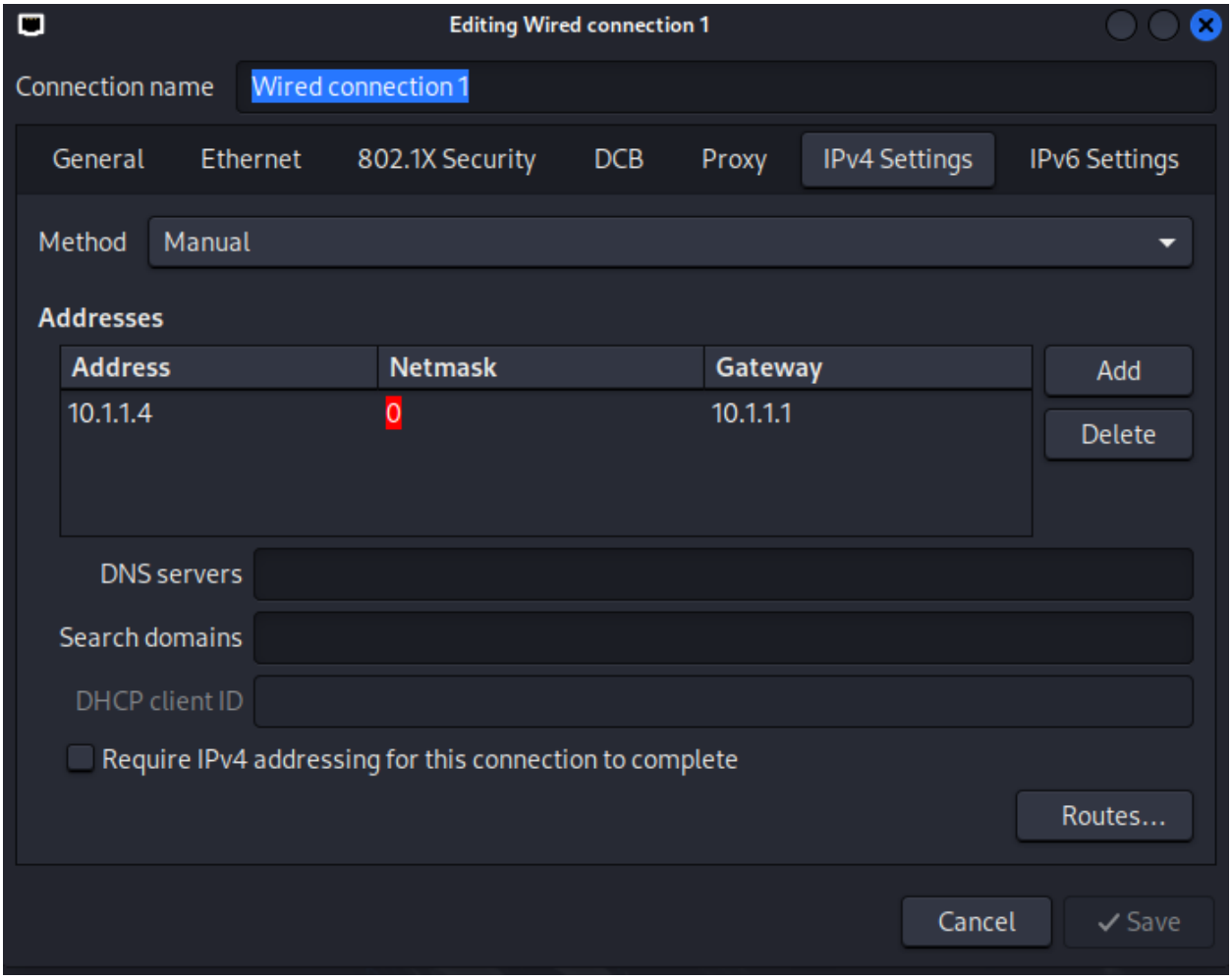
2. IP configuration

Now we do have the correct adapter connected to both the firewall and kali machines but machines typically have automatic IP assignment setup, while we want to do it statically so let's do that:



This interface is the Advanced Network Configuration interface found in settings.

Now, clicking on the Wired connection 1 takes us to this:



Here, I've gone to the IPv4 settings tab and set the method to manual, which allows us to assign a static IP to our machine (in this case, I put the machine IP to 10.1.1.4 and gateway to 10.1.1.1).

3. Nginx setup

By default, we had the apache2 server running, so to convert it to nginx, we first removed the apache2 server:

```
┌──(kali㉿kali)-[~]
└─$ sudo apt remove apache2 --purge
The following packages were automatically installed and are no longer require
d:
  apache2-data  apache2-utils
Use 'sudo apt autoremove' to remove them.

REMOVING:
  apache2*  kali-linux-default*  kali-linux-headless*

Summary:
  Upgrading: 0, Installing: 0, Removing: 3, Not Upgrading: 0
  Freed space: 621 kB

Continue? [Y/n] y
(Reading database ... 416728 files and directories currently installed.)
Removing kali-linux-default (2024.3.3) ...
Removing kali-linux-headless (2024.3.3) ...
Removing apache2 (2.4.62-1) ...
Processing triggers for man-db (2.12.1-2) ...
Processing triggers for kali-menu (2024.3.1) ...
(Reading database ... 416671 files and directories currently installed.)
Purging configuration files for apache2 (2.4.62-1) ...
dpkg: warning: while removing apache2, directory '/var/www/html' not empty so
 not removed
```

By running this command, we have deleted the apache2 server, now we setup nginx.

I've already installed nginx by using the command,

sudo apt install nginx

Now to get it up and running, we use the following commands:

```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl start nginx

┌──(kali㉿kali)-[~]
└─$ sudo systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nginx
```

To check if it is running, we can use the following command,

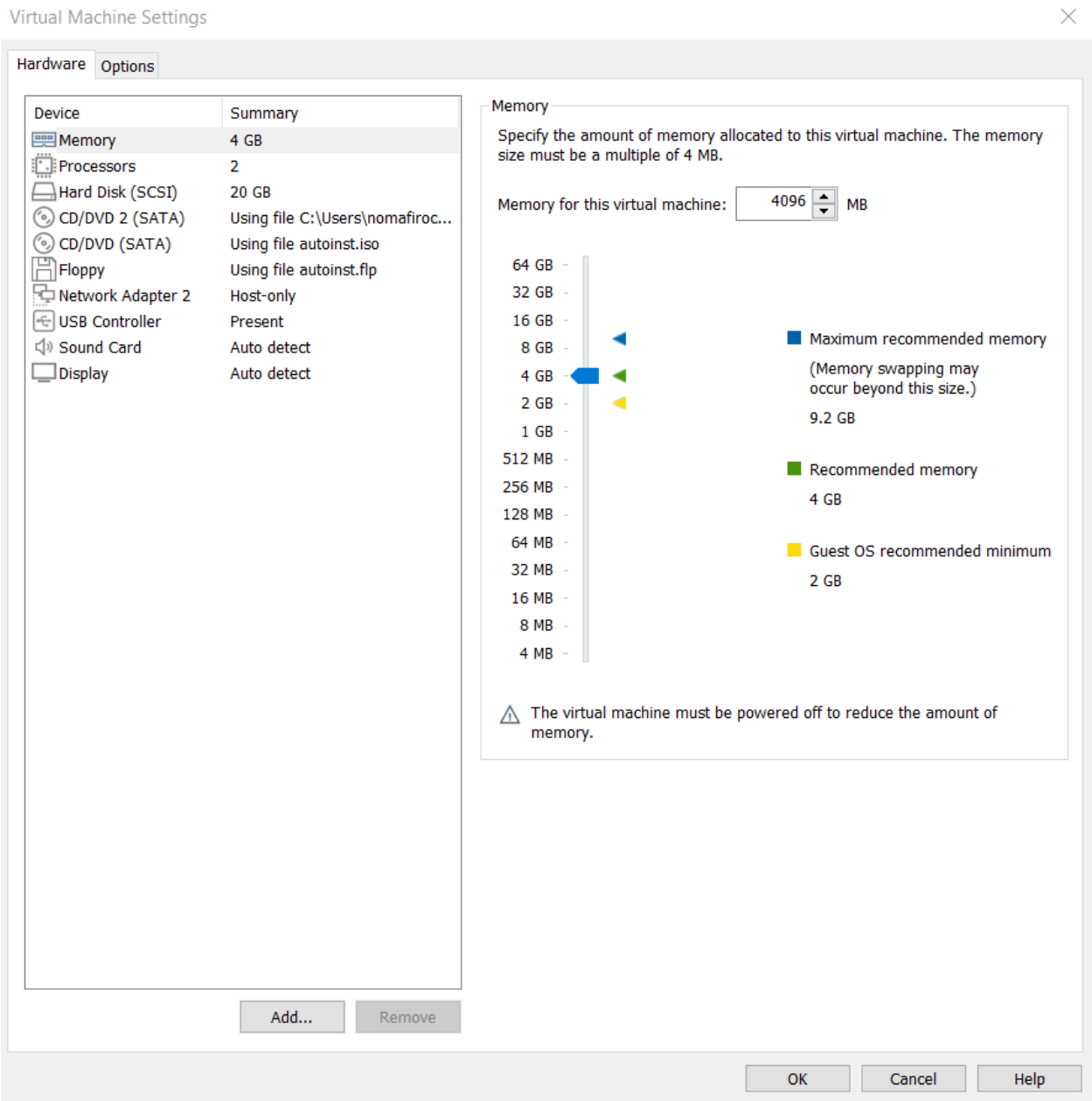
```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset:>
     Active: active (running) since Tue 2024-11-12 03:30:19 UTC; 5h 47min ago
 Invocation: dc2de2e1225244bbabbbd62b0ae5a194
       Docs: man:nginx(8)
   Main PID: 12937 (nginx)
      Tasks: 3 (limit: 3923)
     Memory: 4M (peak: 4.5M)
        CPU: 153ms
     CGroup: /system.slice/nginx.service
             ├─12937 "nginx: master process /usr/sbin/nginx -g daemon on; ma>
             ├─12938 "nginx: worker process"
             └─12939 "nginx: worker process"

Nov 12 03:30:19 kali systemd[1]: Starting nginx.service - A high performance>
Nov 12 03:30:19 kali systemd[1]: Started nginx.service - A high performance >
```

According to this, our nginx server is now hosted!

LAN client configuration

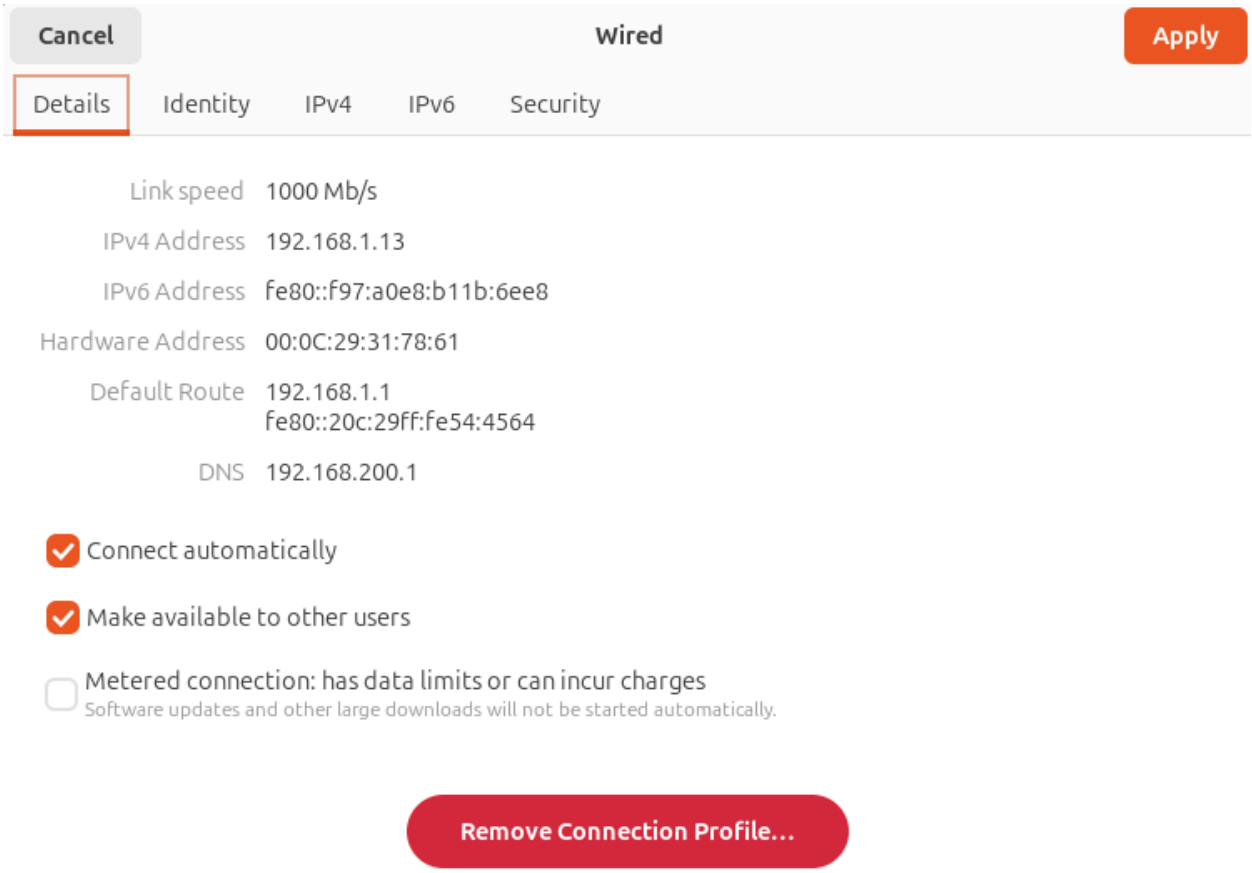
1. Hardware Configuration

   As we did in the server machine hardware configuration, we set the network adapter to the one we will be using for LAN, in this case the Host-only one:



2. IP configuration

For the IP configuration, I put 192.168.1.13 with the gateway 192.168.1.1



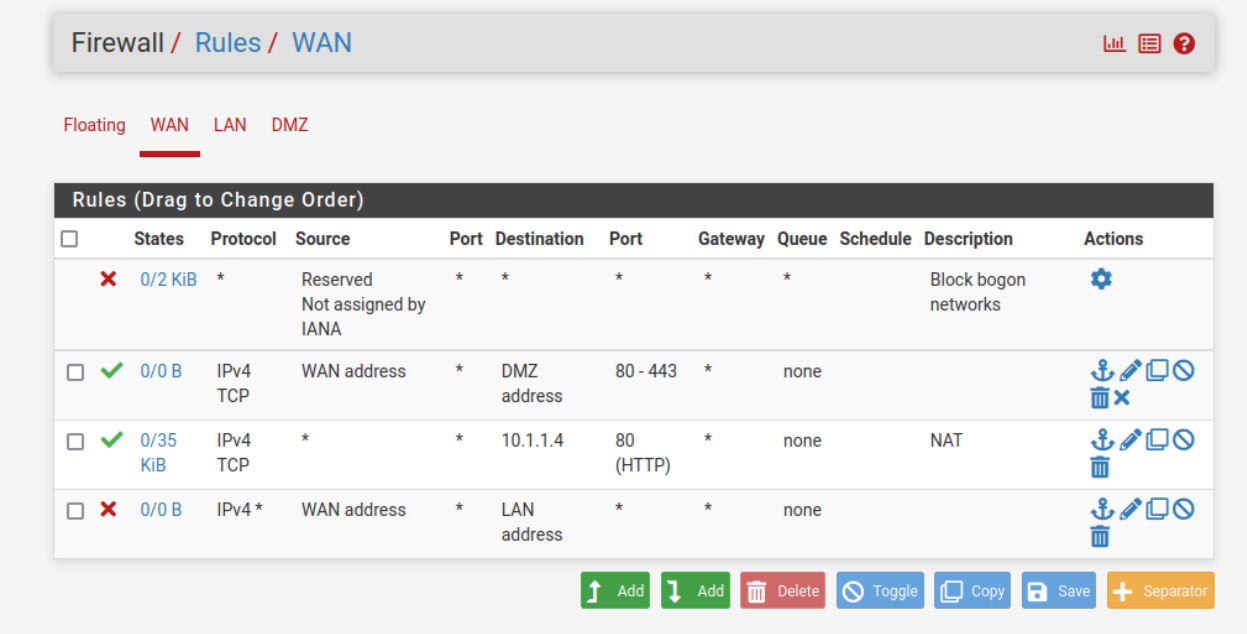
Firewall rules

We were told to apply the following rules:

1. Allow Incoming Traffic from the Internet to DMZ on ports: 80, 443
2. Allow traffic from DMZ to WAN on ports: 53, ICMP and 123 (NTP)
3. Block traffic from LAN to DMZ except ICMP and SSH for administration purposes
4. Block traffic from LAN to DMZ except port 80 to access the web server
5. Allow ICMP (ping) traffic from the LAN network to the WAN network while limiting the rate of ICMP requests to prevent ICMP flooding.
6. Allow DNS (port 53) traffic from the LAN network to specific DNS servers on the Internet.
7. Allow access to limited number of Websites from LAN; Block everything else
8. Block all incoming traffic from WAN to LAN except those mentioned above

All of these have been applied as follows:

1. WAN rules



Allowed:

1. WAN can now communicate with port 80-443 in DMZ (to access web)
2. Implemented port forwarding so the web server can be accessed on WAN

Denied:

Denied everything from WAN to LAN

2. LAN rules

Firewall / Rules / LAN

Floating WAN LAN DMZ

Rules (Drag to Change Order)

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
0/14.12 MiB	*	*	*	LAN Address	443 80	*	*		Anti-Lockout Rule	
0/684 KiB	IPv4 TCP	LAN subnets	*	sites	80 - 443	*	none			
0/250 KiB	IPv4 TCP	LAN subnets	*	*	80 - 443	*	none			
0/0 B	IPv4 ICMP echorep	LAN address	*	WAN address	*	*	none			
0/84.84 MiB	IPv4 *	LAN subnets	*	*	*	*	none		Default allow LAN to any rule	
0/0 B	IPv6 *	LAN subnets	*	*	*	*	none		Default allow LAN IPv6 to any rule	
0/0 B	IPv4 ICMP any	LAN address	*	DMZ address	*	*	none			
0/0 B	IPv4 TCP	LAN address	*	DMZ address	22 (SSH)	*	none			
0/0 B	IPv4 TCP	LAN address	*	DMZ address	80 (HTTP)	*	none			
0/0 B	IPv4 TCP/UDP	LAN address	*	8.8.8.8	53 (DNS)	*	none			
0/0 B	IPv4 *	LAN address	*	DMZ address	*	*	none			

Add Add Delete Toggle Copy Save Separator

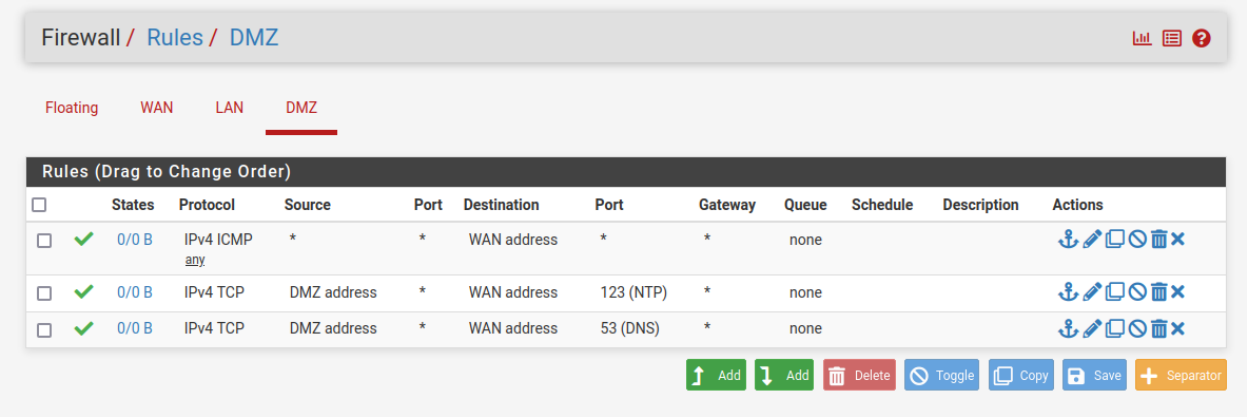
Allowed:

1. Specific sites can be accessed
2. Rate limited ICMP packets from LAN to WAN
3. All ICMP packets from LAN to DMZ
4. SSH from LAN to DMZ
5. Web access to the nginx server (port 80 access from LAN to DMZ)
6. DNS access to 8.8.8.8 for LAN

Denied:

All other packets are denied

3. DMZ rules



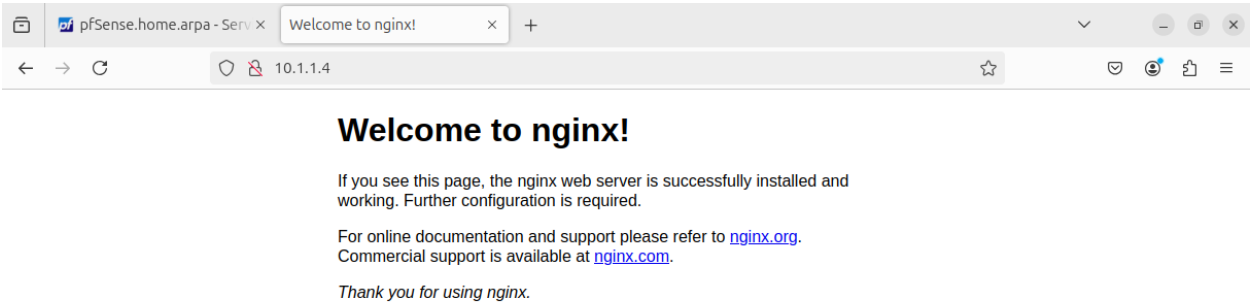
Allowed:

1. Port 123 access from DMZ to WAN
2. Port 53 access from DMZ to WAN

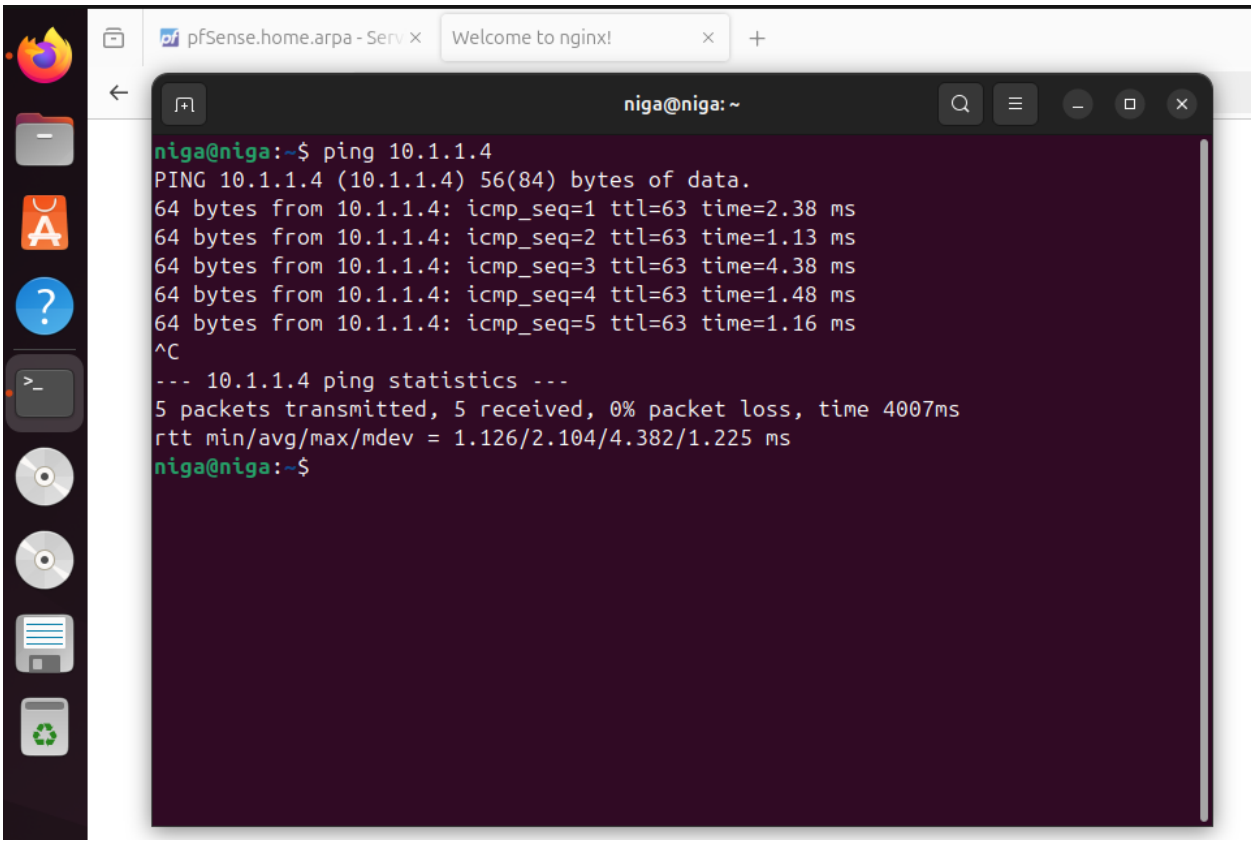
Proof of firewall rules working

1. *LAN*

Web server access

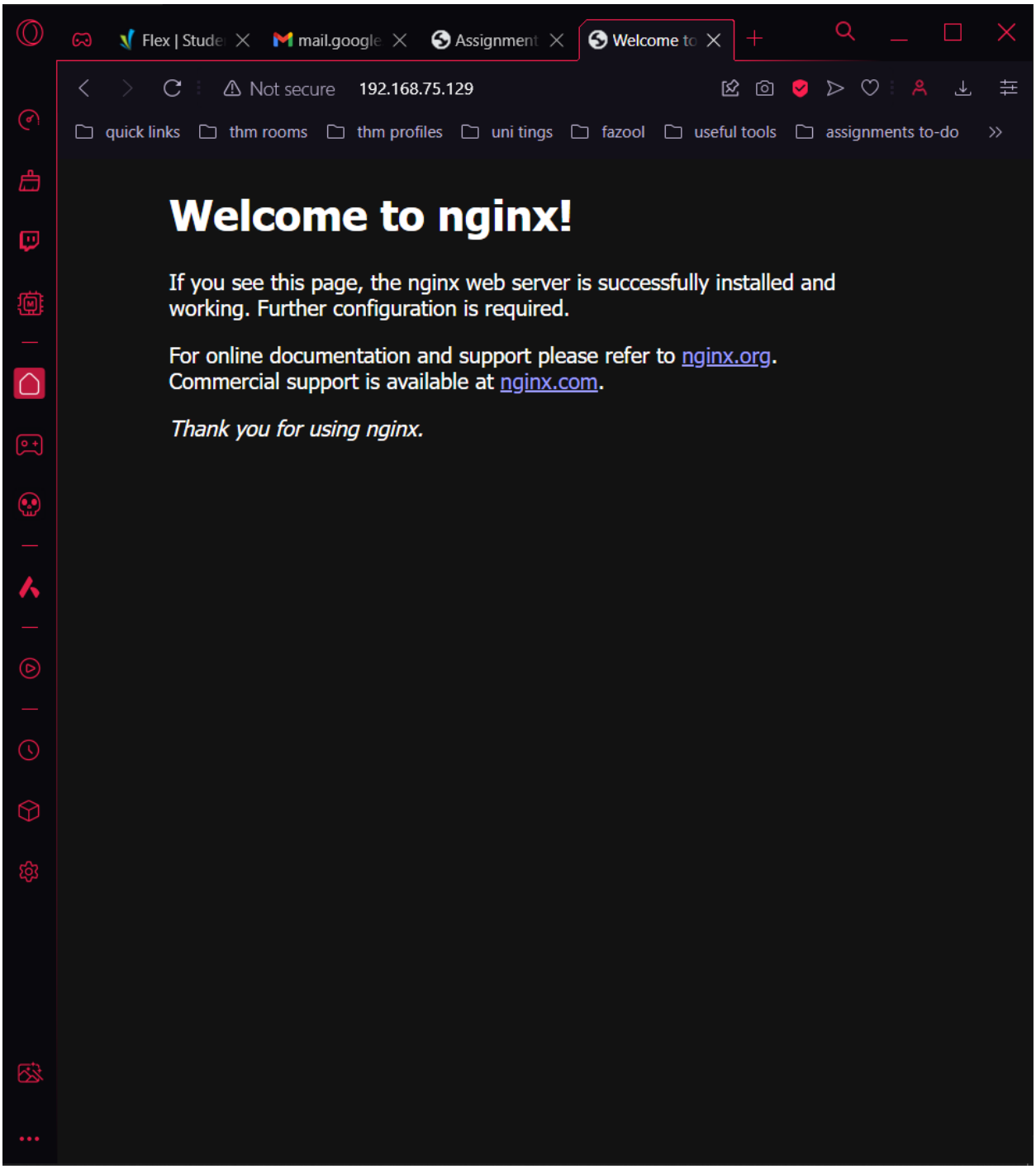


Ability to ping nginx server machine


pfSense.home.arpa - Serv
Welcome to nginx!
niga@niga: ~
niga@niga:~$ ping 10.1.1.4
PING 10.1.1.4 (10.1.1.4) 56(84) bytes of data.
64 bytes from 10.1.1.4: icmp_seq=1 ttl=63 time=2.38 ms
64 bytes from 10.1.1.4: icmp_seq=2 ttl=63 time=1.13 ms
64 bytes from 10.1.1.4: icmp_seq=3 ttl=63 time=4.38 ms
64 bytes from 10.1.1.4: icmp_seq=4 ttl=63 time=1.48 ms
64 bytes from 10.1.1.4: icmp_seq=5 ttl=63 time=1.16 ms
^C
--- 10.1.1.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 1.126/2.104/4.382/1.225 ms
niga@niga:~$

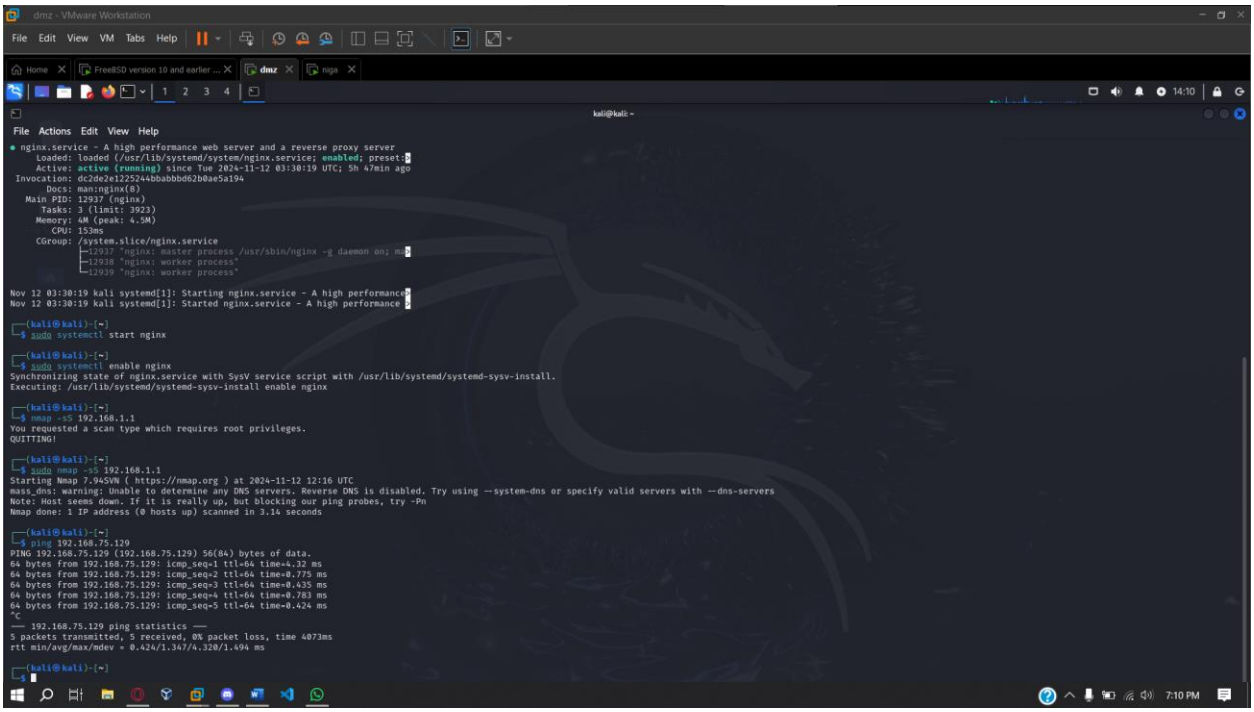
2. *WAN*

I have access to the nginx server through firewall's WAN interface.



3. *DMZ*

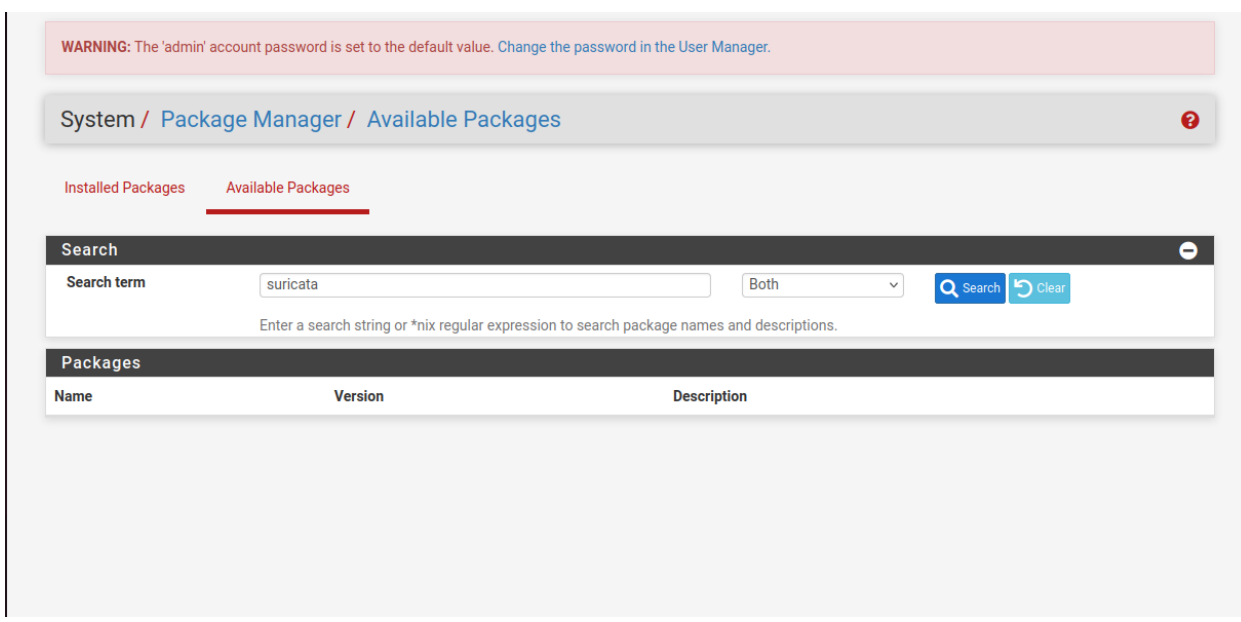
I can ping from DMZ to WAN



Suricata

1. Suricata Download

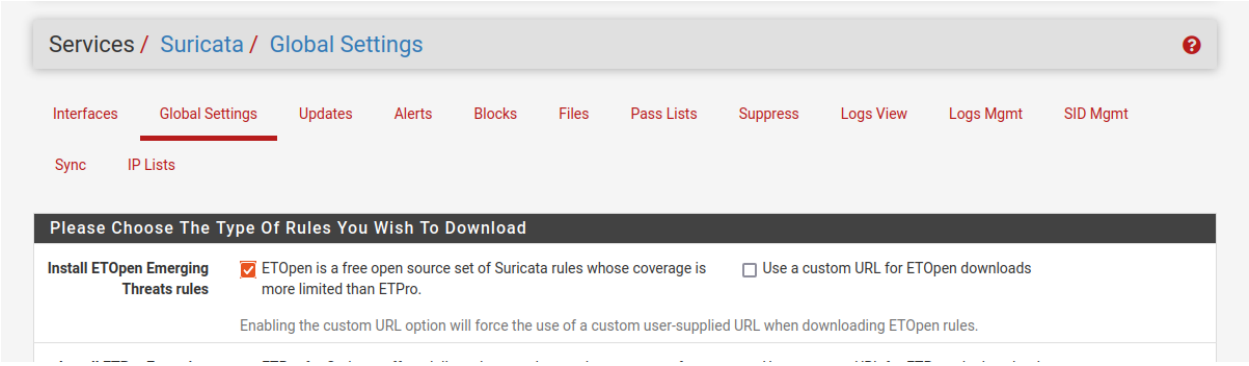
First of all, when we have to setup suricata, we have to download it so we go to the package manager and install suricata:



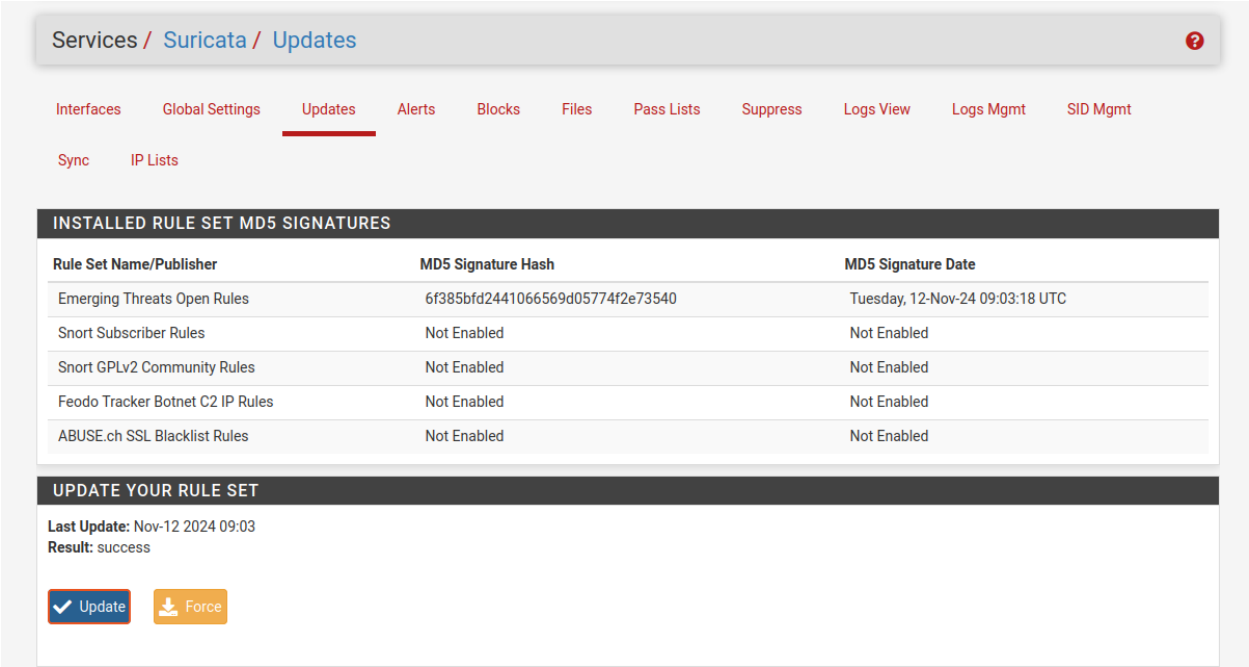
In this case, it's not showing up because I have already downloaded and configured it on my firewall.

2. Downloading Rulesets

To download the rules, we have to first go to Suricata in the services tab:



First of all we select this option to install the rulesets and then save the settings.



After that, we went to the updates tab to update the rules and download them, if not already downloaded.

Now we will select the rules and then check for alerts:

Services / Suricata

Interfaces | Global Settings | Updates | Alerts | Blocks | Files | Pass Lists | Suppress | Logs View | Logs Mgmt | SID Mgmt | Sync | IP Lists

Interface Settings Overview

	Interface	Suricata Status	Pattern Match	Blocking Mode	Description	Actions
☐	DMZ (em2)		AUTO	DISABLED	DMZ	
☐	WAN (em0)		AUTO	DISABLED	WAN	

Add | Delete

WAN Settings | WAN Categories | WAN Rules | WAN Flow/Stream | WAN App Parsers | WAN Variables | WAN IP Rep

Automatic flowbit resolution

Resolve Flowbits ☑ Auto-enable rules required for checked flowbits

Default is Checked. Suricata will examine the enabled rules in your chosen rule categories for checked flowbits. Any rules that set these dependent flowbits will be automatically enabled and added to the list of files in the interface rules directory.

View rules View

Click to view auto-enabled rules required to satisfy flowbit dependencies

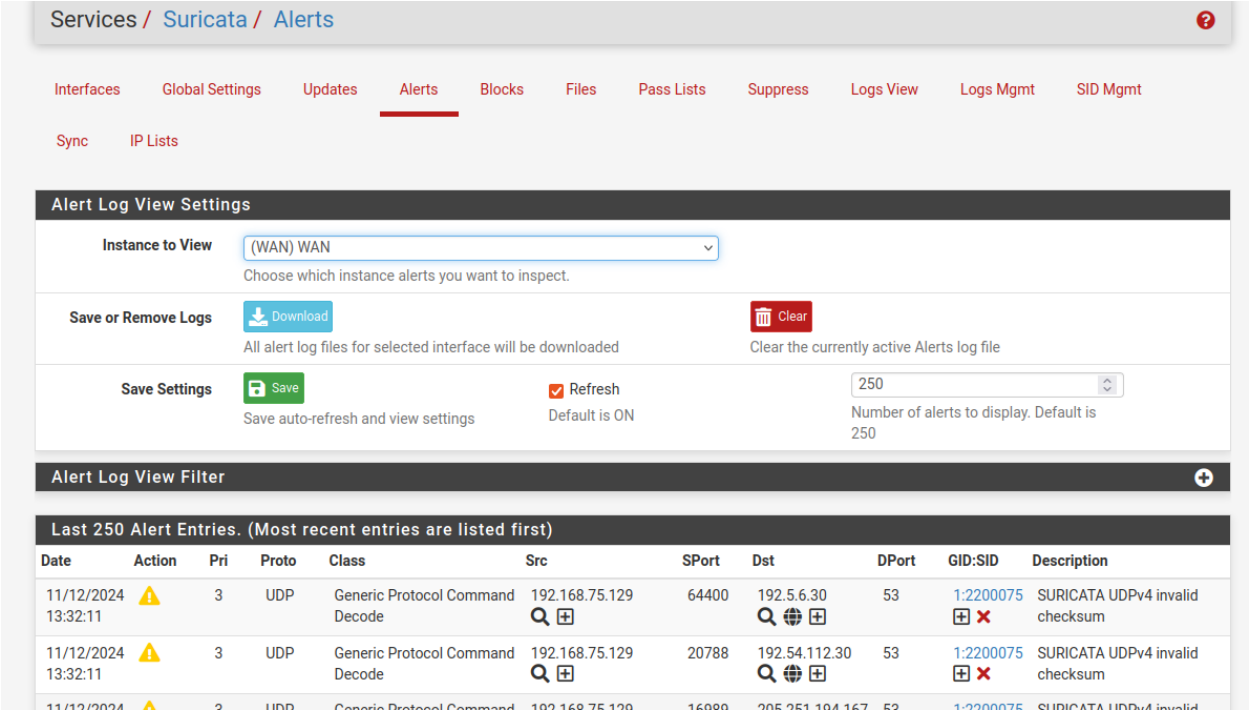
Note: Auto-enabled rules generating unwanted alerts should have their GID:SID added to the Suppression List for the interface.

Select the rulesets (Categories) Suricata will load at startup

- Category is auto-enabled by SID Mgmt conf files
- Category is auto-disabled by SID Mgmt conf files

Select All | Unselect All | Save

Enabled			Ruleset:	
Enabled	**Ruleset: Default Rules**	**Enabled**	**Ruleset: ET Open Rules**	**Snort Rules are not enabled.**
☑	app-layer-events.rules	☑	emerging-3coresec.rules	
☑	decoder-events.rules	☑	emerging-activex.rules	
☑	dhcp-events.rules	☑	emerging-adware_pup.rules	



As you can see, some logs are being formed.

Issues faced

1. PFsense setup

At first, we were setting up the pfsense machine on virtualbox which somehow caused the download to get looped, by the download getting looped, I mean that once the download finished, the machine rebooted and started the configuration and installation menu again, which got fixed when I set the machine up on vmware.

The next issue I personally faced was that I was giving the machine a very small amount of memory which caused the download to keep failing.

These issues caused me to waste a full day on just the setup.

2. Ubuntu network failure

While we were checking if we connected the machine to the correct network adapter, we had to reboot it. While doing so, we lost our network settings as a whole due to which

we had some hours wasted. We fixed this issue by reinstalling the ubuntu machine and then taking a snapshot once everything was in working state.

3. Wrong network interface configuration

After completing half of the assignment, we started setting up the nginx server in such a way that our host machine could access it easily, while doing so, we kept changing our network adapter settings from bridged to NAT and vice versa.

When we were doing this, we accidentally changed our LAN adapter (previously set to Host-only) to NAT which gave us quite a bit of trouble.

4. Website blocking via firewall

There was a firewall rule that we were told to implement that would cause websites to be blocked, at first we were just blocking communication between port 80-443 on both sides which was inadequate upon further inspection. We fixed this by changing the source ports from the previous 80-443 to any.