

NCY-2 Assignment 4&5

PFsense

Submitted by

Abdul Sami Qasim (22i-1725)

Ahmad Abdullah (22i-1609)

Submitted to

Prof. Abdullah Abid

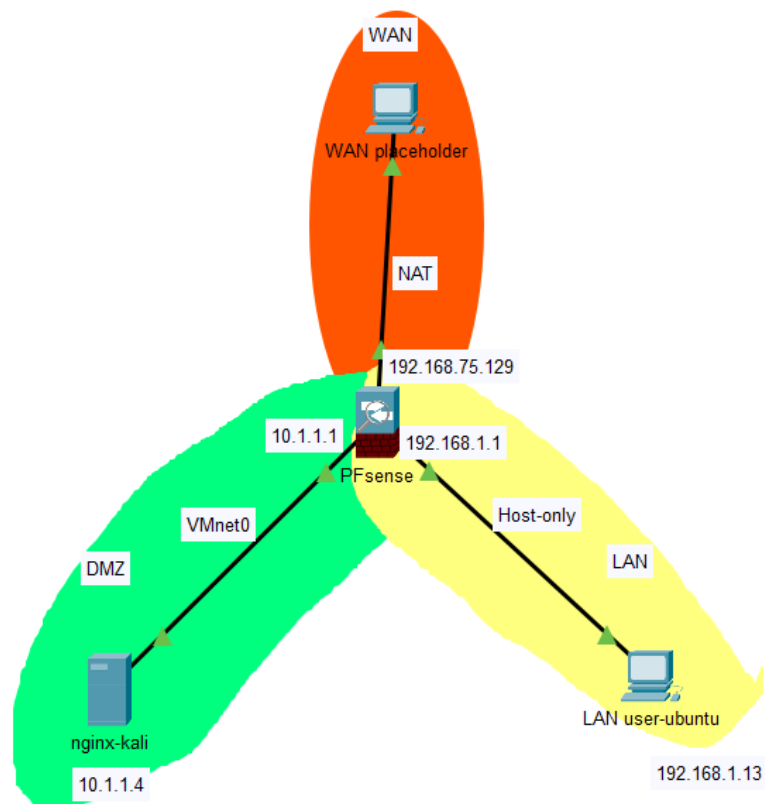
Date of Submission

November 12, 2024

Table of Contents

Network Topology	3
PFsense configuration:	4
1. Installing the iso and machine hardware configurations	4
2. Machine setup	4
Nginx web server configuration	7
1. Kali hardware configurations	7
2. IP configuration	8
3. Nginx setup	9
LAN client configuration	11
1. Hardware Configuration	11
2. IP configuration	12
Firewall rules	13
1. WAN rules	14
2. LAN rules	15
3. DMZ rules	16
Proof of firewall rules working	16
Suricata	20
1. Suricata Download	20
2. Downloading Rulesets	21
Issues faced	23
1. PFsense setup	23
2. Ubuntu network failure	23
3. Wrong network interface configuration	24
4. Website blocking via firewall	24

Network Topology



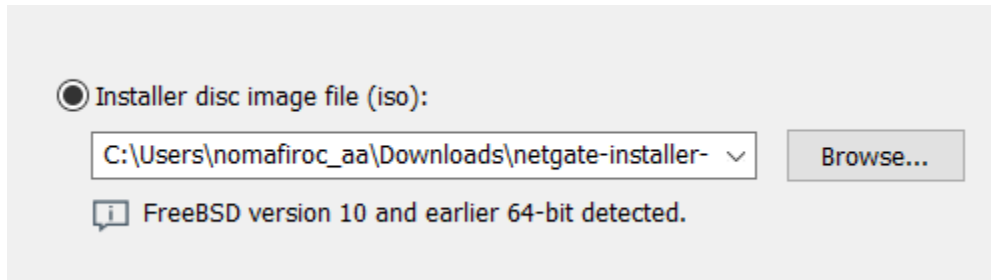
This is the network topology that we made, it consists of

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)

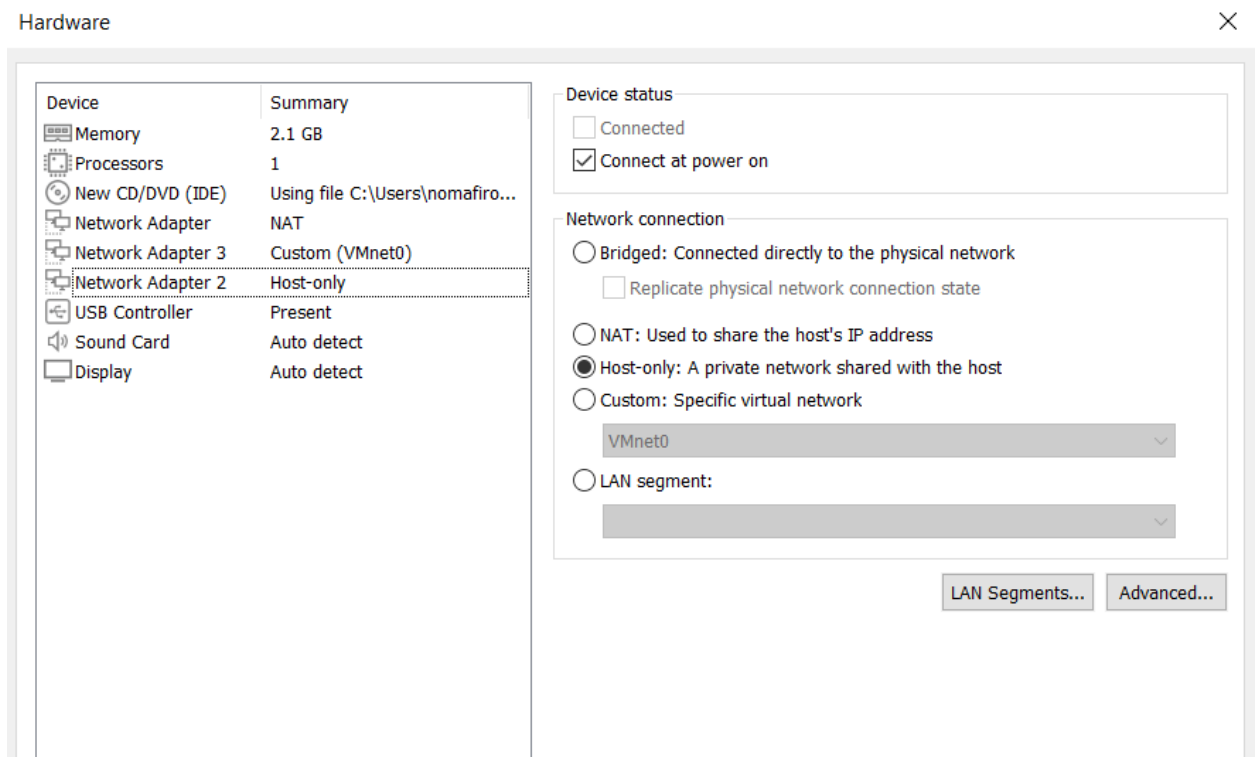
PFsense configuration:

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.



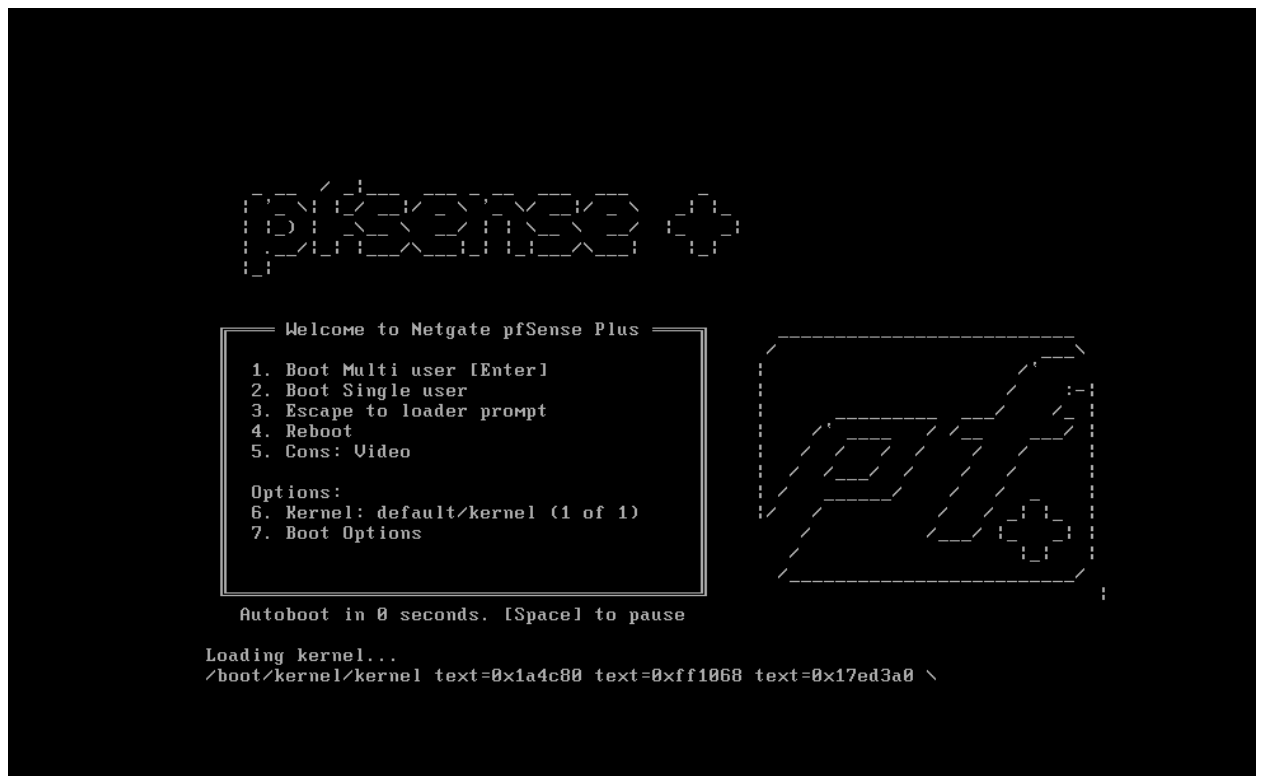
These are the hardware configurations I used for this machine:



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.arp) (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	

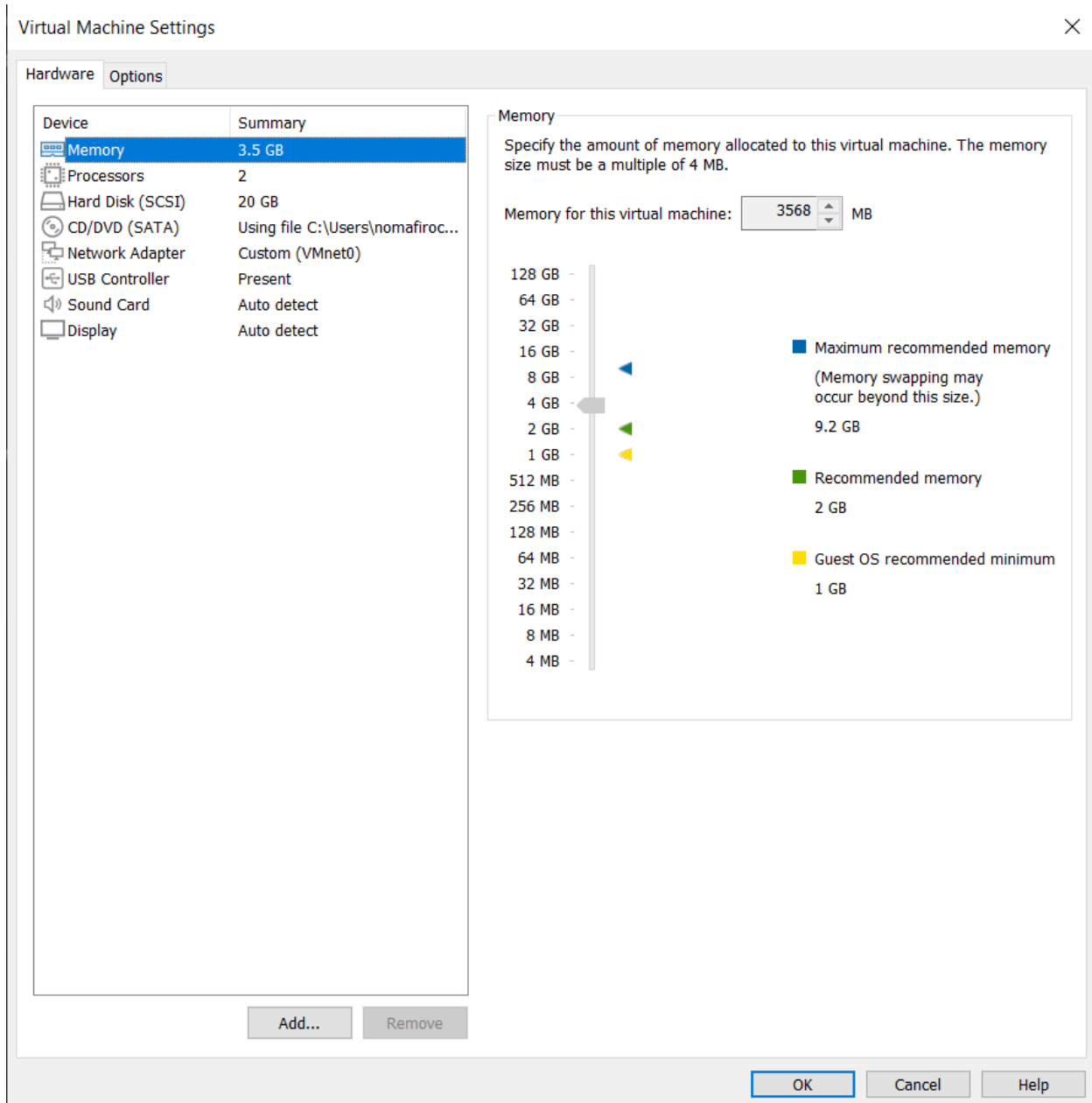
Enter an option: █

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

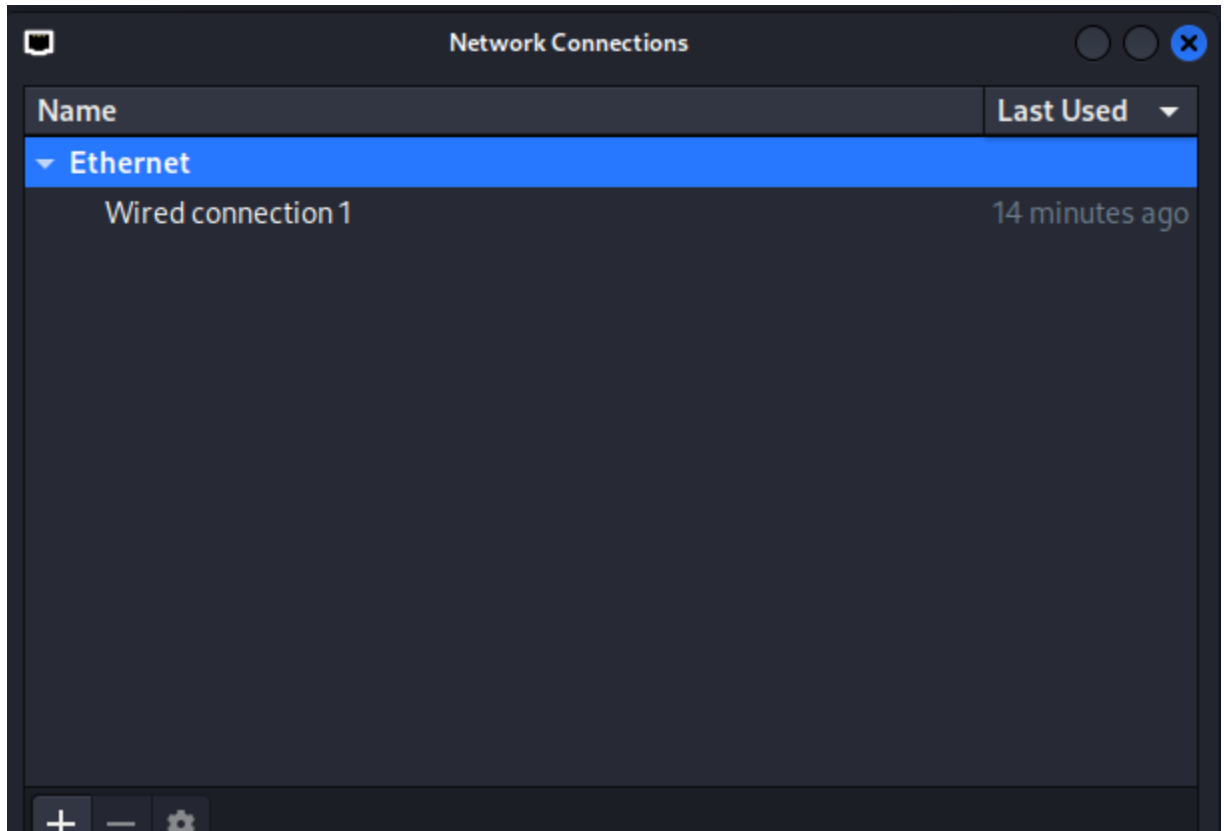
For nginx, we used Kali Linux and while adding the pre-made machine, we changed the network adapter to this:



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.

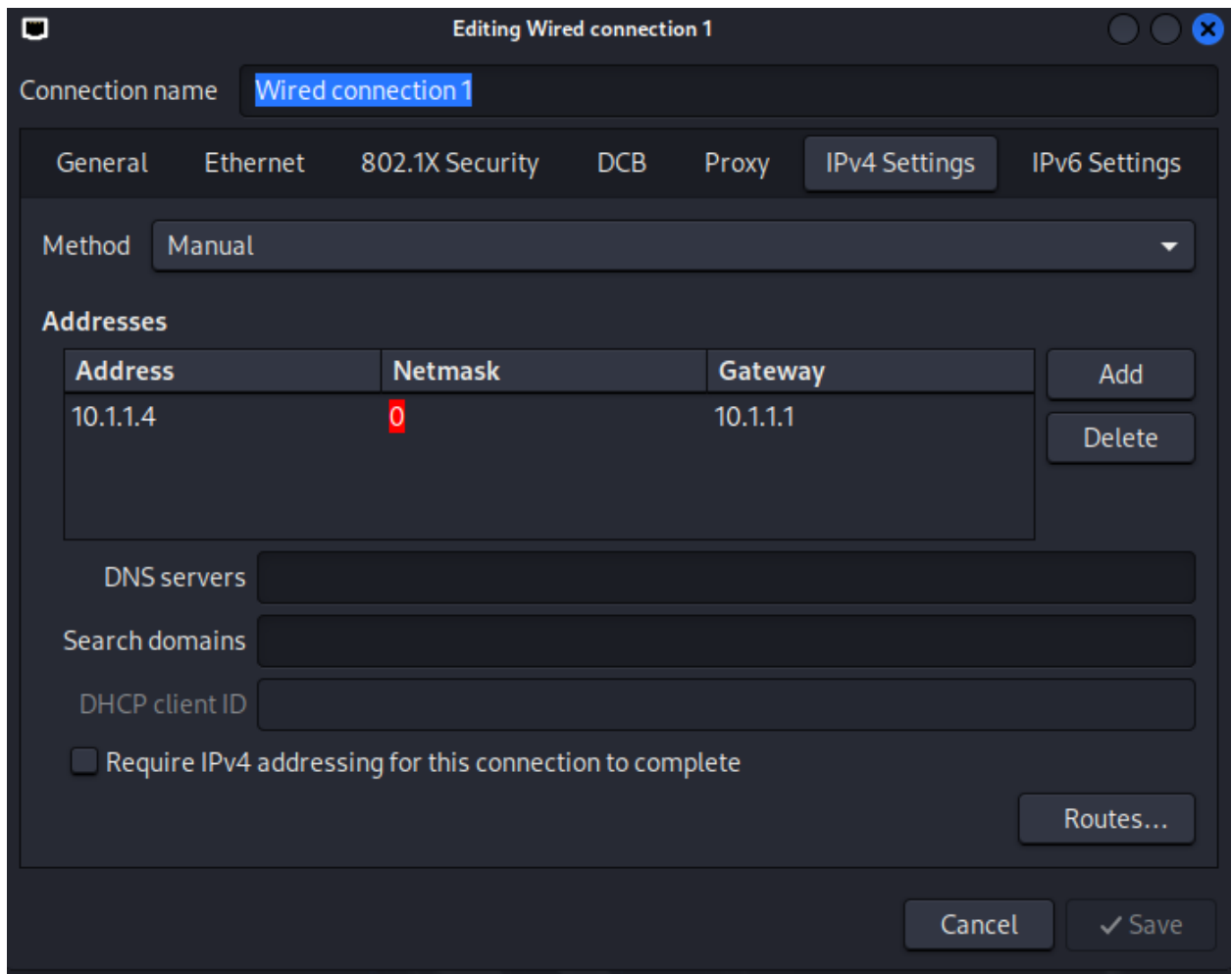
2. IP configuration

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 required:
  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:

Cancel

Wired

Apply

Details

Identity

IPv4

IPv6

Security

Link speed

1000 Mb/s

IPv4 Address

192.168.1.13

IPv6 Address

fe80::f97:a0e8:b11b:6ee8

Hardware Address

00:0C:29:31:78:61

Default Route

192.168.1.1
fe80::20c:29ff:fe54:4564

DNS

192.168.200.1

☒

Connect automatically

☒

Make available to other users

☐

Metered connection: has data limits or can incur charges
Software updates and other large downloads will not be started automatically.

Remove Connection Profile...

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

Firewall / Rules / WAN

Floating

WAN

LAN

DMZ

Rules (Drag to Change Order)

<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✗ 0/2 KiB	*	Reserved Not assigned by IANA	*	*	*	*	*		Block bogon networks	
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	WAN address	*	DMZ address	80 - 443	*	none			
<input type="checkbox"/>	✓ 0/35 KiB	IPv4 TCP	*	*	10.1.1.4	80 (HTTP)	*	none		NAT	
<input type="checkbox"/>	✗ 0/0 B	IPv4 *	WAN address	*	LAN address	*	*	none			

↑ Add

↓ Add

Delete

Toggle

Copy

Save

Separator

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
<input checked="" type="checkbox"/>	✓ 0/14.12 MiB	*	*	*	LAN Address	443 80	*	*		Anti-Lockout Rule	⚙️
<input type="checkbox"/>	✓ 0/684 KiB	IPv4 TCP	LAN subnets	*	sites	80 - 443	*	none			🔗 ⚙️ 📄 🗑️ ✖️
<input type="checkbox"/>	✗ 0/250 KiB	IPv4 TCP	LAN subnets	*	*	80 - 443	*	none			🔗 ⚙️ 📄 🗑️ ✖️
<input type="checkbox"/>	✓ ⚙️ 0/0 B	IPv4 ICMP echorep	LAN address	*	WAN address	*	*	none			🔗 ⚙️ 📄 🗑️ ✖️
<input type="checkbox"/>	✓ 0/84.84 MiB	IPv4 *	LAN subnets	*	*	*	*	none		Default allow LAN to any rule	🔗 ⚙️ 📄 🗑️ ✖️
<input type="checkbox"/>	✓ 0/0 B	IPv6 *	LAN subnets	*	*	*	*	none		Default allow LAN IPv6 to any rule	🔗 ⚙️ 📄 🗑️ ✖️
<input type="checkbox"/>	✓ 0/0 B	IPv4 ICMP any	LAN address	*	DMZ address	*	*	none			🔗 ⚙️ 📄 🗑️ ✖️
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	LAN address	*	DMZ address	22 (SSH)	*	none			🔗 ⚙️ 📄 🗑️ ✖️
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	LAN address	*	DMZ address	80 (HTTP)	*	none			🔗 ⚙️ 📄 🗑️ ✖️
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP/UDP	LAN address	*	8.8.8.8	53 (DNS)	*	none			🔗 ⚙️ 📄 🗑️ ✖️
<input type="checkbox"/>	✗ 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

Firewall / Rules / DMZ

Floating WAN LAN DMZ

Rules (Drag to Change Order)

<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 0/0 B	IPv4 ICMP any	*	*	WAN address	*	*	none			
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	DMZ address	*	WAN address	123 (NTP)	*	none			
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	DMZ address	*	WAN address	53 (DNS)	*	none			

Add Add Delete Toggle Copy Save Separator

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



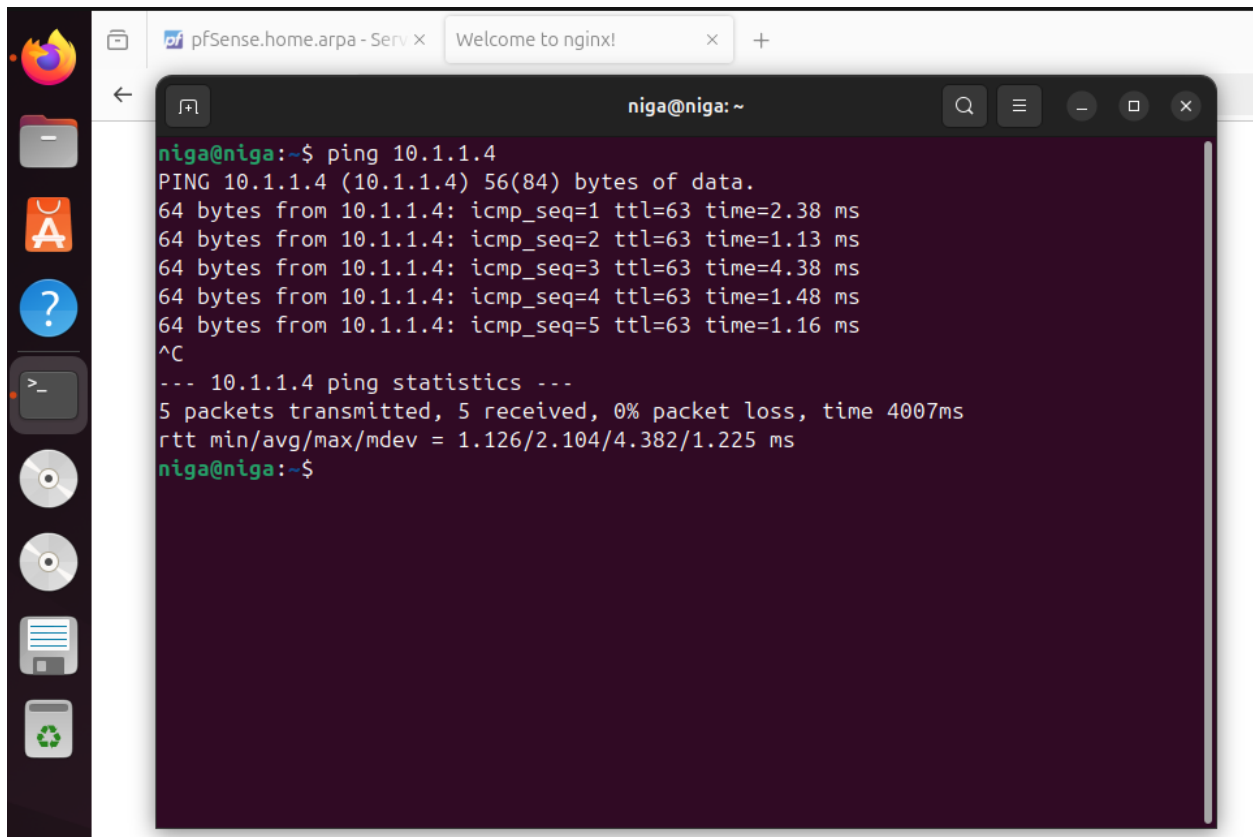
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

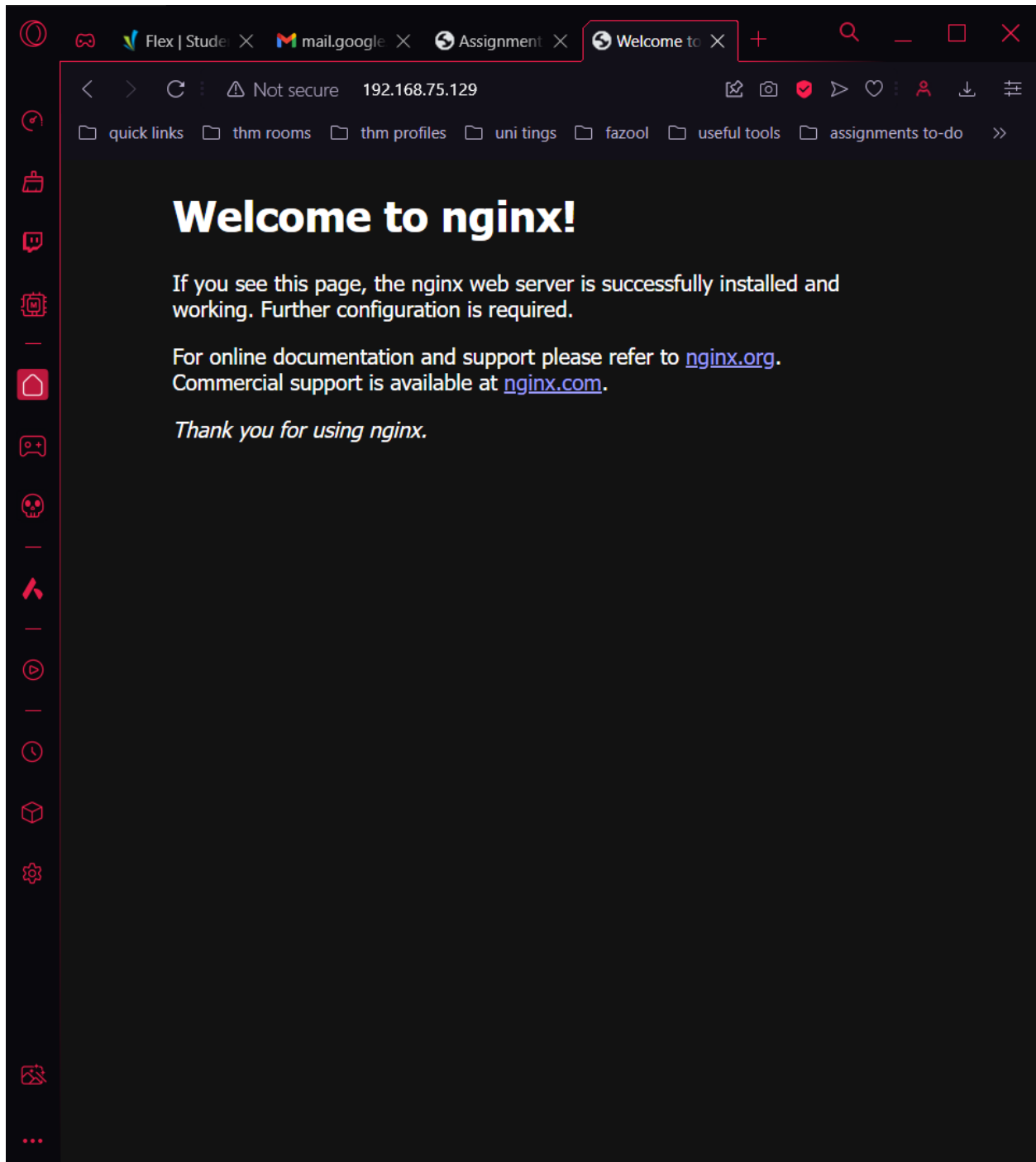
Ability to ping nginx server machine



```
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:~$
```

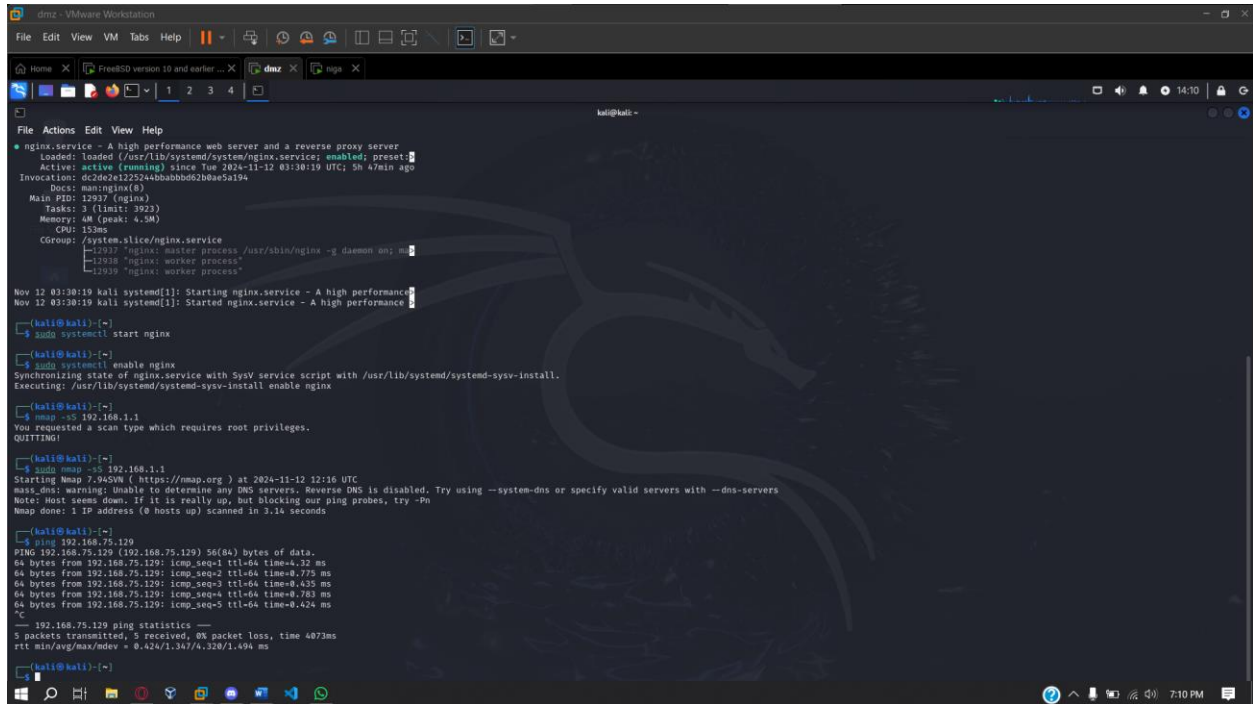
2. WAN

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



3. DMZ

I can ping from DMZ to WAN



```
File Actions Edit View Help
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: dc2de2e1225244bbab0b062b0ae5a194
Dock: main/nginx(3)
Main PID: 12937 (nginx)
Tasks: 3 (limit: 392)
Memory: 4M (peak: 4.5M)
CPU: 153ms
CGroup: /system.slice/nginx.service
├─12937 "nginx: master process /usr/sbin/nginx -g daemon on; m
├─12938 "nginx: worker process"
└─12939 "nginx: worker process"

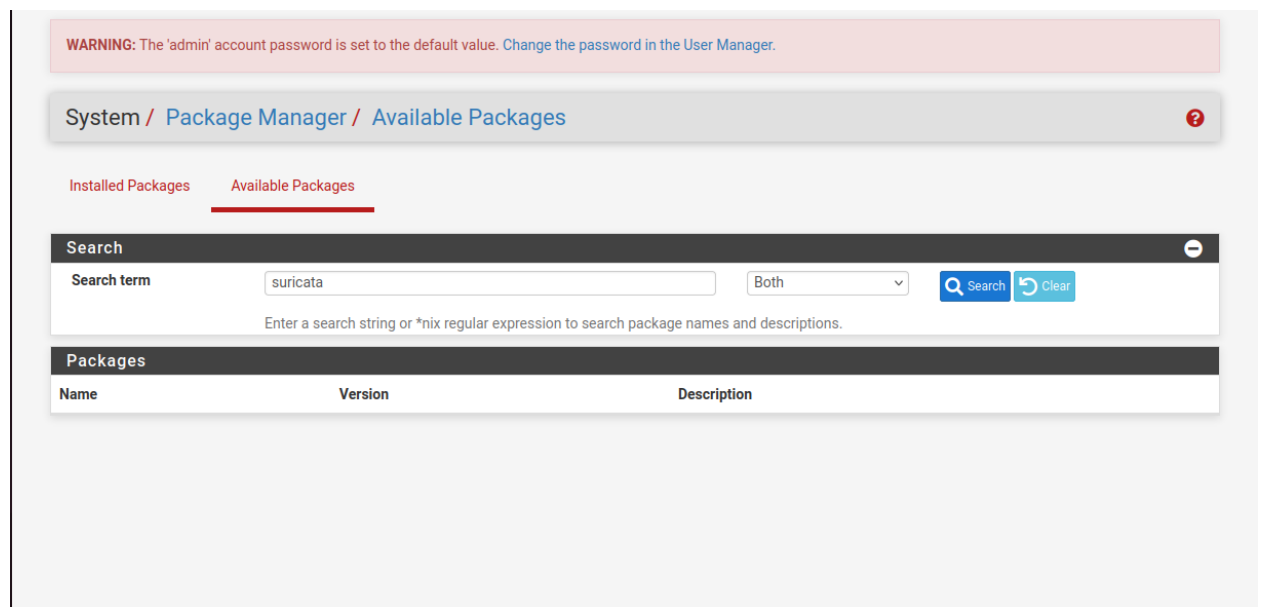
Nov 12 03:30:19 kali systemd[1]: Starting nginx.service - A high performance
Nov 12 03:30:19 kali systemd[1]: Started nginx.service - A high performance

[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
[kali@kali:~]$ nmap -sS 192.168.1.1
You requested a scan type which requires root privileges.
QUITTING!
[kali@kali:~]$ nmap -sS 192.168.1.1
Starting Nmap 7.94506 ( https://nmap.org ) at 2024-11-12 12:16 UTC
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.1s seconds
[kali@kali:~]$ ping 192.168.75.129
PING 192.168.75.129 (192.168.75.129) 56(84) bytes of data:
64 bytes from 192.168.75.129: icmp_seq=1 ttl=64 time=4.32 ms
64 bytes from 192.168.75.129: icmp_seq=2 ttl=64 time=0.775 ms
64 bytes from 192.168.75.129: icmp_seq=3 ttl=64 time=0.435 ms
64 bytes from 192.168.75.129: icmp_seq=4 ttl=64 time=0.783 ms
64 bytes from 192.168.75.129: icmp_seq=5 ttl=64 time=0.424 ms
^C
--- 192.168.75.129 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4073ms
rtt min/avg/max/mdev = 0.424/1.347/4.320/1.494 ms
[kali@kali:~]$
```

Suricata

1. Suricata Download

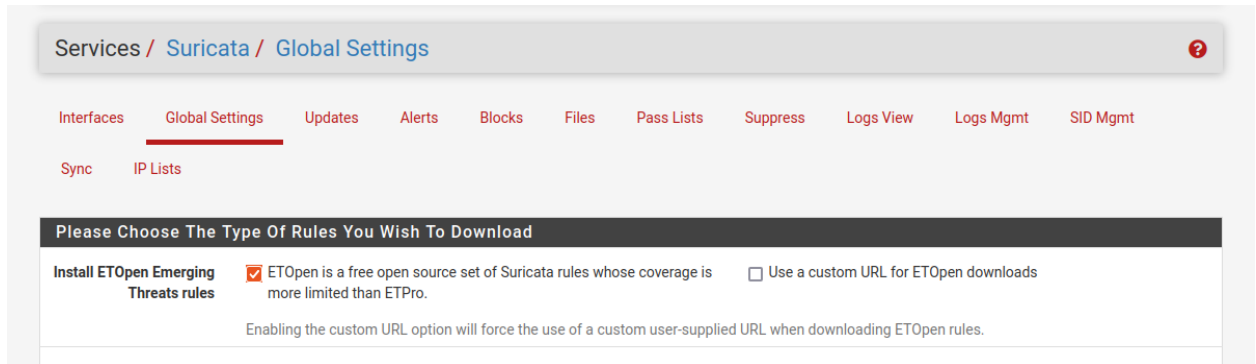
First of all, when we have to setup suricata, we have to download it so we go to the package manager and install suricata:



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

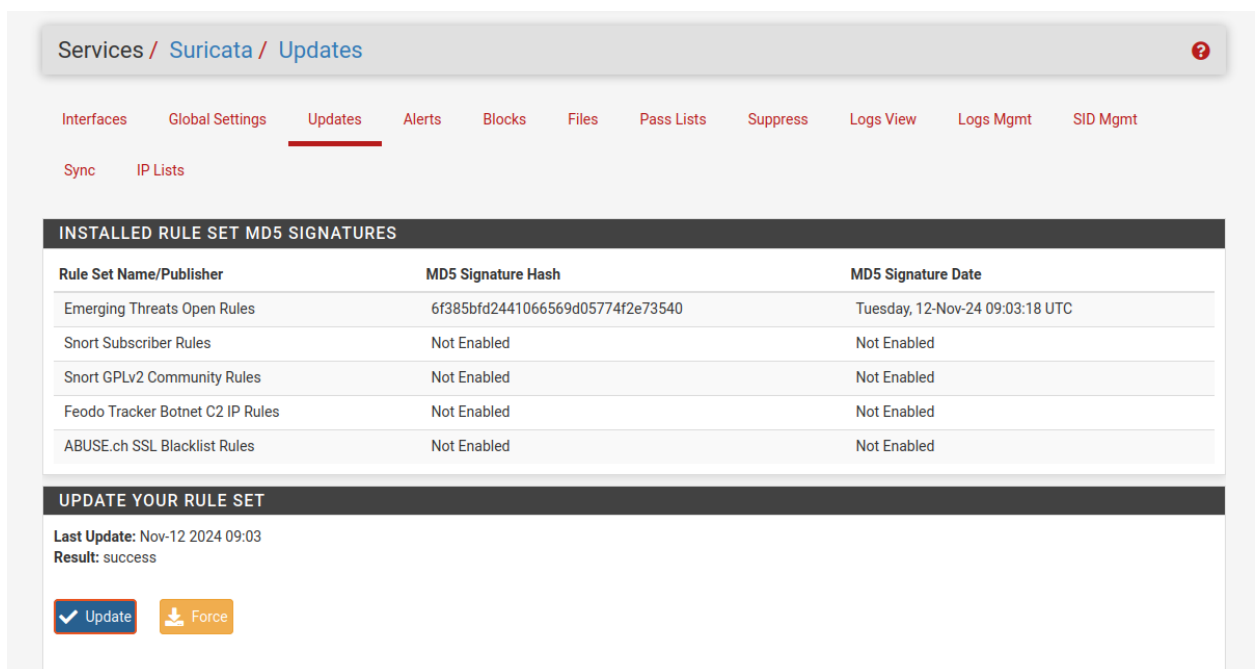
2. Downloading Rulesets

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



The screenshot shows the 'Global Settings' tab for Suricata. The breadcrumb trail is 'Services / Suricata / Global Settings'. The 'Global Settings' tab is selected in the top navigation bar. Below the navigation bar, there is a section titled 'Please Choose The Type Of Rules You Wish To Download'. It contains two main options: 'Install ETOpen Emerging Threats rules' and 'ETOpen is a free open source set of Suricata rules whose coverage is more limited than ETPro.'. The 'ETOpen' option is checked. There is also an unchecked checkbox for 'Use a custom URL for ETOpen downloads'. A note below states: 'Enabling the custom URL option will force the use of a custom user-supplied URL when downloading ETOpen rules.'

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



The screenshot shows the 'Updates' tab for Suricata. The breadcrumb trail is 'Services / Suricata / Updates'. The 'Updates' tab is selected in the top navigation bar. Below the navigation bar, there is a section titled 'INSTALLED RULE SET MD5 SIGNATURES'. It contains a table with the following data:

Rule Set Name/Publisher	MD5 Signature Hash	MD5 Signature Date
Emerging Threats Open Rules	6f385bfd2441066569d05774f2e73540	Tuesday, 12-Nov-24 09:03:18 UTC
Snort Subscriber Rules	Not Enabled	Not Enabled
Snort GPLv2 Community Rules	Not Enabled	Not Enabled
Feodo Tracker Botnet C2 IP Rules	Not Enabled	Not Enabled
ABUSE.ch SSL Blacklist Rules	Not Enabled	Not Enabled

Below the table, there is a section titled 'UPDATE YOUR RULE SET'. It shows the 'Last Update' as 'Nov-12 2024 09:03' and the 'Result' as 'success'. At the bottom, there are two buttons: 'Update' (with a checkmark icon) and 'Force' (with a download icon).

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
<input checked="" type="checkbox"/> DMZ (em2)	<div><div>✓</div><div>↺</div><div>↻</div></div>	AUTO	DISABLED	DMZ	<div><div>✎</div><div>📄</div><div>🗑</div></div>
<input type="checkbox"/> WAN (em0)	<div><div>✓</div><div>↺</div><div>↻</div></div>	AUTO	DISABLED	WAN	<div><div>✎</div><div>📄</div><div>🗑</div></div>

+

 Add

🗑

 Delete

i

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.
<input checked="" type="checkbox"/>	app-layer-events.rules	<input checked="" type="checkbox"/>	emerging-3coresec.rules	
<input checked="" type="checkbox"/>	decoder-events.rules	<input checked="" type="checkbox"/>	emerging-activex.rules	
<input checked="" type="checkbox"/>	dhcp-events.rules	<input checked="" type="checkbox"/>	emerging-adware_pup.rules	

Services / Suricata / Alerts

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

Sync
IP Lists

Alert Log View Settings

Instance to View
(WAN) WAN

Choose which instance alerts you want to inspect.

Save or Remove Logs
Download

All alert log files for selected interface will be downloaded

Save Settings
Save

Refresh
☒

Default is ON

Clear

Clear the currently active Alerts log file

250

Number of alerts to display. Default is 250

Alert Log View Filter

Last 250 Alert Entries. (Most recent entries are listed first)

Date	Action	Pri	Proto	Class	Src	SPort	Dst	DPort	GID:SID	Description
11/12/2024 13:32:11	⚠	3	UDP	Generic Protocol Command Decode	192.168.75.129	64400	192.5.6.30	53	1:2200075	SURICATA UDPv4 invalid checksum
11/12/2024 13:32:11	⚠	3	UDP	Generic Protocol Command Decode	192.168.75.129	20788	192.54.112.30	53	1:2200075	SURICATA UDPv4 invalid checksum
11/12/2024 13:32:11	⚠	3	UDP	Generic Protocol Command Decode	192.168.75.129	16080	205.251.184.167	53	1:2200075	SURICATA UDPv4 invalid checksum

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.