

OUTSIDE-FIREWALL

Overview

This project involved setting up and configuring a multi-VM network environment using VirtualBox, pfSense, Ubuntu Linux, and Splunk Enterprise. The objective was to design and deploy an **outside-firewall (pfSense)**, configure a **DMZ client**, and establish a **Splunk server** for network monitoring and analysis.

The successful implementation provided a functional test environment for virtual networking, packet inspection, and log analysis, forming the foundation for future security labs as shown in our diagram.

1. pfSense – Outside-Firewall

I successfully created a virtual machine with the features described below just as shown by the screenshot attached .I completed a prior step of downloading pfSense firewall from <https://www.pfsense.org/download/> which I used the following information for the specific image as it might differ given differences in geographical area, platform and architecture etc.

-File Type: **Install**

-Architecture: **AMD64 (64-bit)**

-Platform: **CD Image (ISO) Installer**

-Mirror: **New York City, USA**

The purpose of this install was that the machine would have two network interfaces, one for the WAN which connected to the internet and the other for the LAN which locally connected to the virtual machines I created later referencing the Network Diagram. **NOTE:** The file was downloaded as a GZip file and to uncompress the pfSense.gz file I used 7-zip found at <https://www.7-zip.org/> .


VM Creation


- Name: *Outside-Firewall*
- OS Type: Linux (64-bit)
- Memory: 512 MB
- Disk: 10 GB

← Create Virtual Machine

Name and operating system

Name:

Type: 

Version: 

Memory size

MB

4 MB 16384 MB

Hard disk

☐ Do not add a virtual hard disk

☒ Create a virtual hard disk now

☐ Use an existing virtual hard disk file




Figure 1/Vm creation

- Network Interfaces:

Below features show the network configuration accessible by clicking the settings button and navigating to the network icon.

- Adapter 1: I enabled it and attached to NAT (WAN)

Outside-Firewall - Settings

Network

Adapter 1 Adapter 2 Adapter 3 Adapter 4

☒ Enable Network Adapter

Attached to:

Name:

▶ Advanced

Figure 2/Network configuration

- Adapter 2: enabled it and attached to Internal Network. Then I changed the intent name to **DMZ**. It is good to also take note of the MAC address generated under the advanced button which will be used later.

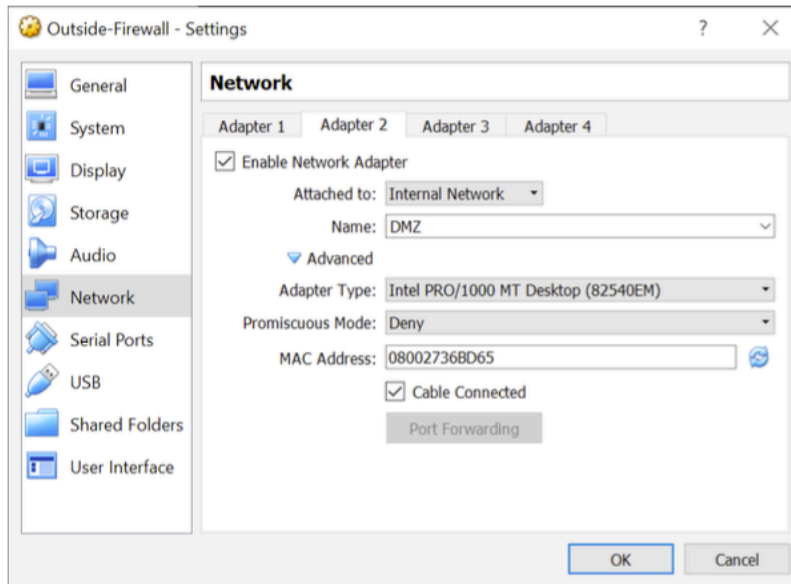


Figure 3/DMZ

Installation

- Deployed the Vm by attaching pfSense-CE-2.5.2-RELEASE-amd64.iso as my virtual optical disk file

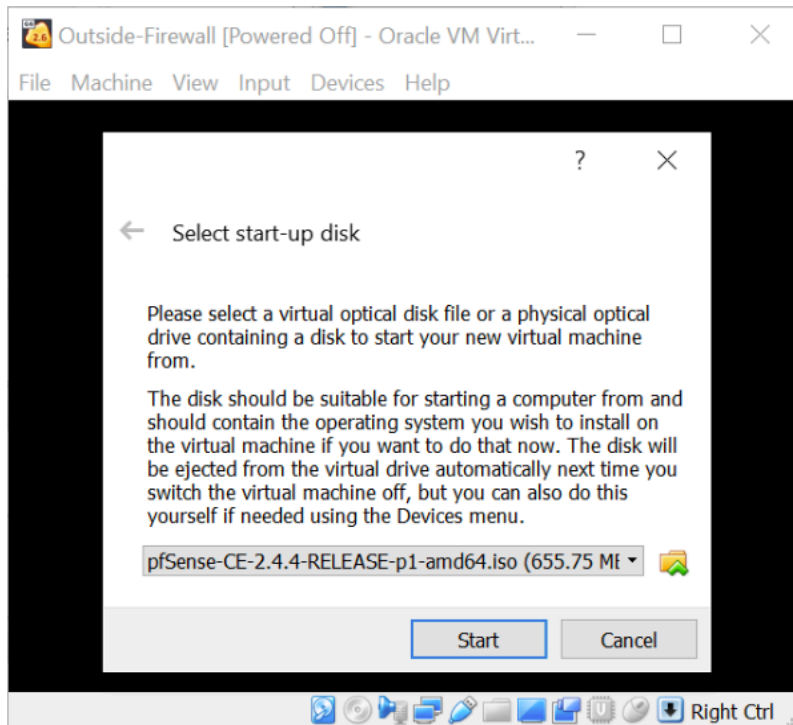


Figure 4/pfSense

After rebooting I removed the virtual CDROM image so that it didn't keep looping back to the same page as shown in the screenshot above.

- Configured pfSense with **WAN (em0)** for internet access and **LAN (em1)** as 192.168.4.1/24.
- Then I disabled DHCP and confirmed internet connectivity using ping tests to external hosts (e.g. google.com, yahoo.com, 8.8.8.8)
- I achieved the above by going through the following steps:

-Selected **Option 2 (Set interfaces IP address)** for the LAN interface as successfully shown in the screenshot below. The following prompts show how I configured it with written answers.

-LAN IPv4 address: **192.168.4.1**

-Subnet Mask: **24**

-LAN upstream gateway: press **<Enter>**

-LAN IPv6 address: press **<Enter>**

-Do you want to enable the DHCP server on LAN? **N (no)**

-Do you want to revert to HTTP as the web Configurator protocol? **N(no)**

-Press **<Enter>** to continue.

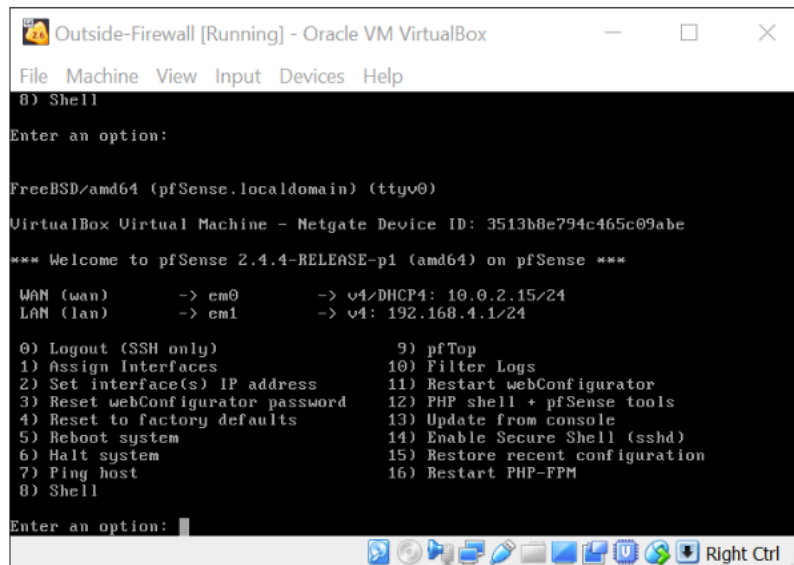


Figure 5/Interface

Web Interface

- I accessed pfSense GUI at: <http://192.168.4.1> using default credentials: admin/pfSense which was vital for the other labs.
- Successfully verified web configurator access.

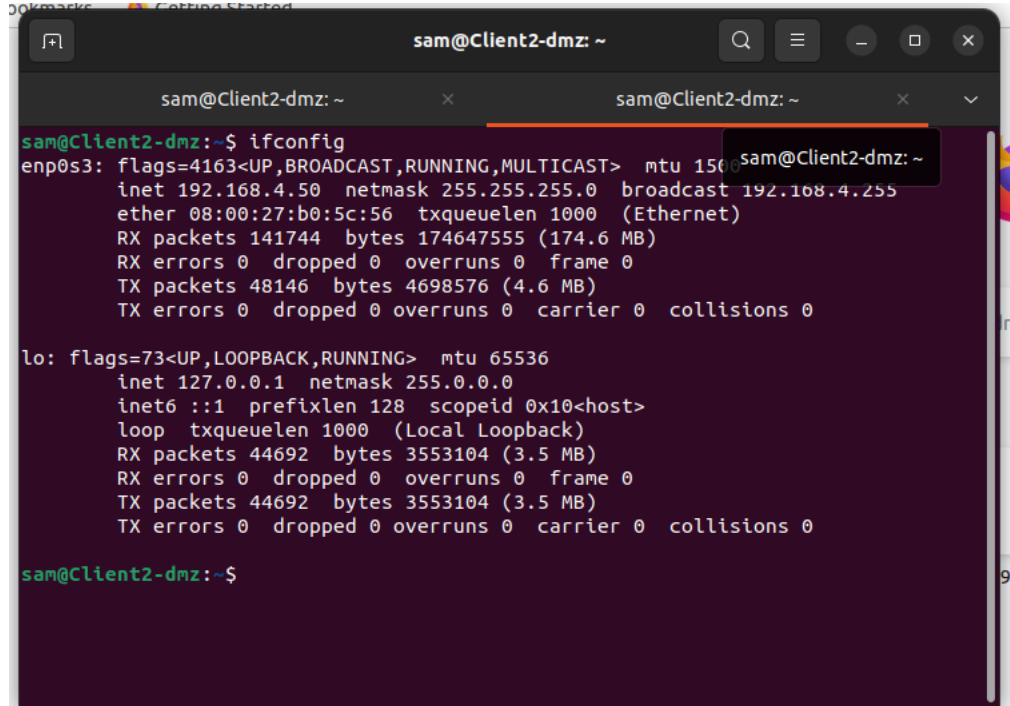
2. Client2-DMZ (Ubuntu Linux)

- **VM Creation**
 - Cloned from existing Ubuntu client1 created earlier.
 - Configured Adapter 1 → Internal Network (*DMZ*).
 - Enabled *Promiscuous Mode: Allow VMs* for traffic sniffing.
- **Networking**
 - Assigned static IP after starting Client2-DMZ and navigated to the networking on the top right of our ubuntu Vm under “Edit Connections” which I populated as follows:
 - IP: 192.168.4.50
 - Netmask: 255.255.255.0
 - Gateway: 192.168.4.1
 - DNS: 8.8.8.8
 - I saved the changes and to make sure that they took effect I ran the following commands on my terminal:

-sudo ifconfig enp0s3 down

-sudo ifconfig enp0s3 up

-ifconfig - This returned out now new Ip address of interface enp0s3



```
sam@Client2-dmz: ~  
$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.4.50 netmask 255.255.255.0 broadcast 192.168.4.255  
    ether 08:00:27:b0:5c:56 txqueuelen 1000 (Ethernet)  
    RX packets 141744 bytes 174647555 (174.6 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 48146 bytes 4698576 (4.6 MB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 44692 bytes 3553104 (3.5 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 44692 bytes 3553104 (3.5 MB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
sam@Client2-dmz: ~$
```

Figure 6/ifconfig

- **Verification**
 - Successfully pinged pfSense LAN IP (192.168.4.1): **sudo ping 192.168.4.1**
 - Accessed pfSense web interface from Ubuntu browser: <http://192.168.4.1> using the default password **admin/pfsense**
- **Traffic Analysis**
 - Installed and ran **Wireshark**: **~\$ sudo wireshark** - This command brought up the wireshark GUI prompting me to accept that I was running it under superuser (sudo). After clicking OK I chose **any** from the interfaces list which began to capture and analyze traffic on all interfaces.
 - Captured ICMP echo reply and echo response traffic while pinged 192.168.4.1. as shown in the wireshark screenshot below.
 - Verified real-time network packet visibility from DMZ client.

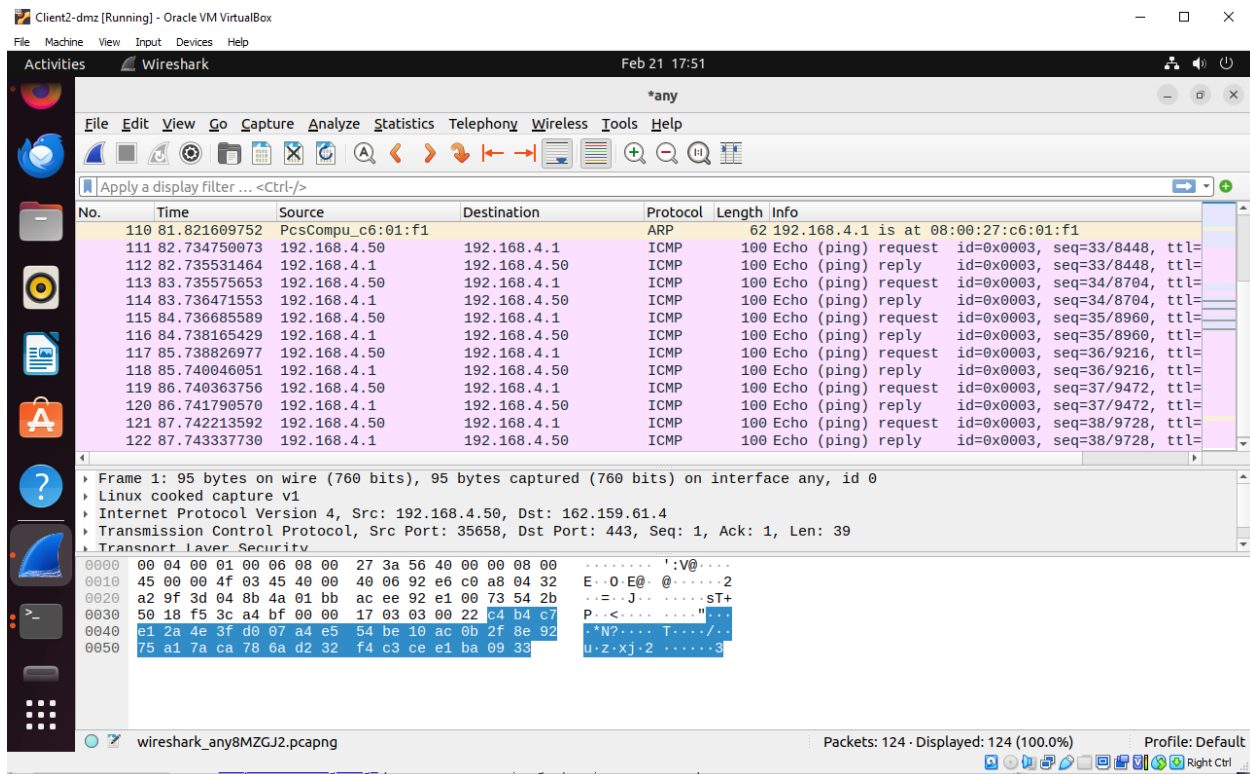


Figure 7/wireshark

3. Splunk-DMZ

- **VM Creation-** The creation of this virtual machine was like the Ubuntu install we did earlier. The only difference is that Splunk-DMZ required more memory and hard drive space as shown below:
 - Name: *Splunk-DMZ*
 - OS: Ubuntu (64-bit)
 - Memory: 2048 MB
 - Disk: 100 GB
 - Network: Internal (*DMZ*)
- **Installation & Configuration**
 - Installed Ubuntu Desktop 18.04 by loading the image, after starting the machine I had created and receiving the window pop that requested for the image to load from. For further instructions to install Ubuntu visit:
<http://www.ubuntu.com/download/desktop/install-ubuntu-desktop>
 - Installed required packages:

```
: ~$ sudo apt-get update
: ~$ sudo apt-get upgrade
: ~$ sudo apt-get install net-tools
```

```
: ~$ sudo apt-get install git
: ~$ sudo apt-get install htop
: ~$ sudo apt-get install traceroute
: ~$ sudo apt-get install nmap
: ~$ sudo apt-get install wireshark
: ~$ sudo apt-get install curl
```

- **Splunk Enterprise Software**

- Downloaded and installed Splunk Enterprise (splunk-*****-amd64.deb and splunkforwarder-*****-amd64.deb) on the Splunk-DMZ Vm from <http://www.splunk.com> .**NOTE:** This download is for the 64-bit Linux/Debian version.
- Configured Splunk to start at boot using the commands below in my terminal window:

```
: ~$ cd /home/<username>/Downloads
: ~$ sudo dpkg -i splunk-*****-amd64.deb
: ~$ cd /opt/splunk/bin
: ~$ sudo /opt/splunk/bin/splunk start --accept-license
: ~$ sudo /opt/splunk/bin/splunk enable boot-start
: ~$ sudo systemctl start splunk
: ~$ sudo systemctl status splunk
```

- Configured credentials for my Splunk GUI which was accessible via <https://localhost:8000> .
- Set hostname: *Splunk-DMZ* by editing the /etc/hosts and /etc/hostname file (sudo nano /etc/hosts, sudo nano /etc/hostname)
- Assigned static IP: 192.168.4.20 and rebooted the Vm to make sure the changes take effect. **NOTE:** The outside firewall should be up and running to allow internet access for this newly created machine.

- **Verification**

- Accessed Splunk Web at: <http://localhost:8000>.
- Logged in with admin credentials created during setup.
- Verified service status with systemctl status splunk.
- Confirmed external connectivity with ping tests (cnn.com, gmail.com):

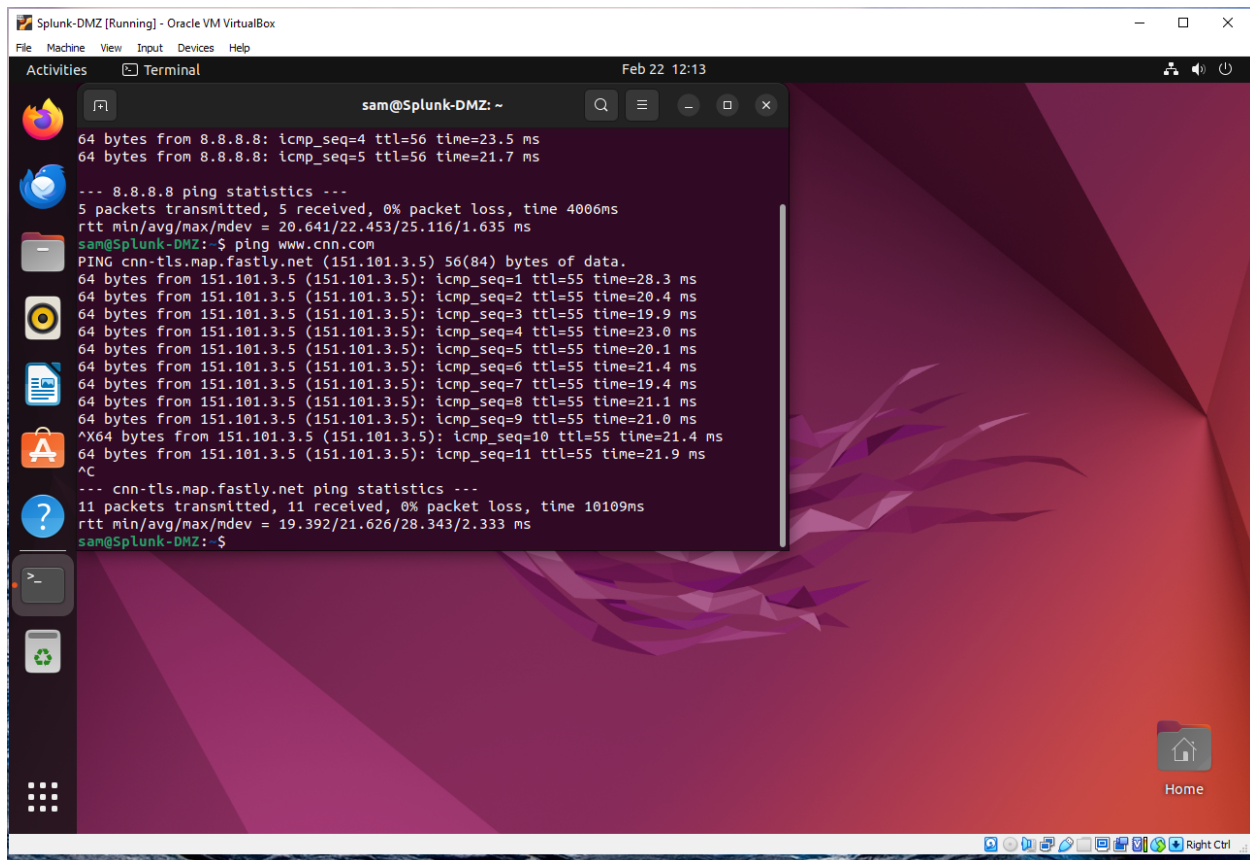


Figure 8/Splunk-DMZ