

CIS 221 Project

Server Security Hardening, Real-World Threats, Policies, and
Incident Response

Armaan Chima #300211286
Date: August 24th, 2025

Will I be designing a Cybersecurity Plan for a PC or a Small Business System?

I will be designing a cybersecurity plan for a **PC (Personal Computer)**

Note: This is done on an Ubuntu Server, not on my own PC

Potential Real-world Threats to a Personal Computer

Potential threats to a personal computer are...

- 1.) **Malware** -> This can range from worms, adware, rootkits, keyloggers, and more. Malware can enter a PC if the user interacts with malicious websites, opens unknown emails from unknown senders (specifically accessing attachments in those emails), downloads unsecure or infected downloads to his or her PC, or inserts infected USBs into their computer. These are just a few examples of how malware can pose a threat to a personal computer if the owner is not careful.
- 2.) **Phishing and Ransomware** -> Despite phishing and ransomware falling under the category of malware, it is still essential to know what each one is and how it poses a threat to your PC, as these two types of malware are credited to be more prevalent compared to other types.
 - **Ransomware** -> Ransomware is a type of malicious software that blocks access to the user's PC until a certain amount of money or ransom is paid. Essentially holding your PC hostage. However, even after the sum is paid, there is still a possibility that your PC will not be restored to its original state. Potentially leaving the user in a lose-lose situation or dilemma.
 - **Phishing** -> Phishing is the process of sending fraudulent emails or messages. However, there is a catch: the sender poses as a legitimate individual from well-known companies like Google or Facebook. As a result, this lures people into giving sensitive or personal information such as passwords, financial information, or any other information that may cause a threat to you.

- 3.) **DOS (Denial-of-Service) Attacks** -> DOS is a type of cyberattack when an attacker disturbs a computer's (in this case PC) normal operation. This is achieved by overwhelming the computer with unnecessary traffic, which overpowers the computer or exploits vulnerabilities, ultimately causing the computer to become unusable.
- 4.) **Zero-day Exploits** -> Zero-day exploits occur when an attacker exploits a vulnerability unknown to the user. An attacker usually finds a loophole in the computer security, and that is used to attack a system (in this case, a PC). It is called a zero-day exploit because it is a type of cyberattack that the user has zero days to prepare for, since it has been newly exploited.
- 5.) **Man-in-the-Middle Attacks** -> A Man-in-the-Middle attack is when an attacker finds poor web-based protocols and exploits them. Specifically, the attacker will put themselves in between the 2 senders in a communication channel, essentially intercepting and stealing any valuable or sensitive data. Usually, the attacker is well hidden in an MITM attack.
- 6.) **Preventative Basics** -> All these examples, in some shape or form, do damage or attack a PC. However, in this section, I want to get into the basics of how people can access PCs in an unauthorized manner. This can be done by using weak or reused passwords to secure accounts, not updating or using obsolete software, and connecting to unsecured Wi-Fi are a few ways that attackers can access your PC. In addition, all of these issues I just mentioned can be easily prevented and do not require extensive knowledge about cyber-related attacks, allowing anyone to keep their PCs safe from attackers by simply practicing the basics.

Security Policy for a Personal Computer

Purpose:

The purpose of this policy is to understand how the PC (the server) will be sheltered and secured from malicious activity, unauthorized access, alteration, or impairment. This policy aims to preserve the PC's integrity. The main objective is to protect sensitive details, reduce security threats, and uphold a reliable and safe security standard.

Scope:

This policy applies only to those who use the PC (specifically, the Ubuntu server on the VirtualBox application).

Policies:

- **Access Control**
 - The server can only be accessed by permitted users
 - Access via SSH is not allowed or permitted
 - If a user fails to enter their credentials correctly in 3 attempts, they will be jailed (IP address is blocked) for 10 minutes until they can try again
- **System Updates**
 - Automatic updates regularly occur due to the unattended-upgrades package installation.
- **Data Privacy**
 - Sensitive or important files will be authorized to members who have root access
 - Any files that need to be copied or reinforced will be done, and preserved
- **Security Monitoring**
 - Snort (IDS) detects and records any activity that is doubtful or suspicious
 - These records are kept for the admin to analyze
- **Firewall Deployment**
 - All ports that are left unused will be closed to minimize the attack surface
 - UFW is implemented to filter traffic, allowing or disallowing traffic into the network.
 - Restricted traffic will only head to open ports

Incident Response Plan

Purpose:

The purpose of an incident response plan is to act in an organized and quick manner to deal with any attacks or breaches that the PC is facing.

Approach:

Before an attack occurs, there are simple things that can be done to stop the attack from occurring and prevent information from being leaked. These things are...

- Having UFW, Snort, and Fail2Ban running
- Having sensitive files backed up
- Constantly having your system updated (the unattended-upgrades command in effect)

Dealing with an Attack

- **Identifying an attack**
 - Analyze Fail2Ban logs. Look for jailed IP addresses with multiple unsuccessful login efforts
 - Inspect Snort logs for suspicious activity
 - Examine system logs (snort/fail2ban logs) on your PC for abnormal behaviour
- **Isolation**
 - If suspicious activity is in play, detach it from your network immediately to avoid further damage to the PC.
- **Removal**
 - Reconfigure any files or services that may have been altered
 - If the PC has software that is unreliable or contains security flaws, update it
 - Get rid of all infected files, malicious software and unwarranted/unauthorized accounts that may be placed on the PC
- **Recovering from an Attack**
 - If your system is saved, recover it to its normal and unaffected state
 - Restart your PC
 - Restart Services and test the function of all logs to ensure they are all functional
- **Realizations after the Attack**
 - Understand where the attack came from
 - Log the attack and log the process by which you took to recover from it
 - Reconfigure the essential services, such as UFW, Fail2ban, and Snort, to ensure attacks such as the one faced do not occur again

File Information and Access

```
total 88
drwx----- 16 ArmaanChima ArmaanChima 4096 Aug 23 20:45 .
drwxr-xr-x  3 root      root     4096 Aug 16 04:19 ..
-rw-----  1 ArmaanChima ArmaanChima  829 Aug 23 13:18 .bash_history
-rw-r--r--  1 ArmaanChima ArmaanChima 220 Mar 31 2024 .bash_logout
-rw-r--r--  1 ArmaanChima ArmaanChima 3771 Mar 31 2024 .bashrc
drwx----- 11 ArmaanChima ArmaanChima 4096 Aug 16 17:54 .cache
drwx----- 12 ArmaanChima ArmaanChima 4096 Aug 16 17:35 .config
drwxr-xr-x  2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Desktop
drwxr-xr-x  2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Documents
drwxr-xr-x  2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Downloads
-rw-rw-r--  1 ArmaanChima ArmaanChima   32 Aug 23 20:45 filename.txt
drwx-----  2 ArmaanChima ArmaanChima 4096 Aug 23 20:18 .gnupg
drwx-----  4 ArmaanChima ArmaanChima 4096 Aug 16 17:29 .local
drwxr-xr-x  2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Music
drwxr-xr-x  2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Pictures
-rw-r--r--  1 ArmaanChima ArmaanChima  807 Mar 31 2024 .profile
drwxr-xr-x  2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Public
drwx-----  3 ArmaanChima ArmaanChima 4096 Aug 16 04:31 snap
drwx-----  2 ArmaanChima ArmaanChima 4096 Aug 16 04:20 .ssh
-rw-r--r--  1 ArmaanChima ArmaanChima    0 Aug 16 04:23 .sudo_as_admin_successfu
l
drwxr-xr-x  2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Templates
drwxr-xr-x  2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Videos
```

Note: ls-la allows you to see all files in thorough detail

The file that will be used to explain file access is filename.txt (Shown in the photo)

In the above photo, we can see that the filename.txt contains the permission -rw-rw-r--, which means that the owner and the members of the filename.txt group have read and write permissions, but no execute permissions, while others (r--) can only read.

```
drwx----- 11 ArmaanChima ArmaanChima 4096 Aug 16 17:54 .cache
drwx----- 12 ArmaanChima ArmaanChima 4096 Aug 16 17:35 .config
drwxr-xr-x  2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Desktop
drwxr-xr-x  2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Documents
drwxr-xr-x  2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Downloads
-rw-rw-r--  1 ArmaanChima ArmaanChima   32 Aug 23 20:45 filename.txt
drwx----- 2 ArmaanChima ArmaanChima 4096 Aug 23 20:18 .gnupg
drwx----- 4 ArmaanChima ArmaanChima 4096 Aug 16 17:29 .local
drwxr-xr-x  2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Music
drwxr-xr-x  2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Pictures
-rw-r----  1 ArmaanChima ArmaanChima  807 Mar 31 2024 .profile
drwxr-xr-x  2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Public
drwx----- 3 ArmaanChima ArmaanChima 4096 Aug 16 04:31 snap
drwx----- 2 ArmaanChima ArmaanChima 4096 Aug 16 04:20 .ssh
-rw-r----  1 ArmaanChima ArmaanChima     0 Aug 16 04:23 .sudo_as_admin_successfu
l
drwxr-xr-x  2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Templates
drwxr-xr-x  2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Videos
-rw-----  1 ArmaanChima ArmaanChima     0 Aug 16 05:24 .Xauthority
-rw-----  1 ArmaanChima ArmaanChima  509 Aug 16 04:30 .xsession-errors
ArmaanChima@ArmaanChima:~$ chmod 640 filename.txt
ArmaanChima@ArmaanChima:~$ ls -l filename.txt
-rw-r----  1 ArmaanChima ArmaanChima 32 Aug 23 20:45 filename.txt
ArmaanChima@ArmaanChima:~$
```

In this photo, we can see that the permissions have changed to -rw-r----, which means that the owner of the file has read and write permissions, and the members of the filename.txt group have only read permissions, while others (r--) have no permissions

Using the command chmod 640 filename.txt, we are changing the permission of the file since the numbers in this command have meaning: 6 means read and write, 4 means read, and 0 means no access.

```
info: Adding new user 'mark' to supplemental / extra groups 'users' ...
info: Adding user 'mark' to group 'users' ...
ArmaanChima@ArmaanChima:~$ sudo usermod -aG ArmaanChima mark
ArmaanChima@ArmaanChima:~$ sudo chown :ArmaanChima filename.txt
ArmaanChima@ArmaanChima:~$ chmod 664 filename.txt
ArmaanChima@ArmaanChima:~$ ls -l
total 40
drwxr-xr-x 2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Desktop
drwxr-xr-x 2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Documents
drwxr-xr-x 2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Downloads
-rw-rw-r-- 1 ArmaanChima ArmaanChima 32 Aug 23 20:45 filename.txt
drwxr-xr-x 2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Music
drwxr-xr-x 2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Pictures
drwxr-xr-x 2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Public
drwx----- 3 ArmaanChima ArmaanChima 4096 Aug 16 04:31 snap
drwxr-xr-x 2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Templates
drwxr-xr-x 2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Videos
ArmaanChima@ArmaanChima:~$ groups
ArmaanChima adm cdrom sudo dip plugdev lxd
ArmaanChima@ArmaanChima:~$ ls -l filename.txt
-rw-rw-r-- 1 ArmaanChima ArmaanChima 32 Aug 23 20:45 filename.txt
ArmaanChima@ArmaanChima:~$ getent group ArmaanChima
ArmaanChima:x:1000:mark
ArmaanChima@ArmaanChima:~$
```

A random user has been created and holds permissions in filename.txt. This shows that the owner can add users to groups.

What Tools can be used to keep a System Safe?

Firewall -> A firewall is a protective tool that functions as a wall or barricade between your PC and the internet. It allows or blocks any incoming or outgoing traffic based on a predefined set of rules and inspects packets, ultimately allowing or disallowing entry to your PC. In addition, firewalls are one of the few tools that cannot be penetrated, making them a valuable asset in keeping computer integrity.

File Permissions -> File permissions determine who can access or modify content within a file or directory. There are three main options that a user can have when it comes to file permissions: they can either read, write or execute a file. Furthermore, an admin can set permissions for each user, determining whether they can read, write, or execute a file, depending on the permissions set.

Encryption -> Encryption is the process of protecting data by turning it into unreadable text. This prevents unauthorized individuals from easily accessing sensitive data. For the intended user to access the encrypted data, that person needs an encryption key. As a result, the data can only be accessed if you have the intended key. Moreover, there are 2 types of encryptions, they are...

- **Symmetric Encryption** -> The sender and the receiver use the same key
- **Asymmetric Encryption** -> The sender and the receiver both have different keys

Screenshots of the Installation and Setup Process for the Ubuntu Server

1.) The first thing I did was set up an **Ubuntu server** on VirtualBox (This acts as the PC)

Once the server was set up and ready to go, I began configuring and adding protective measures that I thought were necessary. Below, I have attached a mix of explanations and photographs to show my hardening process.

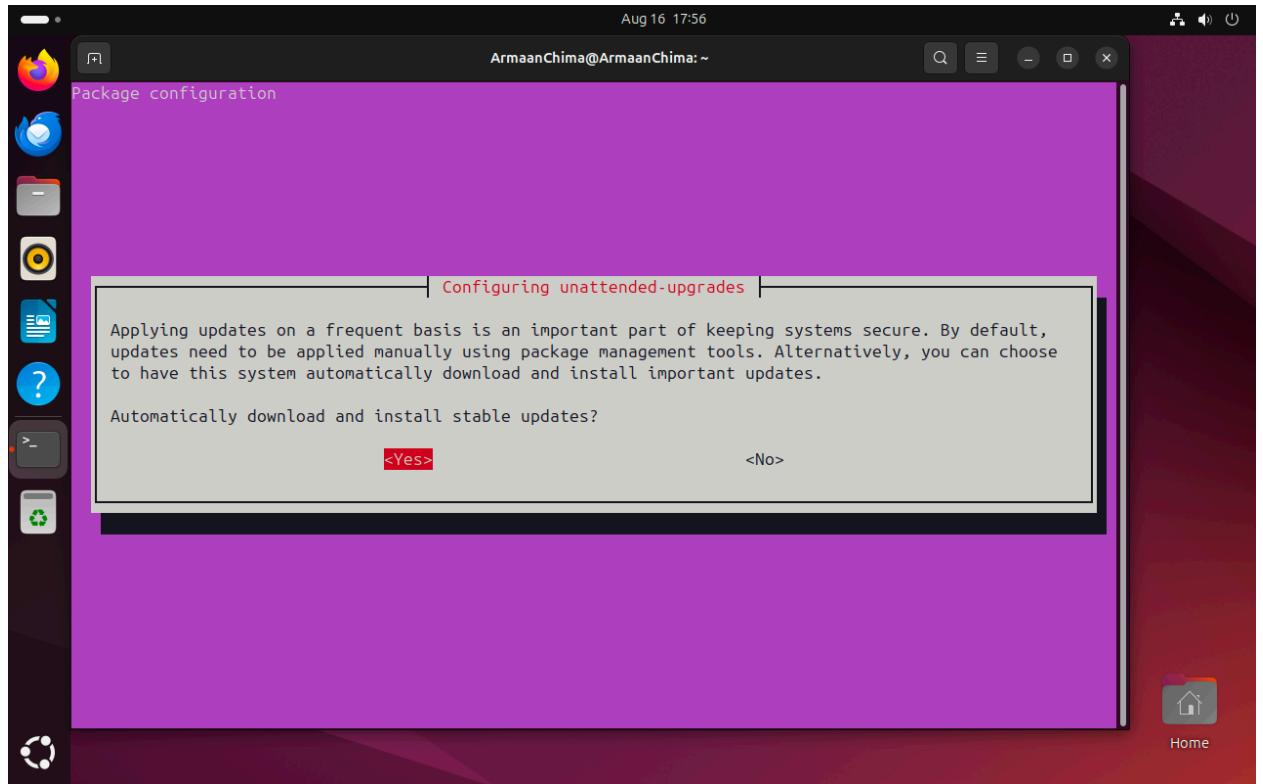
2.) Automatic upgrade configuration and installation

The processor microcode seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.

```
ArmaanChima@ArmaanChima:~$ sudo apt install unattended-upgrades -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
unattended-upgrades is already the newest version (2.9.1+nmu4ubuntu1).
unattended-upgrades set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ArmaanChima@ArmaanChima:~$ sudo dpkg-reconfigure --priority=low unattended-upgrades
Unknown option: prioirty
Usage: dpkg-reconfigure [options] packages
      -u, --unseen-only          Show only not yet seen questions.
      --default-priority        Use default priority instead of low.
      --force                   Force reconfiguration of broken packages.
      --no-reload               Do not reload templates. (Use with caution.)
      -f, --frontend             Specify debconf frontend to use.
      -p, --priority              Specify minimum priority question to show.
      --terse                  Enable terse mode.

ArmaanChima@ArmaanChima:~$ sudo dpkg-reconfigure --priority=low unattended-upgrades
ArmaanChima@ArmaanChima:~$ sudo dpkg-reconfigure --priority=low unattended-upgrades
ArmaanChima@ArmaanChima:~$
```

This image shows me using the command “`sudo apt install unattended-upgrades -y`”. In simple terms, the purpose of this command is to install the unattended-upgrades package on the system. To break it down, **sudo** means the command is run with superuser privileges, **apt** (advanced package tools) is a package manager used to help with installing the unattended-upgrades package, **install** works with apt to install the unattended-upgrades, **unattended-upgrades** is the package name and **-y** basically means that any prompts given to you will be responded to with a yes instead of the user having to type it in manually.



In the previous image, the command “`sudo dpkg-reconfigure --priority=low unattended-upgrades`” was never talked about, as it is correlated to this image. **Sudo**, which was already talked about, means the command is run with superuser privileges. **dpkg-reconfigure** allows you to reconfigure the settings, allowing you to update the unattended-upgrades package. **--priority=low** means that all possible options for configuration will appear when using this command, and unattended-upgrades is the package that is being reconfigured.

3.) UFW Firewall installation and configuration

```
ArmaanChima@ArmaanChima: ~
Get:47 http://ports.ubuntu.com/ubuntu-ports noble-security/multiverse Icons (48x48) [29 B]
Get:48 http://ports.ubuntu.com/ubuntu-ports noble-security/multiverse Icons (64x64) [29 B]
Get:49 http://ports.ubuntu.com/ubuntu-ports noble-security/multiverse Icons (64x64@2) [29 B]
Fetched 13.0 MB in 1s (8,748 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
ArmaanChima@ArmaanChima:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ArmaanChima@ArmaanChima:~$ sudo ufw allow OpenSSH
ERROR: Could not find a profile matching 'OpenSSH'
ArmaanChima@ArmaanChima:~$ sudo ufw enable
Firewall is active and enabled on system startup
ArmaanChima@ArmaanChima:~$ sudo ufw status
Status: active
ArmaanChima@ArmaanChima:~$
```

This screenshot shows me enabling the firewall (UFW) and checking the status of the firewall to confirm that it is up and running.

```
To          Action    From
--          ----     ---
22/tcp      ALLOW     Anywhere
22/tcp (v6) ALLOW     Anywhere (v6)

ArmaanChima@ArmaanChima:~$
```

Sudo ufw status also shows us which ports are open.

The screenshot shows a terminal window titled "ArmaanChima@ArmaanChima:~". The user runs several commands to manage the firewall:

```
ArmaanChima@ArmaanChima:~$ sudo ufw allow port 22
ERROR: Need 'to' or 'from' clause
ArmaanChima@ArmaanChima:~$ sudo ufw allow port 22/tcp
ERROR: Need 'to' or 'from' clause
ArmaanChima@ArmaanChima:~$ ^C
ArmaanChima@ArmaanChima:~$ ls
Desktop Documents Downloads Music Pictures Public snap Templates Videos
ArmaanChima@ArmaanChima:~$ ls -l
total 36
drwxr-xr-x 2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Desktop
drwxr-xr-x 2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Documents
drwxr-xr-x 2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Downloads
drwxr-xr-x 2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Music
drwxr-xr-x 2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Pictures
drwxr-xr-x 2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Public
drwxr----- 3 ArmaanChima ArmaanChima 4096 Aug 16 04:31 snap
drwxr-xr-x 2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Templates
drwxr-xr-x 2 ArmaanChima ArmaanChima 4096 Aug 16 04:30 Videos
ArmaanChima@ArmaanChima:~$ sudo ufw allow OpenSSH
ERROR: Could not find a profile matching 'OpenSSH'
ArmaanChima@ArmaanChima:~$ sudo ufw allow 22/tcp
Rule added
Rule added (v6)
ArmaanChima@ArmaanChima:~$
```

The following shows a picture of me allowing port 22 (SSH) to be open. This is left open as no traffic would be allowed into the server, defeating the purpose of hardening. In addition, I am keeping only one port open since it will help me demonstrate some of the other security measures I have implemented.

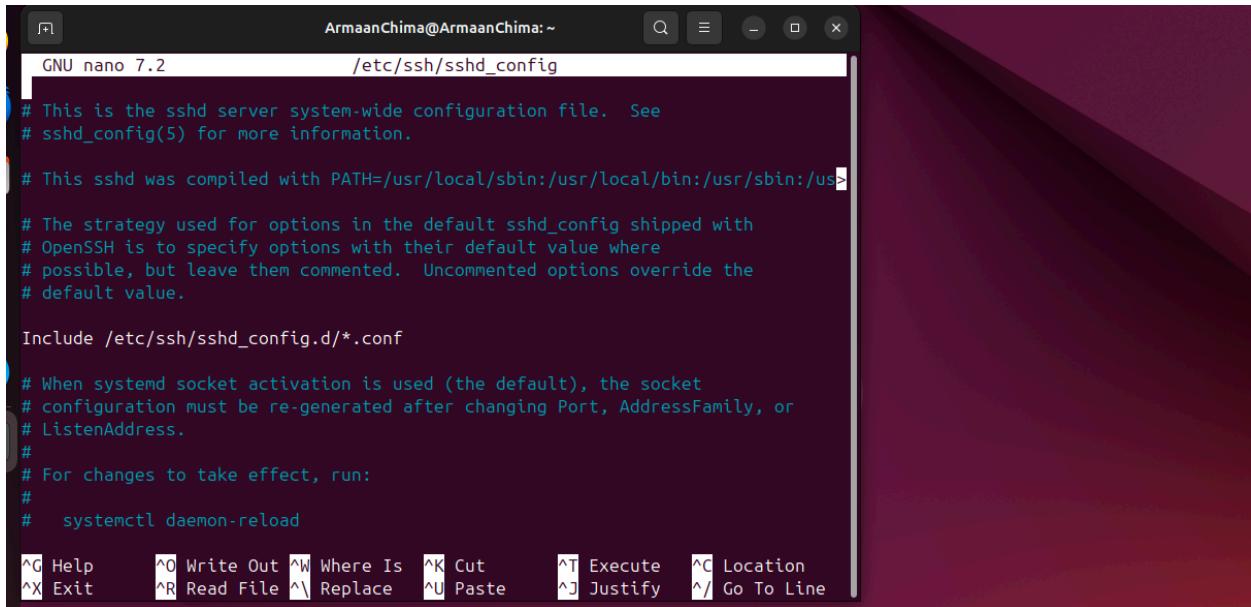
4.) OpenSSH Installation and Configuration

```
ArmaanChima@ArmaanChima:~$ sudo apt install openssh-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 822 kB of archives.
After this operation, 6,771 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports/noble-updates/main arm64 openssh-sftp-server arm64 1:9.6p1-3ubuntu13.13 [36.8 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports/noble-updates/main arm64 openssh-server arm64 1:9.6p1-3ubuntu13.13 [500 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports/noble/main arm64 ncurses-term all 6.4+20240113-1ubuntu2 [275 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports/noble-updates/main arm64 ssh-import-id all 5.11-0ubuntu2.24.04.1 [10.1 kB]
Fetched 822 kB in 1s (664 kB/s)
```

This image shows me using the command “sudo apt install openssh-server -y”. In simple terms, the purpose of this command is to install the openssh-server package on the system. To break it down, **sudo** means the command is run with superuser privileges, **apt** (advanced package tools) is a package manager used to help with installing the unattended-upgrades package, **install** works with apt to install the openssh-server, **openssh-server** is the package name and **-y** basically means that any prompts given to you will be responded to with a yes instead of the user having to type it in manually.

```
ArmaanChima@ArmaanChima:~$ /etc/ssh/sshd_config
bash: /etc/ssh/sshd_config: Permission denied
ArmaanChima@ArmaanChima:~$ sudo nano /etc/ssh/sshd_config
[sudo] password for ArmaanChima: [REDACTED]
```

Using this command, we are going to use superuser privileges and edit the /etc/ssh/sshd_config file



The screenshot shows a terminal window titled "ArmaanChima@ArmaanChima:~". The window contains the contents of the "/etc/ssh/sshd_config" file, which is being edited with the "GNU nano 7.2" text editor. The file includes comments about the configuration file and its compilation, and specifies the inclusion of files from the "/etc/ssh/sshd_config.d/" directory. The terminal also displays a set of keyboard shortcuts at the bottom.

```
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/us

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

# When systemd socket activation is used (the default), the socket
# configuration must be re-generated after changing Port, AddressFamily, or
# ListenAddress.
#
# For changes to take effect, run:
#
#   systemctl daemon-reload

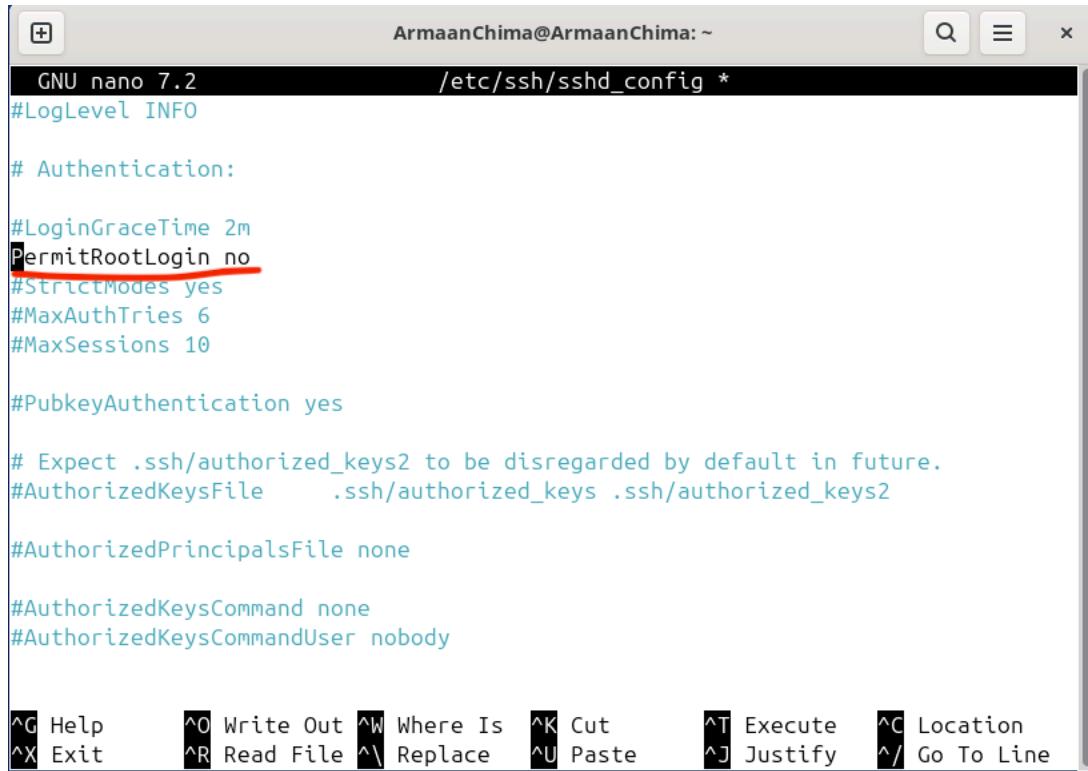
^C Help      ^O Write Out ^W Where Is  ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

The following uncommented text means include all files that end with .conf from the directory named /etc/ssh/sshd_config.d/

Include – Include means that sshd is going to read other config files

/etc/ssh/sshd_config.d/ - this is the directory in which all .conf files will be placed

*.conf – This indicates that all .conf files go in /etc/ssh/sshd_config.d/ and will be processed



```
ArmaanChima@ArmaanChima: ~
GNU nano 7.2          /etc/ssh/sshd_config *

LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit     ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

The following command tells us that Root Login is not allowed at all

```
ArmaanChima@ArmaanChima: ~
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for ufw (0.36.2-6) ...
Scanning processes...
Scanning processor microcode...
Scanning linux images...

Running kernel seems to be up-to-date.

The processor microcode seems to be up-to-date.

No services need to be restarted.

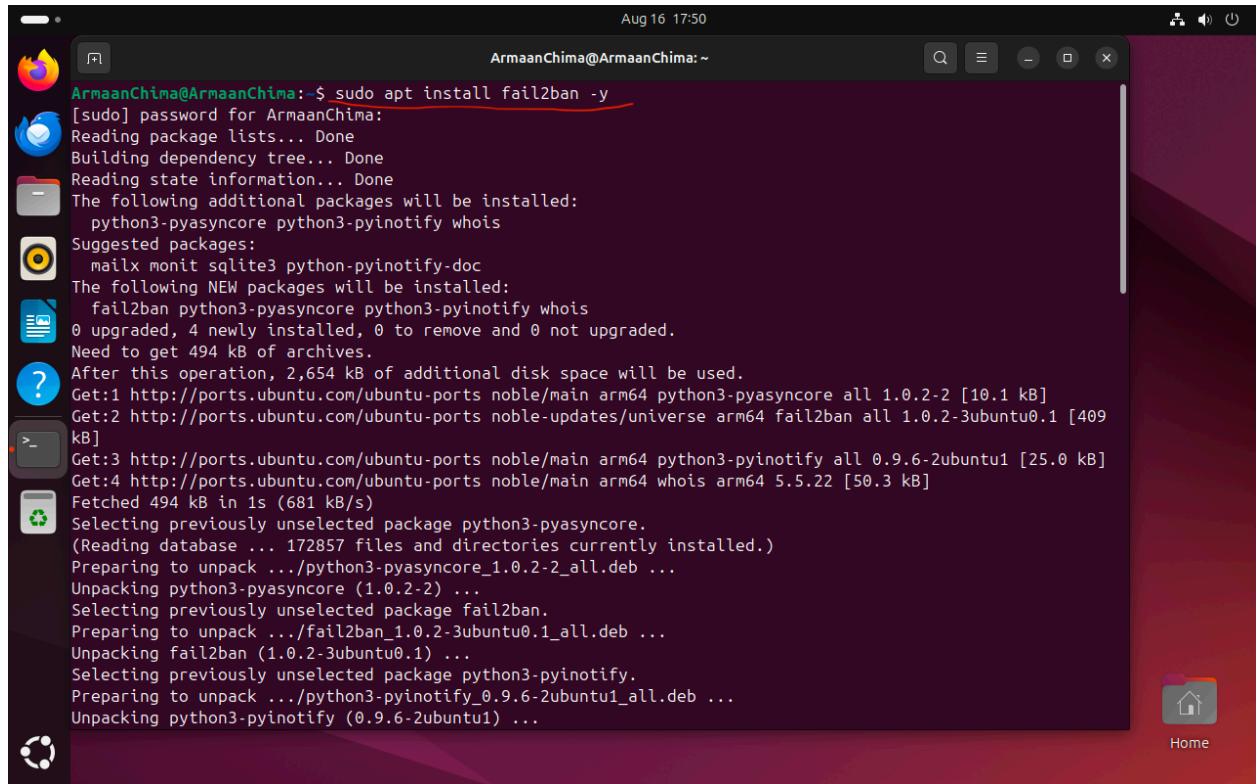
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ArmaanChima@ArmaanChima:~$ sudo nano /etc/ssh/sshd_config
ArmaanChima@ArmaanChima:~$ whoami
ArmaanChima
ArmaanChima@ArmaanChima:~$ sudo whoami
root
ArmaanChima@ArmaanChima:~$ sudo systemctl restart ssh
ArmaanChima@ArmaanChima:~$ █
```

Sudo runs commands using superuser privileges, systemctl can enable, check status, disable, restart, stop or start, and ssh is the service that we are applying this to. In essence, the SSH server is being restarted.

5.) Fail2Ban



ArmaanChima@ArmaanChima:~\$ sudo apt install fail2ban -y
[sudo] password for ArmaanChima:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
 python3-pyasyncore python3-pyinotify whois
Suggested packages:
 mailx monit sqlite3 python-pyinotify-doc
The following NEW packages will be installed:
 fail2ban python3-pyasyncore python3-pyinotify whois
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 494 kB of archives.
After this operation, 2,654 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports noble/main arm64 python3-pyasyncore all 1.0.2-2 [10.1 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports noble-updates/universe arm64 fail2ban all 1.0.2-3ubuntu0.1 [409 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports noble/main arm64 python3-pyinotify all 0.9.6-2ubuntu1 [25.0 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports noble/main arm64 whois arm64 5.5.22 [50.3 kB]
Fetched 494 kB in 1s (681 kB/s)
Selecting previously unselected package python3-pyasyncore.
(Reading database ... 172857 files and directories currently installed.)
Preparing to unpack .../python3-pyasyncore_1.0.2-2_all.deb ...
Unpacking python3-pyasyncore (1.0.2-2) ...
Selecting previously unselected package fail2ban.
Preparing to unpack .../fail2ban_1.0.2-3ubuntu0.1_all.deb ...
Unpacking fail2ban (1.0.2-3ubuntu0.1) ...
Selecting previously unselected package python3-pyinotify.
Preparing to unpack .../python3-pyinotify_0.9.6-2ubuntu1_all.deb ...
Unpacking python3-pyinotify (0.9.6-2ubuntu1) ...

This image shows me using the command “sudo apt install fail2ban -y”. In simple terms, the purpose of this command is to install the fail2ban package on the system. To break it down, **sudo** means the command is run with superuser privileges, **apt** (advanced package tools) is a package manager used to help with installing the fail2ban package, **install** works with apt to install the fail2ban package, **fail2ban** is the package name and **-y** basically means that any prompts given to you will be responded to with a yes instead of the user having to type it in manually.

```
All packages are up to date.  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Calculating upgrade... Done  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
ArmaanChima@ArmaanChima:~$ sudo systemctl enable fail2ban  
Synchronizing state of fail2ban.service with sysV service script with /usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable fail2ban  
ArmaanChima@ArmaanChima:~$ sudo systemctl start fail2ban  
ArmaanChima@ArmaanChima:~$ sudo systemctl status fail2ban  
● fail2ban.service - Fail2Ban Service  
    Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled; preset: enabled)  
    Active: active (running) since Mon 2025-08-18 05:36:32 UTC; 33min ago  
      Docs: man:fail2ban(1)  
      Main PID: 1115 (fail2ban-server)  
         Tasks: 5 (limit: 11903)  
        Memory: 27.6M (peak: 28.5M)  
          CPU: 1.030s  
        CGroup: /system.slice/fail2ban.service  
                └─1115 /usr/bin/python3 /usr/bin/fail2ban-server -xf start  
  
Aug 18 05:36:32 ArmaanChima systemd[1]: Started fail2ban.service - Fail2Ban Service.  
Aug 18 05:36:32 ArmaanChima fail2ban-server[1115]: 2025-08-18 05:36:32,736 fail2ban.configreader >  
Aug 18 05:36:32 ArmaanChima fail2ban-server[1115]: Server ready  
Show Apps 14/14 (END)
```

Sudo runs commands using superuser privileges, systemctl can enable, check status, disable, stop, restart or start, and fail2ban is the package that we are applying this to.

Note: start starts fail2ban, enable enables fail2ban, and status shows us the status of fail2ban (All underlined in red).

Furthermore, in this photo, we can see that fail2ban is also up and running

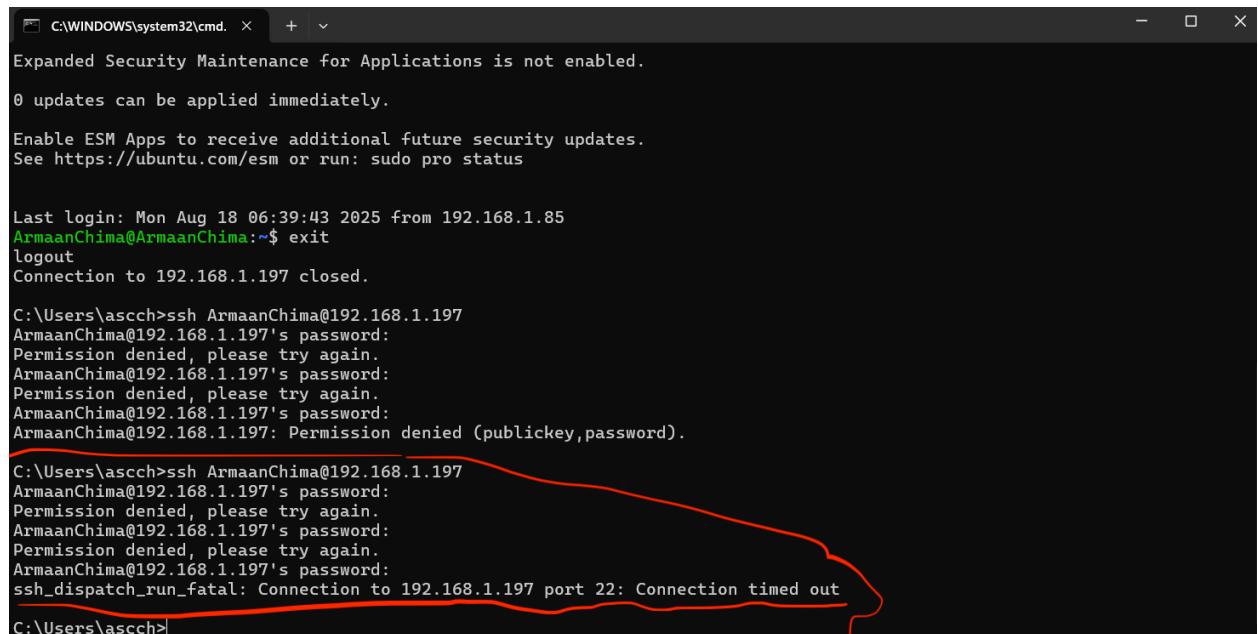
```
Main PID: 1115 (fail2ban-server)  
Tasks: 5 (limit: 11903)  
Memory: 27.6M (peak: 28.5M)  
CPU: 1.030s  
CGroup: /system.slice/fail2ban.service  
        └─1115 /usr/bin/python3 /usr/bin/fail2ban-server -xf start  
  
Aug 18 05:36:32 ArmaanChima systemd[1]: Started fail2ban.service - Fail2Ban Service.  
Aug 18 05:36:32 ArmaanChima fail2ban-server[1115]: 2025-08-18 05:36:32,736 fail2ban.configreader >  
Aug 18 05:36:32 ArmaanChima fail2ban-server[1115]: Server ready  
ArmaanChima@ArmaanChima:~$ sudo fail2ban-client status  
Status  
|- Number of jail:      1  
`- Jail list:    sshd
```

This command allows us to see if fail2ban to ban is running, which IP addresses have been jailed, if any, and shows the number of jails that are up and running.

Sudo – this means the command is being run with superuser privileges

fail2ban-client – allows us to see the number of jails, remove banned IP addresses and check the overall standing of fail2ban

status – shows us the status of fail2ban

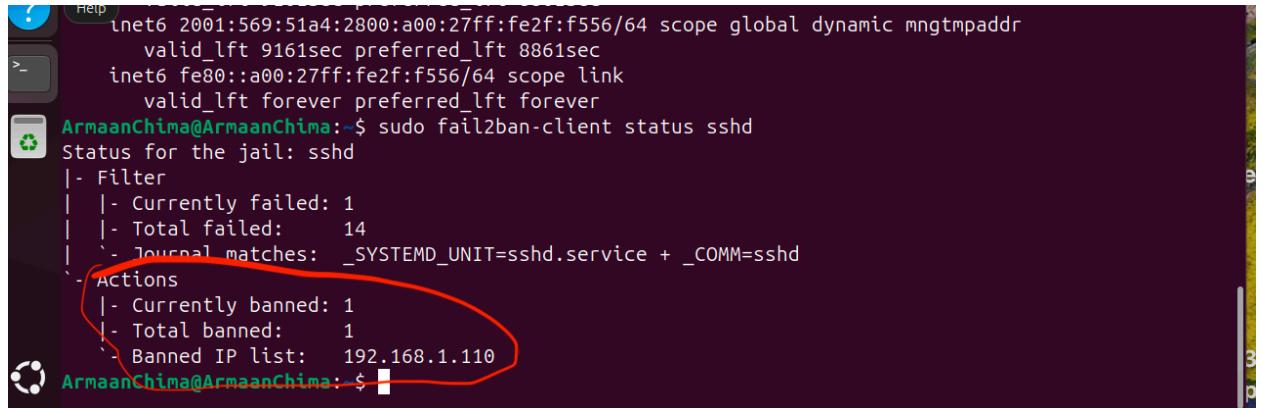


The screenshot shows a Windows Command Prompt window titled 'C:\WINDOWS\system32\cmd.' with the following text:

```
Expanded Security Maintenance for Applications is not enabled.  
0 updates can be applied immediately.  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
Last login: Mon Aug 18 06:39:43 2025 from 192.168.1.85  
ArmaanChima@ArmaanChima:~$ exit  
logout  
Connection to 192.168.1.197 closed.  
  
C:\Users\ascch>ssh ArmaanChima@192.168.1.197  
ArmaanChima@192.168.1.197's password:  
Permission denied, please try again.  
ArmaanChima@192.168.1.197's password:  
Permission denied, please try again.  
ArmaanChima@192.168.1.197's password:  
ArmaanChima@192.168.1.197: Permission denied (publickey,password).  
  
C:\Users\ascch>ssh ArmaanChima@192.168.1.197  
ArmaanChima@192.168.1.197's password:  
Permission denied, please try again.  
ArmaanChima@192.168.1.197's password:  
Permission denied, please try again.  
ArmaanChima@192.168.1.197's password:  
ssh_dispatch_run_fatal: Connection to 192.168.1.197 port 22: Connection timed out  
C:\Users\ascch>
```

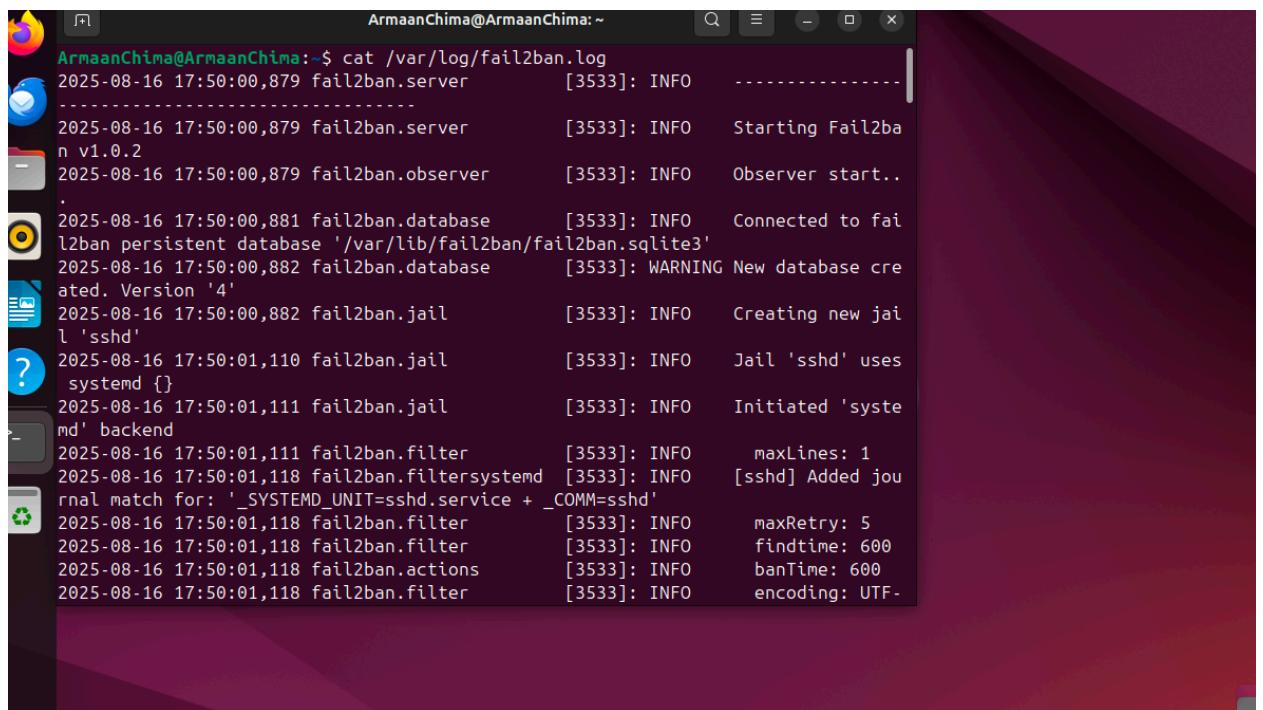
A red oval highlights the final line of the session, 'ssh_dispatch_run_fatal: Connection to 192.168.1.197 port 22: Connection timed out'.

This image represents us trying to access the server from a different computer. As you can see, after three tries, we get a timeout, and we are still unable to access the server. This will lead to the next photo.



```
inet6 2001:569:51a4:2800:a00:27ff:fe2f:f556/64 scope global dynamic mngtmpaddr  
      valid_lft 9161sec preferred_lft 8861sec  
inet6 fe80::a00:27ff:fe2f:f556/64 scope link  
      valid_lft forever preferred_lft forever  
ArmaanChima@ArmaanChima:~$ sudo fail2ban-client status sshd  
Status for the jail: sshd  
|- Filter  
| |- Currently failed: 1  
| |- Total failed: 14  
| `- Journal matches: _SYSTEMD_UNIT=sshd.service + _COMM=sshd  
`- Actions  
  |- Currently banned: 1  
  |- Total banned: 1  
  `- Banned IP list: 192.168.1.110  
ArmaanChima@ArmaanChima:~$
```

In this photo, we can see that the IP address 192.168.1.110 has been jailed, showing that our fail2ban works, resulting in that IP being jailed for 10 minutes



```
ArmaanChima@ArmaanChima:~$ cat /var/log/fail2ban.log  
2025-08-16 17:50:00,879 fail2ban.server [3533]: INFO -----  
-----  
2025-08-16 17:50:00,879 fail2ban.server [3533]: INFO Starting Fail2ba  
n v1.0.2  
2025-08-16 17:50:00,879 fail2ban.observer [3533]: INFO Observer start..  
. .  
2025-08-16 17:50:00,881 fail2ban.database [3533]: INFO Connected to fai  
l2ban persistent database '/var/lib/fail2ban/fail2ban.sqlite3'  
2025-08-16 17:50:00,882 fail2ban.database [3533]: WARNING New database cre  
ated. Version '4'  
2025-08-16 17:50:00,882 fail2ban.jail [3533]: INFO Creating new jai  
l 'sshd'  
2025-08-16 17:50:01,110 fail2ban.jail [3533]: INFO Jail 'sshd' uses  
systemd []  
2025-08-16 17:50:01,111 fail2ban.jail [3533]: INFO Initiated 'syste  
md' backend  
2025-08-16 17:50:01,111 fail2ban.filter [3533]: INFO maxLines: 1  
2025-08-16 17:50:01,118 fail2ban.filtersystemd [3533]: INFO [sshd] Added jou  
rnal match for: '_SYSTEMD_UNIT=sshd.service + _COMM=sshd'  
2025-08-16 17:50:01,118 fail2ban.filter [3533]: INFO maxRetry: 5  
2025-08-16 17:50:01,118 fail2ban.filter [3533]: INFO findtime: 600  
2025-08-16 17:50:01,118 fail2ban.actions [3533]: INFO banTime: 600  
2025-08-16 17:50:01,118 fail2ban.filter [3533]: INFO encoding: UTF-
```

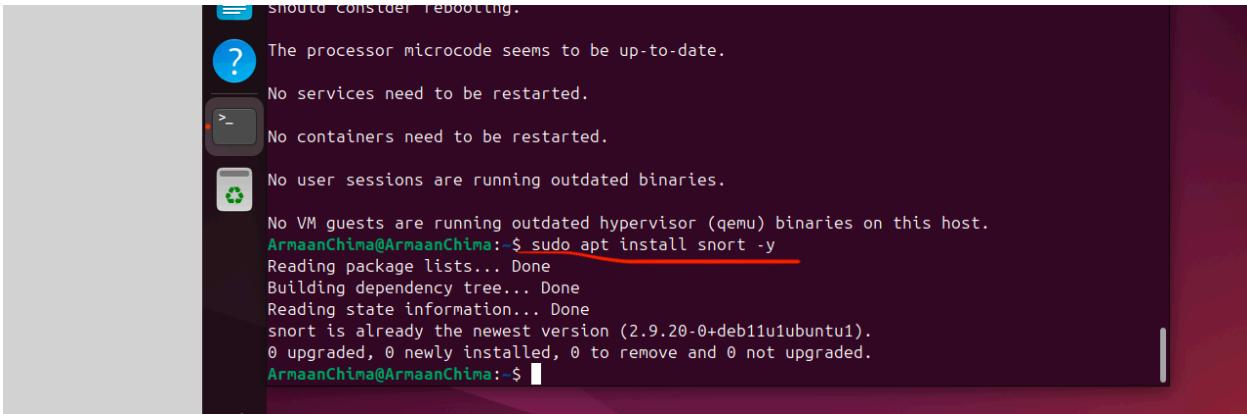
This photo shows us all the events that are “fail2ban-related”. This log tells us and shows us the start of jails, errors, banned IP addresses, and more.

cat - Displays the contents of the fail2ban.log file

/var/log – This is the place where log files are kept

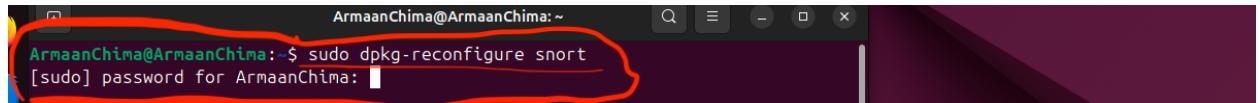
fail2ban.log – This is the file that fail2ban is “in charge of”

6.) Snort Installation and Configuration



```
System should consider rebooting.  
? The processor microcode seems to be up-to-date.  
>- No services need to be restarted.  
Recycle No containers need to be restarted.  
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ArmaanChima@ArmaanChima:~$ sudo apt install snort -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
snort is already the newest version (2.9.20-0+deb11u1ubuntu1).  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
ArmaanChima@ArmaanChima:~$
```

This image shows me using the command “sudo apt install snort -y”. In simple terms, the purpose of this command is to install the Snort package on the system. To break it down, **sudo** means the command is run with superuser privileges, **apt** (advanced package tools) is a package manager used to help with installing the snort package, **install** works with apt to install the snort package, **snort** is the package name and **-y** basically means that any prompts given to you will be responded to with a yes instead of the user having to type it in manually.



```
ArmaanChima@ArmaanChima:~$ sudo dpkg-reconfigure snort  
[sudo] password for ArmaanChima:
```

Sudo, which was already talked about, means the command is run with superuser privileges; **dpkg-reconfigure** allows you to reconfigure the settings, allowing you to update the Snort package, and Snort is the package that we are reconfiguring.

Configuring snort

Please use the CIDR form - for example, 192.168.1.0/24 for a block of 256 addresses or 192.168.1.42/32 for just one. Multiple values should be comma-separated (without spaces).

You can leave this value empty and configure HOME_NET in /etc/snort/snort.conf instead. This is useful if you are using Snort in a system which frequently changes network and does not have a static IP address assigned.

Please note that if Snort is configured to use multiple interfaces, it will use this value as the HOME_NET definition for all of them.

Address range for the local network:

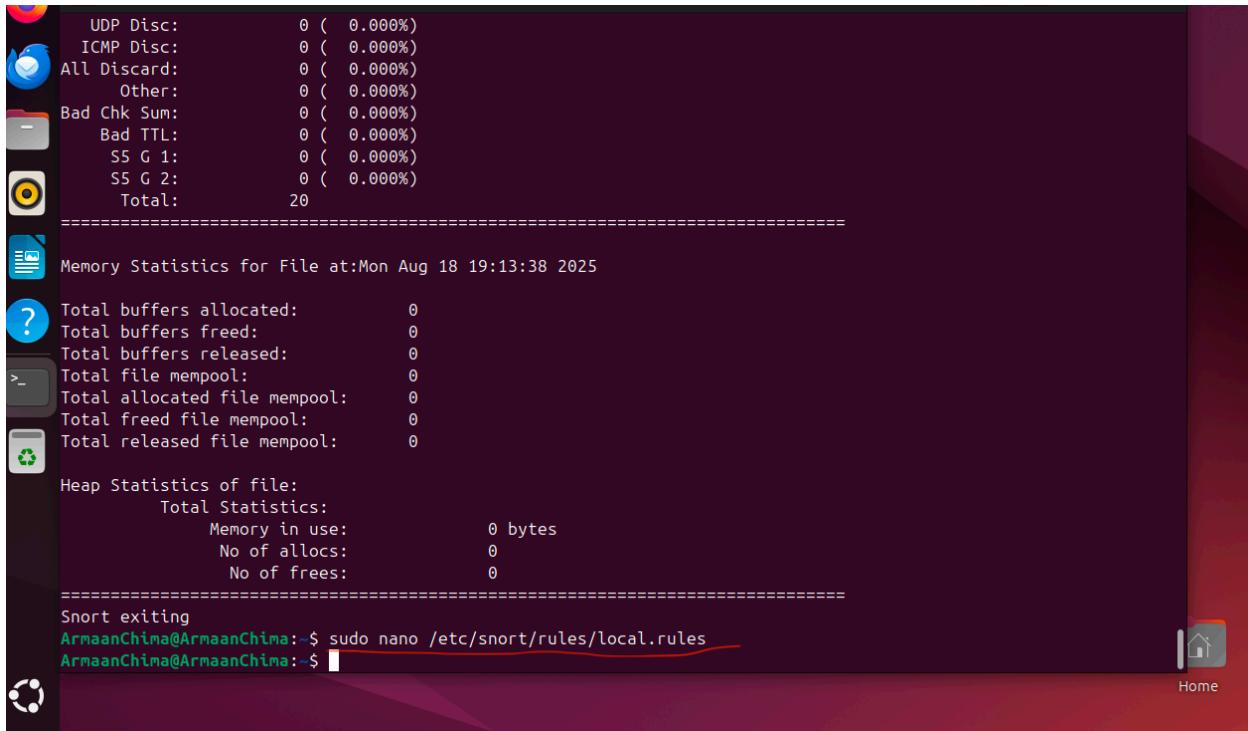
192.168.1.197/24

<0k>

After typing the command sudo dpkg-reconfigure snort, I was taken here where I inserted my IP address. Doing this allowed me to have a functional Snort log, which enabled me to analyze the feedback I was receiving from it (that feedback being traffic and alerts presented in the log). I will talk about this in the following pictures to come.

```
net 192.168.1.197/24 brd 192.168.1.255 scope global dynamic mngtmpc txroute c
np0s8
    valid_lft 85759sec preferred_lft 85759sec
    inet6 2001:569:51a4:2800:34ea:e279:a18c:5ae7/64 scope global temporary dynam
ic
        valid_lft 8872sec preferred_lft 8572sec
        inet6 2001:569:51a4:2800:a00:27ff:fe2f:f556/64 scope global dynamic mngtmpad
dr
    valid_lft 8872sec preferred_lft 8572sec
    inet6 fe80::a00:27ff:fe2f:f556/64 scope link
        valid_lft forever preferred_lft forever
ArmaanChima@ArmaanChima:~$ snort -V
,,-      -*> Snort! <*-
o"_)~ Version 2.9.20 GRE (Build 82)
    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
    Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reser
ved.
    Copyright (C) 1998-2013 Sourcefire, Inc., et al.
    Using libpcap version 1.10.4 (with TPACKET_V3)
    Using PCRE version: 8.39 2016-06-14
    Using ZLIB version: 1.3
ArmaanChima@ArmaanChima:~$
```

The command snort -V shows the user the version of Snort that is running on their system



UDP Disc: 0 (0.000%)
ICMP Disc: 0 (0.000%)
All Discard: 0 (0.000%)
Other: 0 (0.000%)
Bad Chk Sum: 0 (0.000%)
Bad TTL: 0 (0.000%)
SS G 1: 0 (0.000%)
SS G 2: 0 (0.000%)
Total: 20
=====

Memory Statistics for File at:Mon Aug 18 19:13:38 2025

Total buffers allocated:	0
Total buffers freed:	0
Total buffers released:	0
Total file mempool:	0
Total allocated file mempool:	0
Total freed file mempool:	0
Total released file mempool:	0

Heap Statistics of file:
Total Statistics:
 Memory in use: 0 bytes
 No of allocs: 0
 No of frees: 0
=====

Snort exiting

```
ArmaanChima@ArmaanChima:~$ sudo nano /etc/snort/rules/local.rules
ArmaanChima@ArmaanChima:~$
```

Home

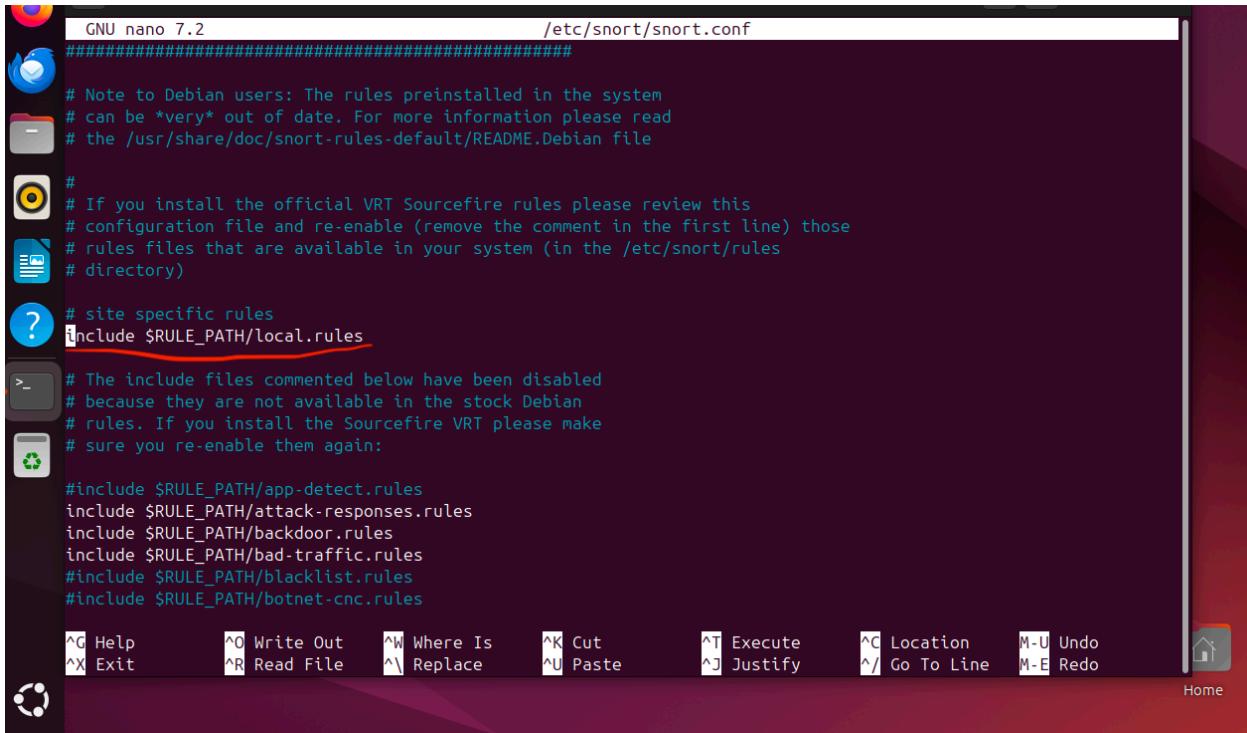
The terminal window displays Snort's packet processing statistics, memory usage, and heap statistics. It then exits. Below the terminal, a command is shown being typed: "sudo nano /etc/snort/rules/local.rules". A red line highlights the "/etc/snort/rules/local.rules" part of the command. The user is identified as "ArmaanChima" and the command is run from the root directory (~\$).

The command sudo means we are running the command with superuser privileges, nano is a text editor that we will need to edit the file in question, and /etc/snort/rules/local.rules is the path that we typed in order to get to the file that we need to edit. In addition, local.rules is a file that does not come with any signatures, allowing you to add your own additions to the file.

```
GNU nano 7.2                               /etc/snort/rules/local.rules *
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.
alert icmp any any -> any any (msg: "Ping Detected (ICMP)"; sid:1000001; rev:1;)

File Name to Write: /etc/snort/rules/local.rules
^G Help      M-D DOS Format    M-A Append    M-B Backup File
^C Cancel    M-M Mac Format    M-P Prepend   ^T Browse
Home
```

In the custom file (local.rules), I added “alert icmp any any -> any any (msg: “Ping Detected (ICMP)”; sid:1000001; rev:1;). This shows an alert anytime a ping happens in the middle of any destination and any source. To break this down, **alert** means Snort creates an alert, an alert happens when the rule is met, resulting in one being logged. **ICMP** is the protocol used for ping responses and requests, the first any in **any any** (before the arrow) means any IP address while the other any means any port, the arrow tells us the direction in which the traffic is flowing, and the first any after the arrow means any destination IP address while the second any means any destination port. Furthermore, **msg: “Ping Detected (ICMP)**”, you will see this message if a packet matches, **sid:1000001** is Snort’s ID for this rule, and rev: 1 is the revision number, meaning anytime you update or change the rule, this number goes up by 1.



```
GNU nano 7.2                               /etc/snort/snort.conf
#####
# Note to Debian users: The rules preinstalled in the system
# can be *very* out of date. For more information please read
# the /usr/share/doc/snort-rules-default/README.Debian file
#
# If you install the official VRT Sourcefire rules please review this
# configuration file and re-enable (remove the comment in the first line) those
# rules files that are available in your system (in the /etc/snort/rules
# directory)
#
# site specific rules
include $RULE_PATH/local.rules
#
# The include files commented below have been disabled
# because they are not available in the stock Debian
# rules. If you install the Sourcefire VRT please make
# sure you re-enable them again:
#
#include $RULE_PATH/app-detect.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/bad-traffic.rules
#include $RULE_PATH/blacklist.rules
#include $RULE_PATH/botnet-cnc.rules

^G Help      ^O Write Out   ^W Where Is    ^K Cut        ^T Execute   ^C Location   M-U Undo
^X Exit      ^R Read File   ^\ Replace     ^U Paste      ^J Justify   ^/ Go To Line M-E Redo
                                         Home
```

In addition, I also accessed the snort.conf file where I uncommented include \$RULE_PATH/local.rules. In short, this command means look inside the rules directory and include the local.rules file too. Make sure all the rules, together with the other ones, are applied. To break this down, **include** indicates to include an additional rule file (in this case, local.rules), **\$RULE_PATH** is a variable telling Snort to go to the directory where all stored rules are, and **local.rules** is the file in question that we want to include.

```
ArmaanChima@ArmaanChima:~$ sudo snort -A console -q -c /etc/snort/snort.conf -i enp0s8
ERROR: /etc/snort/rules/local.rules(7) Invalid argument to 'sid' rule option: 1000001: rev:1. Must be a positive integer.
Fatal Error, Quitting..
ArmaanChima@ArmaanChima:~$ sudo nano /etc/snort/rules/local.rules
ArmaanChima@ArmaanChima:~$ sudo snort -A console -q -c /etc/snort/snort.conf -i enp0s8
08/19/04:37:29.903560  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} fe80::a2b5:3cff:fe
24:85dd -> ff02::1:ff08:490d
08/19/04:37:30.927058  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} fe80::a2b5:3cff:fe
24:85dd -> ff02::1:ff9d:71cd
08/19/04:37:30.927059  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} fe80::a2b5:3cff:fe
24:85dd -> ff02::1:ff78:5330
08/19/04:37:30.927059  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} fe80::a2b5:3cff:fe
24:85dd -> ff02::1:ff03:4617
08/19/04:37:30.927293  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} fe80::a2b5:3cff:fe
24:85dd -> ff02::1:ff3e:e9fd
08/19/04:37:30.927579  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} fe80::a2b5:3cff:fe
24:85dd -> ff02::1:ff8f:c6d0
08/19/04:37:30.927943  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} fe80::a2b5:3cff:fe
24:85dd -> ff02::1:ff8c:5265
08/19/04:37:30.927943  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} fe80::a2b5:3cff:fe
24:85dd -> ff02::1:ff8d:d4bc
08/19/04:37:30.928596  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} fe80::a2b5:3cff:fe
24:85dd -> ff02::1:ff25:c167
08/19/04:37:31.234158  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} fe80::a2b5:3cff:fe
24:85dd -> fe80::a00:27ff:fe2f:f556
08/19/04:37:31.234182  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} fe80::a00:27ff:fe2
f:f556 -> fe80::a2b5:3cff:fe24:85dd
```

The command `sudo snort -A console -q -c /etc/snort/snort.conf -i enp0s8` allows us to see all the traffic we are getting (pings). Here is a more detailed rundown...

Sudo – this means the command is being run with superuser privileges

Snort – this is the service in question

-A console --A tells Snort the way the logs are supposed to be displayed (telling snort how to alert essentially), while the console command prints the traffic to the terminal (only traffic that matches the rules)

-q – this means quiet mode

-c /etc/snort/snort.conf - -c tells snort what conf file snort should be applied, and /etc/snort/snort.conf is the configuration file in question

-i enp0s8 - -i specifies which network interface should be watched, and enp0s8 is the network interface that is being watched

```
08/19-04:38:44.349500  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::1:ff41:781b
08/19-04:38:44.350711  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} fe80::1046:f158:4c41:781b -> ff02::16
08/19-04:38:44.351275  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
08/19-04:38:44.351275  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} fe80::1046:f158:4c41:781b -> ff02::2
08/19-04:38:44.451597  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} fe80::1046:f158:4c41:781b -> ff02::16
08/19-04:38:44.451598  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} :: -> ff02::1:ff23:e4a3
08/19-04:38:44.451598  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::1:ff23:e4a3
08/19-04:38:44.452252  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} fe80::1046:f158:4c41:781b -> ff02::16
08/19-04:38:44.452252  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} :: -> ff02::1:ff39:7f20
08/19-04:38:44.452252  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} :: -> ff02::1:ff39:7f20
08/19-04:38:44.453202  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} fe80::1046:f158:4c41:781b -> ff02::2
08/19-04:38:44.913628  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {ICMP} 192.168.1.174 -> 192.168.1.197
08/19-04:38:44.913650  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {ICMP} 192.168.1.197 -> 192.168.1.174
08/19-04:38:45.487578  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {IPV6-ICMP} fe80::1046:f158:4c41:781b -> ff02::16
08/19-04:38:45.918713  [**] [1:1000001:1] Ping Detected (ICMP) [**] [Priority: 0] {ICMP} 192.168.1.174 -> 192.168.1.197
```

A picture showing a separate computer trying to ping the server and an alert being generated.

Short Reflection

Over the course of this semester, I have learned and touched on many security-related topics. I have had the opportunity to understand security threats such as phishing, shoulder surfing, ransomware, zero-day attacks, and more, along with prevention methods like encryption, firewalls, and IDS software. Additionally, I enjoyed the chance to harden a server, which I found particularly interesting because it tested the knowledge I've gained from this class and previous ones, helping me sharpen existing skills and learn new ones in the field of security. This course has taught me what to expect in future classes and has truly sparked my interest in the security field. Overall, my experience has been both enjoyable and informative.

