CIS 221 Project

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

Armaan Chima

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:

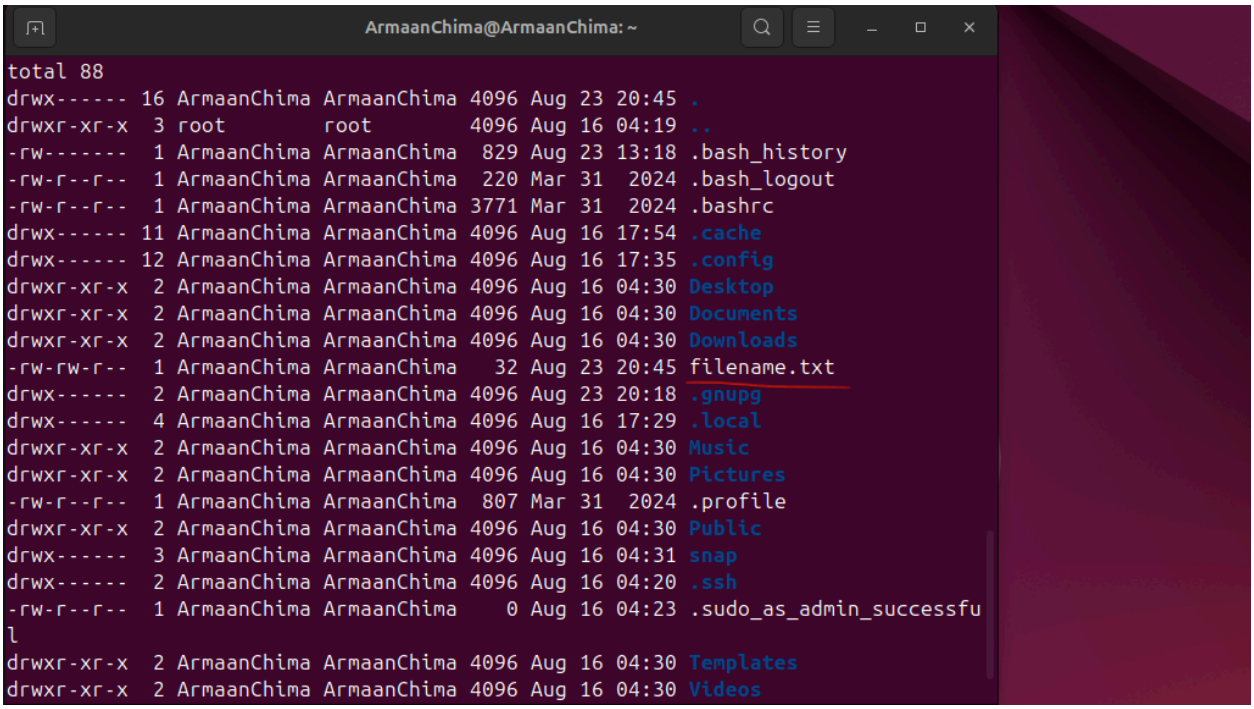
Before an attack occurs, there are simple things that can be done to stop the attack from occurring and prevent information from being leaked. These things are…

- 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

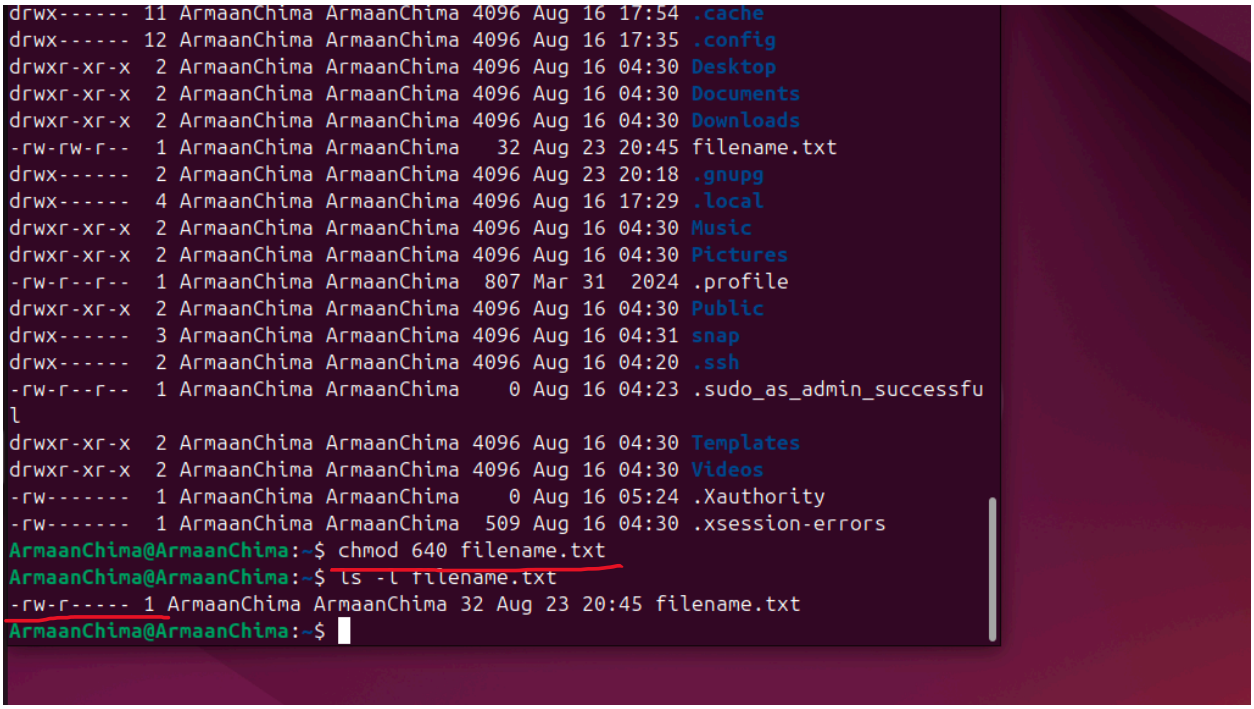
File Information and Access



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

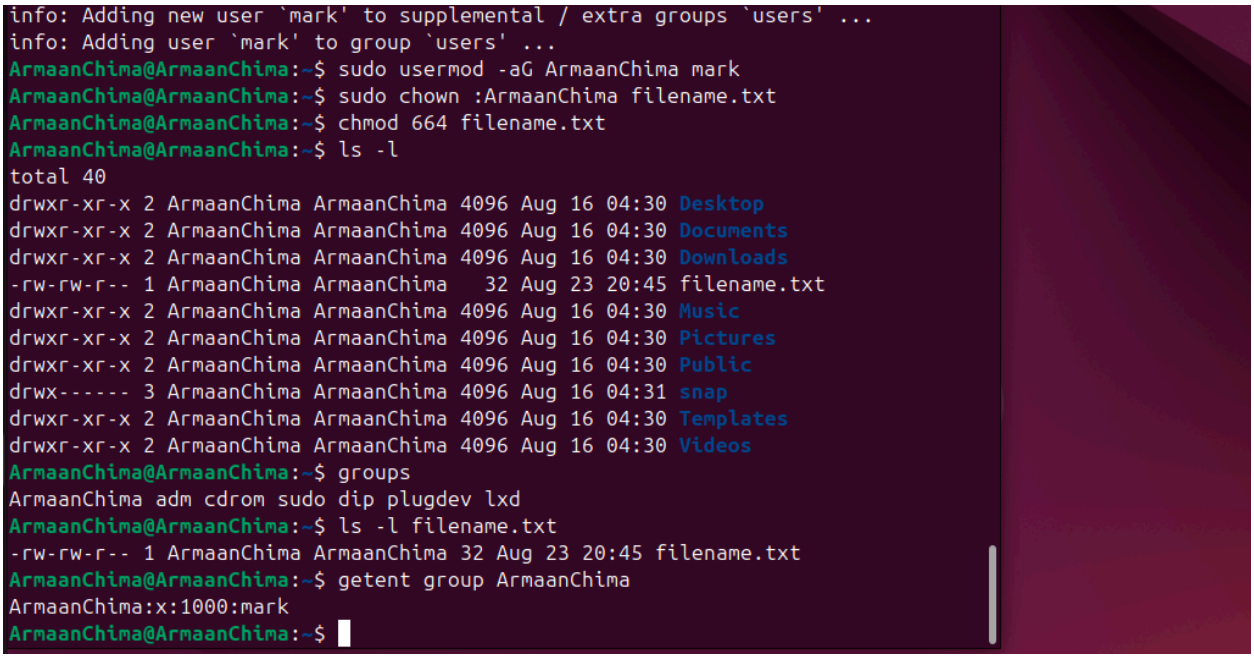
The file that will be used to explain file access is filename.txt (Shown in the photo)

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.



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.



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

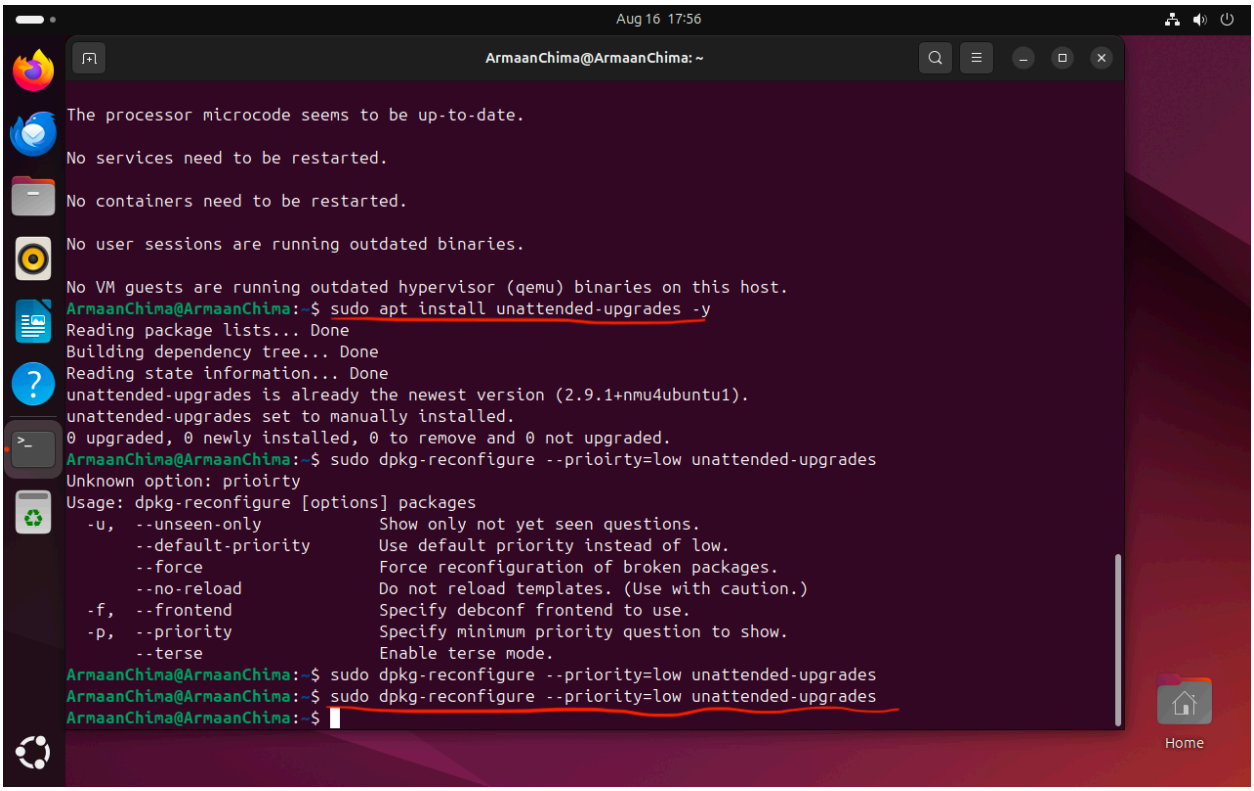
- **Asymmetric Encryption ->** The sender and the receiver both have different keys

Screenshots of the Installation and Setup Process for the Ubuntu Server

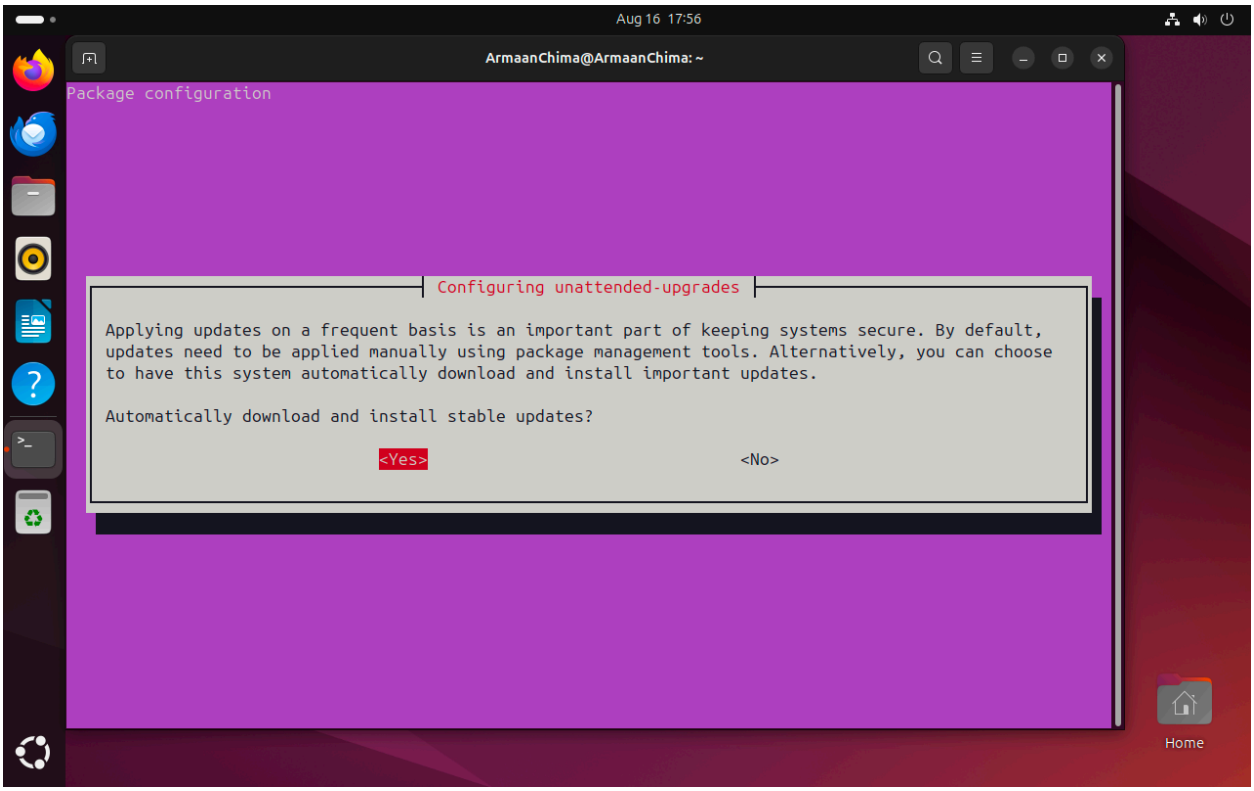
1.) The first thing I did was set up an **Ubuntu server** on VirtualBox (This acts as the PC)

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

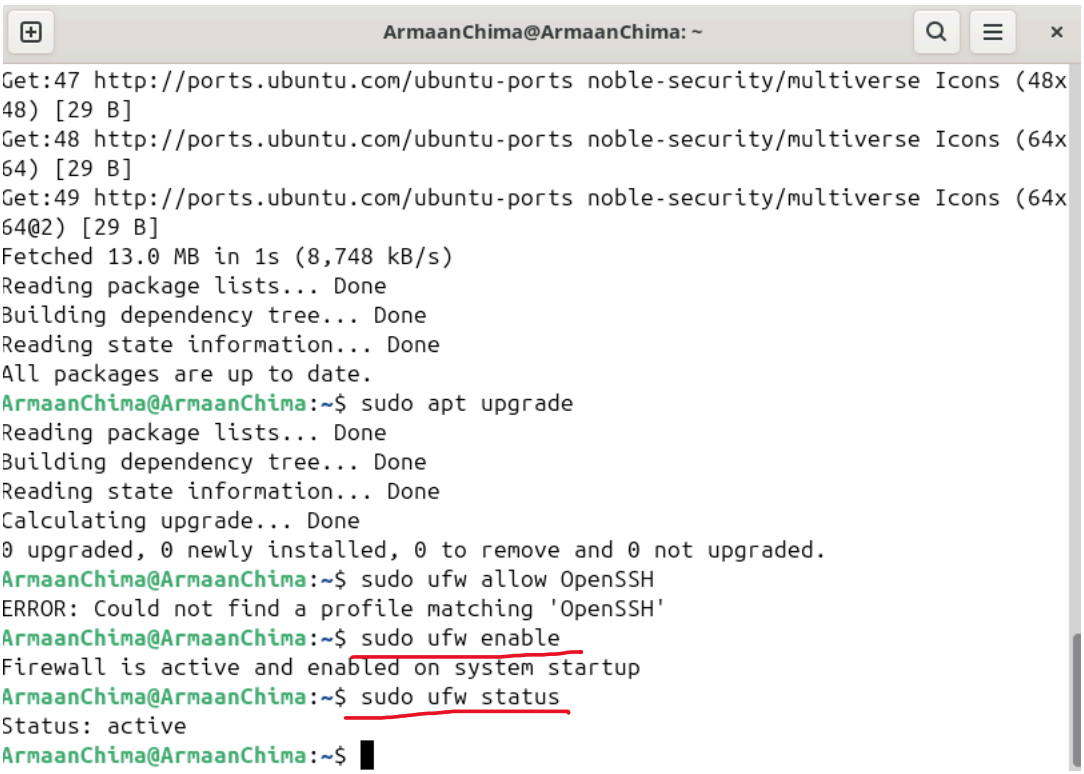


This image shows me using the command “sudo apt install unattended-upgrades -y”. In simple terms, the purpose of this command is to install the unattended-upgrades package on the system. To break it down, **sudo** means the command is run with superuser privileges, **apt** (advanced package tools) is 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.

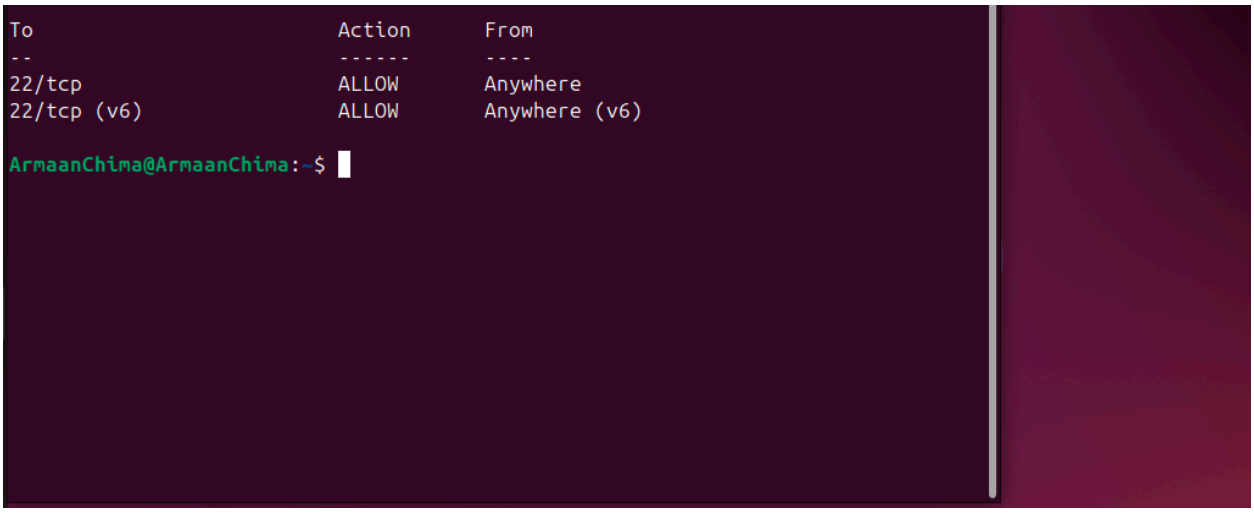


In the previous image, the command “sudo dpkg-reconfigure –priority=low unattended upgrades” was never talked about, as it is correlated to this image. **Sudo**, which was already talked about, means the command is run with superuser privileges. **dpkg-reconfigure** allows you to reconfigure the settings, 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.

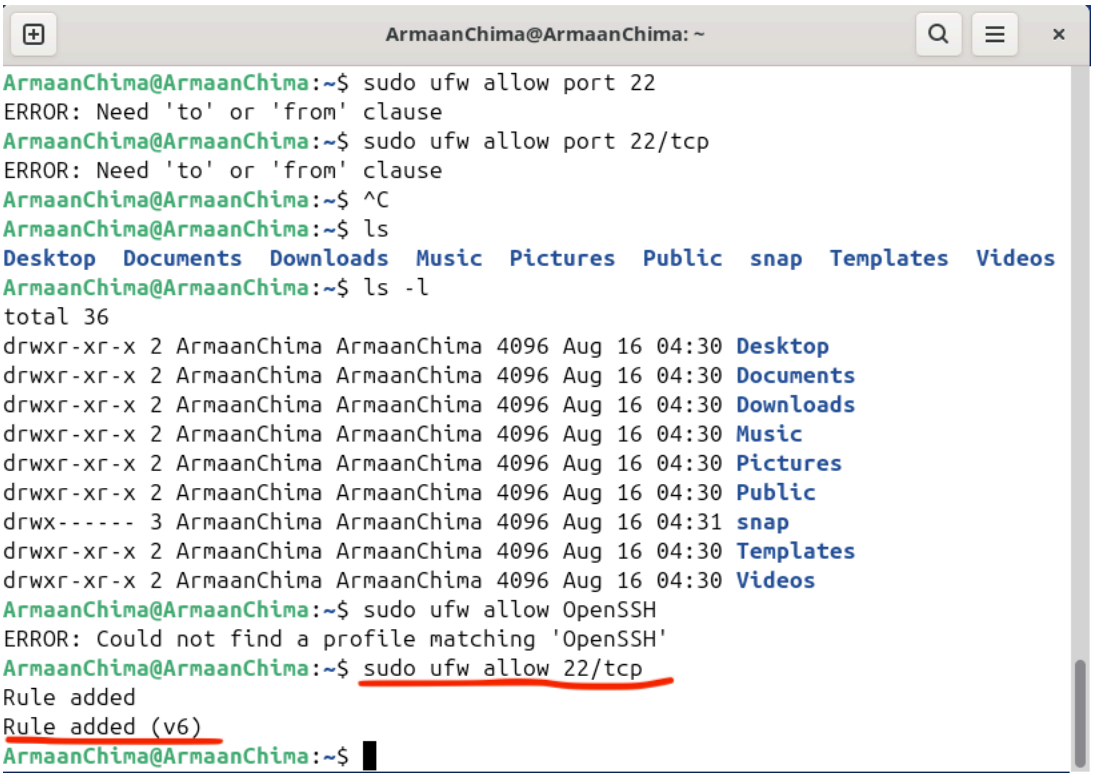
3.) UFW Firewall installation and configuration



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

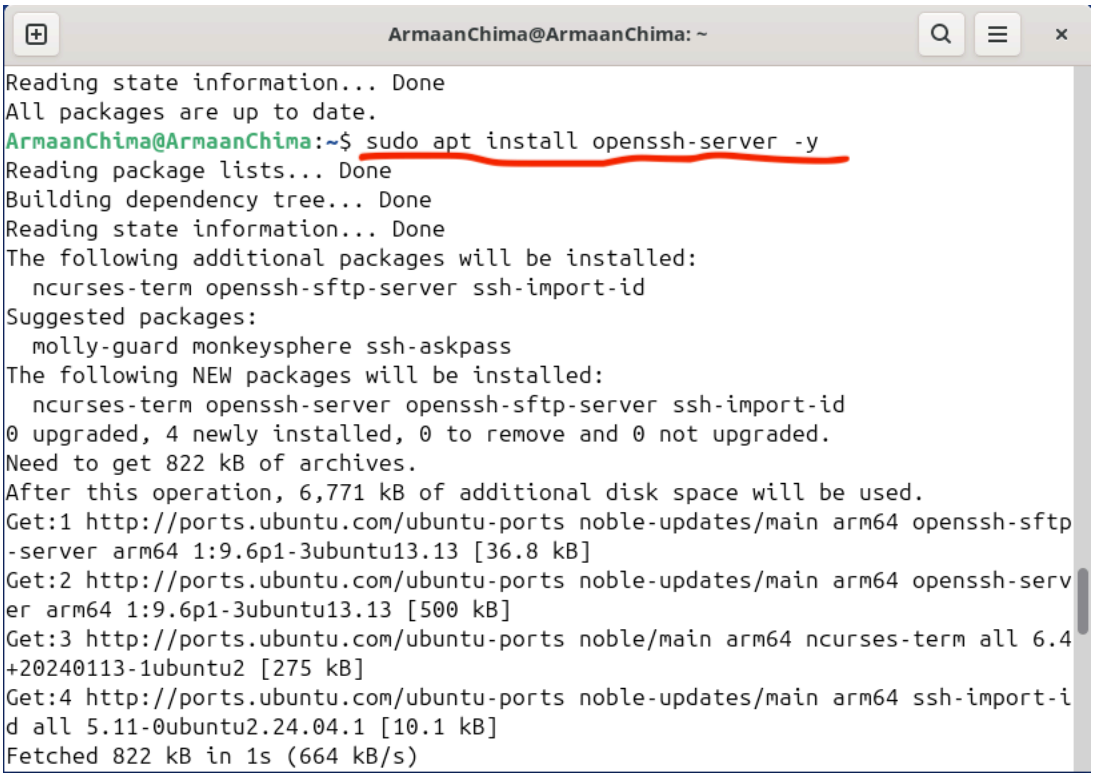


Sudo ufw status also shows us which ports are open.

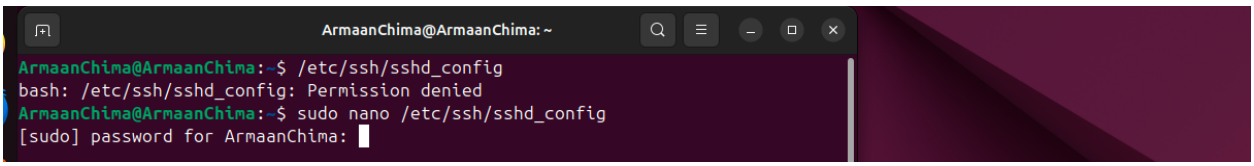


The following shows a picture of me allowing port 22 (SSH) to be open. This is left open as no traffic would be allowed into the server, defeating the purpose of hardening. In addition, I am keeping only one port open since it will help me demonstrate some of the other security measures I have implemented.

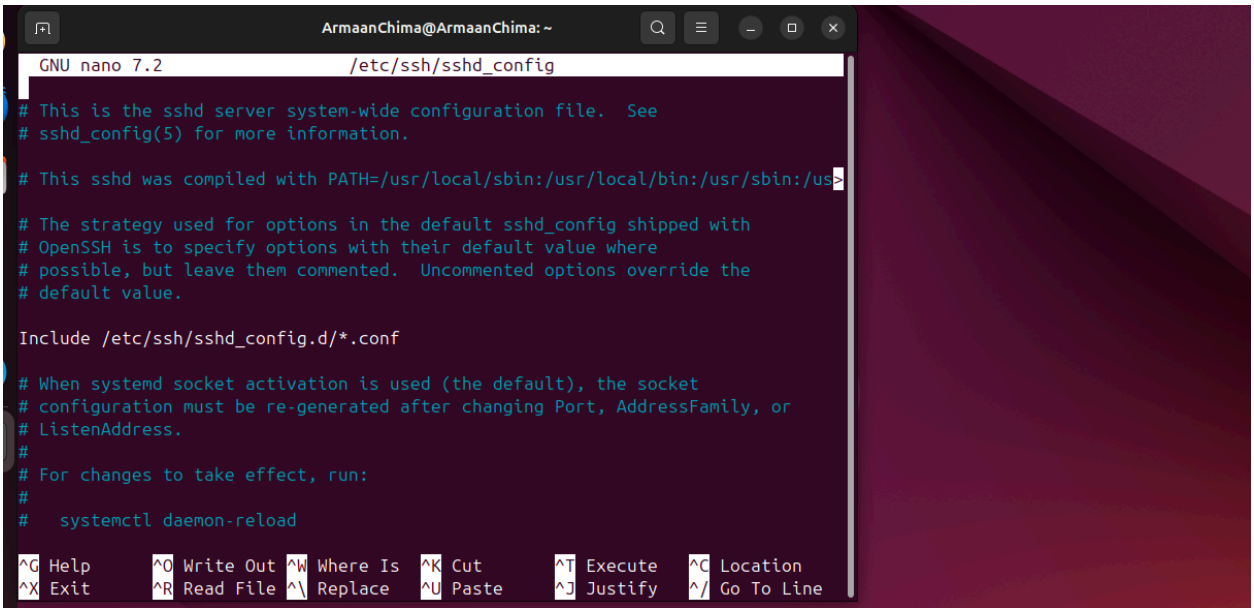
4.) OpenSSH Installation and Configuration



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.



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

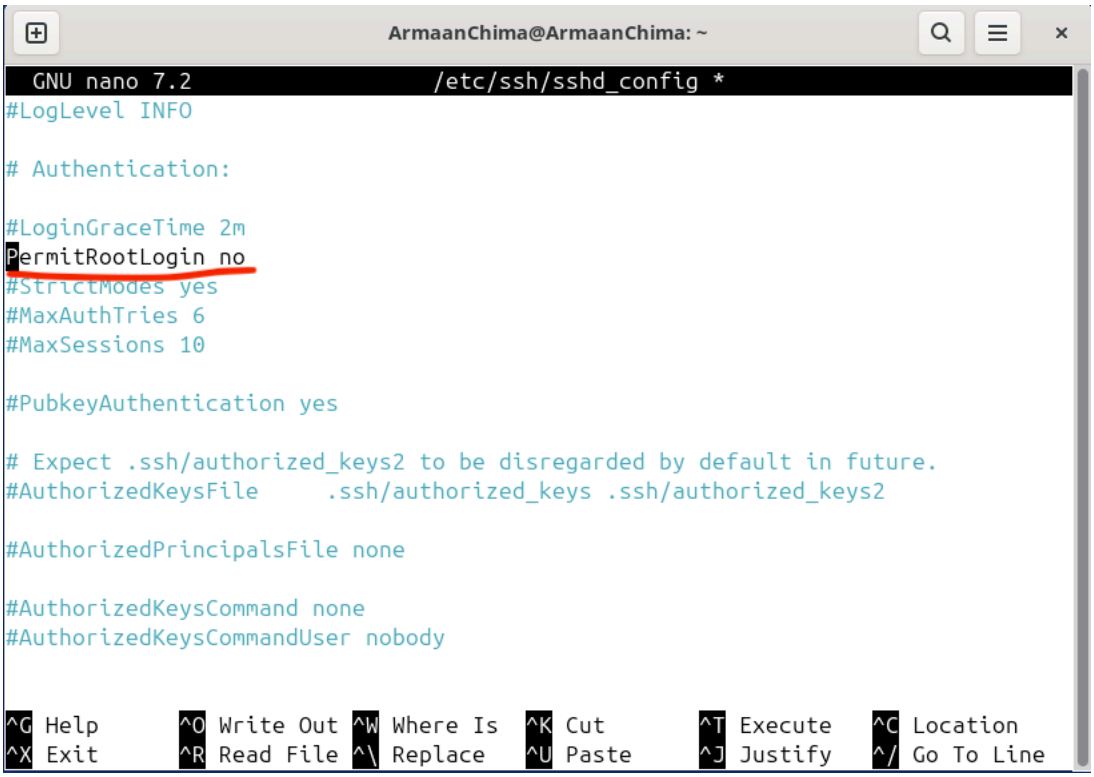


The following uncommented text means include all files that end with .conf from the directory named /etc/ssh/sshd_config.d/

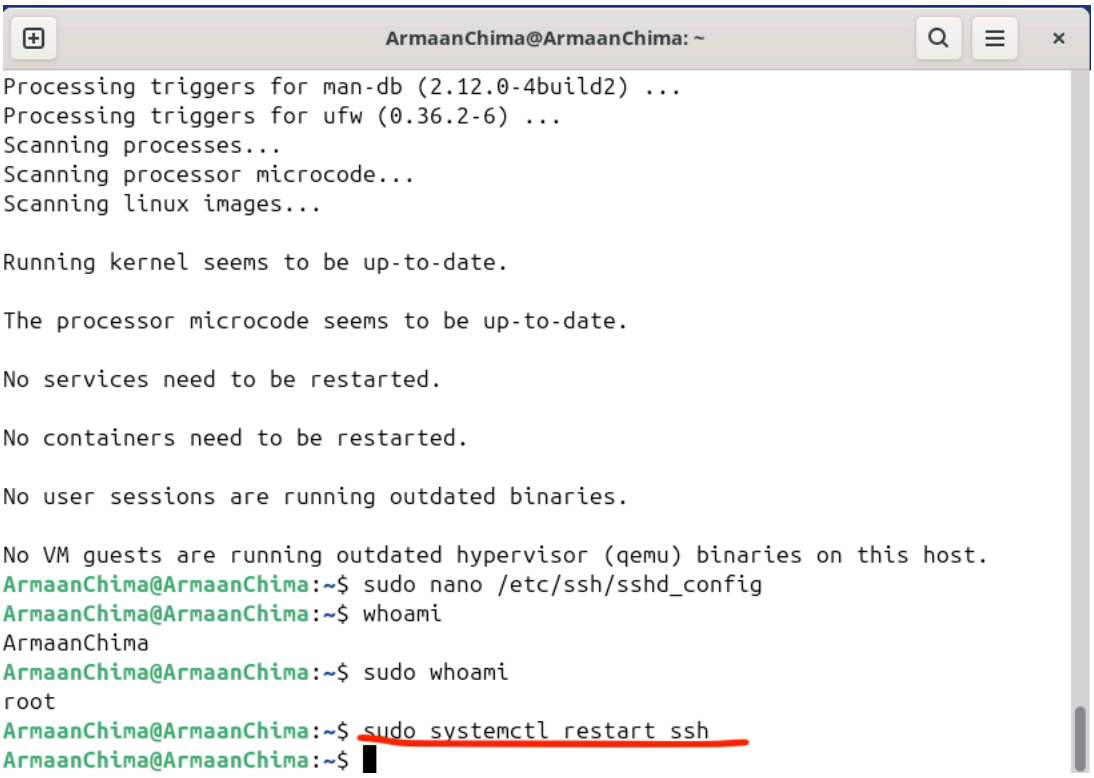
Include – Include means that sshd is going to read other config files

/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

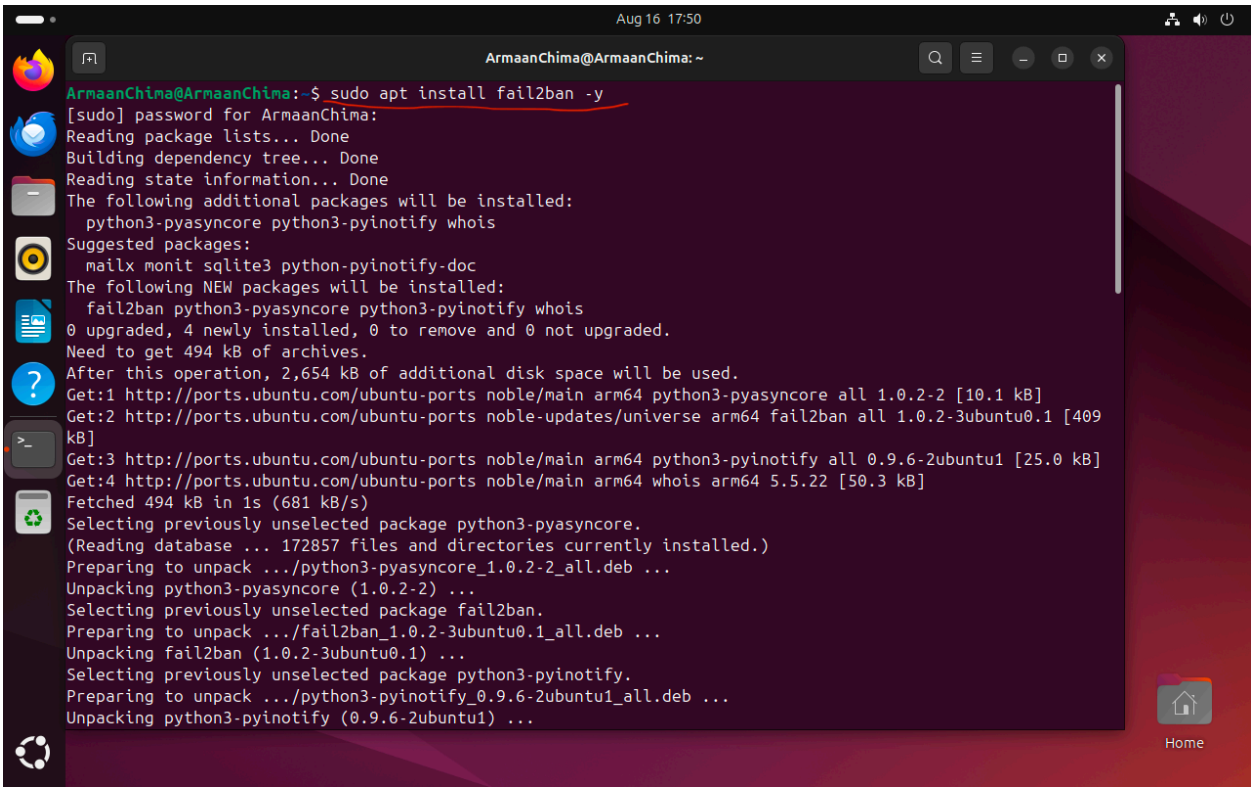


The following command tells us that Root Login is not allowed at all

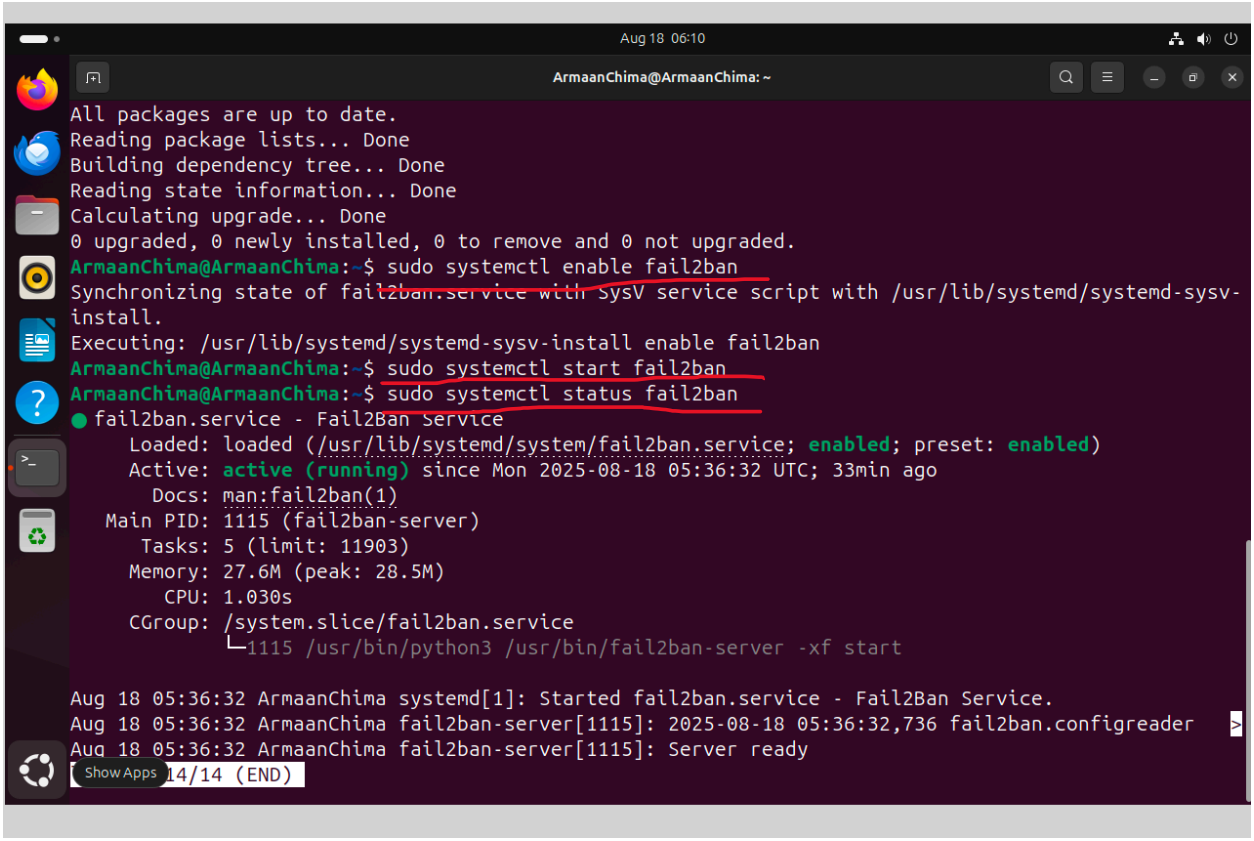


Sudo runs commands using superuser privileges, systemctl can enable, check status, disable, restart, stop or start, and ssh is the service that we are applying this to. In essence, the SSH server is being restarted.

5.) Fail2Ban



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.



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).

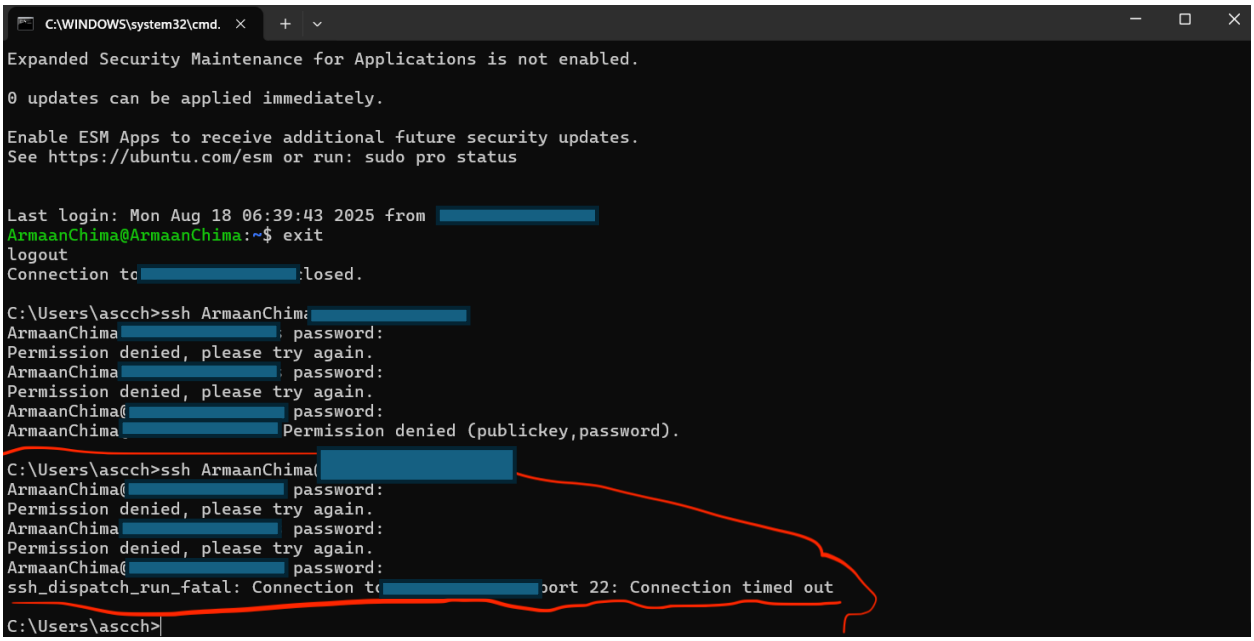
Furthermore, in this photo, we can see that fail2ban is also up and running

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.

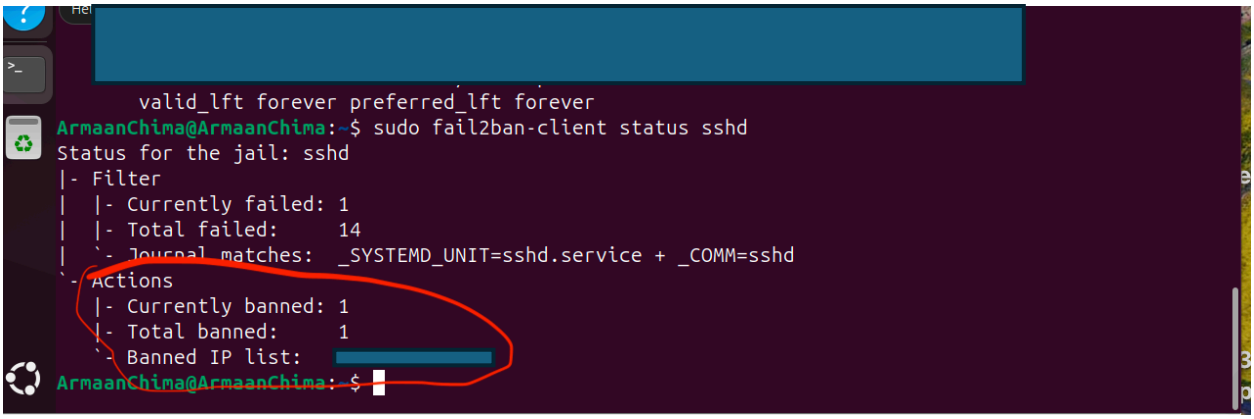
Sudo – this means the command is being run with superuser privileges

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

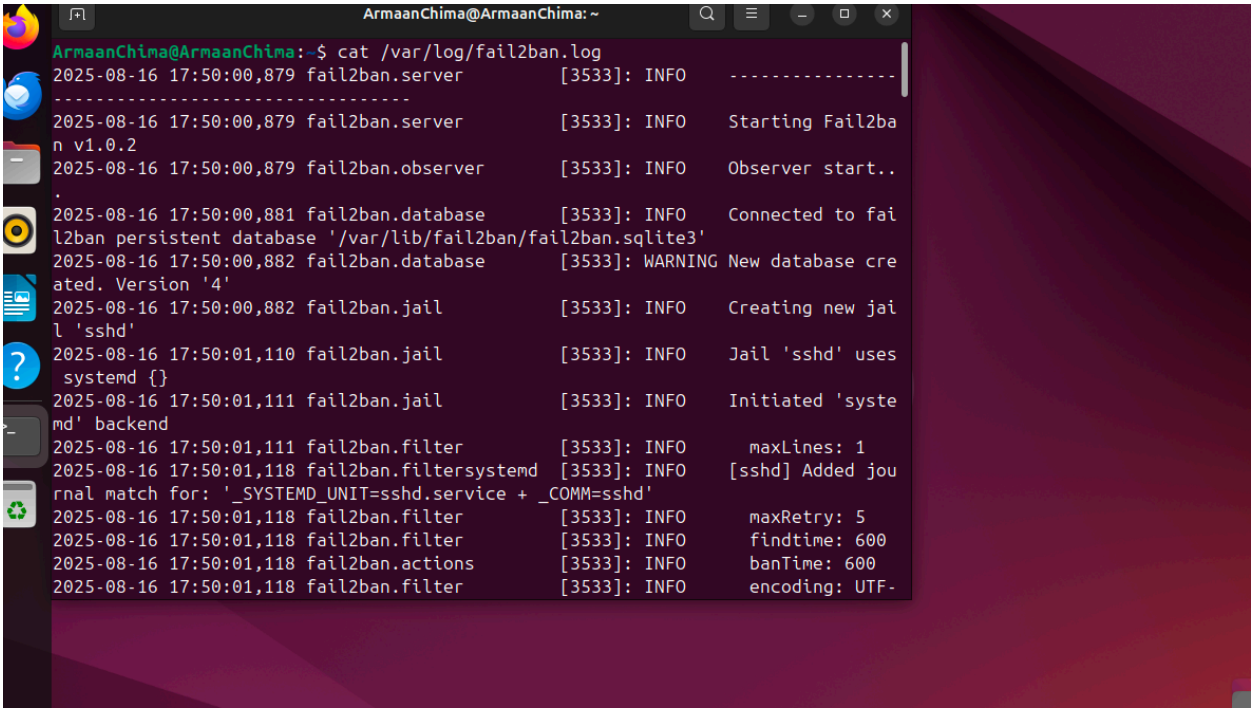
status – shows us the status of fail2ban



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.



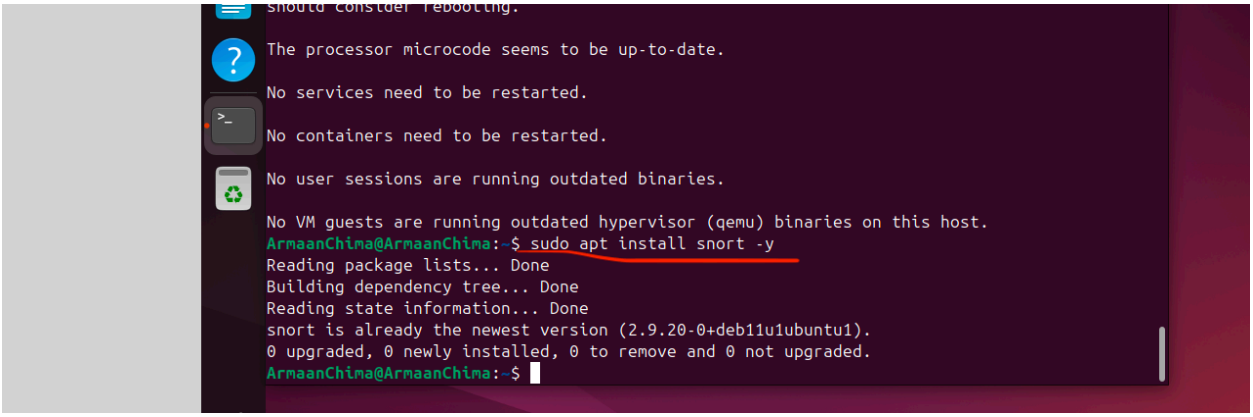
In this photo, we can see that the IP address 192.168.10.10 (Placeholder IP address) has been jailed, showing that our fail2ban works, resulting in that IP being jailed for 10 minutes



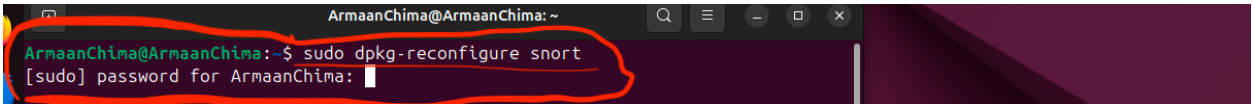
This photo shows us all the events that are “fail2ban-related”. This log tells us and shows us the start of jails, errors, banned IP addresses, and more.

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”

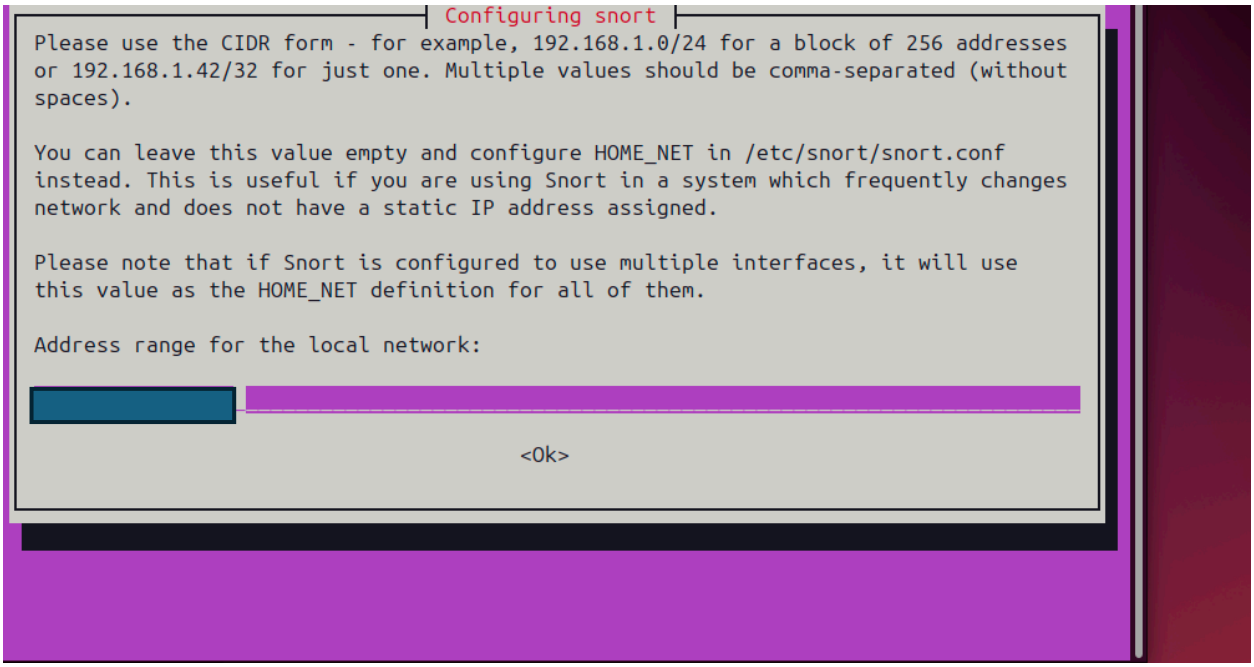
6.) Snort Installation and Configuration



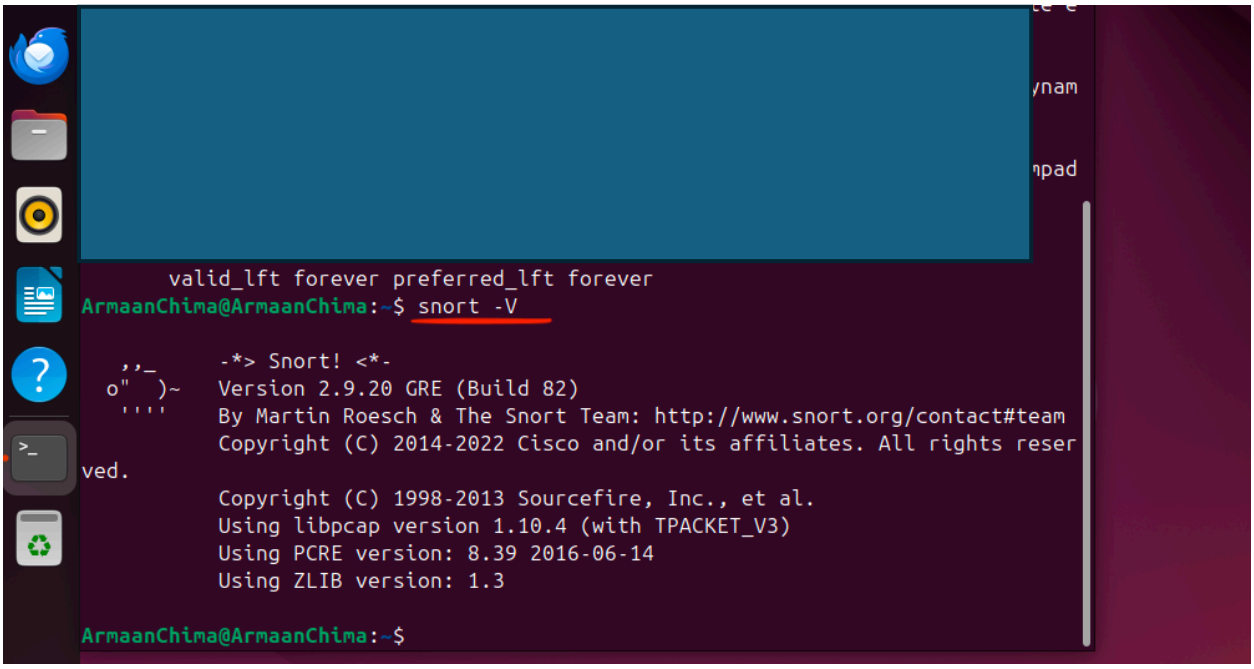
This image shows me using the command “sudo apt install snort -y”. In simple terms, the purpose of this command is to install the Snort package on the system. To break it down, **sudo** means the command is run with superuser privileges, **apt** (advanced package tools) is a package manager used to help 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.



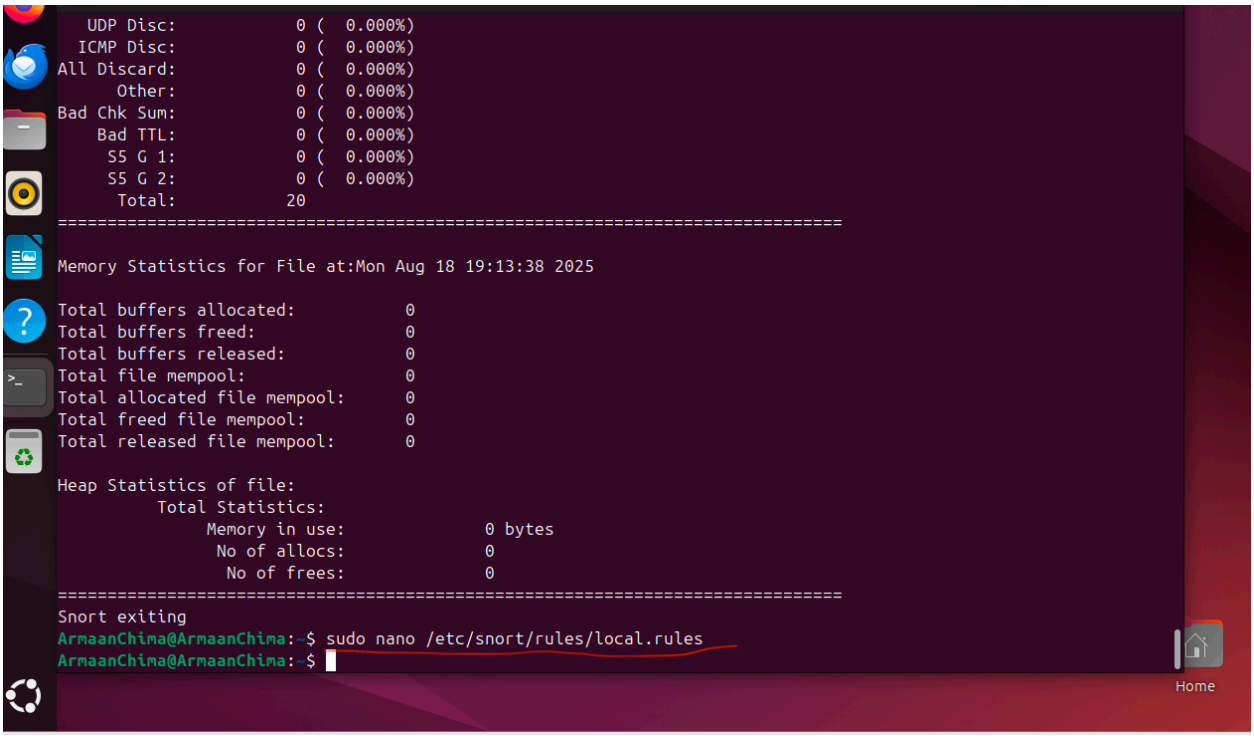
Sudo, which was already talked about, means the command is run with superuser privileges; **dpkg-reconfigure** allows you to reconfigure the settings, allowing you to update the Snort package, and Snort is the package that we are reconfiguring.



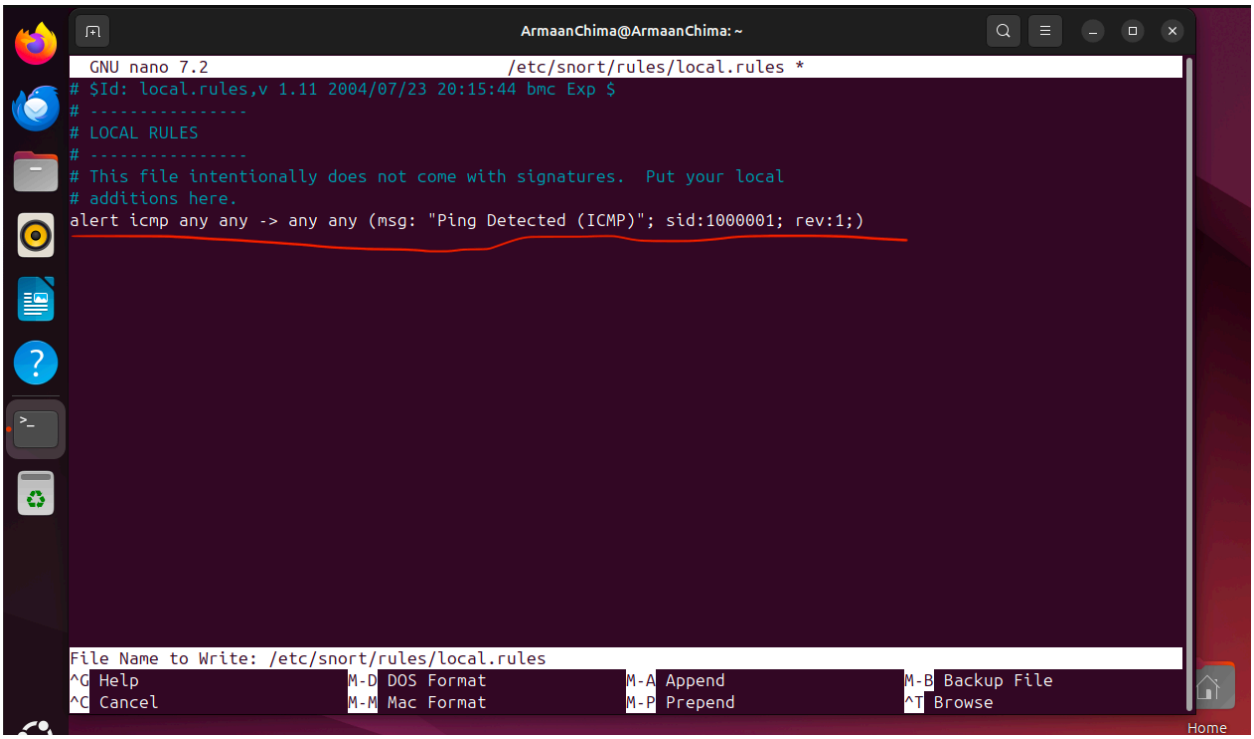
After typing the command sudo dpkg-reconfigure snort, I was taken here where I inserted my IP address. Doing this allowed me to have a functional Snort log, which enabled me to analyze the feedback I was receiving from it (that feedback being traffic and alerts presented in the log). I will talk about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



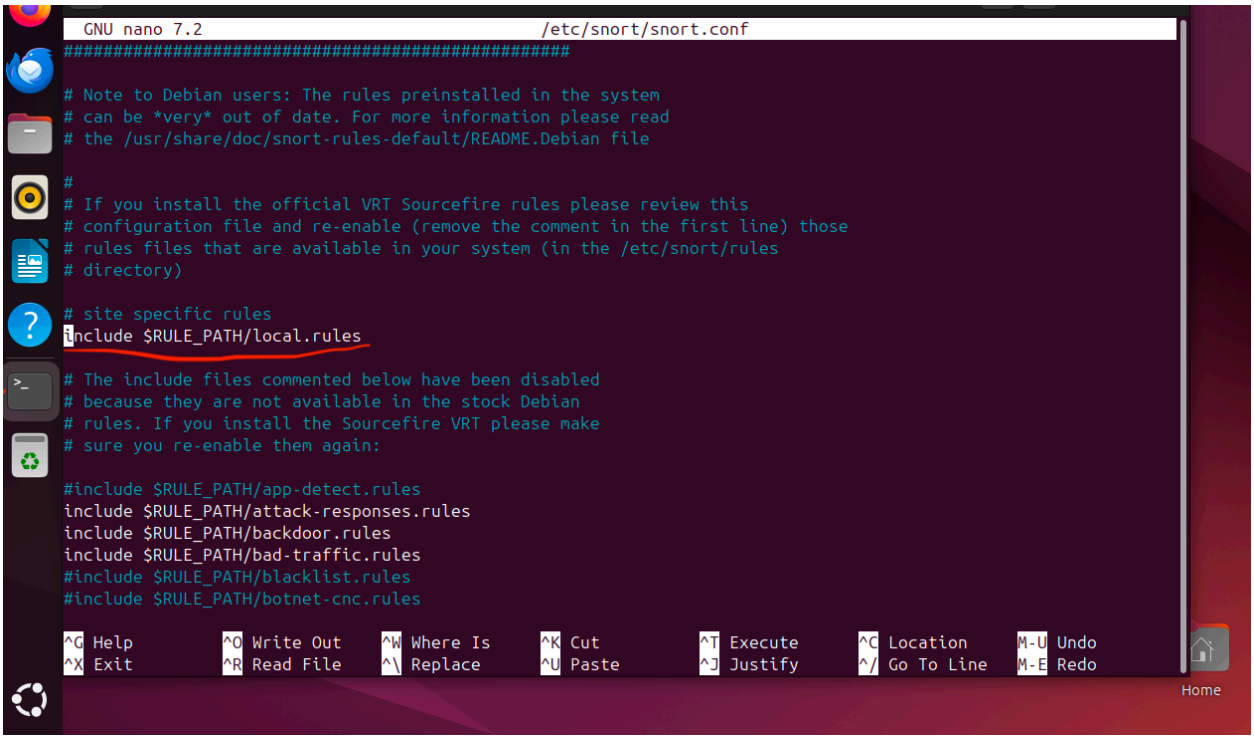
The command snort -V shows the user the version of Snort that is running on their system



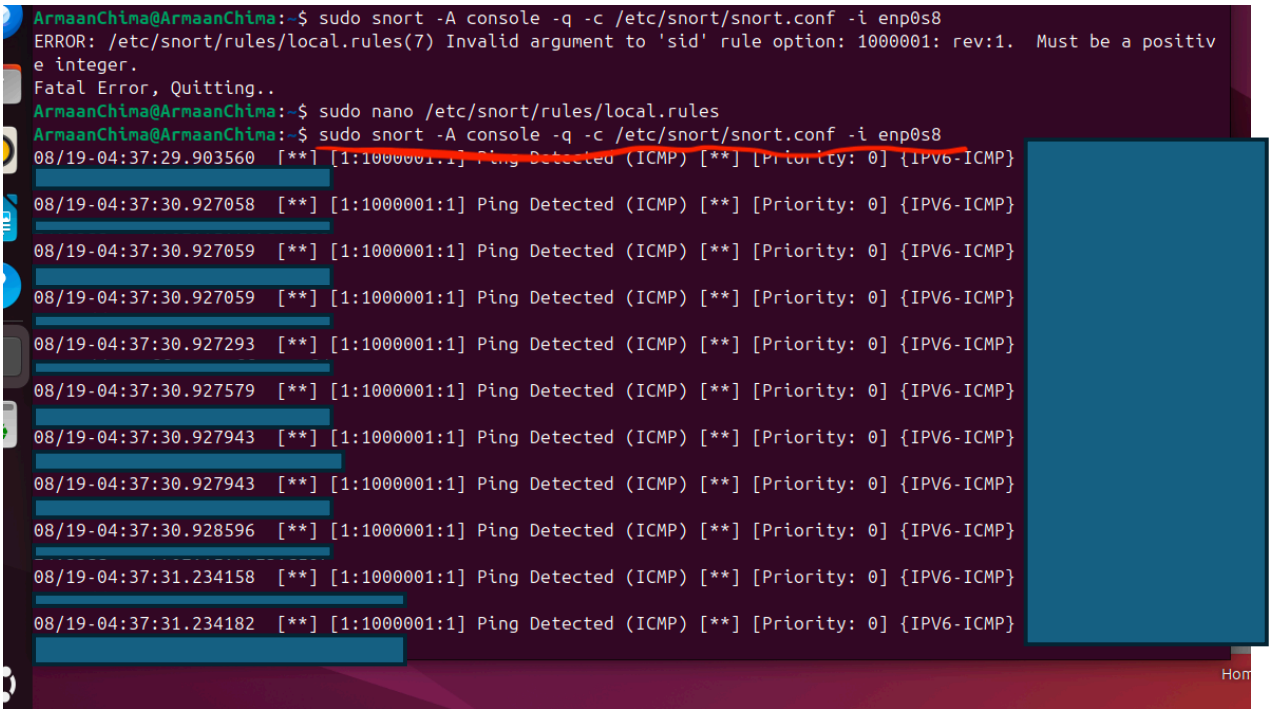
The command sudo means we are running the command with superuser privileges, nano is a text editor that we will need to edit the file in question, and /etc/snort/rules/local.rules is the path that we typed in order to get to the file that we need to edit. In addition, local.rules is a file that does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



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.



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.



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

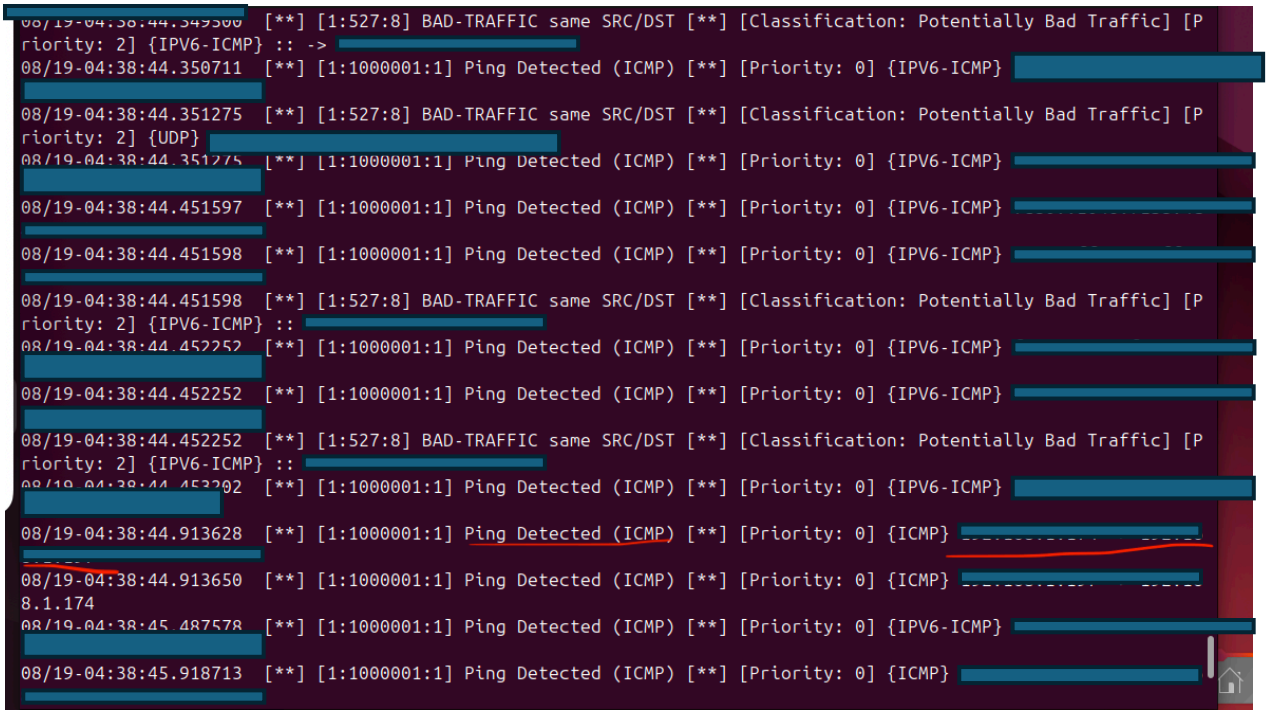
Snort – this is the service in question

-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



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.