Module 12: Evading IDS, Firewalls, and Honeypots

Lab 1: Perform Intrusion Detection using Various Tools

Lab Scenario

The goal of the Intrusion Detection Analyst is to find possible attacks against a network. Recent years have witnessed a significant increase in Distributed Denial-of-Service (DDoS) attacks on the Internet, making network security a great concern. Analysts search for possible attacks by examining IDS logs and packet captures and corroborating them with firewall logs, known vulnerabilities, and general trending data from the Internet. IDS attacks are becoming more sophisticated; automatically reasoning the attack scenarios in real-time, and categorizing them has become a critical challenge. These processes result in huge amounts of data, which analysts must examine to detect a pattern. However, the overwhelming flow of events generated by IDS sensors make it difficult for security administrators to uncover hidden attack plans.

To become an expert penetration tester and security administrator, you must possess sound knowledge of network IPSs, IDSs, malicious network activity, and log information.

Lab Objectives

- Detect intrusions using Snort
- Deploy Cowrie honeypot to detect malicious network traffic

Overview of Intrusion Detection Systems

Intrusion detection systems are highly useful as they monitor both the inbound and outbound traffic of the network and continuously inspects the data for suspicious activities that may indicate a network or system security breach. The IDS checks traffic for signatures that match known intrusion patterns and signals an alarm when a match is detected. It can be categorized into active and passive, depending on its functionality: an IDS is generally passive and is used to detect intrusions, while an intrusion prevention system (IPS) is considered as an active IDS, as it is not only used to detect the intrusion on the network, but also prevent them.

Main Functions of IDS:

- Gathers and analyzes information from within a computer or a network, to identify the possible violations of security policy
- Also referred to as a “packet-sniffer,” which intercepts packets traveling along various communication mediums and protocols
- Evaluates traffic for suspected intrusions and signals an alarm after detection

Task 1: Detect Intrusions using Snort

Snort is an open-source network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis and content

searching/matching and is used to detect a variety of attacks and probes such as buffer overflows, stealth port scans, CGI attacks, SMB probes, and OS fingerprinting attempts. It uses a flexible rules language to describe traffic to collect or pass, as well as a detection engine that utilizes a modular plug-in architecture.

Uses of Snort:

- Straight packet sniffer such as tcpdump
- Packet logger (useful for network traffic debugging, etc.)
- Network intrusion prevention system

Here, we will use Snort to detect network intrusions.

1. Click on Windows 11 to switch to **Windows 11** machine. Click Ctrl+Alt+Delete to activate the machine and login using **Admin/Pa$$w0rd**.

Alternatively, you can also click **Pa$$w0rd** under **Windows 11** machine thumbnail in the **Resources** pane.

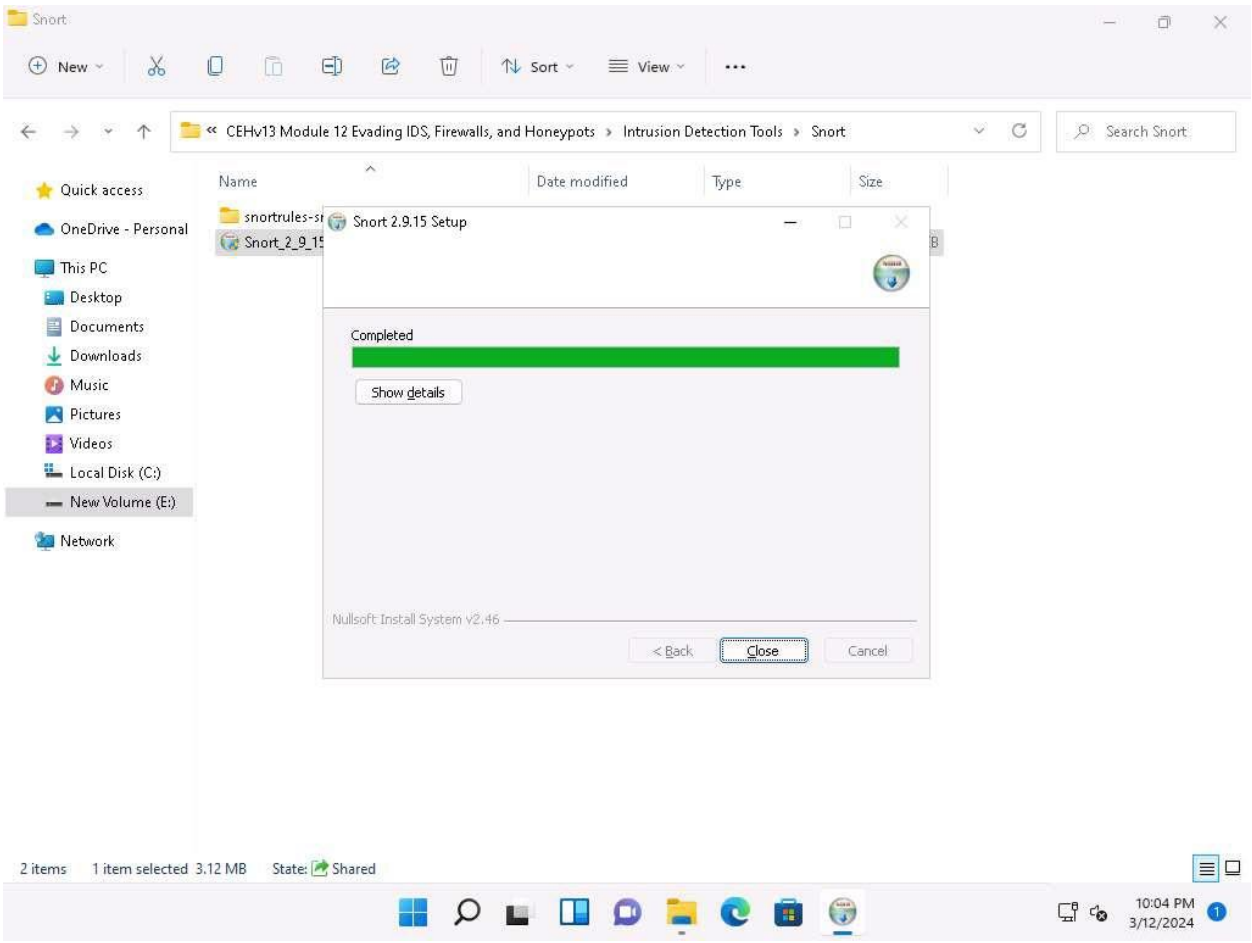
Networks screen appears, click **Yes** to allow your PC to be discoverable by other PCs and devices on the network.

2. Navigate to **E:\CEH-Tools\CEHv13 Module 12 Evading IDS, Firewalls, and Honeypots\Intrusion Detection Tools\Snort** and double-click the **Snort_2_9_15_Installe.x64.exe** file to start the Snort installation.

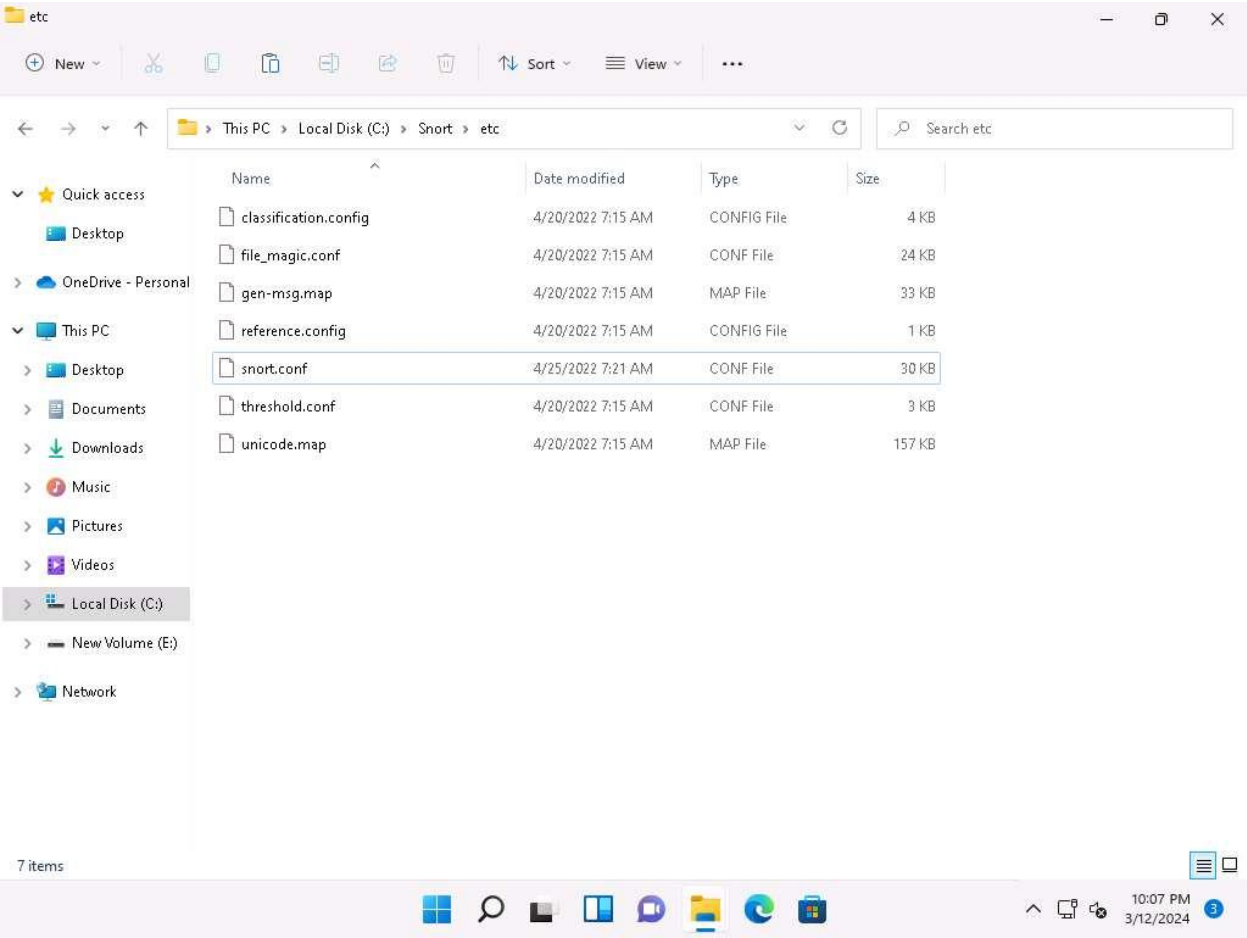
If an **Open File - Security warning** pop-up window appears, click **Yes**.

3. Accept the **License Agreement** and install Snort by selecting the default options that appear step by step in the wizard.
4. A window appears after the successful installation of Snort; click **Close**.
5. Click **OK** to exit the **Snort Installation** window.

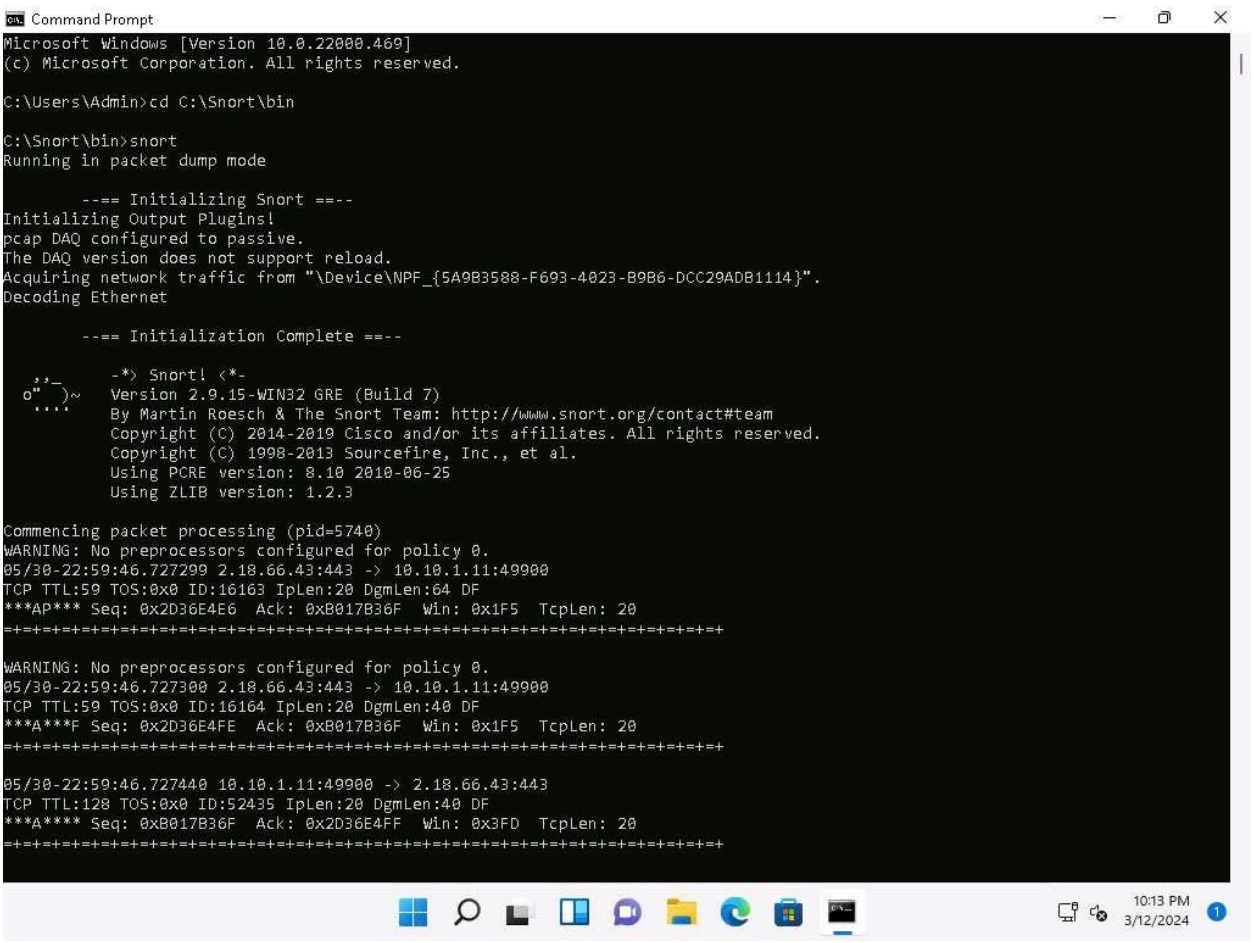
Snort requires **WinPcap** to be installed on your machine. In this task environment, we have already installed WinPcap drivers for packet capturing.



6. By default, Snort installs itself in **C:\Snort** (C:\ or D:\, depending on the disk drive in which the OS is installed).
7. Navigate to the **etc** folder in the specified location, **E:\CEH-Tools\CEHv13 Module 12 Evading IDS, Firewalls, and Honeypots\Intrusion Detection Tools\Snort\snortrules-snapshot-29150\etc** of the Snort rules; copy **snort.conf** and paste it in **C:\Snort\etc**.
8. **snort.conf** is already present in **C:\Snort\etc**; replace the file with the newly copied file.

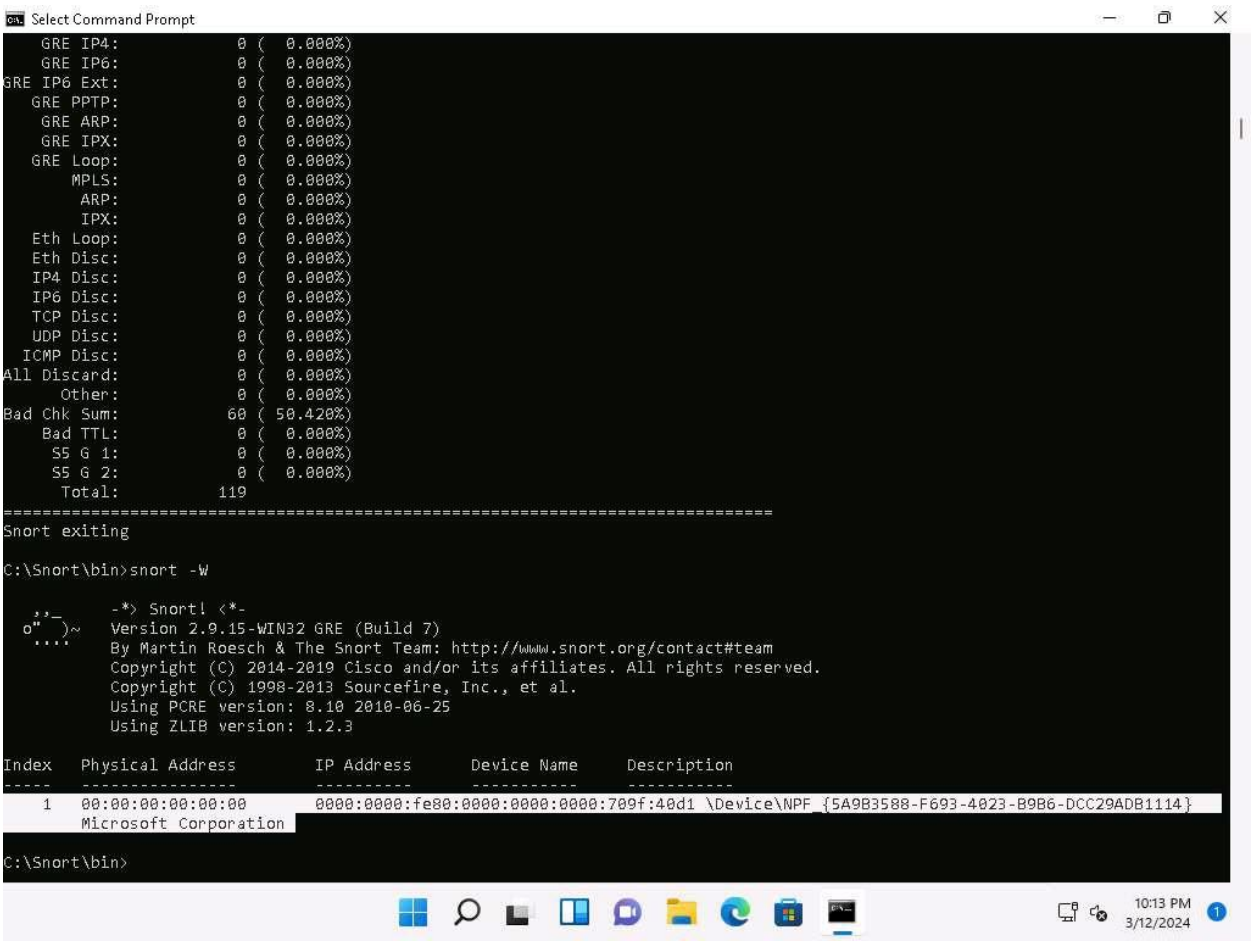


9. Copy the **so_rules** folder from **E:\CEH-Tools\CEHv13 Module 12 Evading IDS, Firewalls, and Honeypots\Intrusion Detection Tools\Snort\snortrules-snapshot-29150** and paste into **C:\Snort**.
10. Copy the **preproc_rules** folder from **E:\CEH-Tools\CEHv13 Module 12 Evading IDS, Firewalls, and Honeypots\Intrusion Detection Tools\Snort\snortrules-snapshot-29150**, and paste it into **C:\Snort**. The **preproc_rules** folder is already present in **C:\Snort**; replace this folder with the **preproc_rules** folder taken from the specified location.
11. Using the same method, copy the **rules** folder from **E:\CEH-Tools\CEHv13 Module 12 Evading IDS, Firewalls, and Honeypots\Intrusion Detection Tools\Snort\snortrules-snapshot-29150** and paste into **C:\Snort**.
12. Now, right-click on the **Windows Start** icon and click **Run** from the menu.
13. The **Run** window appears; type **cmd** in the **Open** field and click **OK** to launch command prompt window.
14. The **Command Prompt** window appears; type **cd C:\Snort\bin** and press **Enter** to access the bin folder in the command prompt. Run **snort** command to initiate snort.

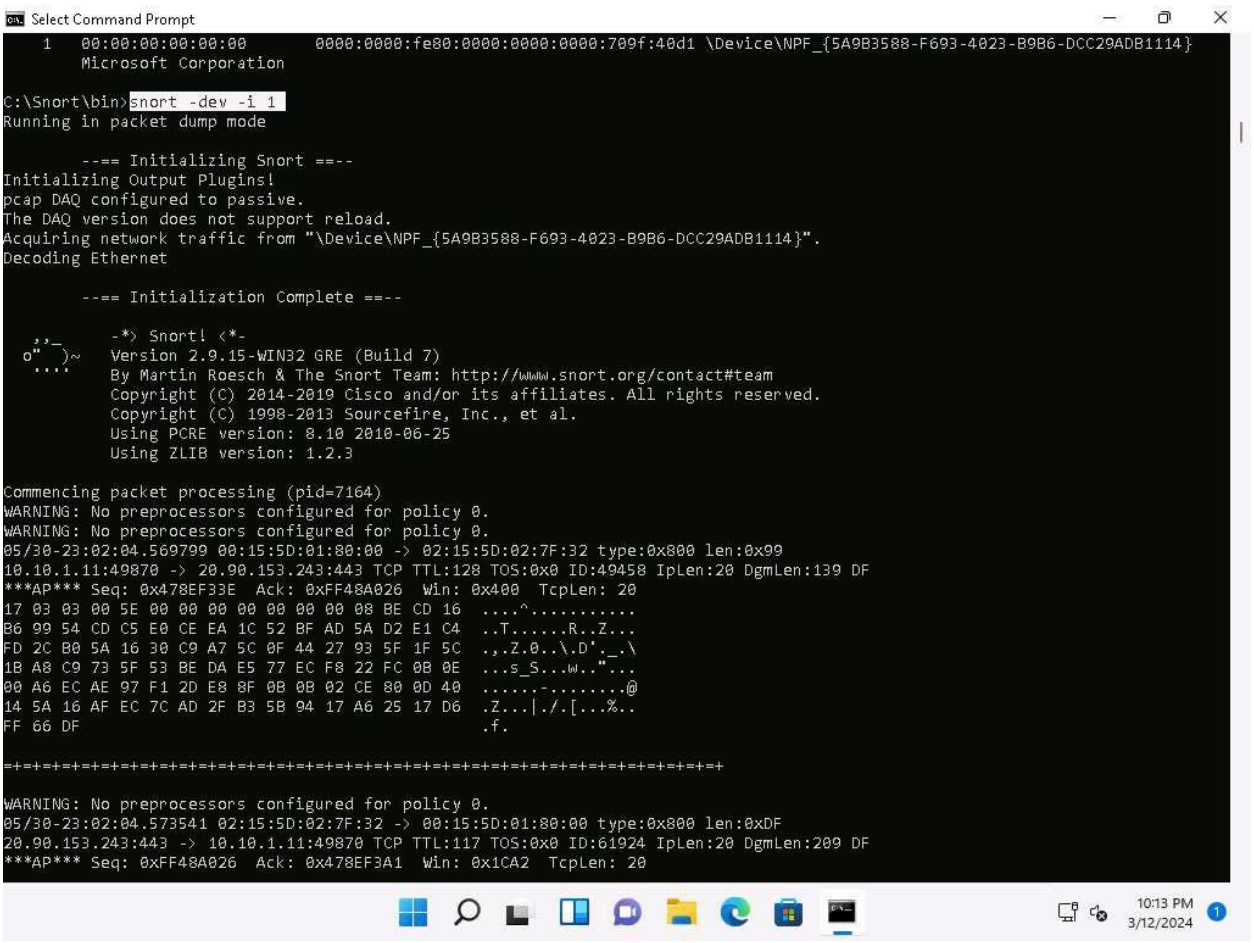


15. Snort initializes; wait for it to complete. Press **Ctrl+C** after some time, Snort exits and comes back to **C:\Snort\bin**.

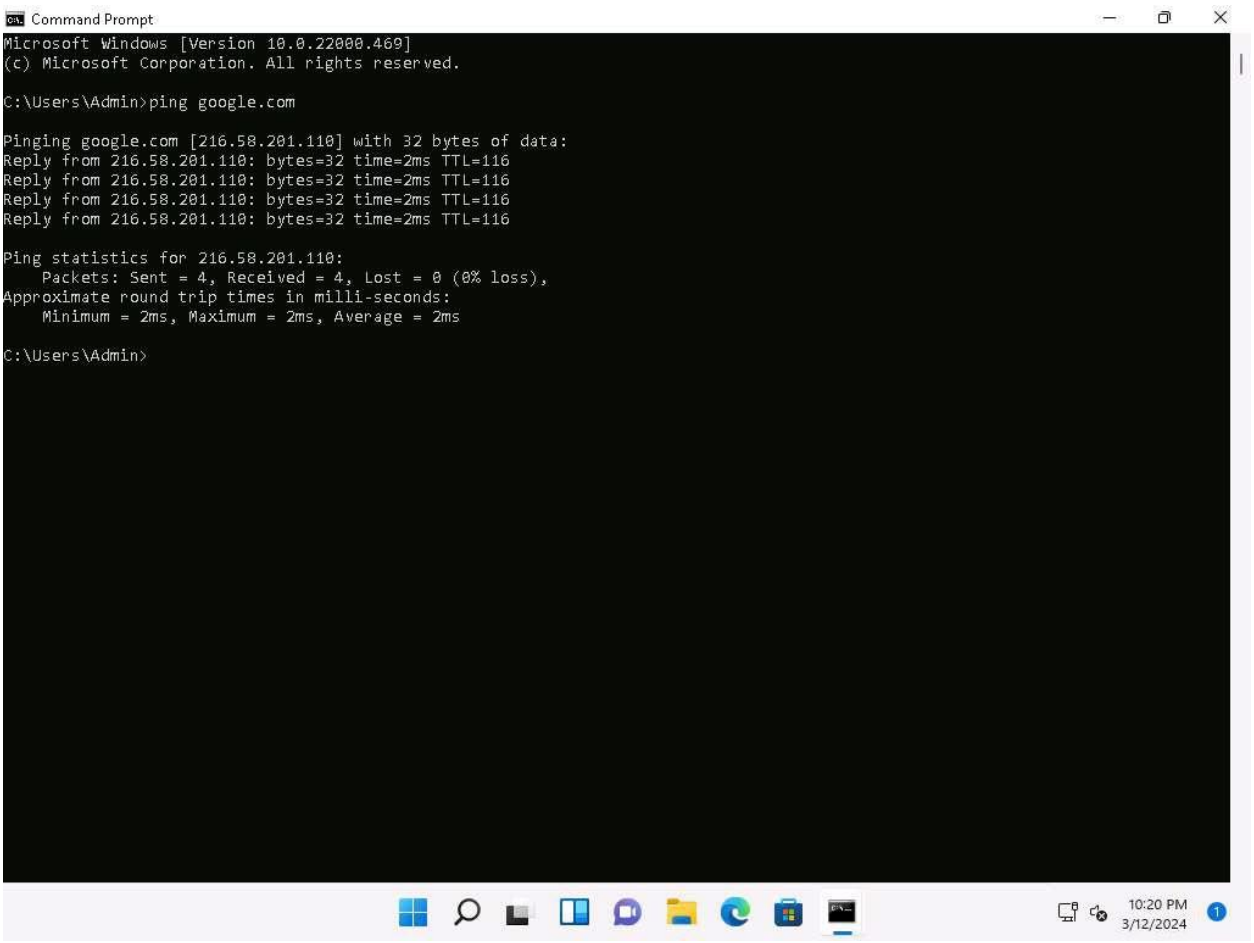
16. Now type **snort -W**. This command lists your machine's physical address, IP address, and Ethernet Drivers, but all are disabled by default.



17. Observe your Ethernet Driver **index number** and write it down (in this task, it is **1**).
18. To enable the Ethernet Driver, in the command prompt, run command **snort -dev -i 1**.
19. You see a rapid scroll text in the command prompt, which means that the Ethernet Driver is enabled and working properly.

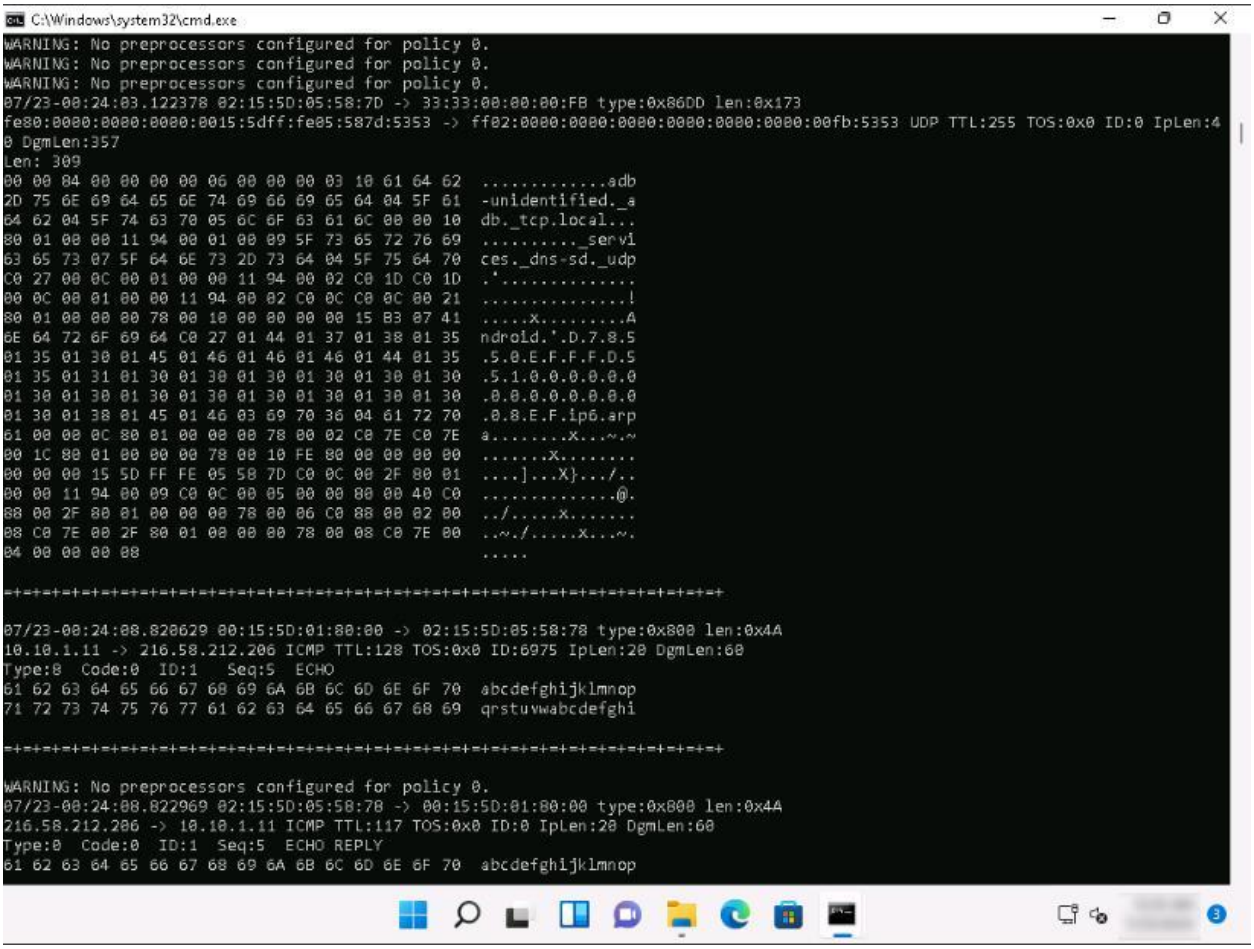


20. Leave the Snort command prompt window open, and launch another command prompt window.
21. In a new command prompt, run **ping google.com** command.



22. This ping command triggers a Snort alert in the Snort command prompt with rapid scrolling text.

The Google IP address will differ when you perform this task.



23. Close both command prompt windows. The verification of Snort installation and the triggering alert is complete, and Snort is working correctly in verbose mode.

24. Configure the **snort.conf** file, located at **C:\Snort\etc**.

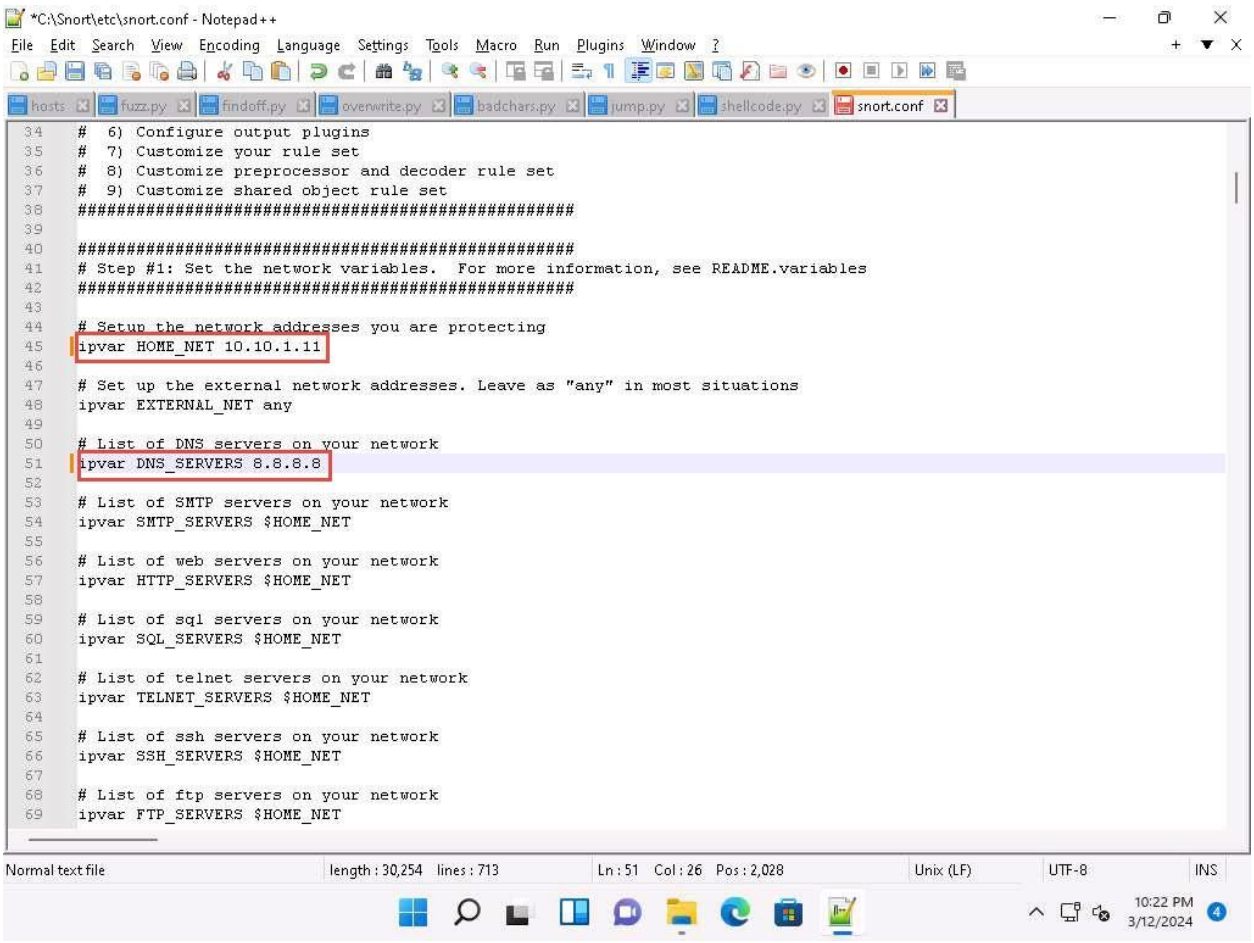
25. Open the **snort.conf** file with **Notepad++**.

26. Scroll down to the **Step #1: Set the network variables** section (Line 41) of the **snort.conf** file. In the **HOME_NET** line (Line 45), replace **any** with the IP addresses of the machine (target machine) on which Snort is running. Here, the target machine is **Windows 11** and the IP address is **10.10.1.11**.

27. Leave the **EXTERNAL_NET any** line as it is.

28. If you have a **DNS Server**, then make changes in the **DNS_SERVERS** line by replacing **$HOME_NET** with your DNS Server IP address; otherwise, leave this line as it is.

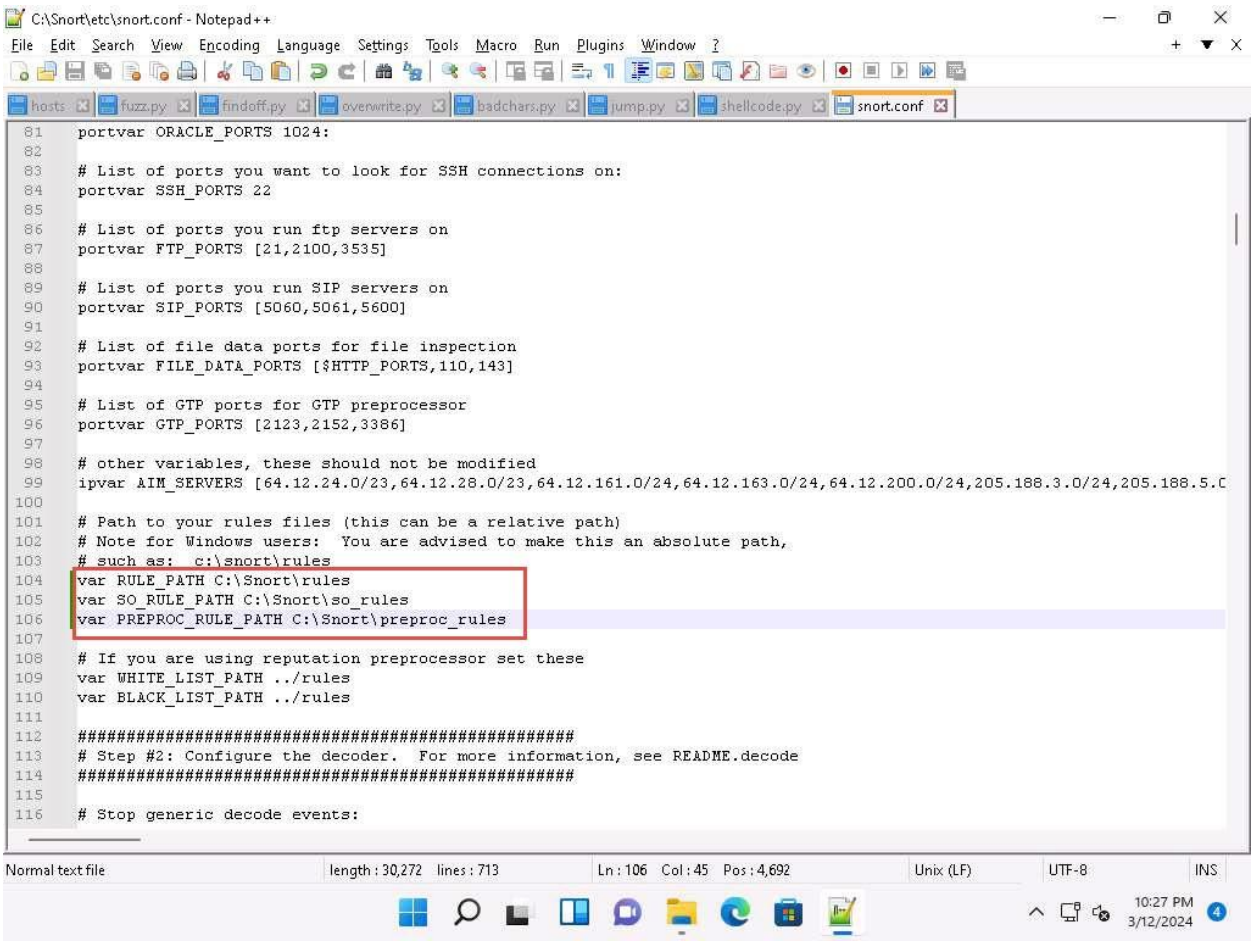
Here, the DNS server is **8.8.8.8**.



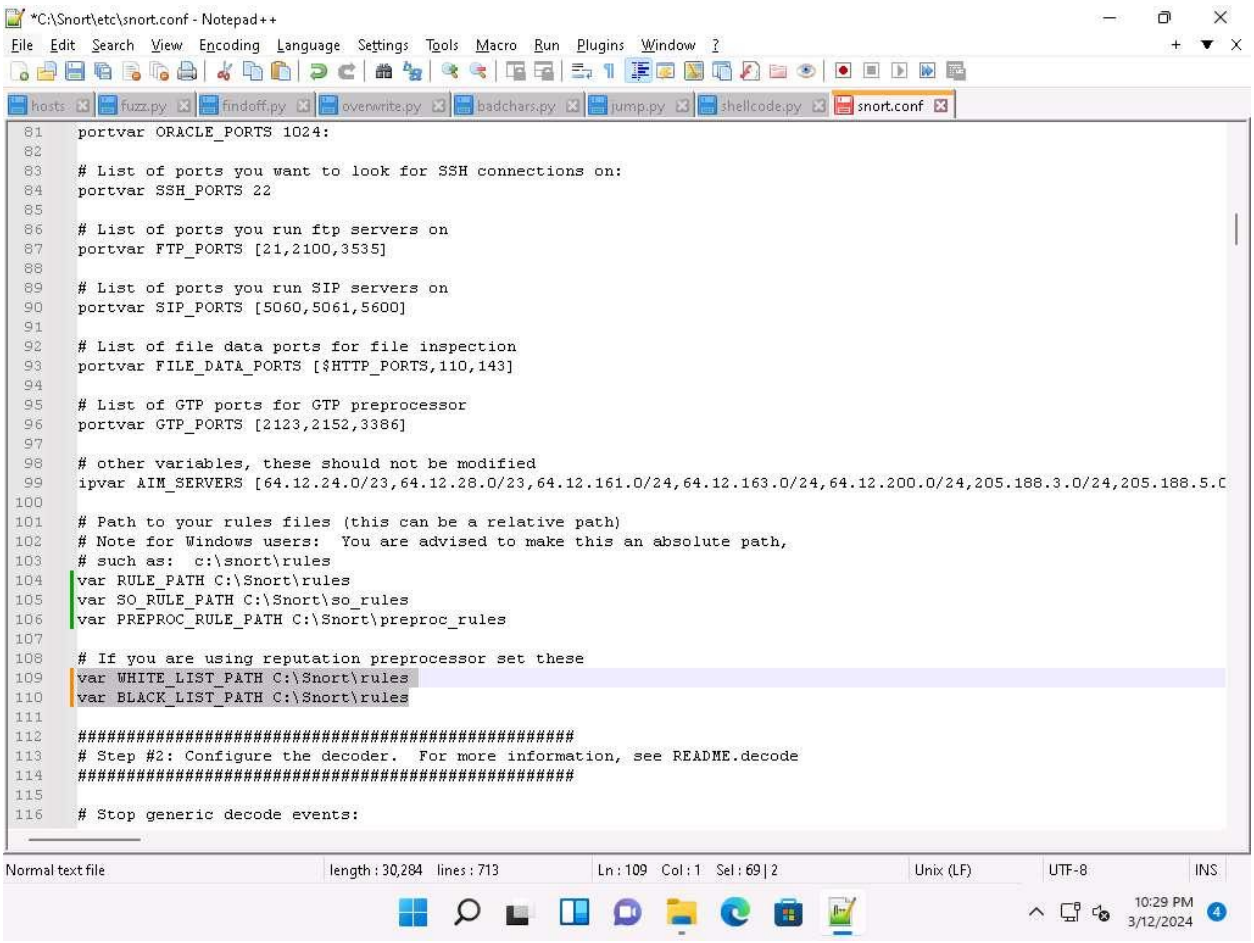
29. The same applies to SMTP_SERVERS, HTTP_SERVERS, SQL_SERVERS, TELNET_SERVERS, and SSH_SERVERS.

30. Remember that if you do not have any servers running on your machine, leave the line as it is. **DO NOT** make any changes in that line.

31. Scroll down to **RULE_PATH** (Line 104). In Line 104, replace **../rules** with **C:\Snort\rules** in Line 105, replace **../so_rules** with **C:\Snort\so_rules** and in Line 106, replace **../preproc_rules** with **C:\Snort\preproc_rules**.



32. In Lines 109 and 110, replace **../rules** with **C:\Snort\rules**. Minimize the **Notepad++** window.



33. Navigate to **C:\Snort\rules**, and create two text files; name them **white_list** and **black_list** and change their file extensions from **.txt** to **.rules**.

To create a text file, right-click anywhere inside the rules window and navigate to **New --> Text Document**.

34. While changing the extension, if any pop-up appears, click **Yes**.

35. Switch back to **Notepad++**, scroll down to the **Step #4: Configure dynamic loaded libraries** section (Line 238). **Configure dynamic loaded libraries** in this section.

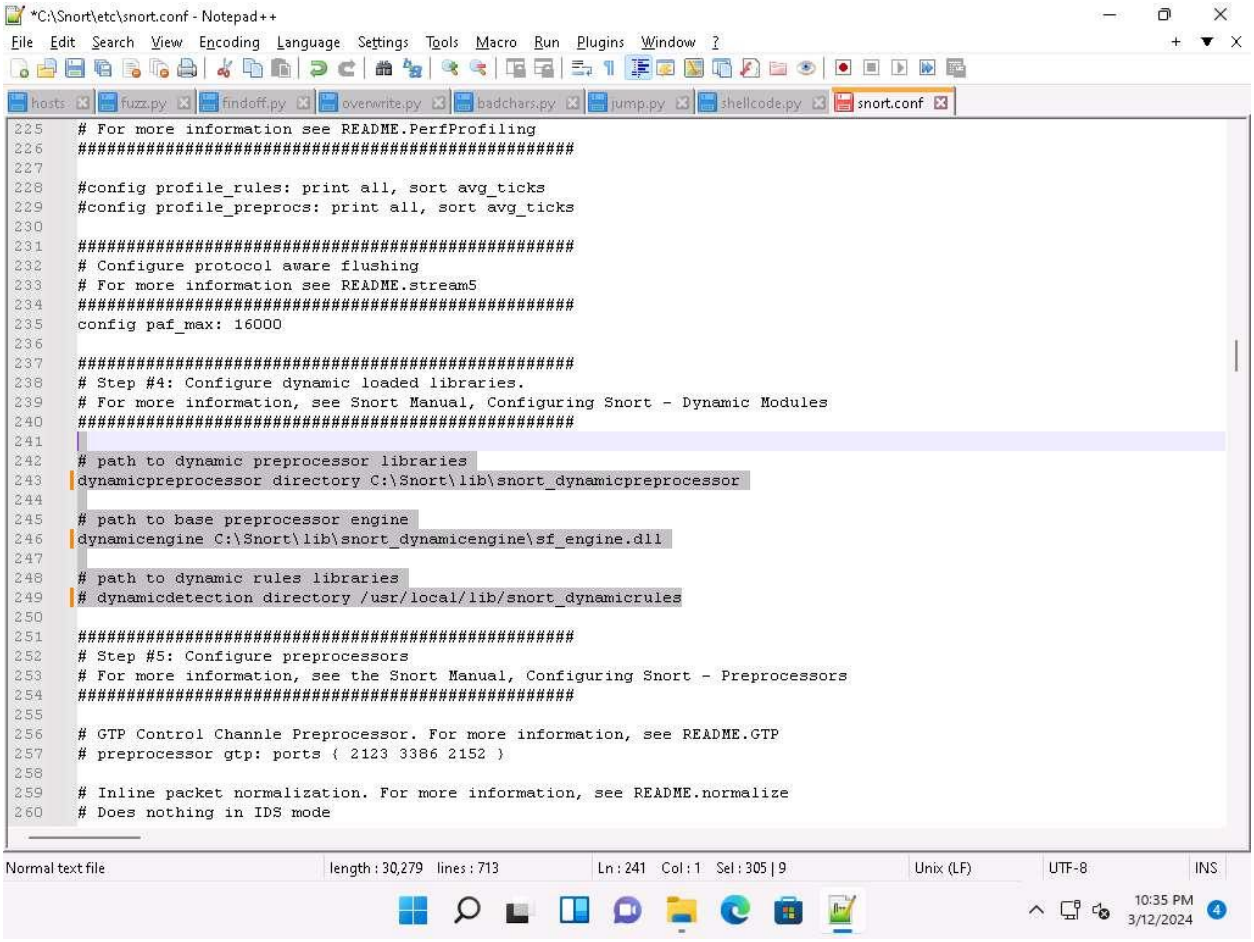
36. Add the path to dynamic preprocessor libraries (Line 243);
replace **/usr/local/lib/snort_dynamicpreprocessor/** with your dynamic preprocessor libraries folder location.

37. In this task, the dynamic preprocessor libraries are located at **C:\Snort\lib\snort_dynamicpreprocessor**.

38. At the path to base preprocessor (or dynamic) engine (Line 246),
replace **/usr/local/lib/snort_dynamicengine/libsf_engine.so** with your base preprocessor engine **C:\Snort\lib\snort_dynamicengine\sf_engine.dll**.

39. Ensure that the dynamic rules libraries (Line 249) is commented out, as you have already configured the libraries in dynamic preprocessor libraries.

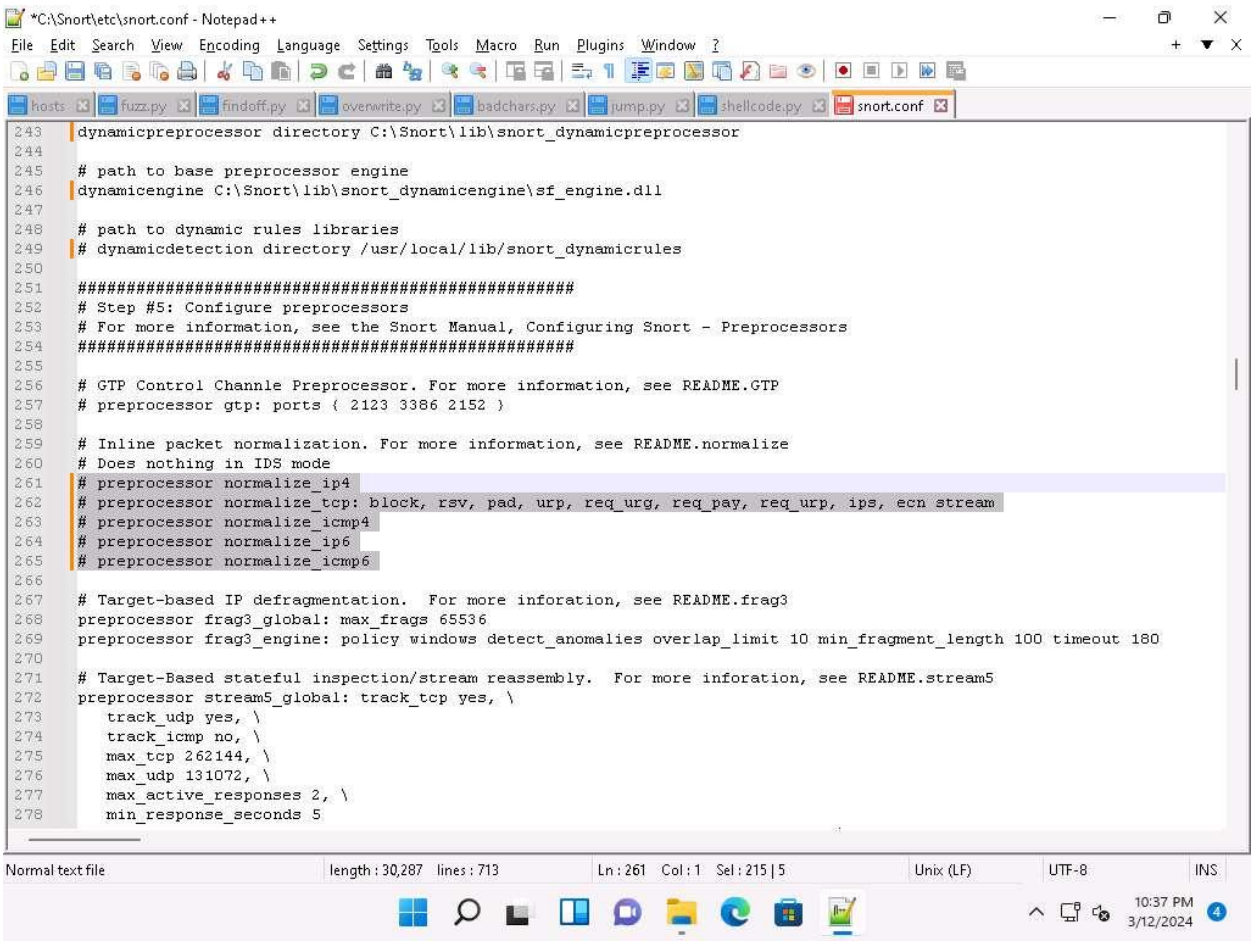
Add (**space**) in between **#** and dynamicdetection (Line 249).



40. Scroll down to the **Step #5: Configure preprocessors** section (Line 253), the listed preprocessor. This does nothing in IDS mode, however, it generates errors at runtime.

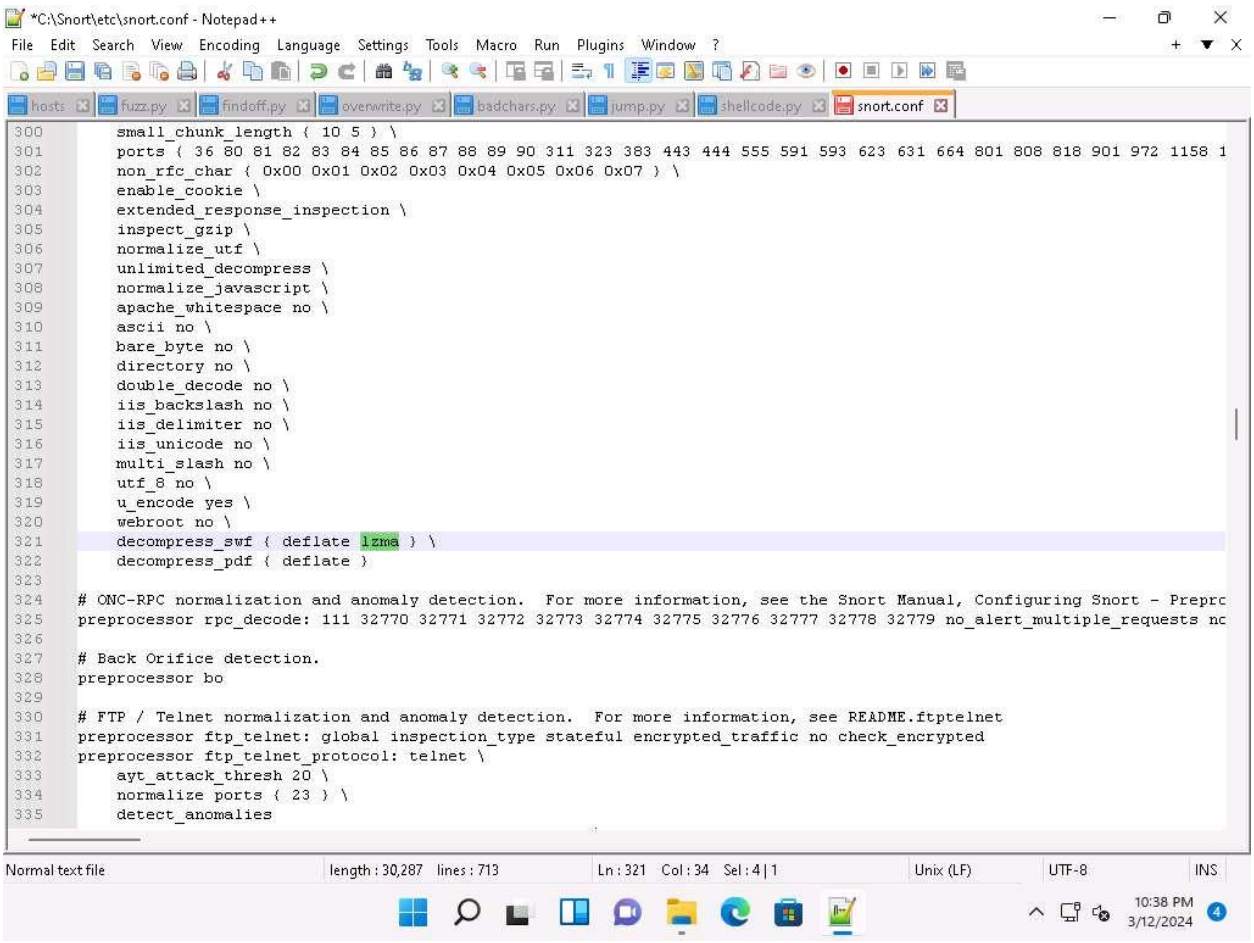
41. Comment out all the preprocessors listed in this section by adding '**#**' and (**space**) before each preprocessor rule (261-265).

To 'comment out' is to render a block of code inert by turning it into a comment.



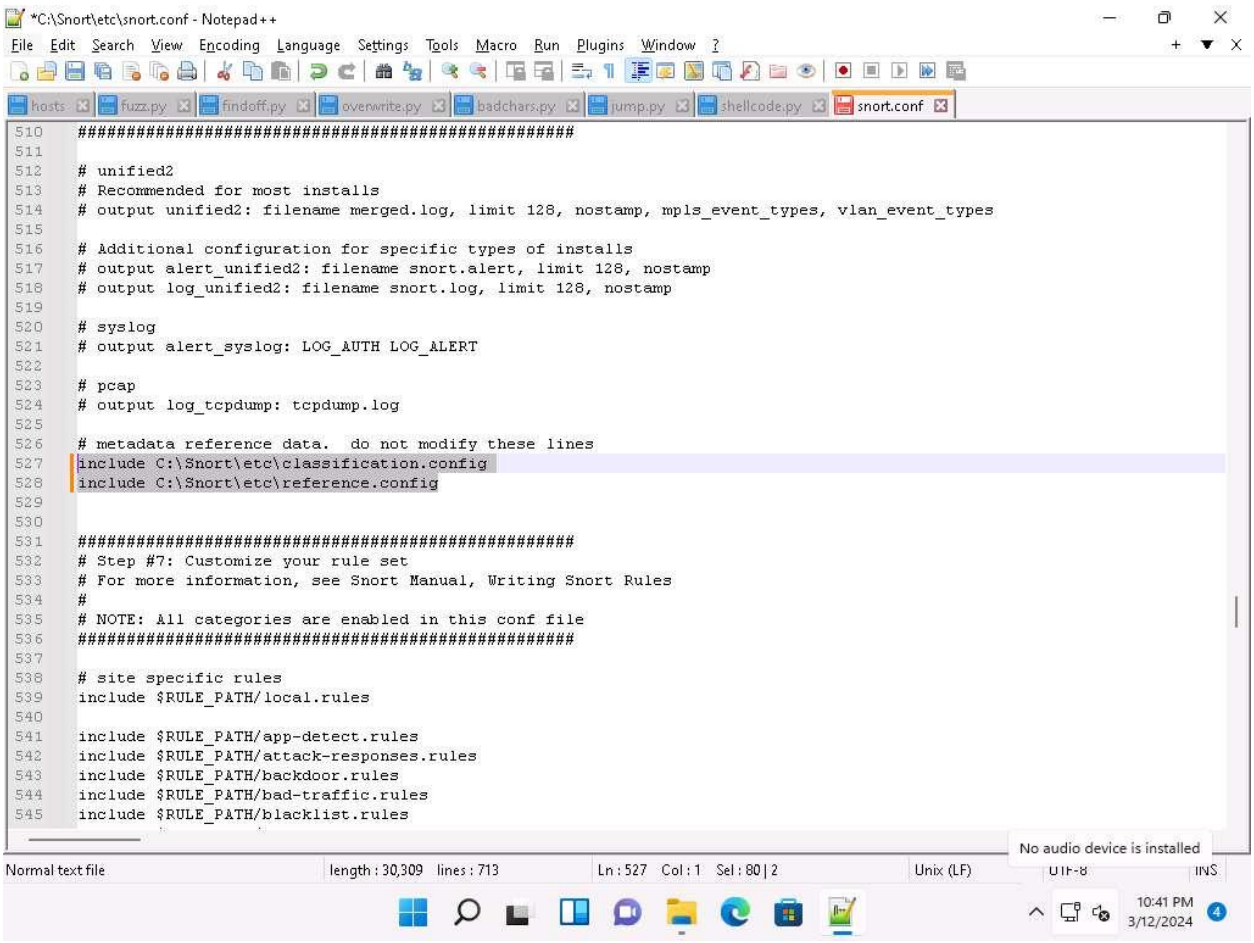
42. Scroll down to line 321 and delete **lzma** keyword and a (**space**).

Make sure you only delete **"lzma"** keyword.

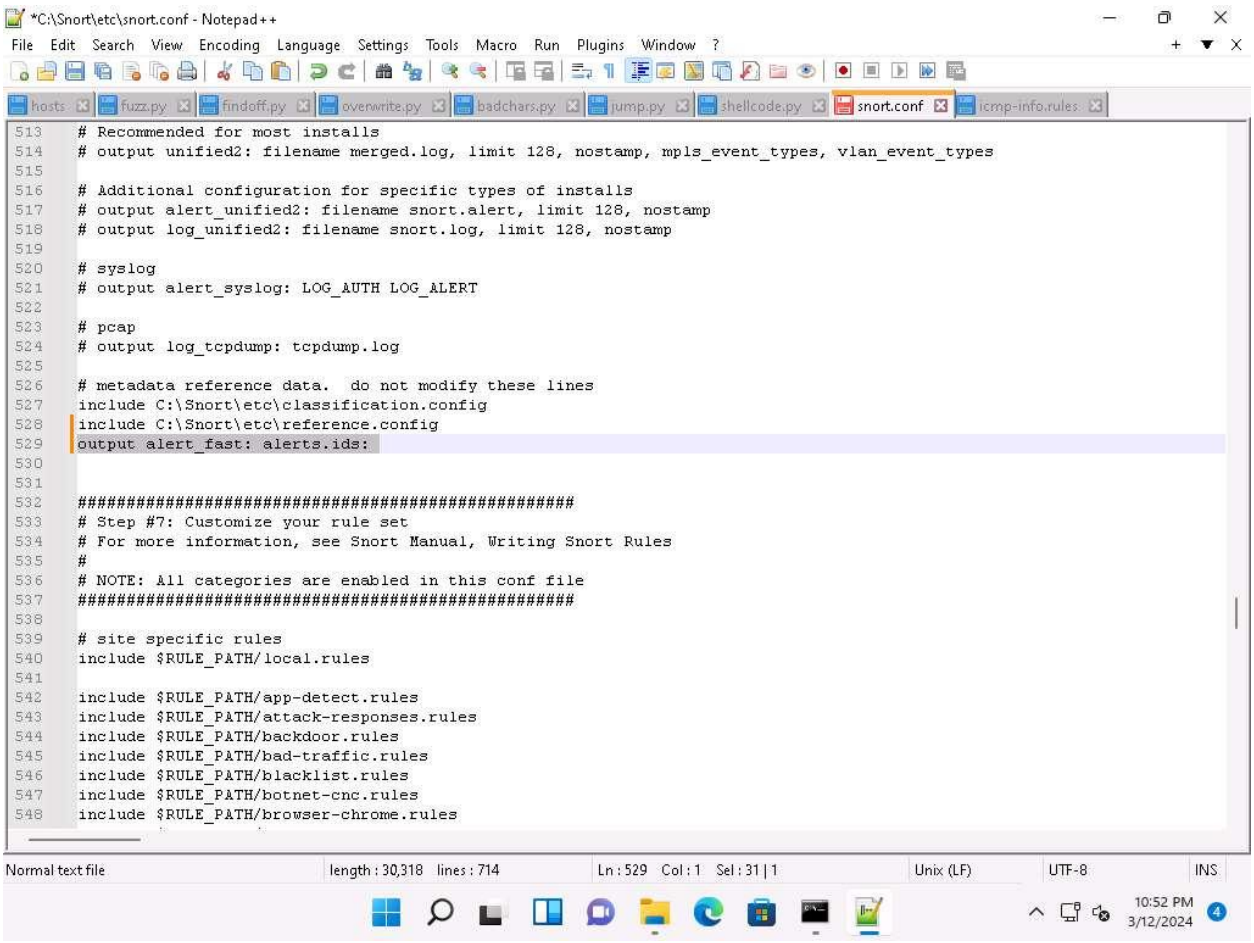


43. Scroll down to **Step #6: Configure output plugins** (Line 513). In this step, provide the location of the **classification.config** and **reference.config** files.

44. These two files are in **C:\Snort\etc**. Provide this location of files in the configure output plugins (in Lines 527 and 528)
(i.e., **C:\Snort\etc\classification.config** and **C:\Snort\etc\reference.config**).



45. In **Step #6**, add to line (529) **output alert_fast: alerts.ids:** this command orders Snort to dump all logs into the **alerts.ids** file.

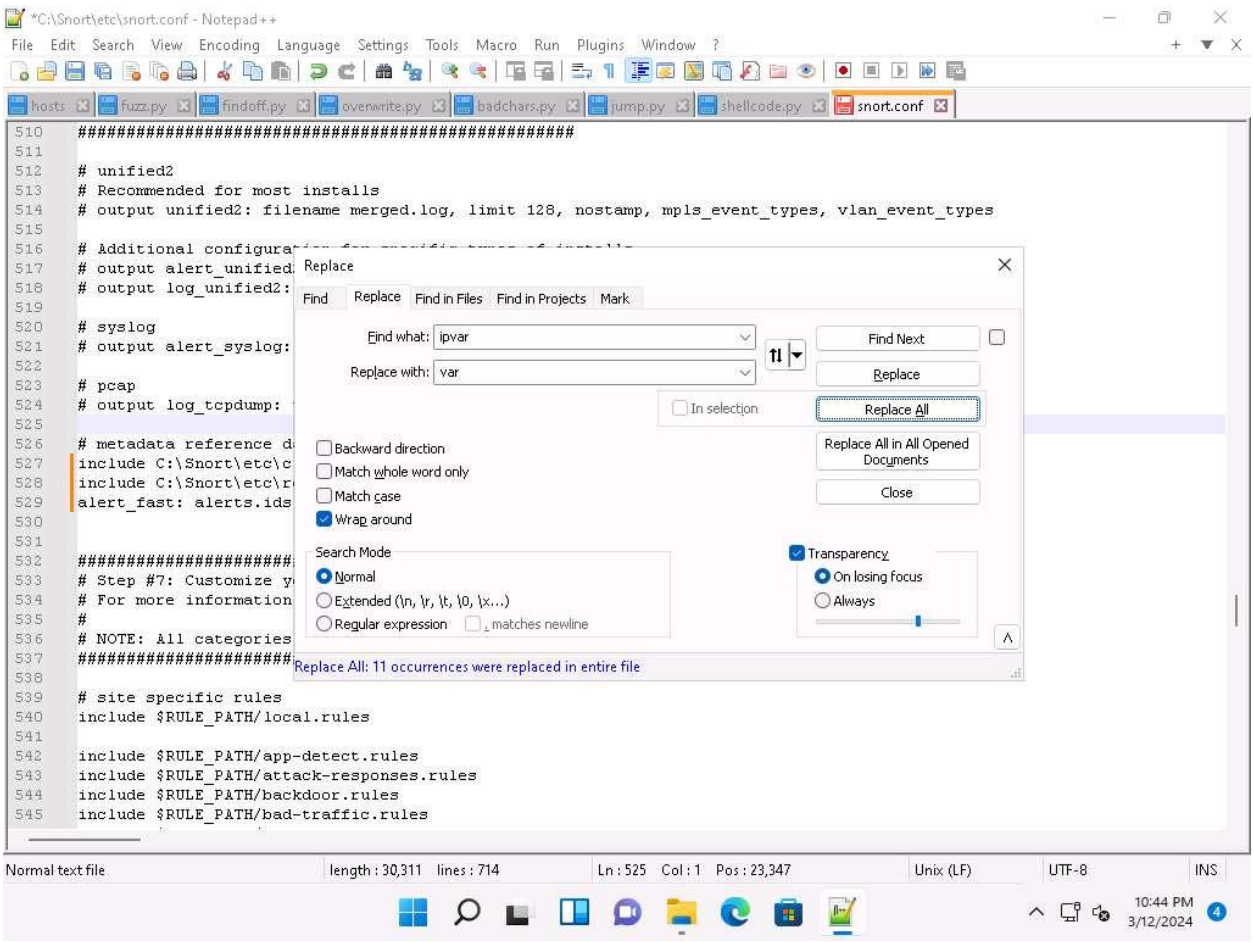


46. In the **snort.conf** file, find and replace the **ipvar** string with **var**. To do this, press **Ctrl+H** on the keyboard. The **Replace** window appears; enter **ipvar** in the **Find what** : text field, enter **var** in the **Replace with** : text field, and click **Replace All**.

You will get a notification saying 11 occurrences were replaced.

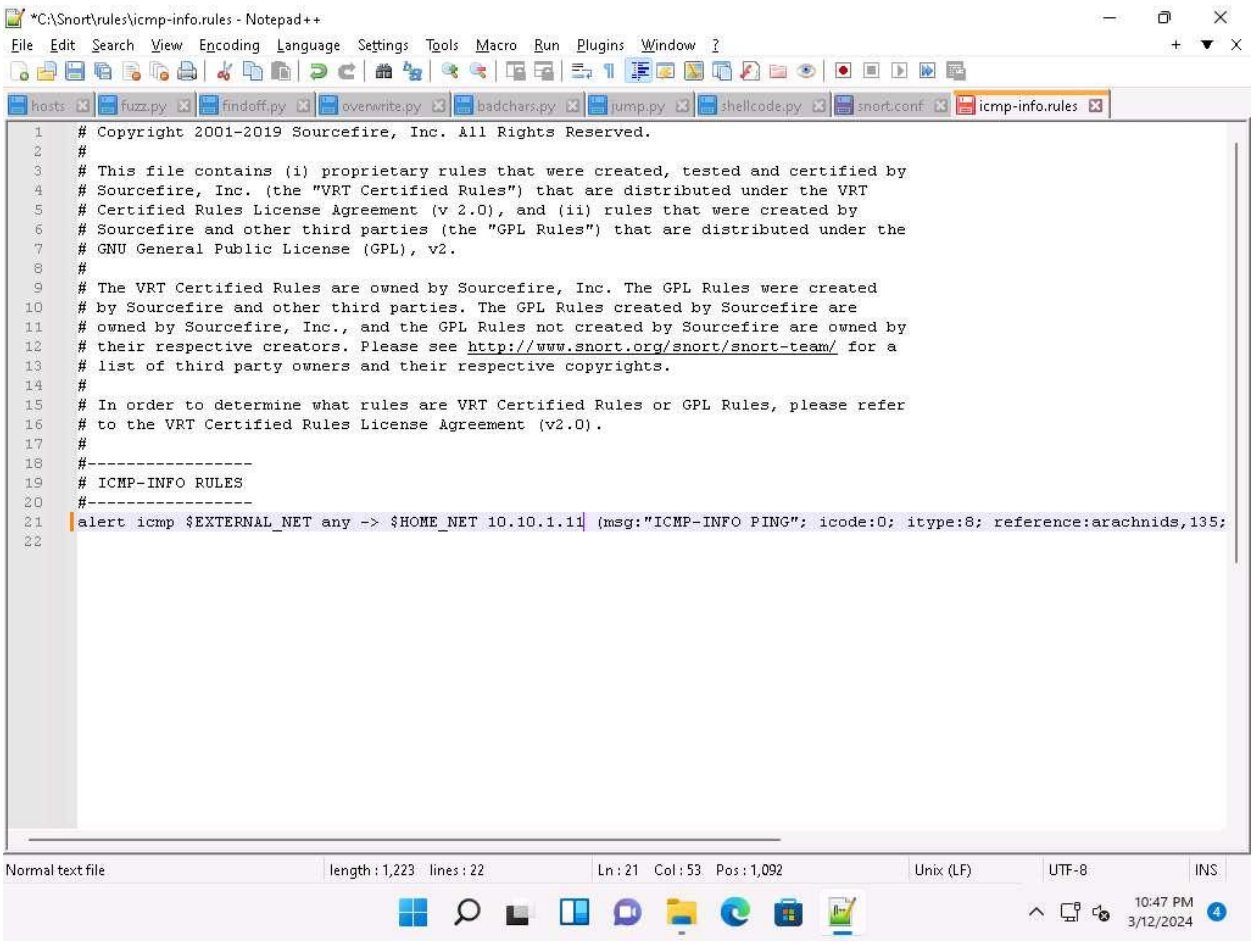
47. By default, the string is **ipvar**, which is not recognized by Snort: replace with the **var** string, and then **close** the window.

Snort now supports multiple configurations based on VLAN Id or IP subnet within a single instance of Snort. This allows administrators to specify multiple snort configuration files and bind each configuration to one or more VLANs or subnets rather than running one Snort for each configuration required.

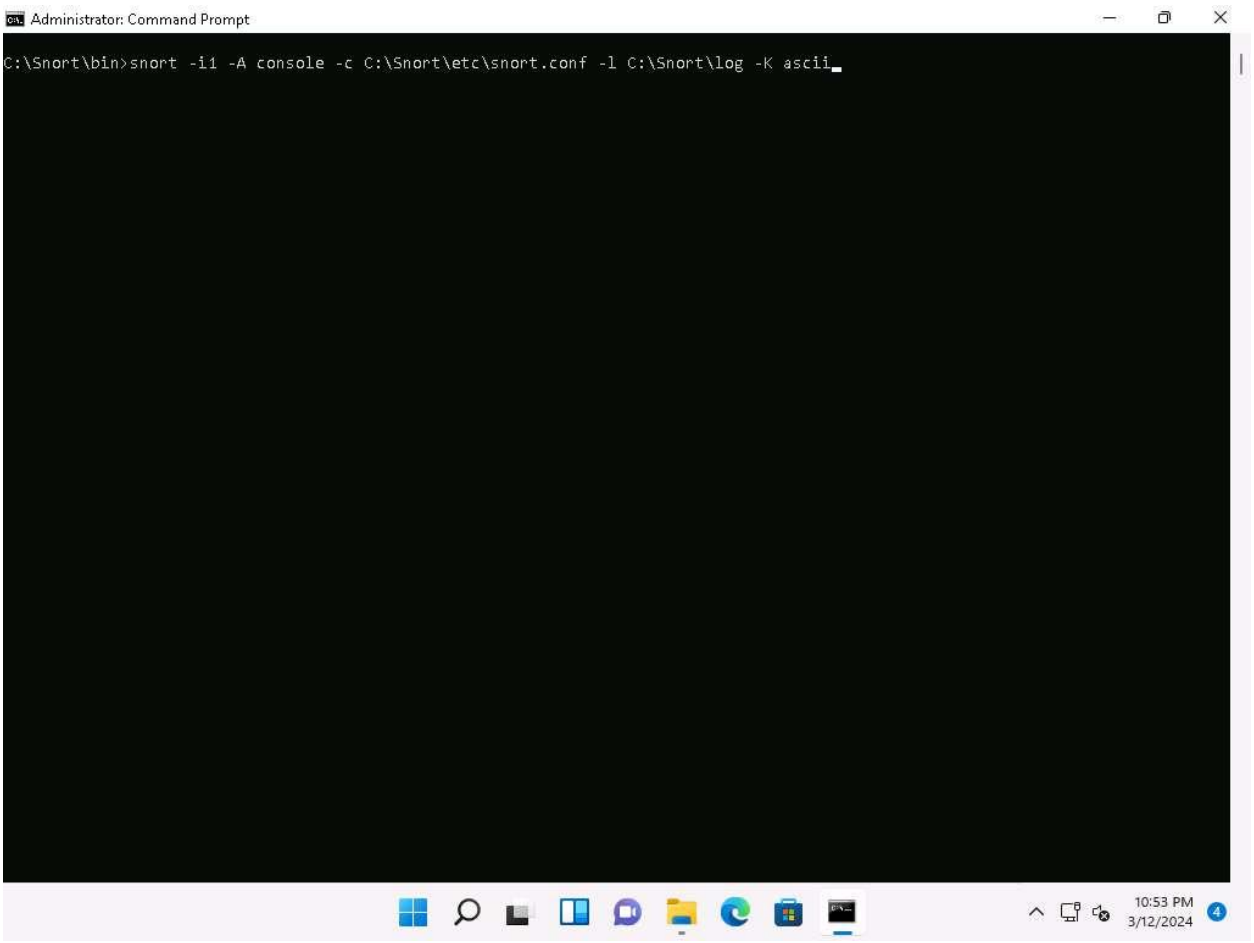


48. Click **Close** to close the **Replace** window.
49. Save the **snort.conf** file by pressing **Ctrl+S** and close Notepad++ window.
50. Before running Snort, you need to enable detection rules in the Snort rules file. For this task, we have enabled the ICMP rule so that Snort can detect any host discovery ping probes directed at the system running Snort.
51. Navigate to **C:\Snort\rules** and open the **icmp-info.rules** file with **Notepad++**.
52. In line 21, type **alert icmp $EXTERNAL_NET any -> $HOME_NET 10.10.1.11 (msg:"ICMP-INFO PING"; icode:0; itype:8; reference:arachnids,135; reference:cve,1999-0265; classtype:bad-unknown; sid:472; rev:7;)** and save. Close the **Notepad++** window.

The IP address (10.10.1.11) mentioned in $HOME_NET may vary when you perform this task.



53. Now right-click on the **Windows Start** icon and click **Run** from the menu.
54. In the **Run** window, type **cmd** in the **Open** field and press **Enter**: This will launch a command prompt window.
55. In the command prompt window, type **cd C:\Snort\bin** and press **Enter**.
56. Run command **snort -iX -A console -c C:\Snort\etc\snort.conf -l C:\Snort\log -K ascii** to start Snort (replace **X** with your device index number; in this task: **X** is 1).

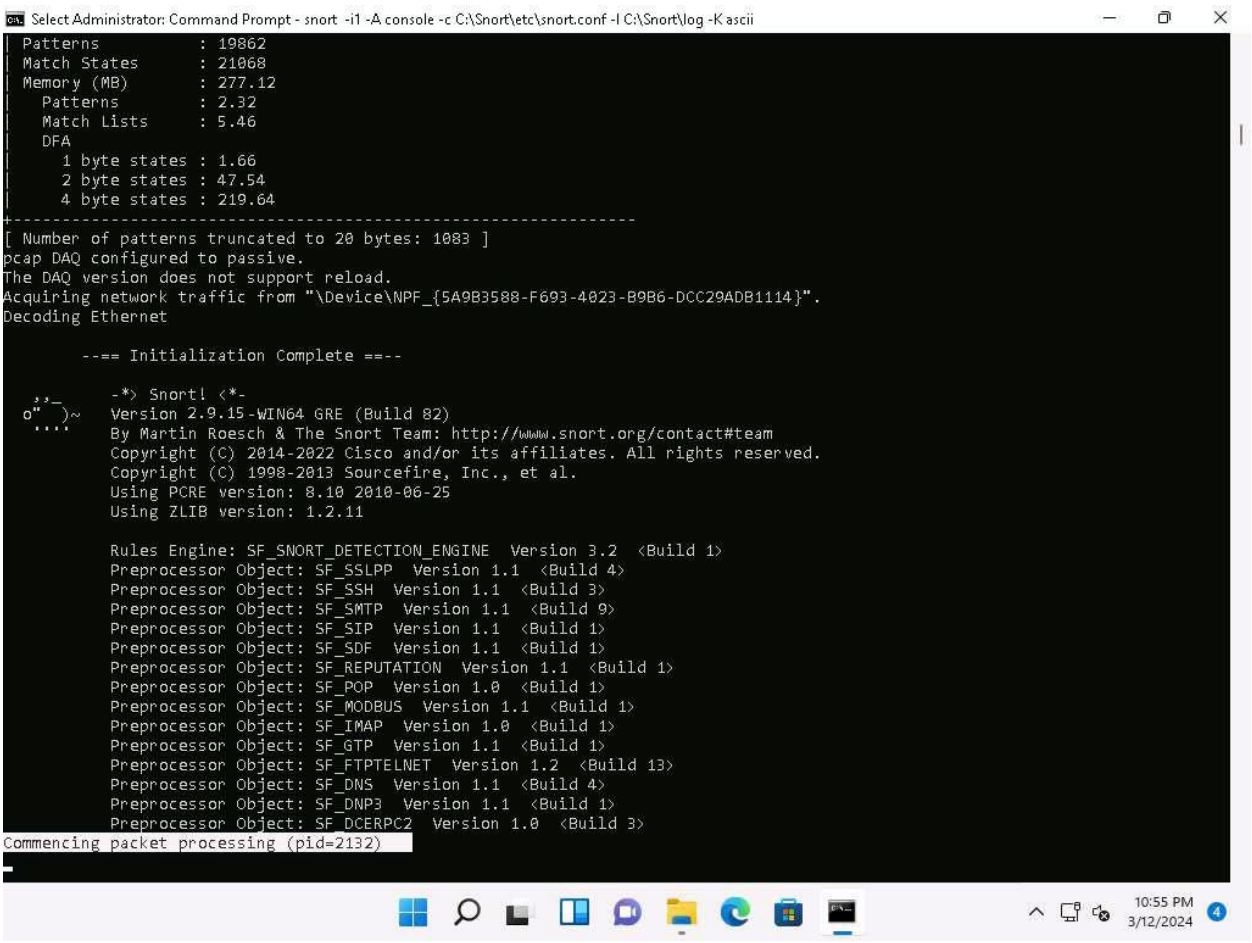


57. If you receive a **fatal error**, you should first **verify** that you have typed all modifications correctly into the **snort.conf** file, and then search through the file for **entries** matching your fatal error message.

58. If you receive an error stating "**Could not create the registry key**," then run the command prompt as **Administrator**.

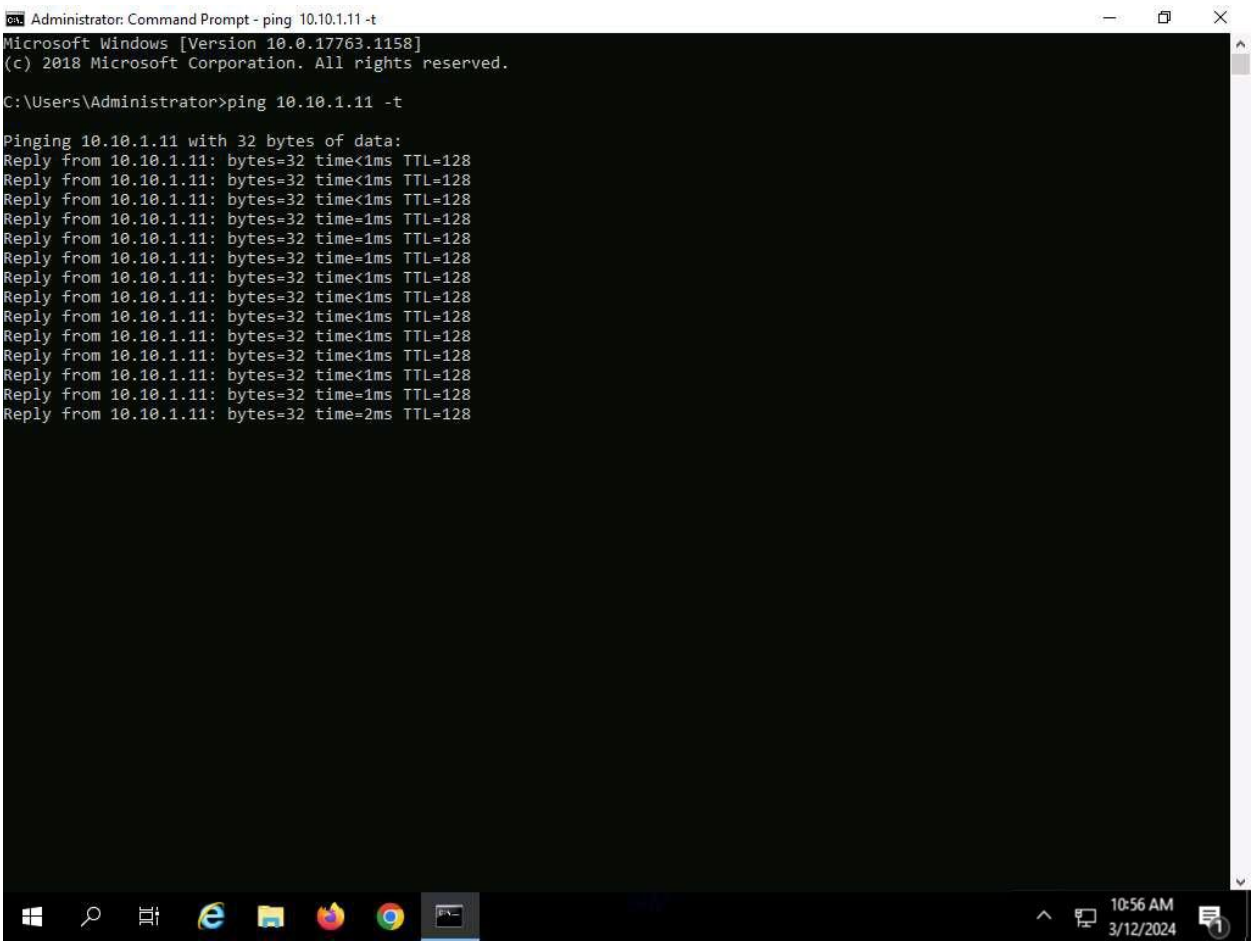
59. Snort starts running in IDS mode. It first initializes output plug-ins, preprocessors, plug-ins, loads dynamic preprocessors libraries, rule chains of Snort, and then logs all signatures.

60. If you have entered all command information correctly, you receive a comment stating **Commencing packet processing (pid=xxxx)** (the value of xxxx may be any number; in this task, it is 2132), as shown in the screenshot.

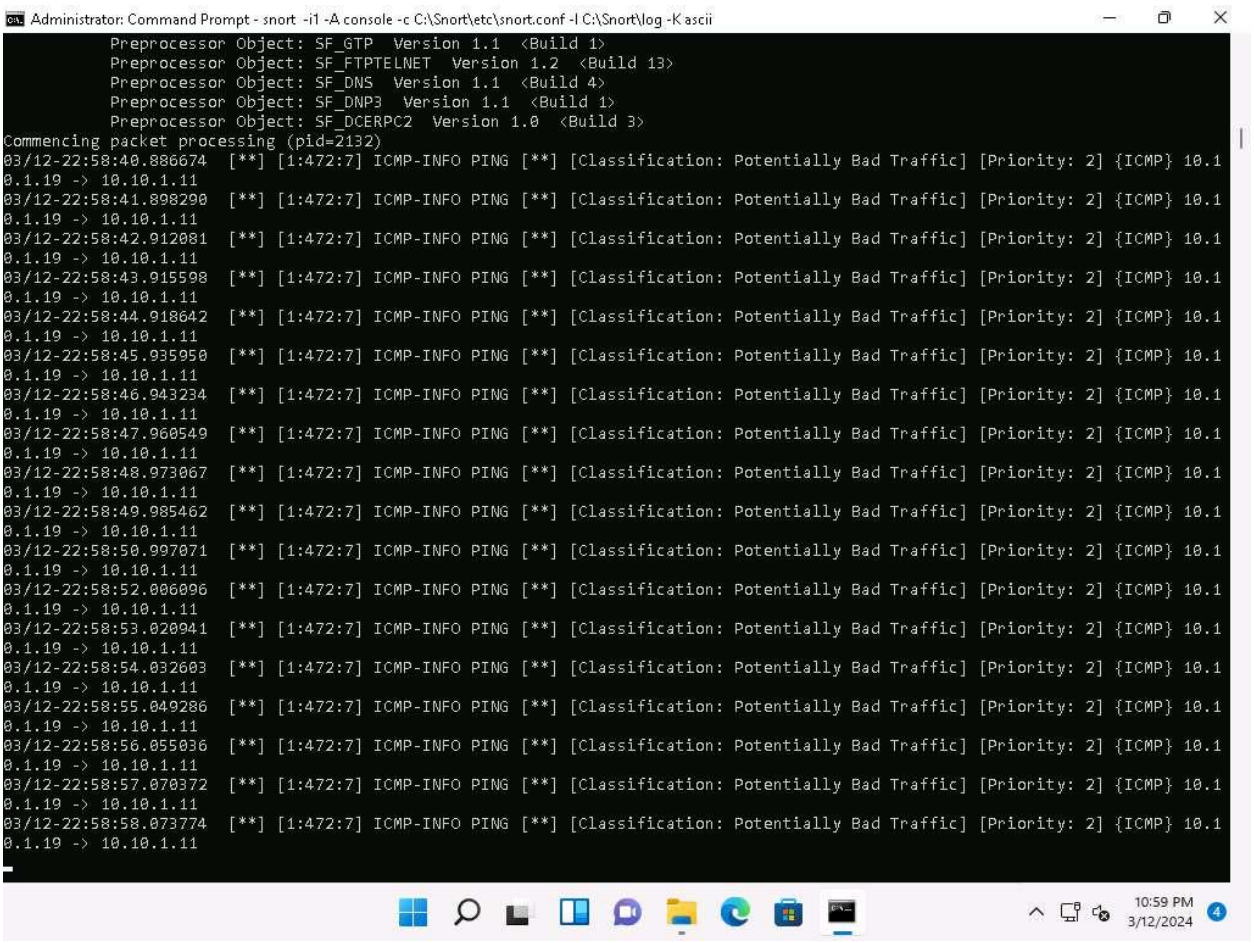


61. After initializing interface and logged signatures, Snort starts and waits for an attack and triggers alerts when attacks occur on the machine.
62. Leave the Snort command prompt running.
63. Attack your own machine, and check whether Snort detects it or not.
64. Now, click on Windows Server 2019 to switch to the **Windows Server 2019** machine (**Attacker Machine**). Click Ctrl+Alt+Delete to activate the machine and login with **Administrator**/**Pa$$w0rd**.
65. Open the command prompt and issue the command **ping 10.10.1.11 -t** from the **Attacker Machine**

10.10.1.11 is the IP address of the Windows11. This IP address may differ when you perform the task.



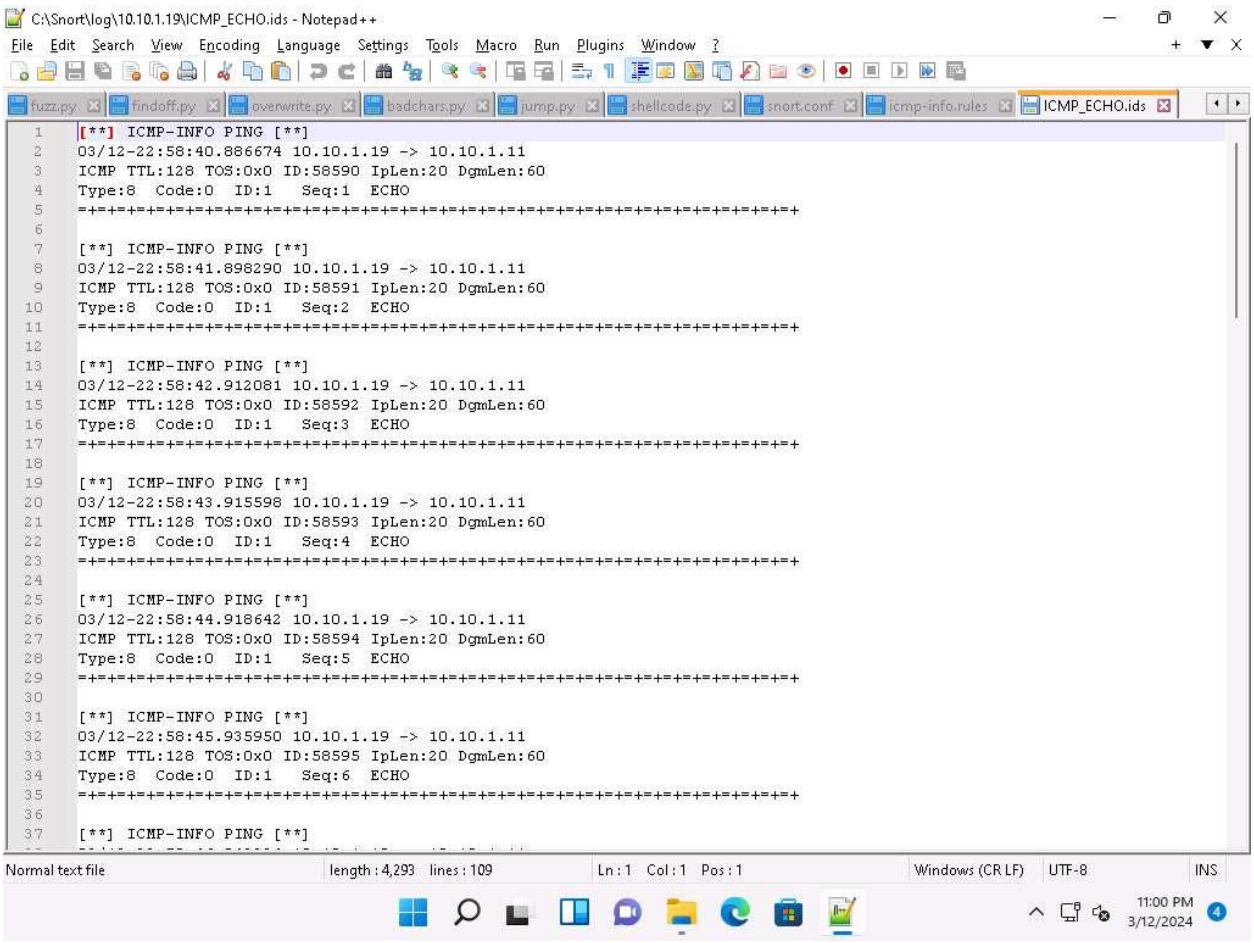
66. Click Windows 11 to return to the **Windows 11** machine. Observe that Snort triggers an alarm, as shown in the screenshot:



67. Press **Ctrl+C** to **stop** Snort; snort exits.

68. Go to the **C:\Snort\log\10.10.1.19** folder and open the **ICMP_ECHO.ids** file with **Notepad++**. You see that all the log entries are saved in the **ICMP_ECHO.ids** file.

The folder name **10.10.1.19** might vary when you perform the task, depending on the IP address of the **Windows 11** machine.



This means that whenever an attacker attempts to connect or communicate with the machine, Snort immediately triggers an alarm

This will make you aware of the intrusion and can thus take certain security measures to disconnect the lines of communication with the attacker's machine.

69. Close all open windows in the **Windows 11** and **Windows Server 2019** machines.

Question 12.1.1.1

Install Snort in the Windows Server 2019 machine. The necessary files are available at Z:\CEHv13 Module 12 Evading IDS, Firewalls, and Honeypots\Intrusion Detection Tools\Snort. Configure and initialize the Snort tool. Initialize the Snort interfaces and attack a target machine from the attacker machine (10.10.1.11) to check whether Snort detects it or not. Enter the Snort command to view the index number of the Ethernet driver.

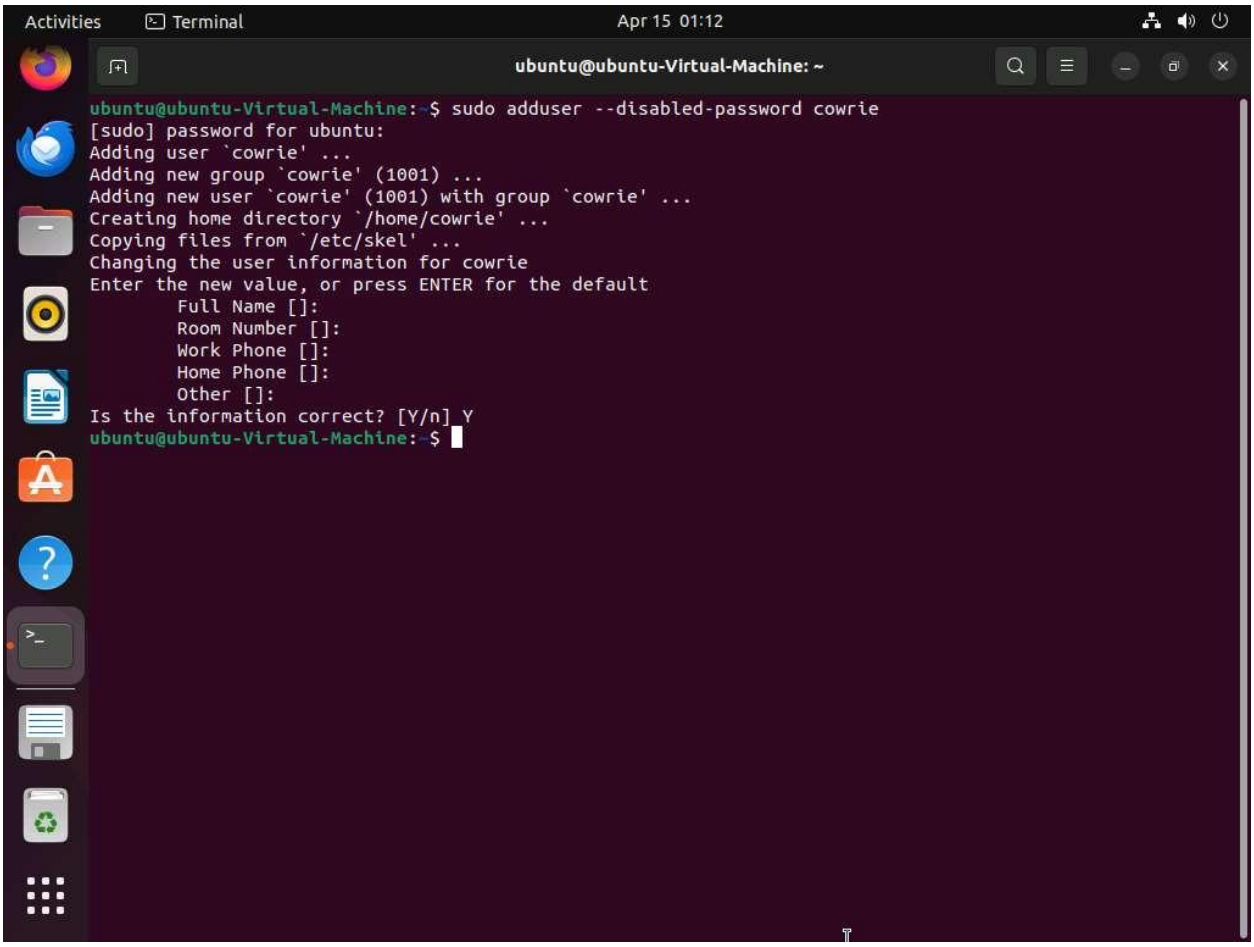
Task 2: Deploy Cowrie Honeypot to Detect Malicious Network Traffic

Cowrie serves as an SSH and Telnet honeypot, capable of capturing brute-force attacks and the actions taken by attackers within the shell. When operating in medium interaction mode, it replicates UNIX behavior using Python. In high interaction mode, acting as a proxy, it allows observation of attacker actions on another system via SSH and Telnet.

Here, we will use Cowrie honeypot to capture incoming malicious traffic from the attacker's machine (here, Parrot Security).

1. Click Ubuntu to switch to the **Ubuntu** machine and login with **Ubuntu**/**toor**.
2. In the **Ubuntu** machine, we will deploy **Cowrie** honeypot.
3. Open a **Terminal** window and execute **sudo adduser --disabled-password cowrie** to create a new user named **cowrie** without password (When prompted, enter the password **toor**, The password that you type will not be visible). Close the terminal.

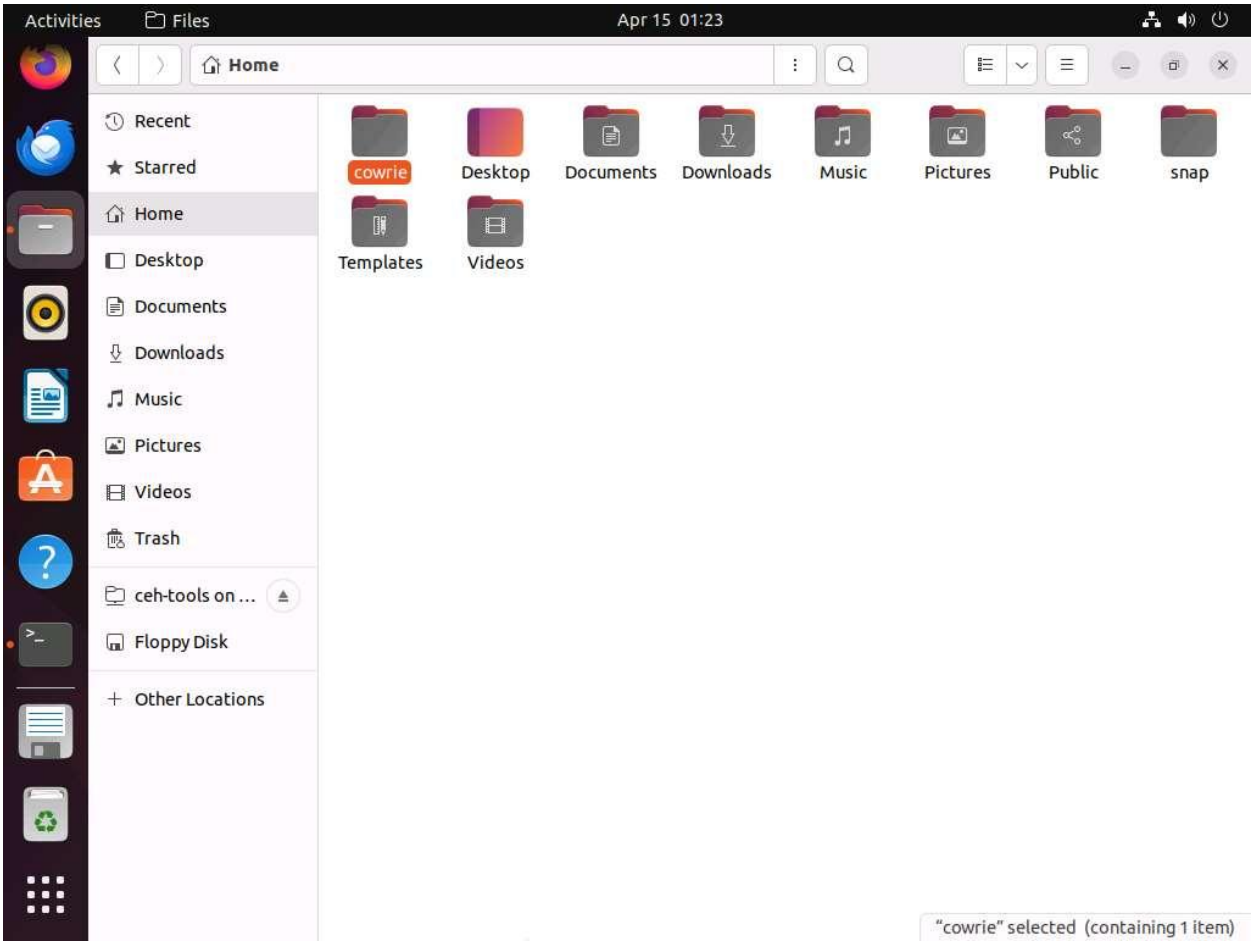
Leave Full Name, Room Number, Work Phone, Home Phone, Other blank and type **Y** when prompted **Is the information correct? [Y/n]**.



4. Click on **Files** icon and navigate to **ceh tools on 10.10.1.11/CEHv13 Module 12 Evading IDS, Firewalls, and Honeypots/Honeypot Tools** and copy **cowrie** folder and paste it into **/home/ubuntu**.

If **ceh-tools on 10.10.1.11** option is not present then follow the below steps to access **CEH-Tools** folder:

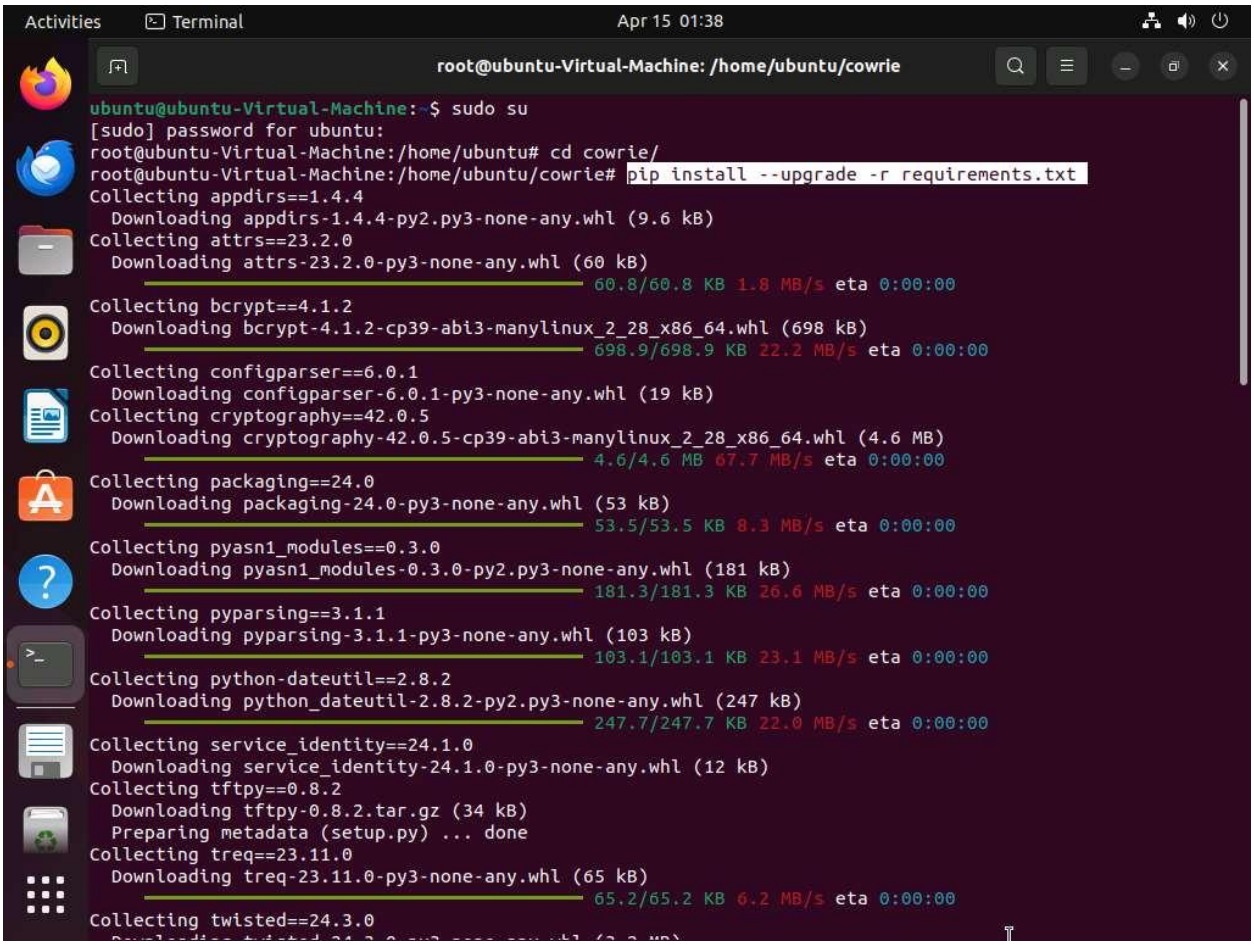
- Open **Files** and navigate to the **+ Other Locations** from the left pane
- In the **Connect to Server** field, type **smb://10.10.1.11** and press **Enter** to access **Windows 11** shared folders.
- The security pop-up appears; enter the **Windows 11** machine credentials (**Admin**/**Pa$$w0rd**) and click **Connect**.
- The **Windows shares on 10.10.1.11** window appears; double-click the **CEH-Tools** folder.



5. Open a new terminal and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**). Now, jump into Cowrie directory using **cd cowrie** command.

The password that you type will not be visible.

6. Run **pip install --upgrade -r requirements.txt** to install all the required dependencies.

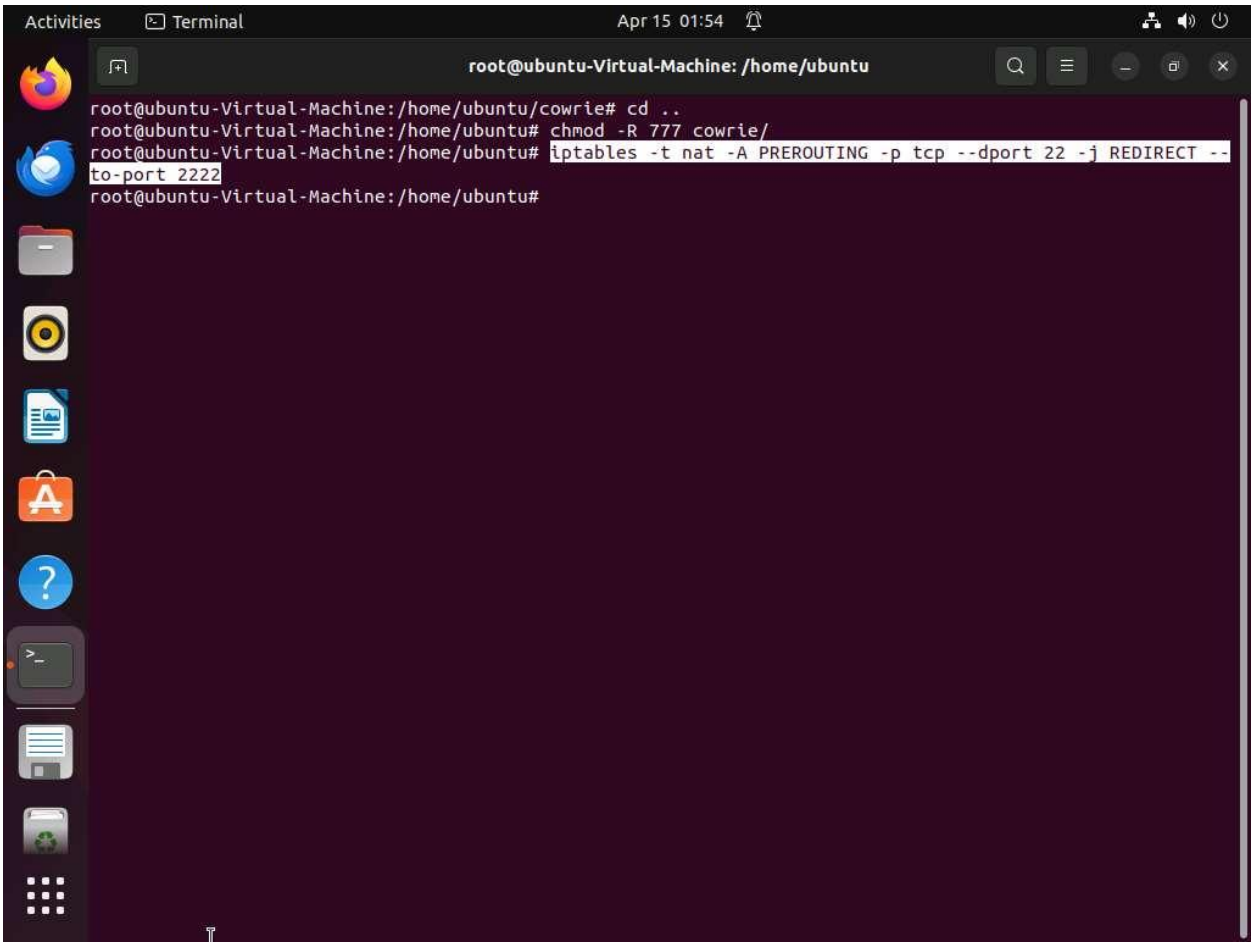


7. Run **cd ..** command to jump back to **/home/ubuntu** and run **chmod -R 777 cowrie** to modify the file permissions of cowrie folder.
8. Run **iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 2222** to redirect the traffic coming on port 22 to port 2222. This redirection channels incoming traffic to a Cowrie honeypot, ensuring the protection of the primary system by segregating potential risks.

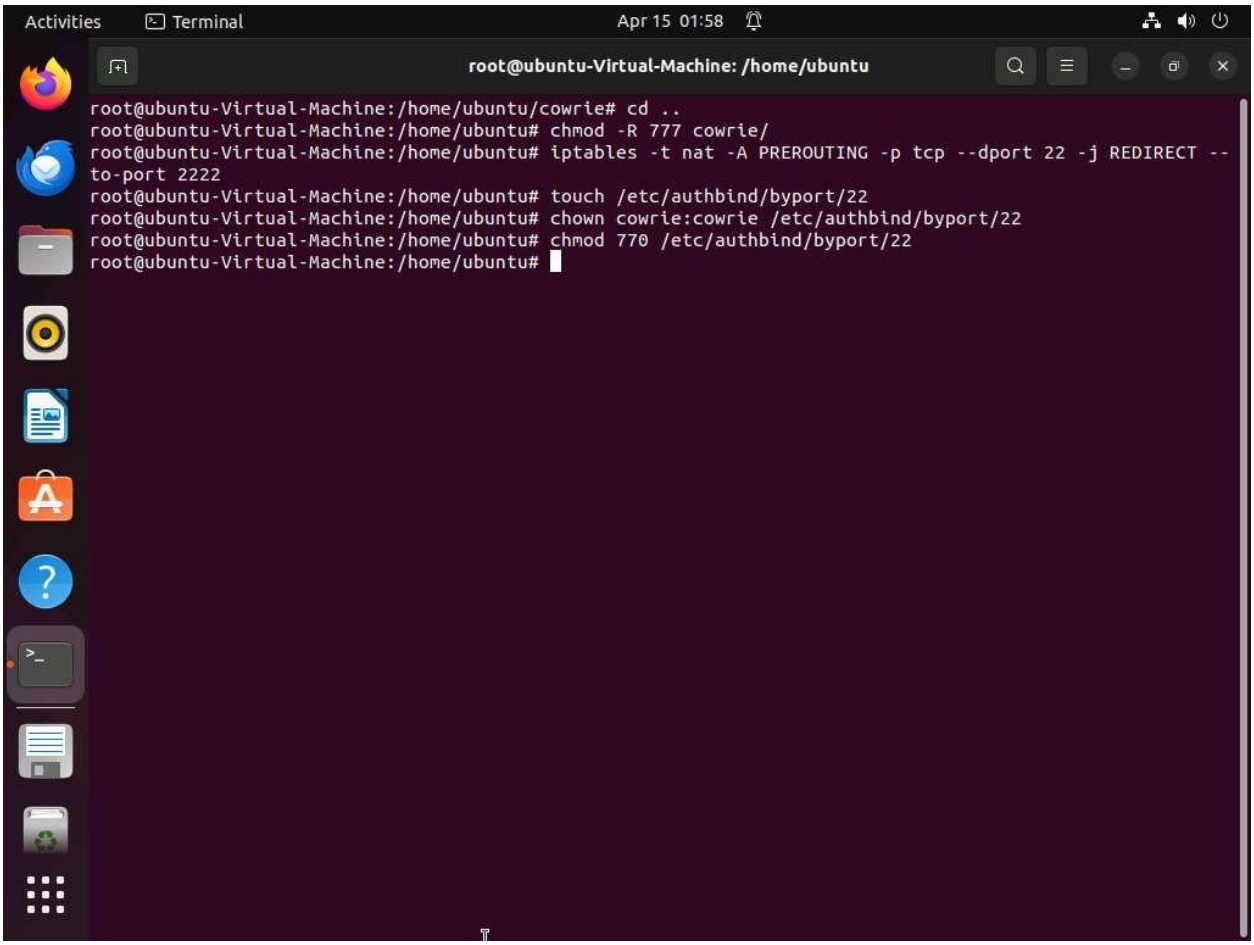
Here,

- **-t nat** specifies the table in which the rule should be added. Here, it is the network address translation (NAT) table.
- **-A PREROUTING** specifies that the rule should be appended to the PREROUTING chain. The PREROUTING chain is traversed by packets as soon as they come in, before any routing decisions are made.
- **-p tcp** specifies the protocol to which the rule should apply. Here, it is TCP.
- **--dport 22** specifies the destination port. Here, it is port 22, which is commonly used for SSH (Secure Shell) connections.

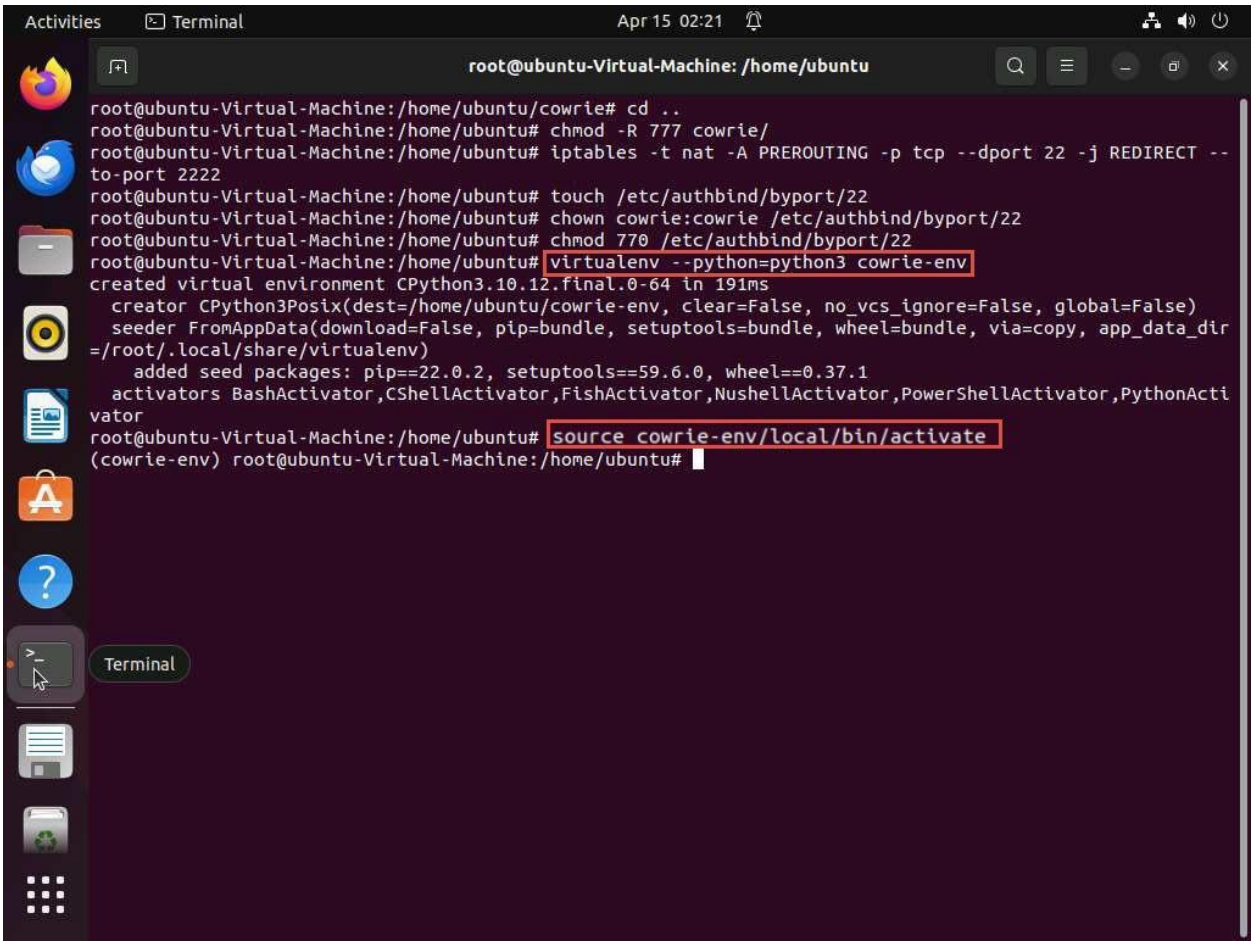
- **-j REDIRECT** specifies the target of the rule. It instructs iptables to redirect the packet to another destination instead of its original destination. Here, the destination will be redirected.
- **--to-port 2222** specifies the port to which the packet should be redirected. Here, it is port 2222. So, any incoming TCP packets to port 22 will be redirected to port 2222.



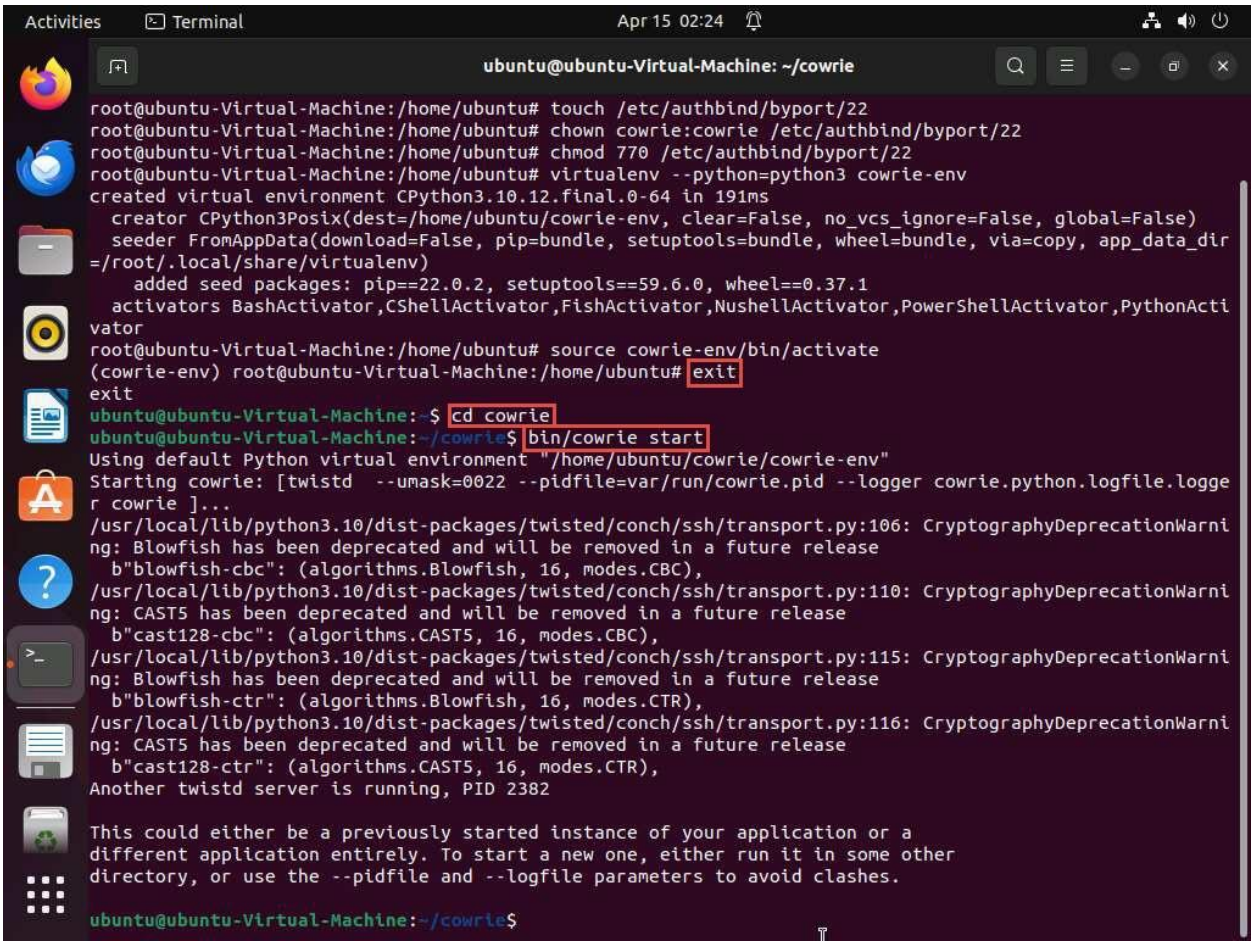
9. To facilitate enhanced security measures, run the following commands to configure authbind, enabling the Cowrie honeypot to operate on port 22 without requiring root privileges:
    - Run **touch /etc/authbind/byport/22** to create an authbind configuration file for port 22, essential for granting permission to the Cowrie honeypot to listen on that specific port.
    - Run **chown cowrie:cowrie /etc/authbind/byport/22** to change the ownership of the authbind configuration file for port 22 to the user and group "cowrie," ensuring proper access permissions for the Cowrie honeypot to operate on that port.
    - Run **chmod 770 /etc/authbind/byport/22** to set appropriate permissions on the authbind configuration file for port 22, ensuring that the Cowrie honeypot has the necessary access to operate on the port without requiring root privileges.



10. Run **virtualenv --python=python3 cowrie-env** to create a virtual environment. Run **source cowrie-env/local/bin/activate** to activate the environment.

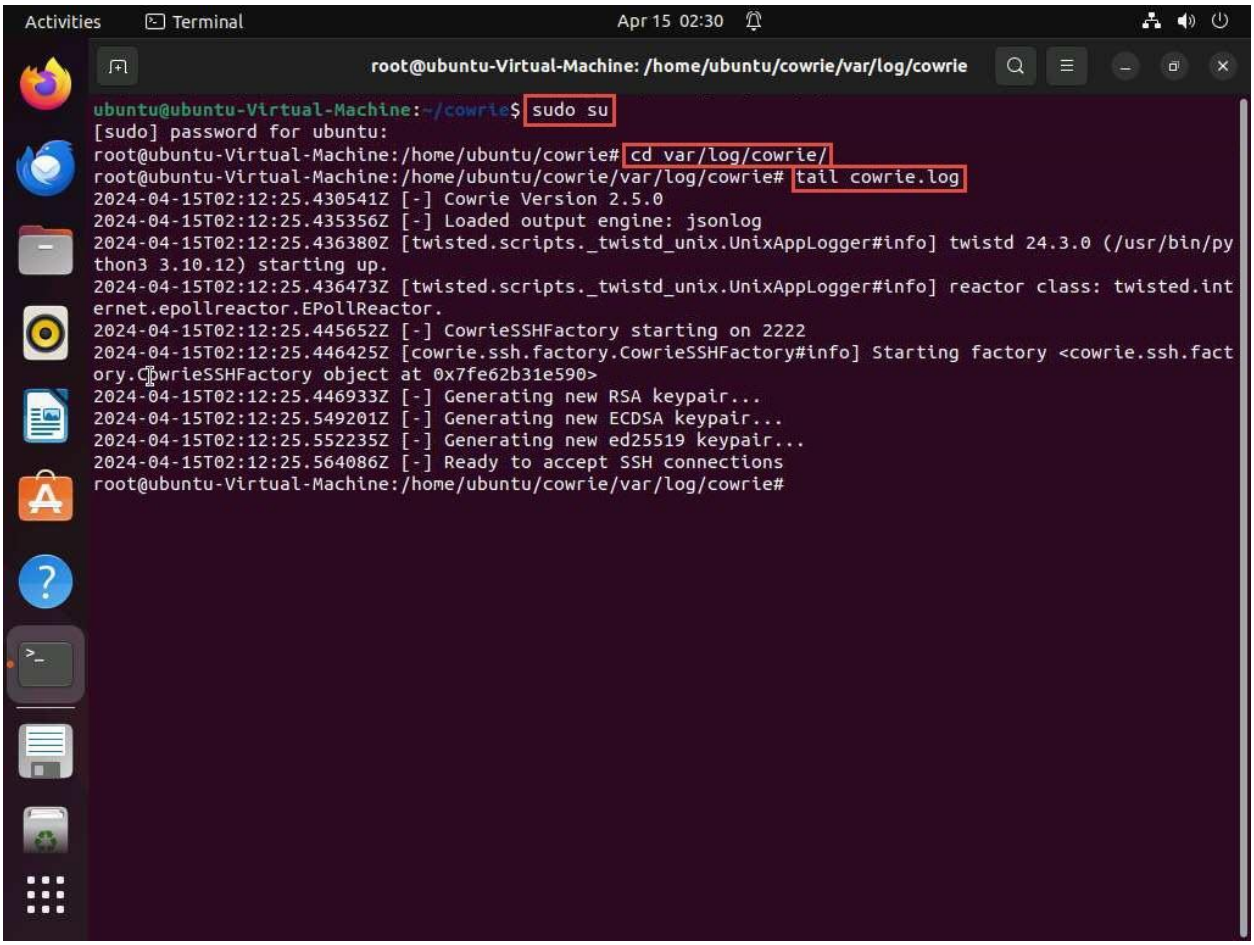


11. Exit the root privileged terminal using **exit** command. Navigate to cowrie directory using **cd cowrie** command. Here, run **bin/cowrie start** to initiate the Cowrie honeypot and begin monitoring and logging incoming traffic for security analysis and threat detection purposes.



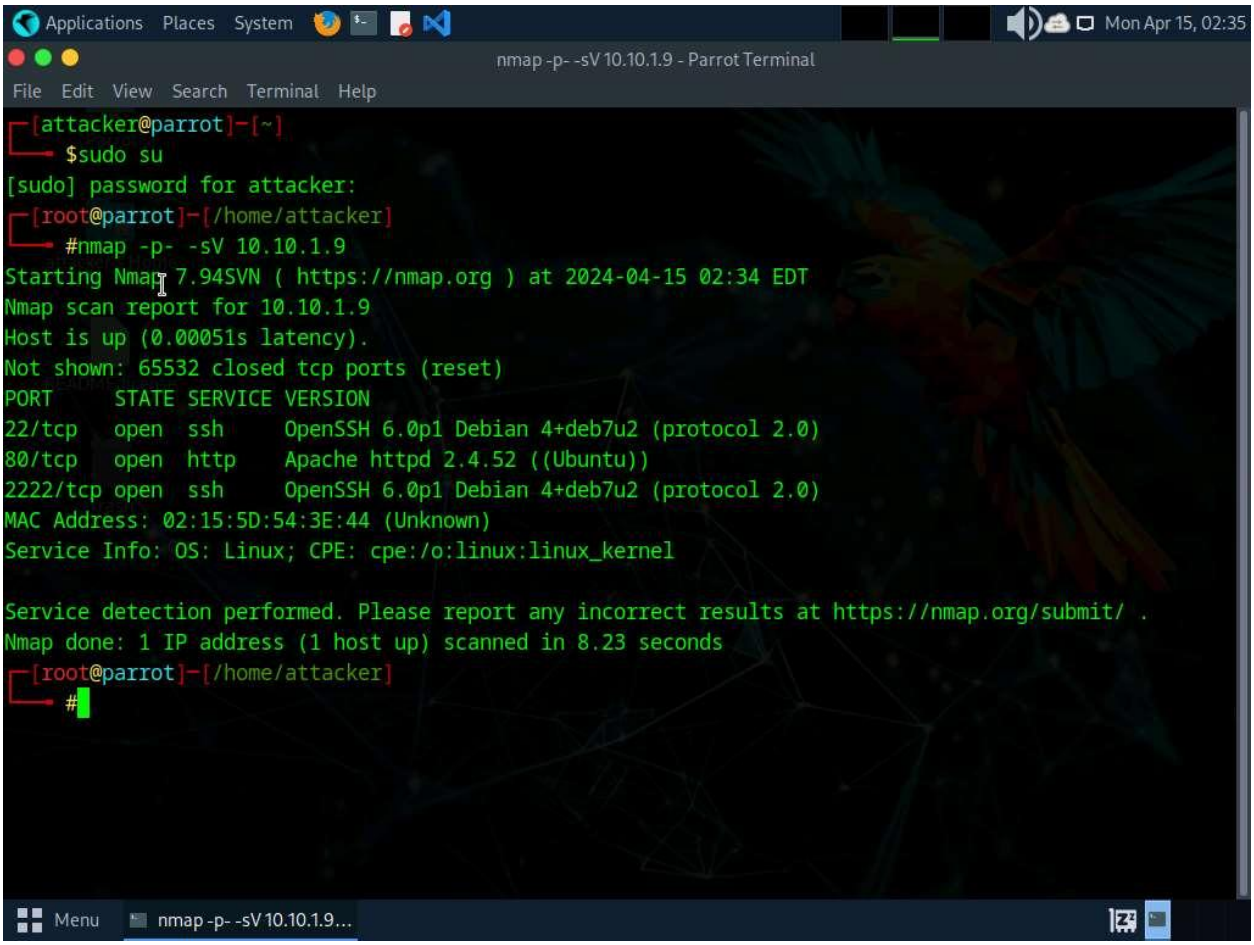
12. In the terminal, acquire root privileges using **sudo su** command and execute **cd var/log/cowrie/** to navigate to **var/log/cowrie**. Now, run **tail cowrie.log** to view the log file.

Here, **Ready to accept SSH connections** confirms that the honeypot is successfully setup.

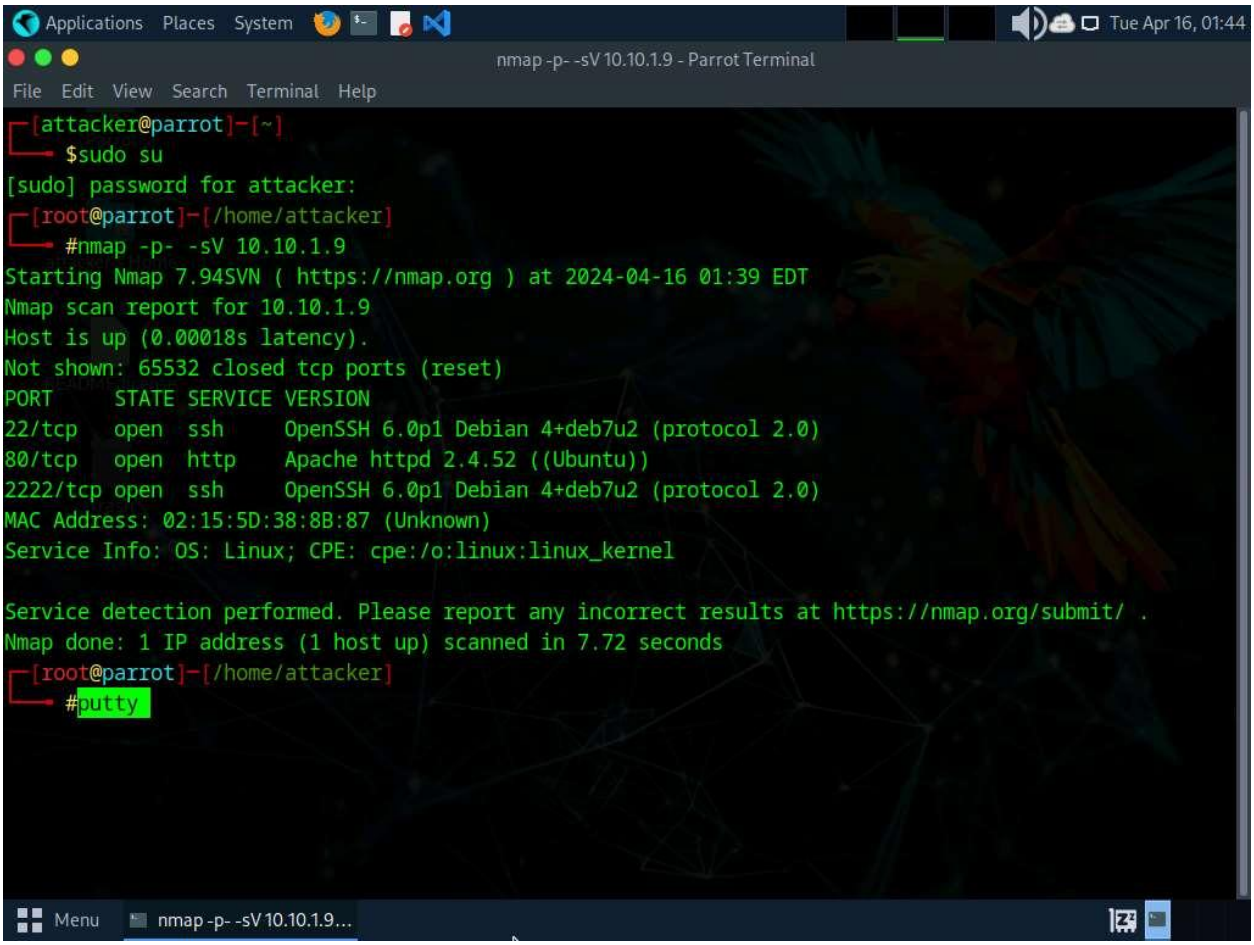


13. Now, we will use **Parrot Security** machine as an attacker to attack the honeypot running SSH service. To do so, firstly we will scan the target machine for open SSH port.
14. Click Parrot Security to switch to **Parrot Security** machine and use **toor** as password. Open a **Terminal** window and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**).
15. In the terminal, run **nmap -p- -sV 10.10.1.9** command to discover open ports and services.
16. You can observe that the port 22 is open and is the default port used for SSH connection. Since the port is open we will try to connect to the machine using SSH.

Here, you can also observe the SSH service running on port 2222 that is a honeypot that we have deployed. Attackers will target the SSH service running on default SSH port (22).

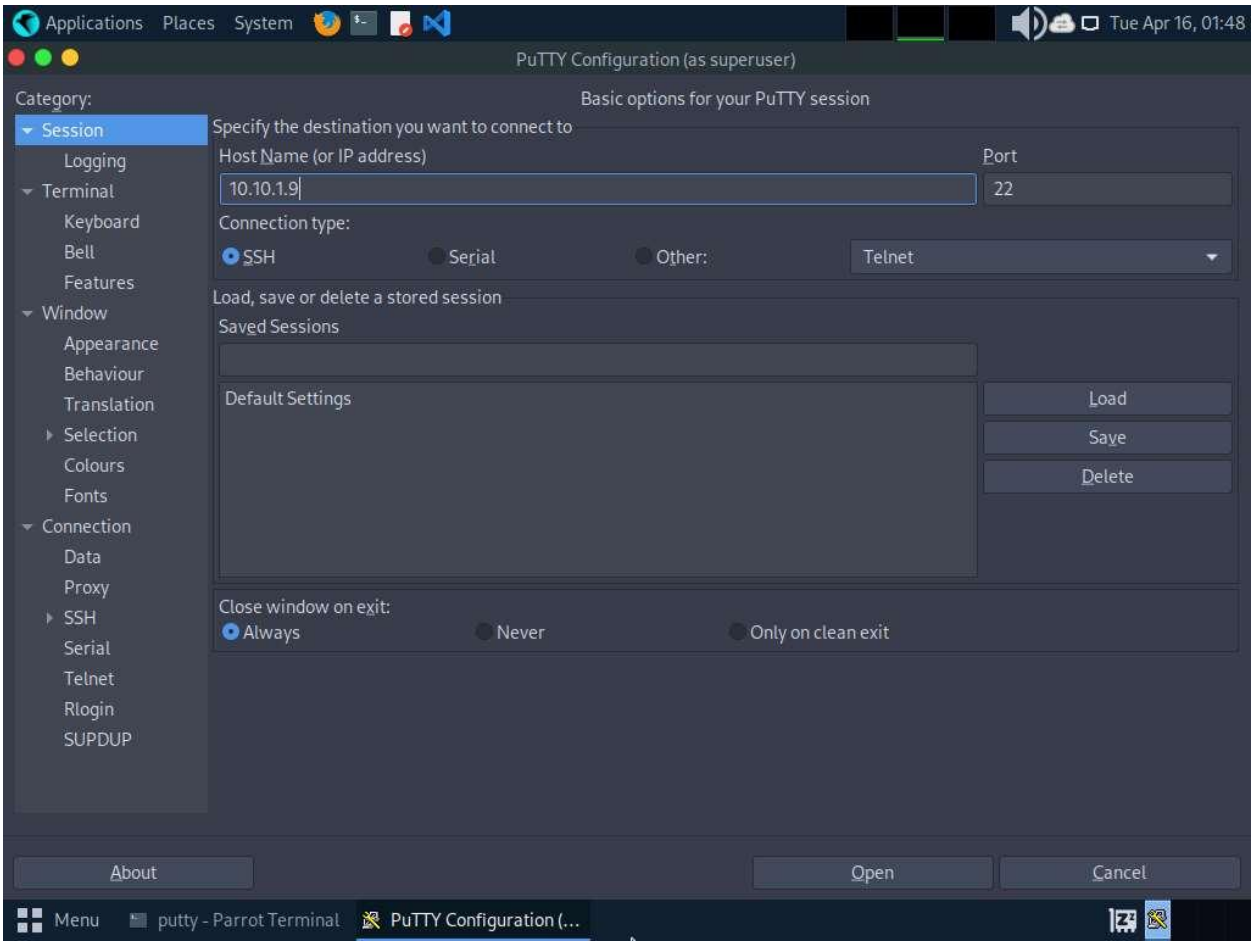


17. To establish a connection to the target machine via SSH, we will use PuTTY. Initiate the PuTTY interface by executing **putty** command.

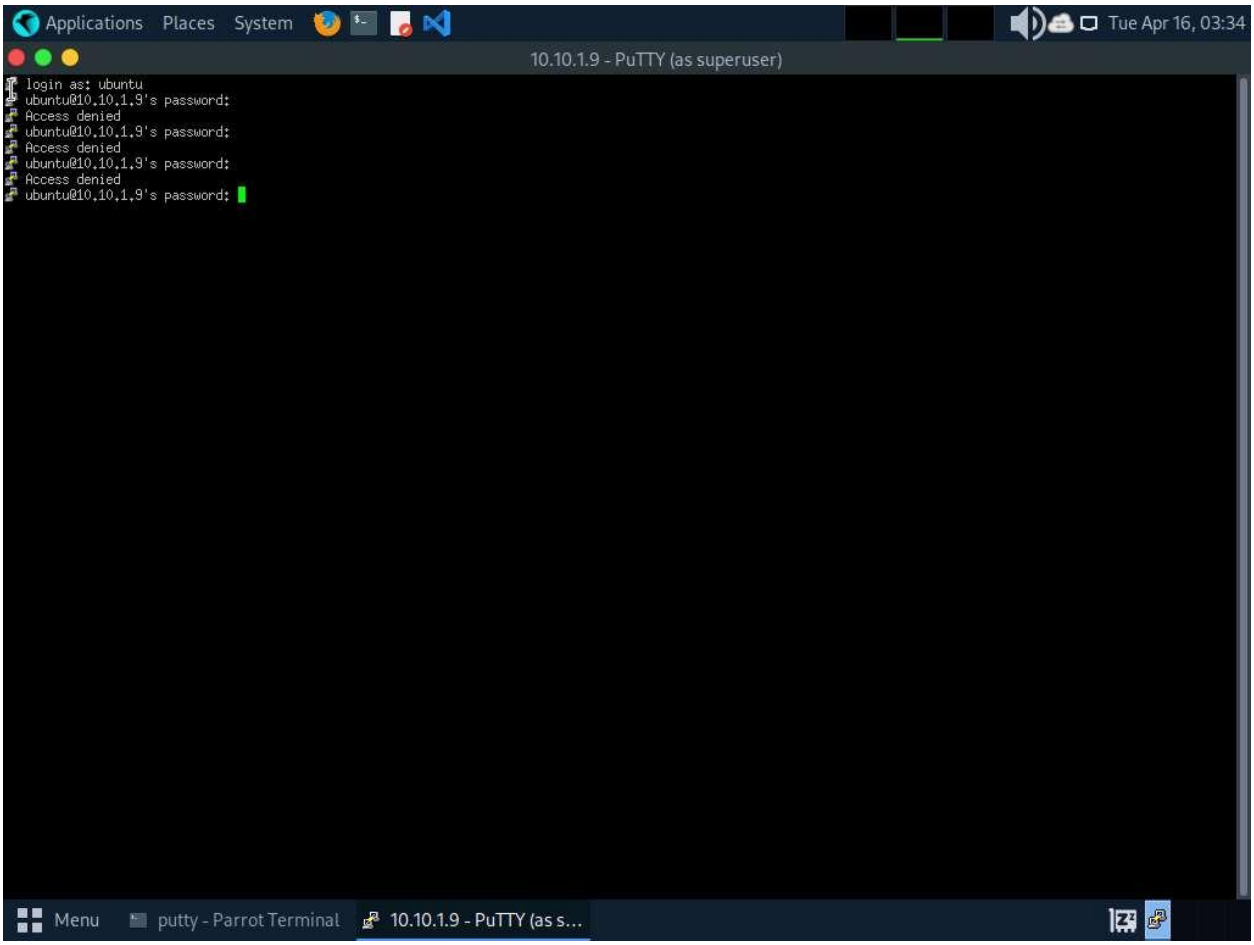


18. The **PuTTY Configuration (as superuser)** window appears. Enter the IP address of the target machine (here, **10.10.1.9**) in the **Host Name (or IP address)** field, and proceed by clicking on **Open** button.

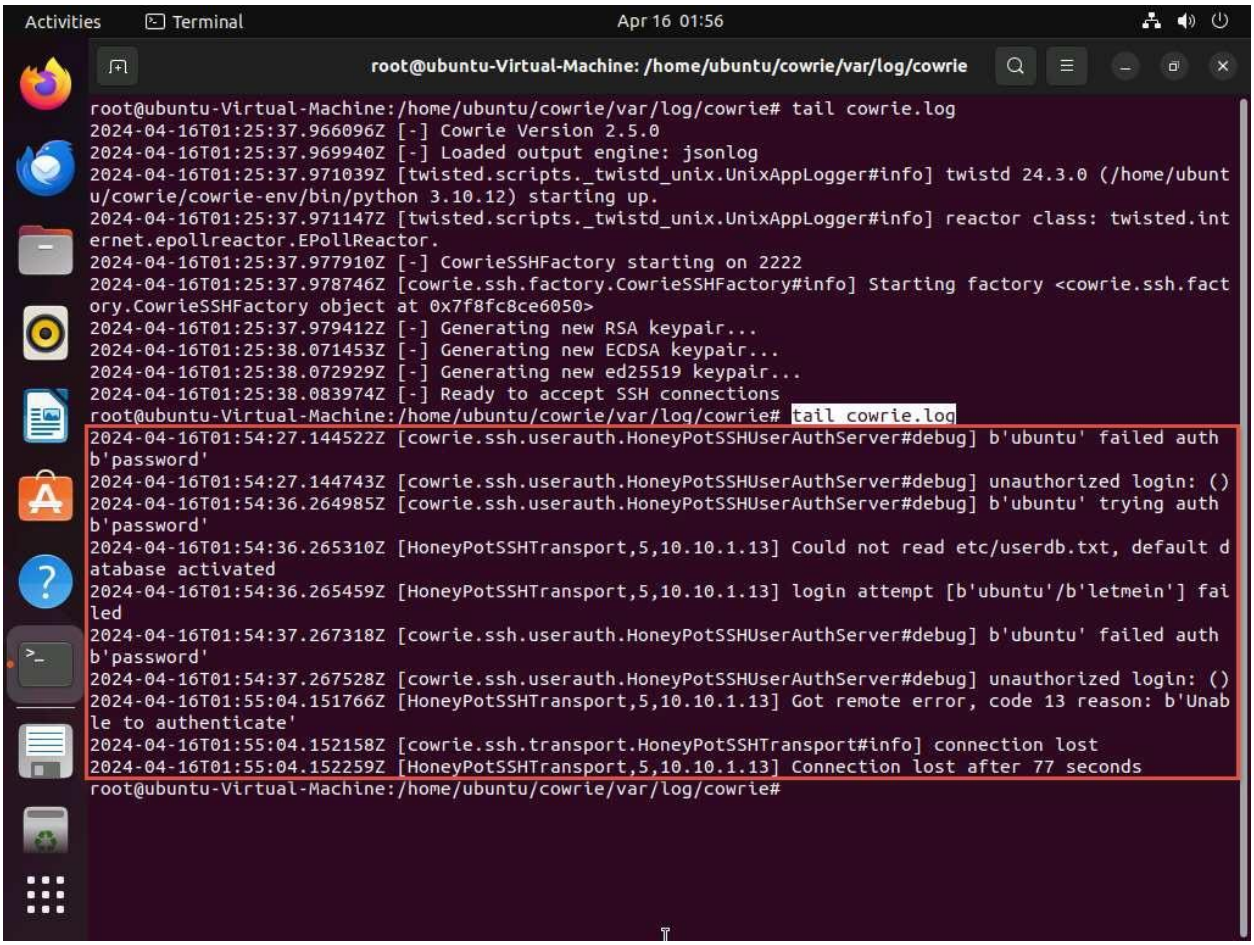
PuTTY Security Alert (as superuser) window appears, click **Accept**.



19. In **10.10.1.9 - PuTTY (as superuser)** window, type **ubuntu** in **login as** field and password as **toor**, it responds with **Access denied**. Try a random set of passwords to bruteforce into the target machine.



20. Click Ubuntu to switch back to **Ubuntu** machine, and run **tail cowrie.log**. Here, observe the logs generated when the attacker tried to connect to the SSH service on the **Ubuntu** machine using PuTTY. These log entries serves as evidence of the attacker's unauthorized access attempts.

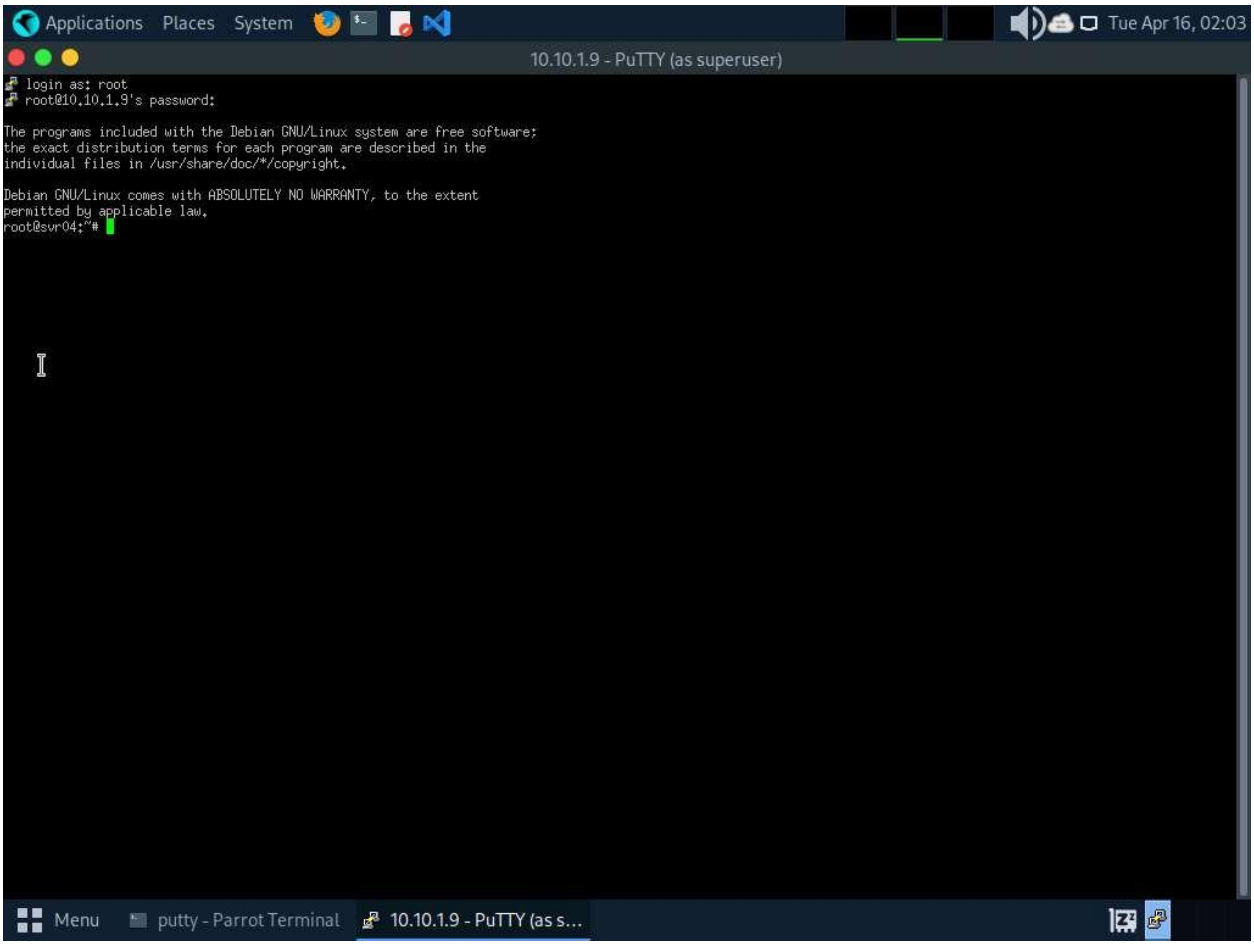


21. Click Parrot Security to switch back to **Parrot Security** machine. In the terminal window, open the PuTTY interface by executing **putty** command.

22. In the **PuTTY Configuration (as superuser)** window, type the IP address of the target machine (here, **10.10.1.9**) under **Host Name (or IP address)** section and click on **Open**.

If **PuTTY Security Alert (as superuser)** window appears, click **Accept**.

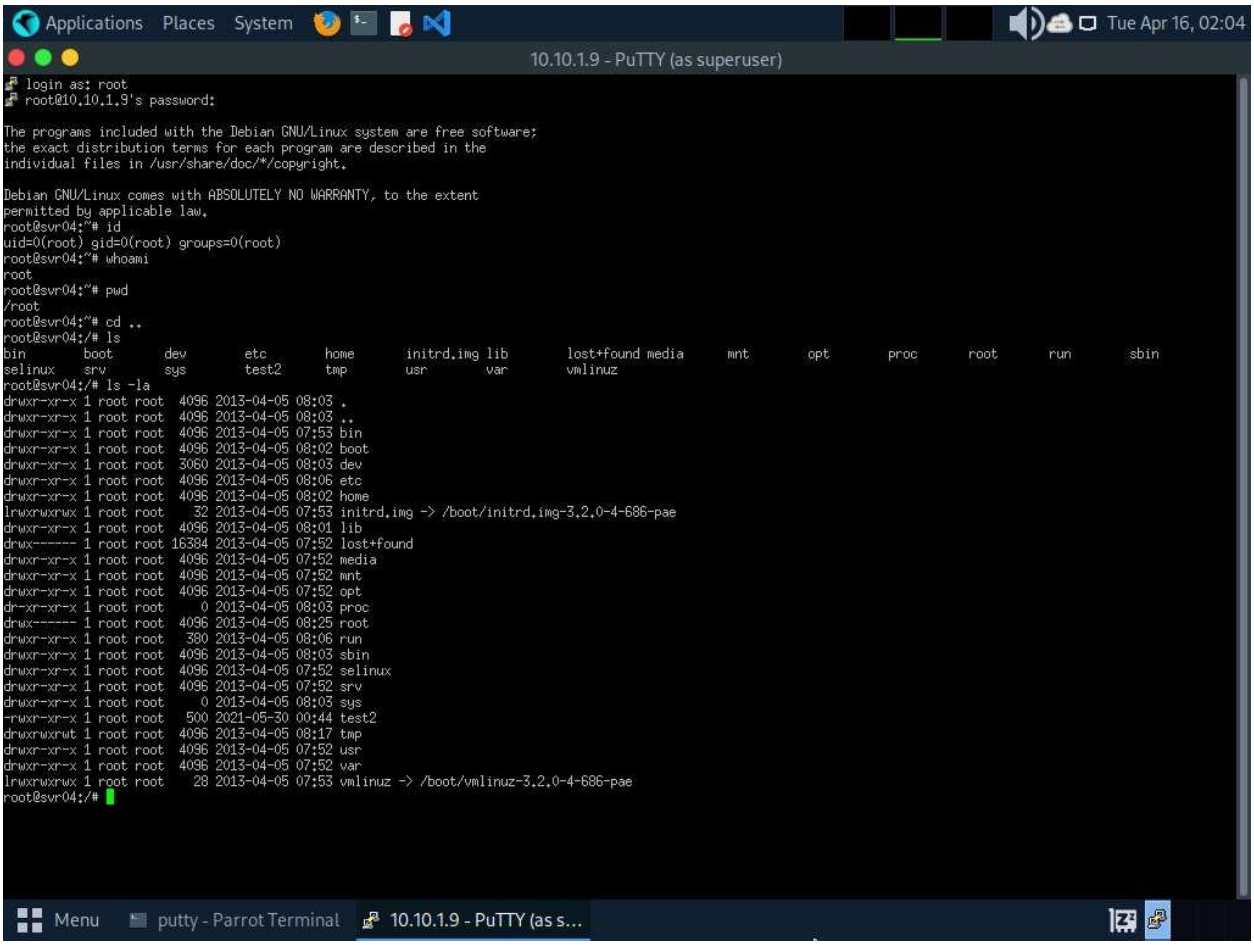
23. In **10.10.1.9 - PuTTY (as superuser)** window, type login as **root** and type a random password of your choice.



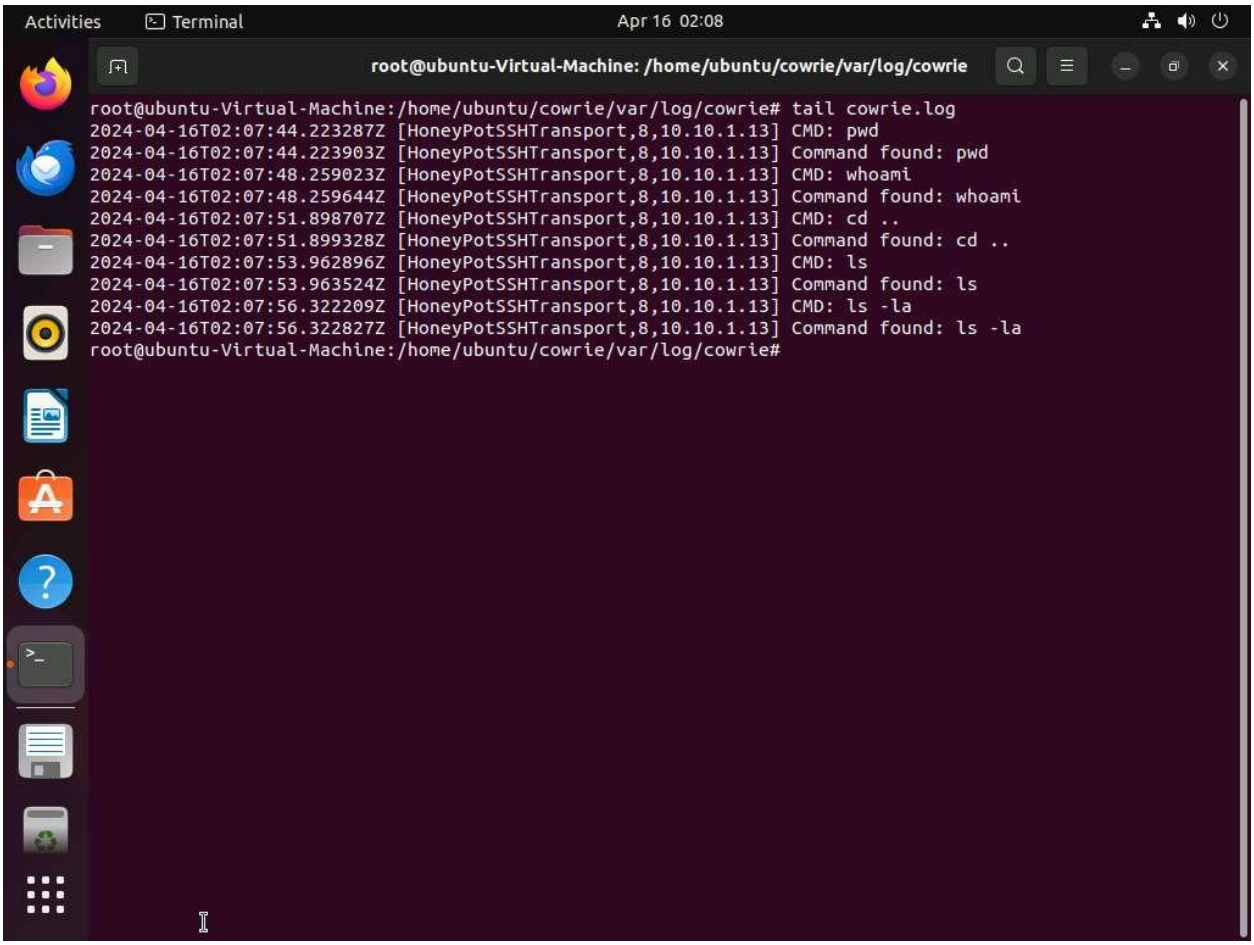
24. You are now in the honeypot, execute various commands to gather intelligence and explore the system.
    - Run **id** command to retrieve information about the current user and group.
    - To identify the username associated with the active session, execute **whoami** command.
    - Run **pwd** to determine the present working directory within the system.
    - Run **cd ..** to navigate to the root directory.
    - Run **ls** to list the files available in the directory.
    - Continue the reconnaissance, run **ls -la** command to list detailed information about files and directories in the current location.

The Cowrie honeypot has default credentials set to **root** / ***** (with root as username and taking any character or word as a correct password). After obtaining access to the SSH service, attacker runs the above commands giving him a sense of compromising the target system successfully. However, all these interactions are captured by the virtual environment which was created while setting up Cowrie honeypot, without affecting actual SSH service running on the target system.

more...



25. Click Ubuntu to switch back to the **Ubuntu** machine and execute **tail cowrie.log** command to view the logs. These logs will capture the activities performed by the attacker within the Cowrie honeypot environment, including the execution of commands.



This will notify you of the intrusion, allowing you to implement specific security measures to sever communication with the attacker's machine.

26. This concludes the demonstration of detecting malicious network traffic using Cowrie.
27. Close all open windows and document all acquired information.

Question 12.1.2.1

In ubuntu machine (10.10.1.9) setup Cowrie honeypot to capture incoming malicious traffic on port 22 from the attacker's machine (10.10.1.13). Enter the port number to which the port 22 traffic is redirected to.