IDS/IPS with Snort and log analysis using Splunk.

Done By: Monarch Giri

Abstract

In the world of the advancement of technology, everyone needs to be safe from threats and the bad guys (black hat hackers). All single devices and networks are connected to the internet, so there should be proper security and monitoring tools in place to keep us safe. There should be proper detection, analysis, and monitoring of our networks, and implementing Snort IDS/IPS and analyzing logs in Splunk is one of the proper ways to tackle these problems.

To demonstrate this project, a virtual environment is used where this simulation can be done efficiently. It uses a virtual box as a VM, and 4 machines are running inside it which are Kali Linux, Ubuntu, Metasploit, and Windows 7.

This report focuses on the introduction of the system and technology, development, and testing of the system. It identifies the problem domain, and the aims and objectives of the project are defined and mentioned here targeting the end users. Proper screenshots are provided with the details about the working of the system and its potential.

Table Of Contents

Table of Figures

1. Introduction

The vast field of network security includes a variety of procedures and methods for defending computer networks, systems, and data against intrusions, attacks, and disruptions. Ensuring the availability, confidentiality, and integrity of data and resources within a network is the aim of network security. Network security is a continuous process that calls for a blend of user awareness, policies, and technical solutions. Maintaining an edge over constantly changing cyber threats requires a comprehensive and proactive strategy. For robust network security, there needs to be proper configuration and maintenance of the security components. Some key aspects and components of network security are as follows:

- Access control
- Firewalls
- IDS/IPS
- VPNs
- Encryption
- Network segmentation
- Incident response and forensic
- Security patching and updates
- Logging and monitoring

These are some key factors that help to build a robust network. This report does not cover all of these components and tools but it mainly focuses on IDS/IPS using Snort and logging and monitoring the logs in Splunk.

1.1 Virtual Box

Oracle Virtual Machine An open-source virtualization program called VirtualBox, or simply VirtualBox, enables users to run several operating systems on a single physical computer. Within a host operating system, it offers a platform for establishing and controlling virtual machines (VMs). Numerous guest operating systems, such as Windows, Linux, macOS, and others, are supported by VirtualBox. Software development, testing, running legacy apps, experimenting with different operating systems, and setting up isolated environments for security testing are just a few of the many uses for VirtualBox that are frequently found. For individuals who need to run multiple operating systems on a single physical machine, it offers an affordable and adaptable solution. This project is also done in Virtual Box.

1.2 Snort

An open-source intrusion detection and prevention system (IDS/IPS) called Snort keeps an eye on network traffic and spots potentially dangerous activity on the networks. Snort is a rule-based language that enables organizations to block or prevent potential attack vectors and identify malicious packets in network traffic. It combines protocol, signature, and anomaly-based inspection methods. In addition, Snort uses behavior-based techniques to identify real vulnerabilities by comparing network activity to a pre-established set of Snort rules. This allows it to identify sophisticated emerging threats that signature-based methods alone might not have been able to identify in the past.

Snort Modes

One of three flags that can be set up for Snort will dictate how it operates:

- Snort reads TCP/IP packets in sniffer mode (-v flag), printing packet information on the console.
- In packet logger mode (-l flag), Snort records incoming TCP/IP packets for later analysis in a disk-based logging directory.
- Network intrusion detection and prevention system (NIDS) mode (-c flag): Based on the rule type entered in the configuration file, Snort decides whether to act on network traffic.

Types of Snort Rules

Instead of evaluating rules according to the order in which they appear in the configuration file, Snort examines them according to the rule type, which indicates what should happen when Snort discovers a packet that satisfies the rule criteria.

- Alert rules: Upon detecting a suspicious packet, Snort sends out an alert.
- Block rules: Snort stops all packets in the network flow after the suspicious packet.
- Drop rules: As soon as an alert is generated, Snort drops the packet.
- Logging rules: As soon as an alert is generated, Snort logs the packet.
- Pass rules: Snort classifies the suspicious packet as passed and ignores it.

1.3 Splunk

Splunk is a software platform that facilitates the search, monitoring, and analysis of data generated by machines. Gathering, organizing, and analyzing vast amounts of data from various sources, it enables businesses to obtain insights and make data-driven decisions. In many IT environments, Splunk is utilized for data analysis, log management, and security information and event management (SIEM).

Important Splunk features include:

- Machine-generated data is gathered and indexed by Splunk from a variety of sources, such as log files, databases, apps, sensors, and more.
- Splunk's robust search and analysis features are its main advantages. With the help of the Splunk Search Processing Language (SPL), users can search and analyze data in real time, finding trends, patterns, and anomalies.
- Splunk offers interactive dashboard and visualization creation tools that simplify the interpretation and presentation of complex data. Custom dashboards can be created by users to track particular metrics and KPIs.
- With Splunk, users can create alerts based on pre-established criteria. Splunk has the ability to send out alerts to administrators or other pertinent stakeholders when specific events or patterns are found in the data.

- A popular Security Information and Event Management (SIEM) tool is Splunk. It aids in the monitoring and analysis of security events, the identification of threats, and the maintenance of industry compliance.

Applications for Splunk can be found in many different fields and use cases, such as business analytics, cybersecurity, DevOps, IT operations, and Internet of Things (IoT) data analysis. It offers a centralized platform for organizing and deriving insightful information from various data sources, enhancing operational effectiveness and facilitating well-informed decision-making.

1.4 IDS and IPS

There are two main modes of operation for Snort: Intrusion Prevention System (IPS) and Intrusion Detection System (IDS). Although they both concentrate on spotting possible threats using a database of recognized attack patterns and signatures, how these modes address these threats is very different.

Snort monitors network traffic in Intrusion Detection System (IDS) mode, comparing it to a predefined set of rules intended to spot suspicious activity. Depending on how it is configured, Snort can notify administrators when it detects a possible threat and logs the event. Snort does not, however, take any action to halt or alter network traffic when operating in IDS mode. It monitors and reports on possible security threats, serving as a kind of surveillance system.

Similar to IDS mode, Snort actively analyses network traffic in Intrusion Prevention System (IPS) mode. However, it also has the ability to block or alter packets that match known malicious traffic patterns. This implies that Snort can serve as a gatekeeper by preventing the detected threat from reaching its intended target.

Comparing IDS and IPS Modes

- Detection vs. Prevention: An IPS can act quickly to stop a threat from causing harm, whereas an IDS only detects and warns about possible threats.
- Deployment: Whereas IPS needs to be installed inline to actively interfere with traffic flow, IDS can be installed anywhere in the network where it can monitor traffic.
- Effect on Network Performance: Compared to an IDS, which merely monitors traffic, an IPS may have a bigger effect on network performance due to its requirement for real-time analysis and possible modification of traffic.
- Risk of False Positives: When legitimate traffic is mistakenly blocked in IPS mode due to a false positive—a mistaken identification of legitimate traffic as malicious—network operations may be affected. In IPS mode, this risk makes precise rule configuration and tuning even more important.

The network architecture, risk management plan, and particular security requirements of an organization all play a role in which IDS or IPS mode is selected. To achieve complete security coverage, some environments might find it advantageous to deploy Snort as both an IPS and an IDS in separate network segments. This way, they can take advantage of each technology's advantages.

2. Aims and Objective

2.1 Aims

The primary goal of this project is to successfully configure Snort and Splunk in a virtual environment for intrusion detection and prevention of our home network, as well as to further analyze Splunk logs to better understand the workings and importance of Snort and log analysis and how it can help with network security.

2.2 Objectives

The objectives are as follows:

- To create a virtual environment where different machines are installed to simulate an attack and detect those intrusions.
- To step up Snort to detect and prevent any intrusion in the home network.
- To analyze the log captured by Snort in Wireshark and use Splunk for further investigation.

3. Project development

3.1 Network Designing

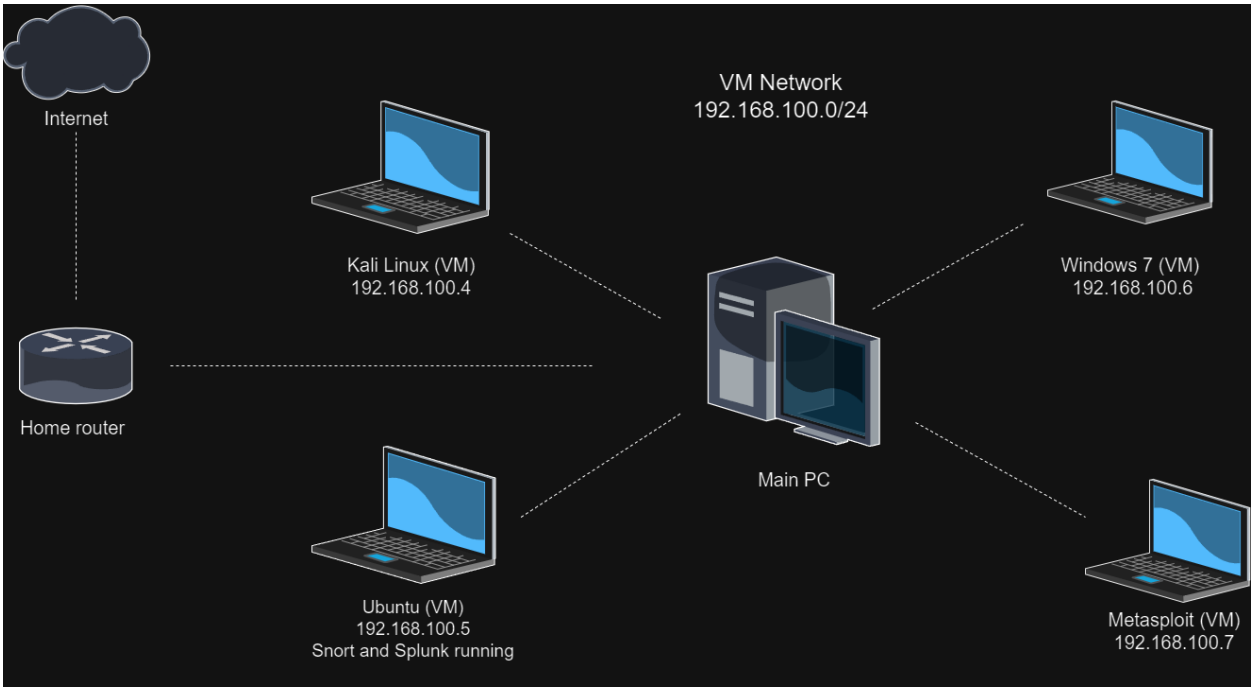


Figure 1 Network design

Here as you can see 4 virtual machines are working inside a virtual box with a network of 192.168.100.0/24. There is a main PC where all the virtualizations are done and connected to the internet through a router. The Ubuntu machine configures the Snort and Splunk, where all the packets and logs are monitored for the entire network. Kali Linux is the main attacking machine, and most of the attacks are performed on the Metasploit and Windows machines, whereas the Ubuntu machine, where Snort and Splunk are installed, captures and stores all the files and logs for further analysis.

3.2 Virtual Machines inside the Virtual Box.

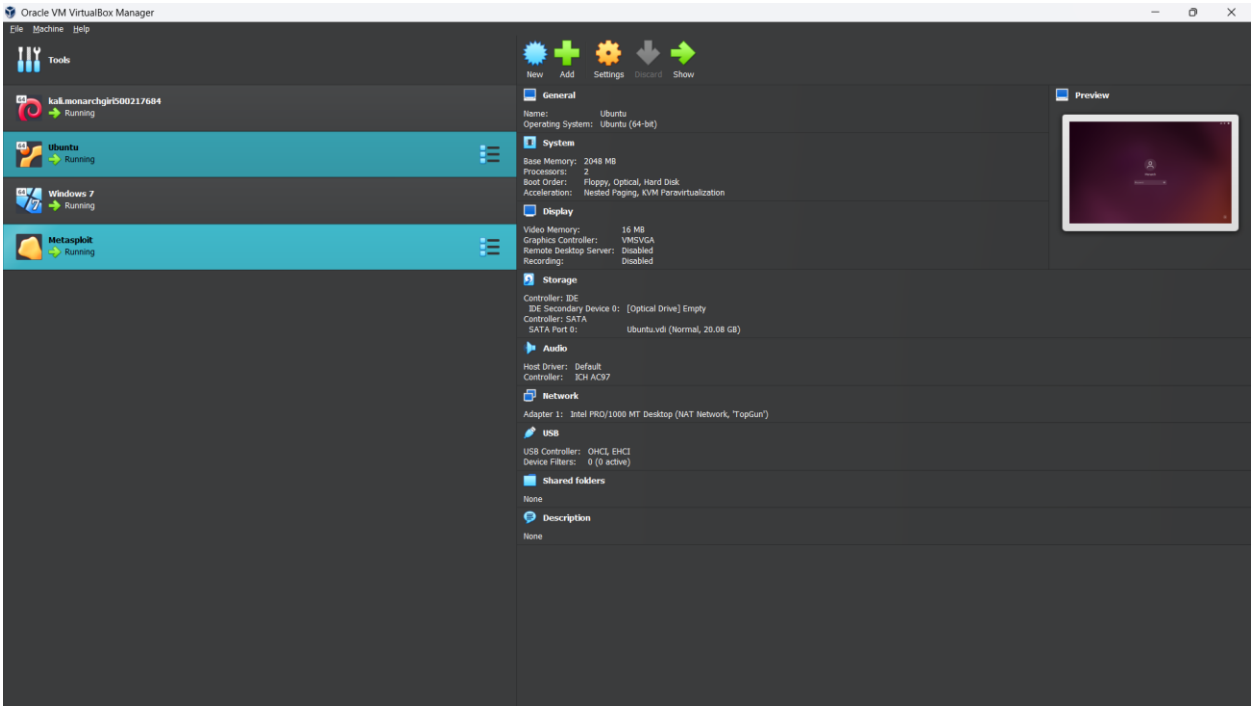


Figure 2 Virtual machines

The above screenshot shows that 4 machines are running inside a virtual box, there are Kali Linux, Ubuntu, Windows 7, and Metasploit which are using a NAT network "TopGun" (192.168.100.0/24). Each of the machines is installed so that they can communicate with each other, and the connection is tested with an ICMP request and response.

3.3 Snort in Ubuntu machine

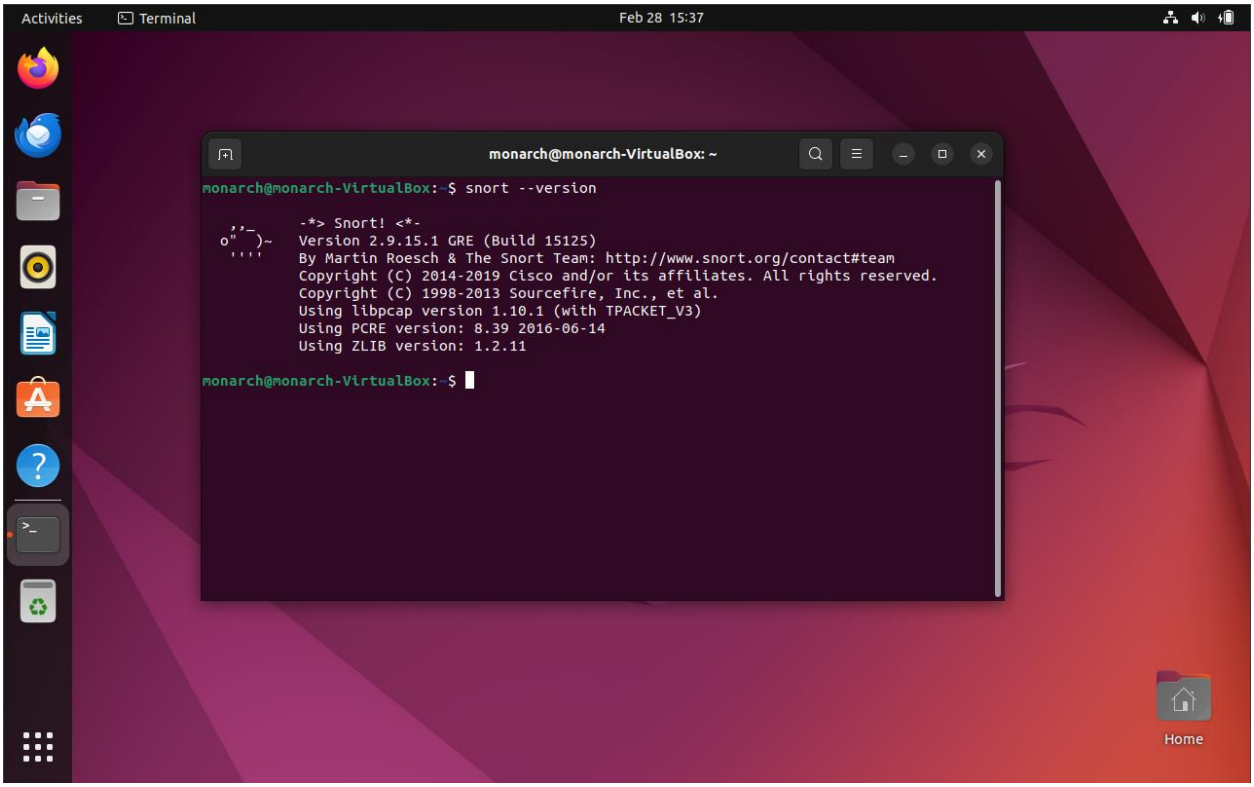


Figure 3 Snort installation

Snort is installed in Ubuntu as this would be the main focus of the project. The installation process is simple and after that to confirm the installation, we can just check its version, the snort installed here is snort 2.9.15.1.

3.4 Snort Rules

In Snort, there are 3 types of rules:

- Community rules – These are free rules created by the Snort community.
- Registered rules – These are the free rules created by Talos and we need to register for an account to use these rules.
- Subscription rules – These are the rules that need a paid subscription to access them.

The Snort comes with a predefined rule set to work on but for this project, we are not going to work with the predefined rule, because we are going to write our own local rules. So that we can practice writing our own rules and trigger the rules of our choice.

Writing a basic snort rule is not that difficult. Below is a snort rule syntax.

Figure 4 Snort rule structure

Here the alert is the action to perform, it can be either drop, block, log, or reject, in this case, it's alert. After that, a protocol is set and then first there is the source IP and source port. Here any means that the IP and port can be anything. The arrow is the direction of the packet flow, after that, we set the destination IP and destination port. Inside the brackets we can put any msg we want to display, a reference, we can define sid as well as rev.

So, we are going to write some basic local rules that we can play around with.

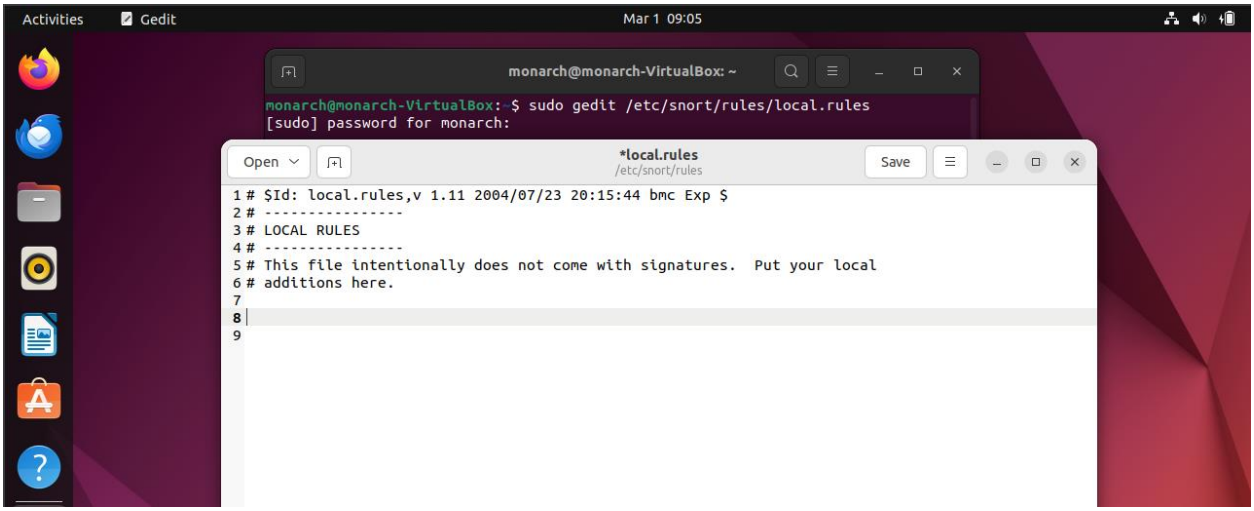


Figure 5 Local. rules

To go to the local rules file, we need root privileges, and the command shown in the screenshot above opens the local.rules file and any text editor will work, in this case, gedit is used. Here we can write rules any rules we want to play around with.

3.5 Writing local Snort IDS rules and triggering those rules

3.5.1 Writing and triggering ICMP rule

Figure 6 ICMP rule

This is a simple ICMP ping request rule which gives alerts when any ICMP packet travels in the network or any user wants to send an ICMP packet to any other user in the network. The protocol here is ICMP, source IP, and port can be anything as it is defined as “any any”, the destination is our home network (192.168.100.0/24) i.e. $HOME_NET, and the destination port can be any port. The message that would be displayed when the rule is triggered would be “ICMP ping detected”, and the sid and rev are given respectively.

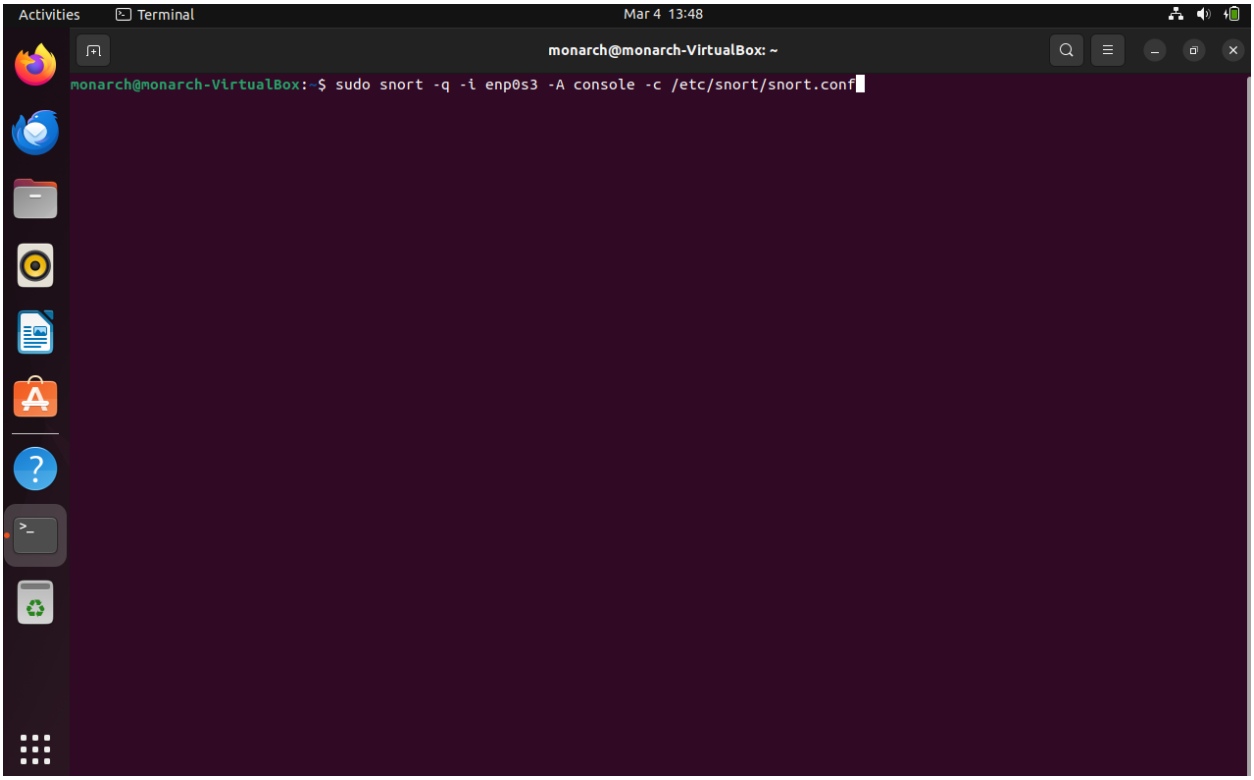


Figure 7 Command to run snort

Here on the Ubuntu machine, the snort is in listening mode to start the packet capture. It's on quiet mode, the interface enp0s3 is specified so that the snort knows what interface to look at and for now the packet capture is shown in the console and then the configuration file is mentioned.

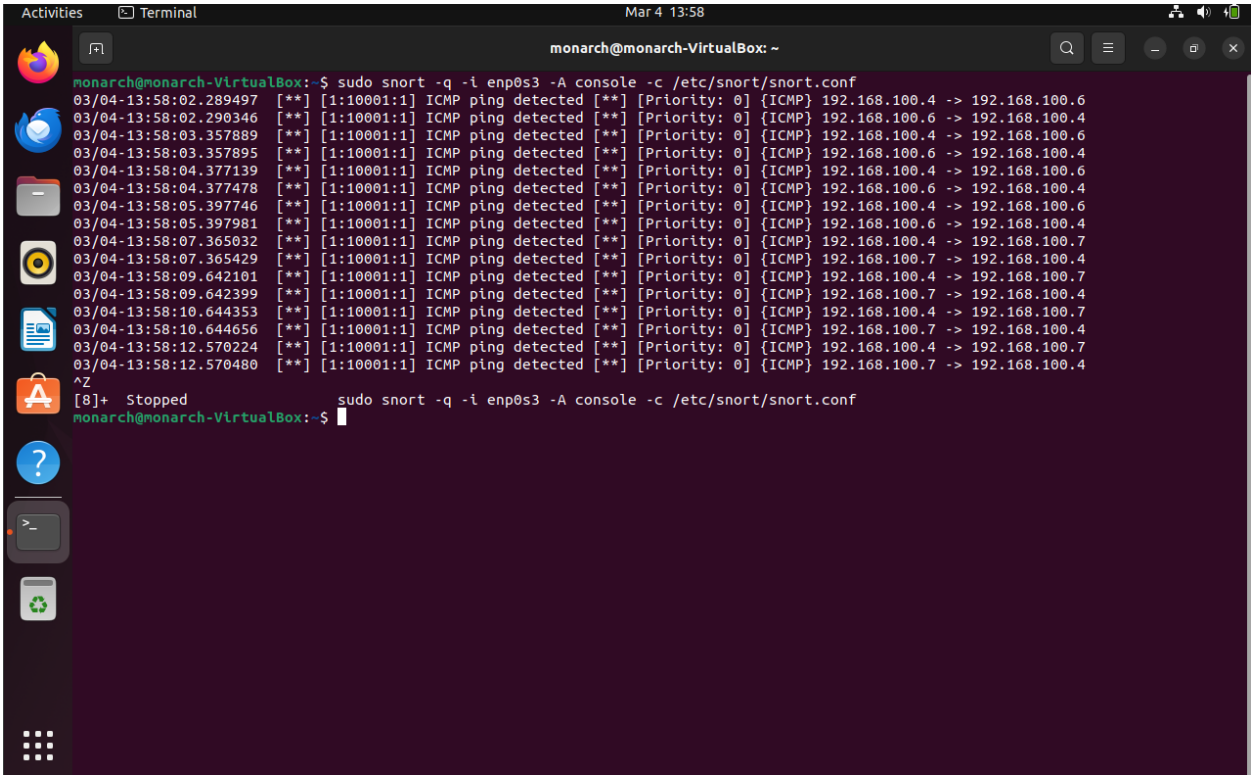


Figure 8 ICMP ping detected

Now the snort captured some traffic as I started the ping request from Kali Linux to Metasploit and Windows machine. The message says ICMP ping detected as we defined in our rule, it gives the protocol and the IP address of the machine as well. This is a simple way that we can write a rule and trigger that rule. Let's see some more examples.

3.5.2 Writing and triggering different FTP rules

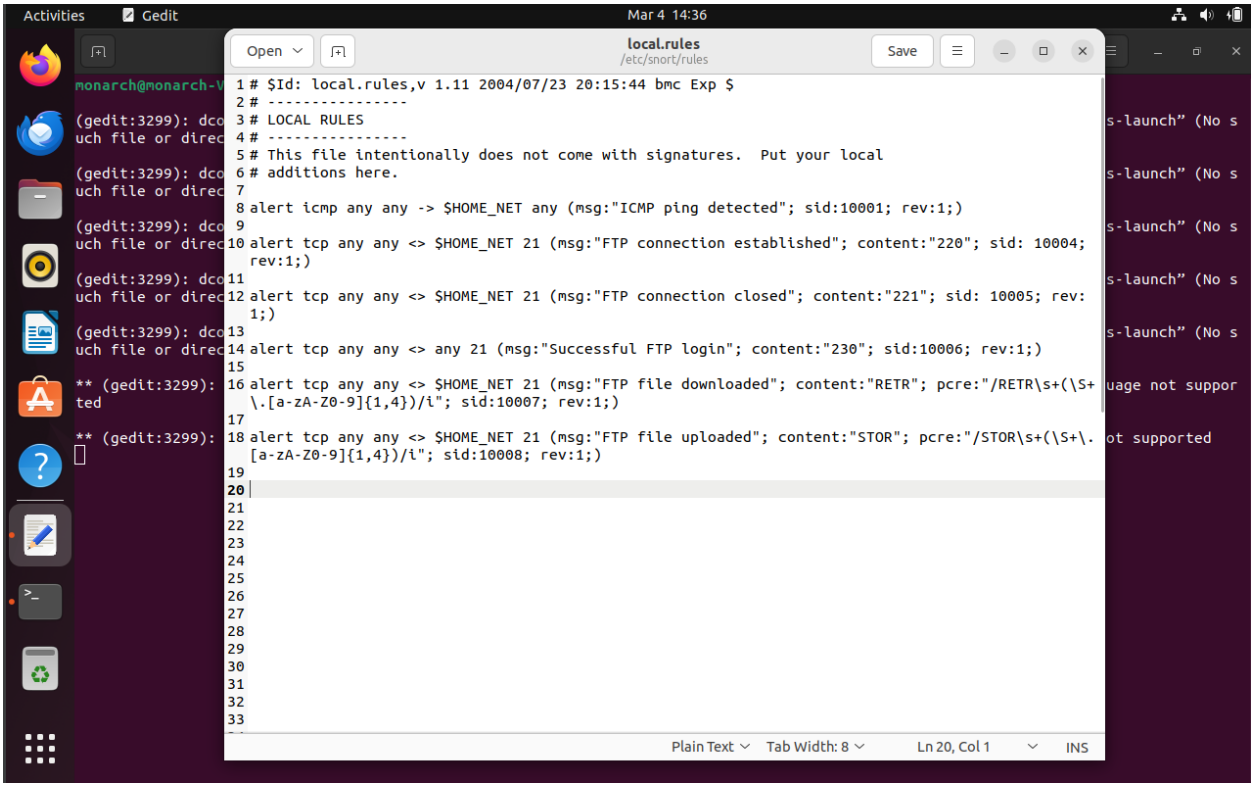


Figure 9 FTP rules

Here are some rules written for FTP.

- The first rule is when an FTP connection is established. The protocol that FTP uses is TCP and it listens on port 21 which is specified in the rule. The content here is 220, the “content” keyword specifies the content or pattern that snort should look for in the payload of the packet. Content 220 states that the FTP server is ready for a new login.
- The second rule is when an FTP connection is closed. The difference here compared to the first rule is only the content value which is 221, it states that the FTP connection is set to be closed.
- The third rule is for a successful FTP login. When the user enters the correct username and password, this rule is triggered so that we know someone has successfully connected to the FTP service in the network. For this rule, the content value is 230.
- The fourth rule is triggered when a user downloads any files from the end devices. The content for file download is “RETR” which is commonly used for file retrieval or

download. The “pcre” part matches the "RETR" command with a filename using a Perl Compatible Regular Expression (PCRE). This simple regular expression assumes that the filename has a 1 to 4-character file extension.

- The fifth rule is triggered when a user uploads any files from the end devices. The content for file upload is “STOR” which is commonly used for file upload. The “pcre” part matches the "STOR" command with a filename using a Perl Compatible Regular Expression (PCRE). This simple regular expression assumes that the filename has a 1 to 4-character file extension.

Triggering the rules:

Figure 10 Attack from kali

For the FTP rules to be triggered I’ve performed an attack from Kali Linux to the Metasploit machine as it has an FTP port open. As shown in the screenshot above, the FTP connection is attempted on the machine, the login is successful, and also the files are downloaded and uploaded successfully through FTP. After that, the connection to the Metasploit machine is closed.

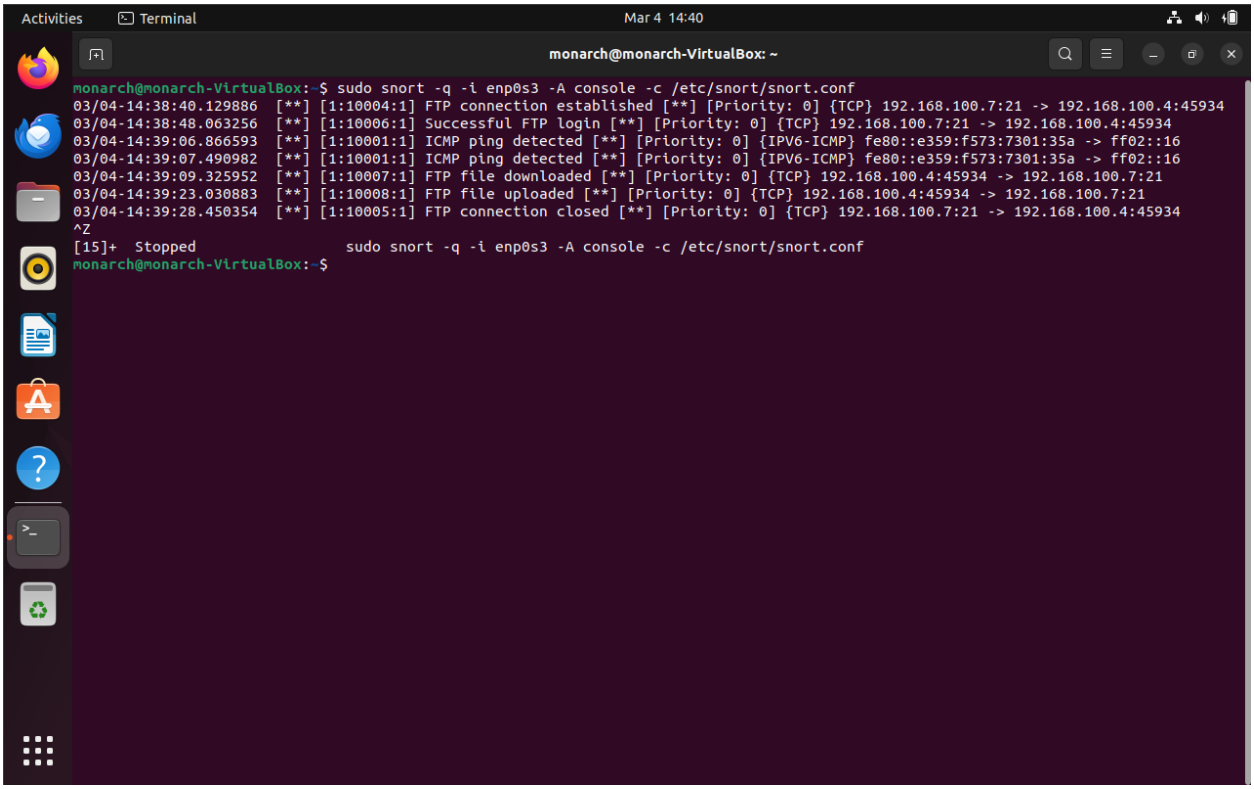


Figure 11 FTP detection

As the rules were defined the snort did its work and it captured all the traffic. First, it recorded the FTP connection, then the successful user login, the upload and download of the file from Kali Linux, and at last it also captured the packet when the collection is closed.

FTP is one of the used protocols and it is also vulnerable to attack. If you have the username and password combination, you're in. So, there are many predefined community rules for FTP in snort as well, and keeping this protocol safe is important. As in the example above, uploading and downloading files is easy using FTP, the file that I uploaded from Kali Linux was a simple text file, but it could have been a malicious file which can be a threat to the Metasploit machine. So, using snort to detect that traffic is important, or using IPS we can even block that traffic, which we will look at later in this report.

3.5.3 Writing and triggering different NMAP rules

NMAP (Network Mapper) is one of the most popular and used scanning tools for active reconnaissance. Scanning the network, system, and ports is one of the first things that any hacker would do to find information about the targeted network or the system. NMAP is not only used for scanning to find valuable information instead it can also be used for launching different attacks by using NSE (Nmap scripting engine). Implementing IDS/IPS to detect those types of scans is very crucial for maintaining the confidentiality of our data.

There are many ways to perform an NMAP scan, according to our needs and the information we want. There are ways the hackers use the NMAP to perform their task without wanting to trigger rules in the firewall and IDS/IPS. So, let's write some rules so that we can detect some common NMAP scans.

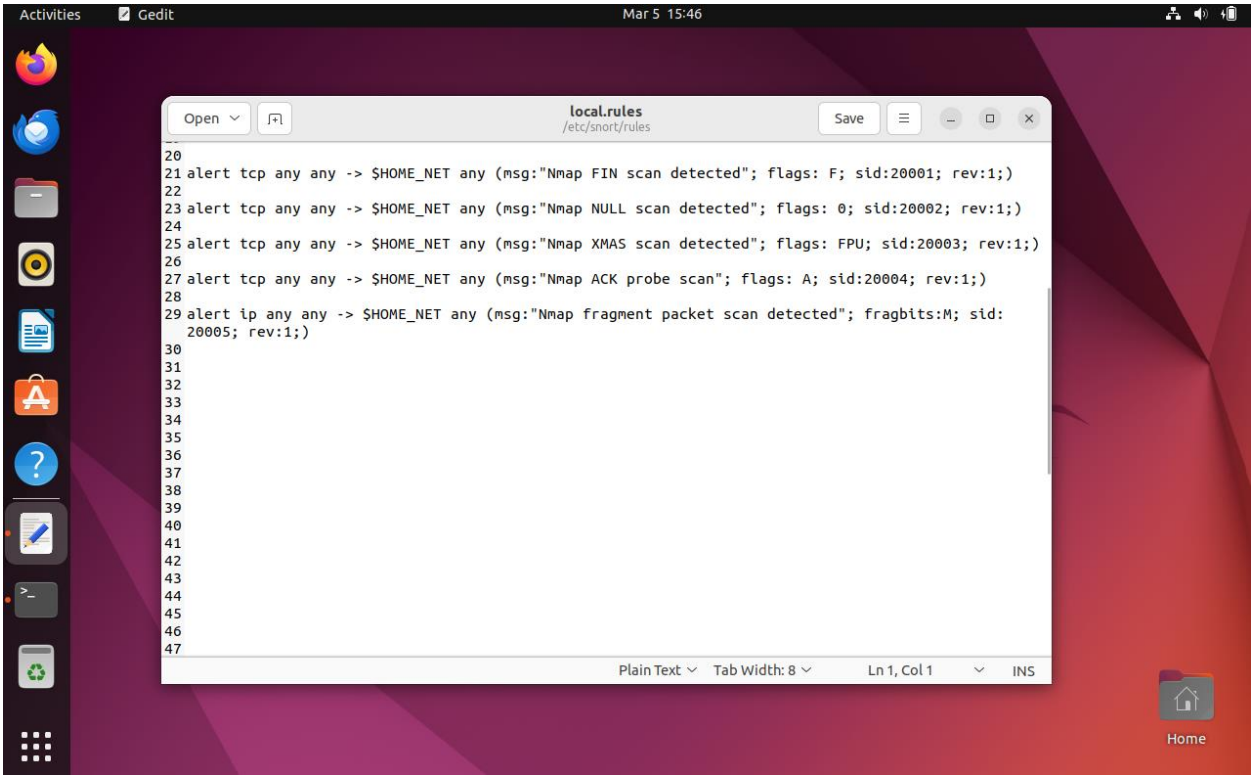


Figure 12 IDS rules for NMAP

These are some of the important rules that help us detect NMAP inverse TCP flag scanning, ACK probe scan, and fragment scan. Hackers typically use these types of scans to remain undetected by

firewalls and IDS/IPS as these systems usually only detect TCP connect scans or other stealth scans. But these rules written are especially here to detect those kinds of NMAP scans.

- The first rule is to detect any NMAP FIN scan. In the FIN scan, NMAP sends the FIN flag set to the target. The flag used in the rule is “F”.
- The second rule is to detect any NMAP NULL scan. Here the NMAP sends the packet with no flag set. The flag used in the rule is “0”.
- The third rule is to detect any NMAP XMAS scan. NMAP sends packets with the FIN, URG, and PSH flags set in the XMAS scan. Similar to the FIN and NULL scans, the response pattern is observed to determine the state of a port. The flag used in the rule is “FPU”.
- The fourth rule differs from the others that have been covered thus far. Firewall rulesets are mapped out using it to identify which ports are filtered and whether or not they are stateful. ACK probe scans are usually performed to know the status of the firewall of the target. Here, NMAP sends an ACK flag and the flag used in the rule is “A”.
- The fifth rule is to detect any fragmented or MTU scans that would be performed in NMAP. Here the rule set “fragbits: M” is mentioned to indicate that when the packet is part of a fragmented IP datagram and the "More Fragments" (M) flag is set, the rule will be triggered.

Triggering the rules:

1) FIN scan

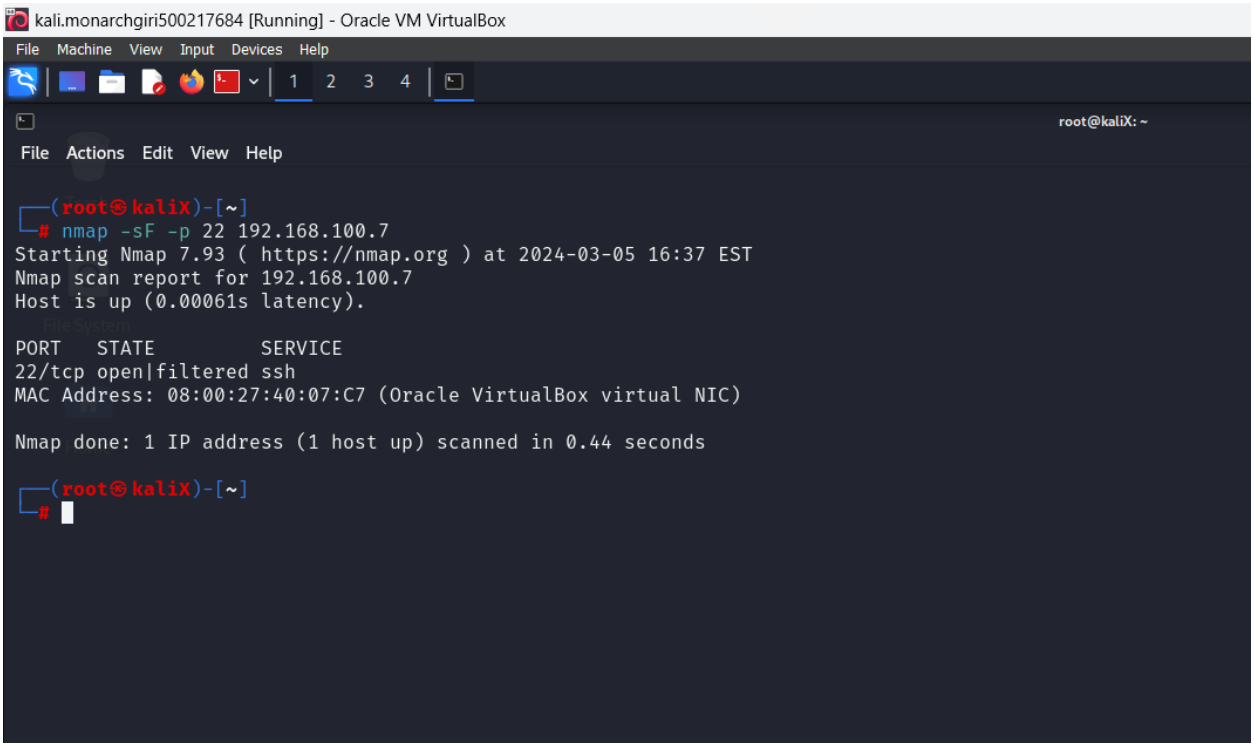


Figure 13 FIN scan

Here I've launched an NMAP scan on the Metasploit machine from my Kali machine and I'm using a FIN scan on port 22, it would be easy to do on a single port rather than scanning all the ports because if I were to perform a scan on all the ports there would be a lot of triggers and NMAP would also take time to perform the attack. It is only a demonstration, so I did it for only one port.

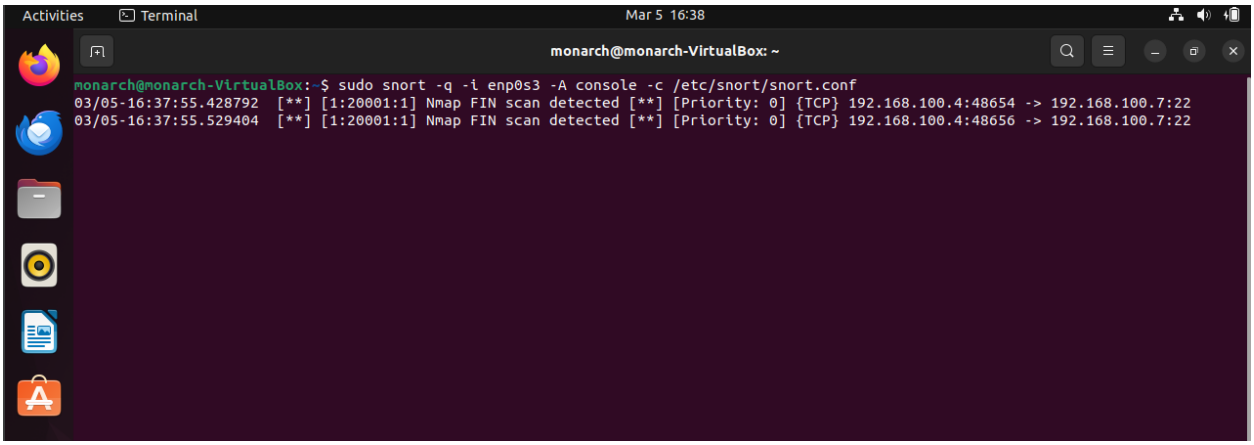


Figure 14 FIN scan detected

Now the rule is triggered as expected and the message is displayed saying that a FIN scan was performed from 100.4 to 100.7 in port 22. So, the rule is working properly.

2) NULL scan

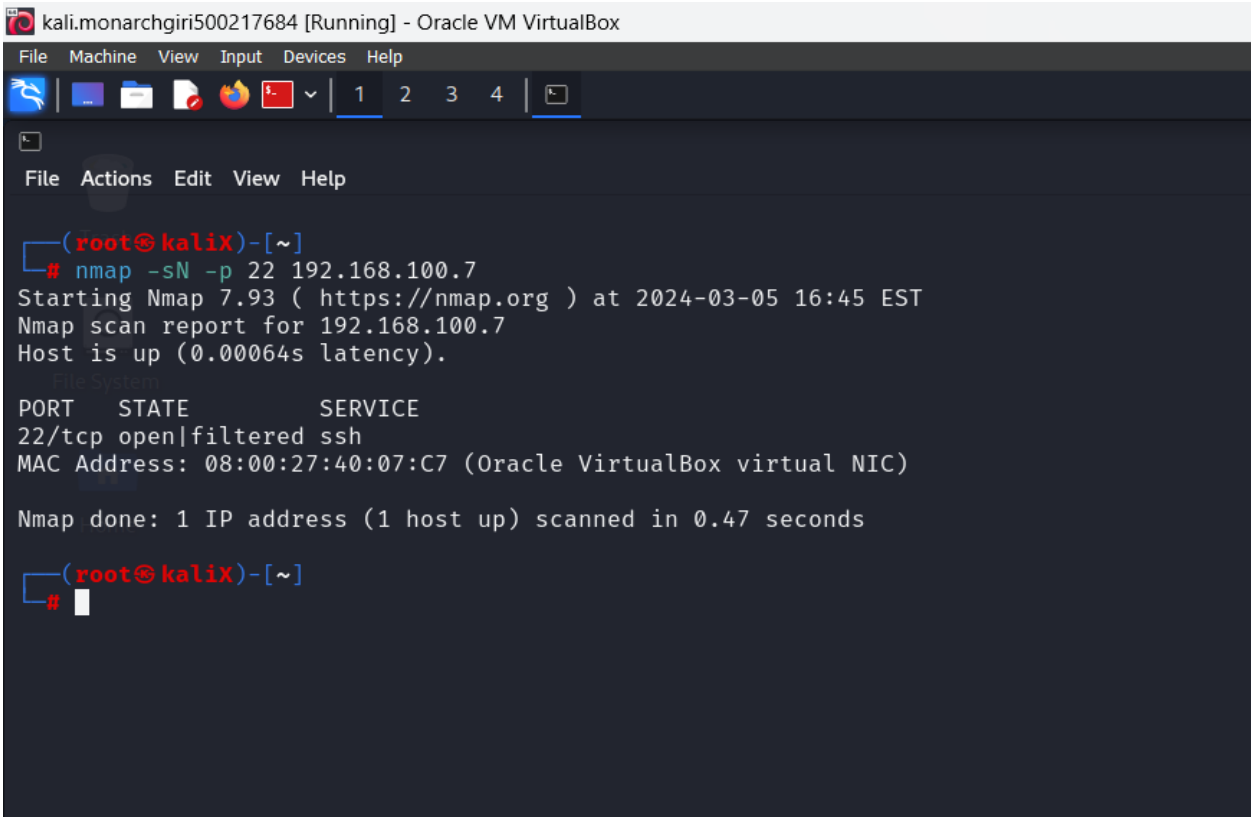


Figure 15 NULL scan

This is for the NMAP null scan.

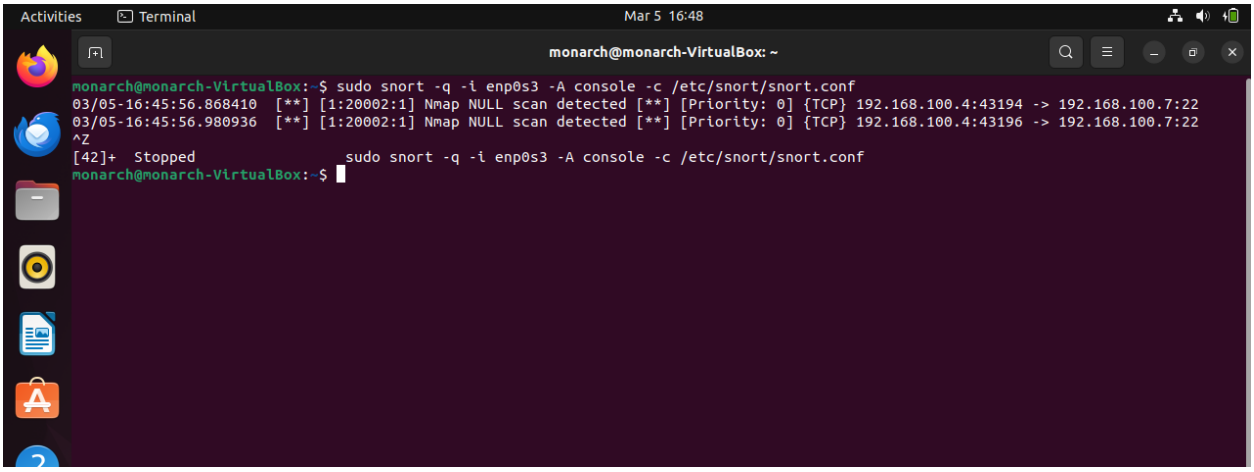


Figure 16 NULL scan detected

Here the rule for null scan is also successfully triggered.

3) XMAS scan

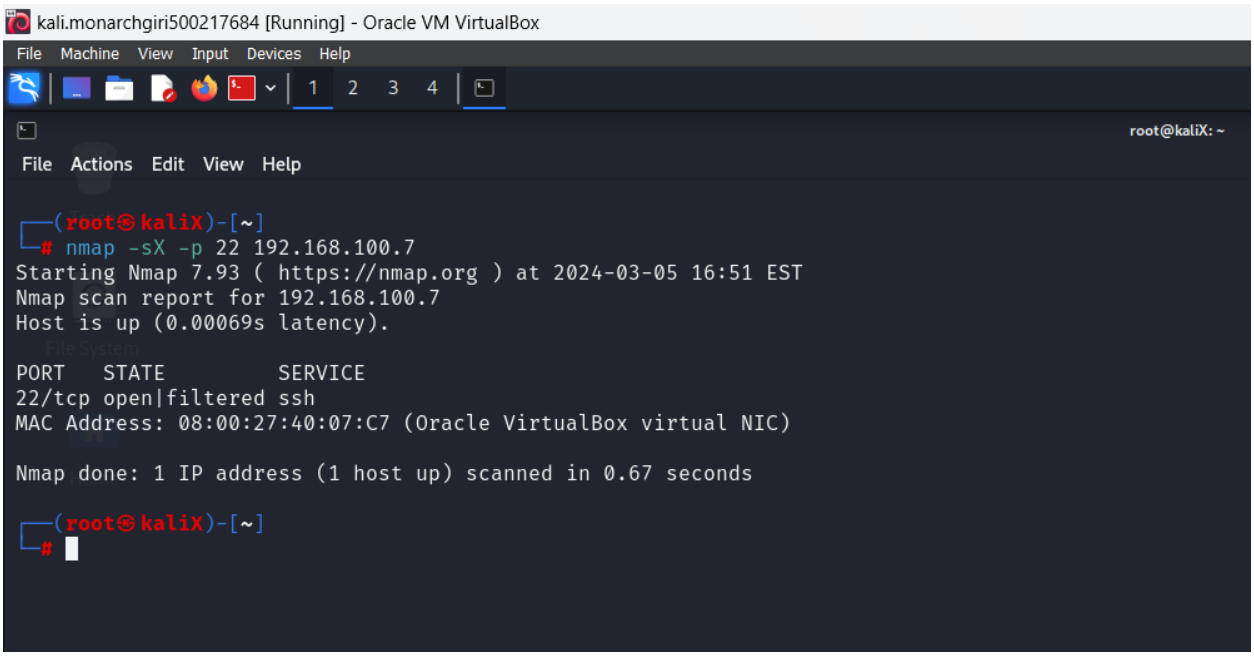


Figure 17 XMAS scan

Figure 18 XMAS scan detected

The XMAS scan is also done, and the rule is also successfully triggered.

4) ACK probe scan

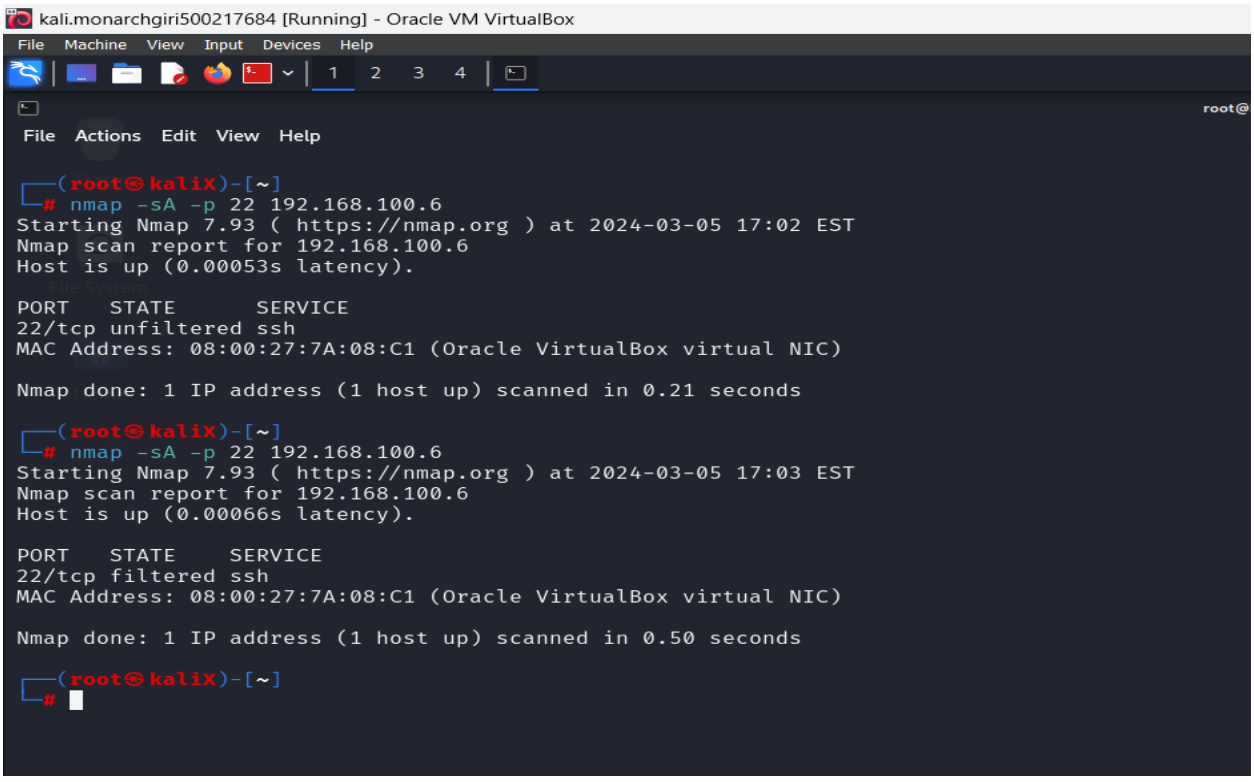


Figure 19 ACK probe scan

Here, I've targeted my Windows 7 machine, the first scan is done when the firewall is turned off, and as you see the status says unfiltered because there are no firewalls to filter the packets sent by

NMAP. The second scan is done when the firewall is turned on, and it says filtered because there is a firewall to filter the packets.

Figure 20 ACK probe detected

The rule is also triggered when the scan is performed. The attacking machine is 100.4 and the target machine is 100.6 which is the Windows 7 machine.

5) Using MTU scan.

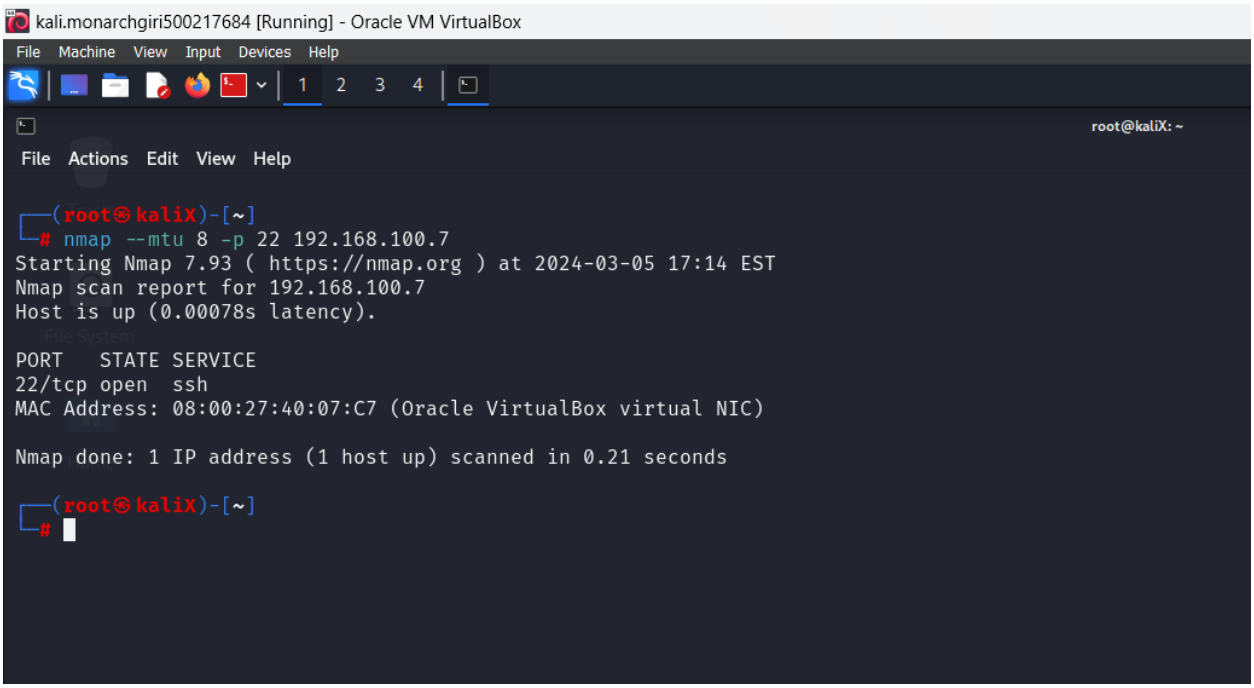


Figure 21 Fragment scan

Using MTU (minimum transmission unit) is a useful way to specify the transmission unit. We can send packets in terms of factors of 8, which is 8, 16, and 24 bytes. Here I've used mtu 8 to send 8 bytes of fragmented packets.

Figure 22 Fragment scan detected

The attack successfully triggered the snort rule.

3.6 Snort log files and Wireshark analysis

Snort log files are important as they store every triggered rule, and they can be further analyzed in tools like Wireshark. The log files can be found in var/log/snort. These are some of the log files that snort generated which are in computer/var/log/snort.

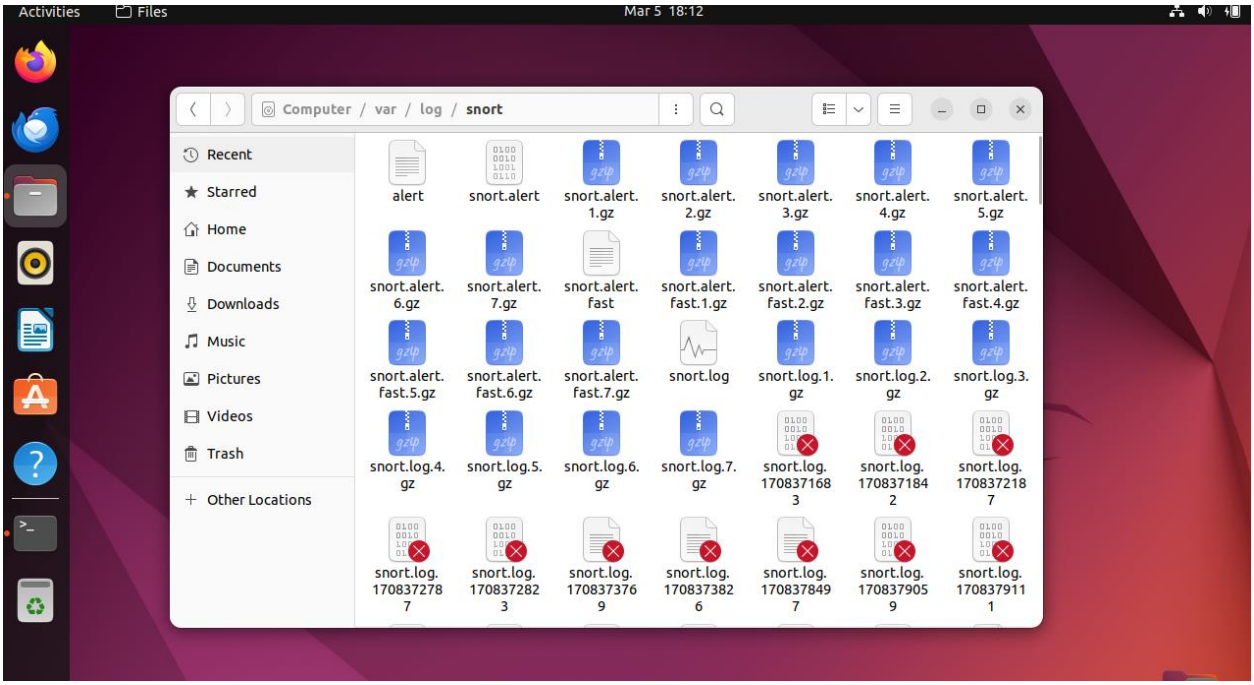


Figure 23 Wireshark files

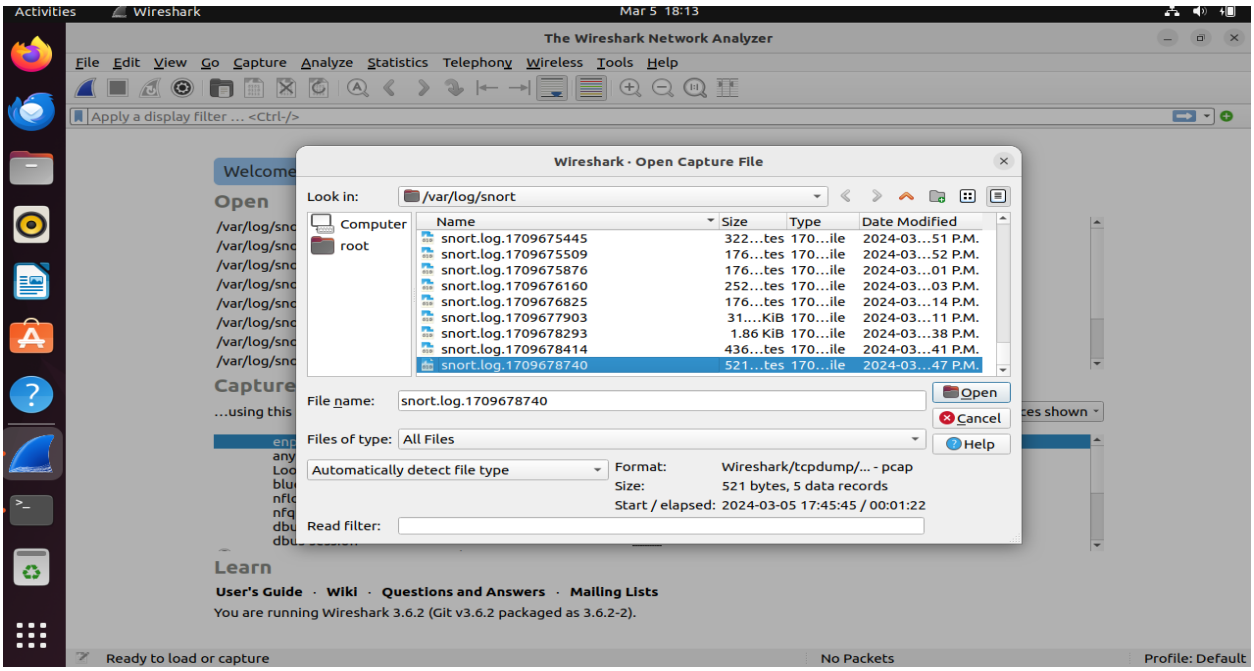


Figure 24 Files selection

We can also go through the Wireshark itself and open the snort log files from there as well.

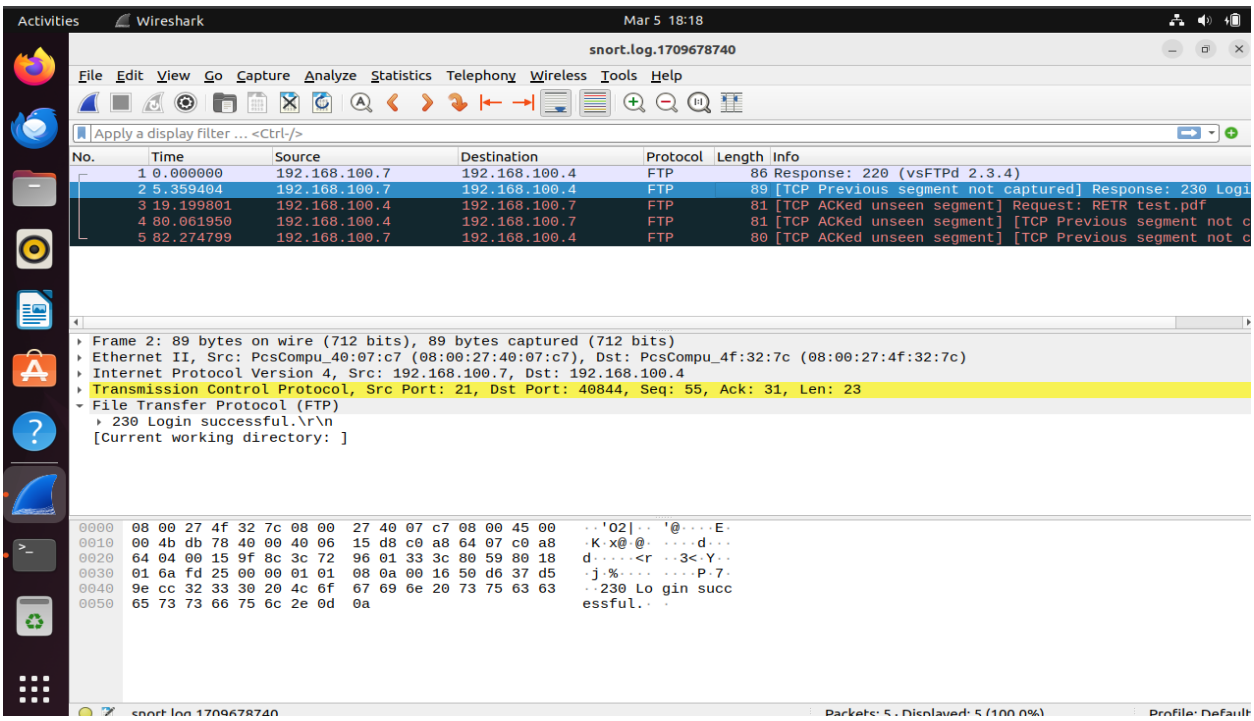


Figure 25 FTP login

This is one of the log files where FTP is used to log in to the Metasploit machine where files were downloaded and uploaded. As FTP is not a secure protocol and using Wireshark, we can see all

the activity that happened in that particular session. Here the above screenshot shows a successful login in FTP.

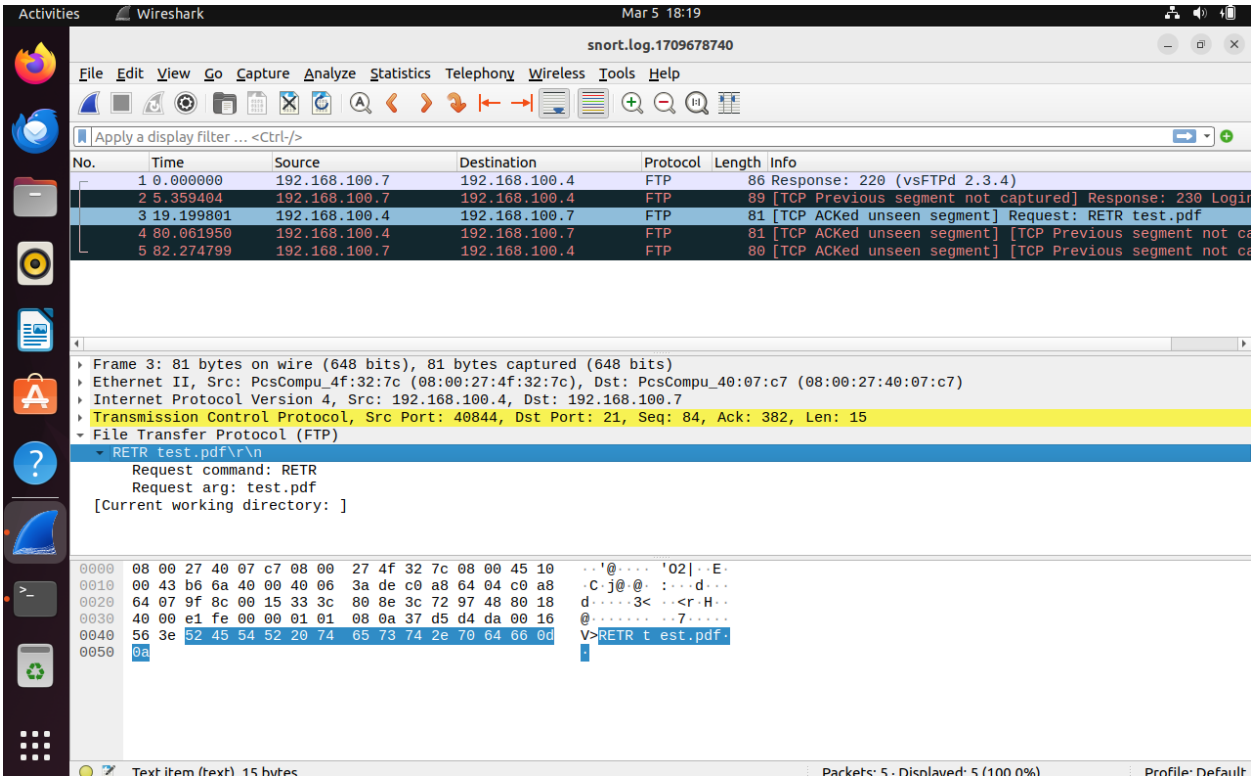


Figure 26 FTP download

Here we can see which file was downloaded using FTP. The request arg shows that the test.pdf file was downloaded through FTP.

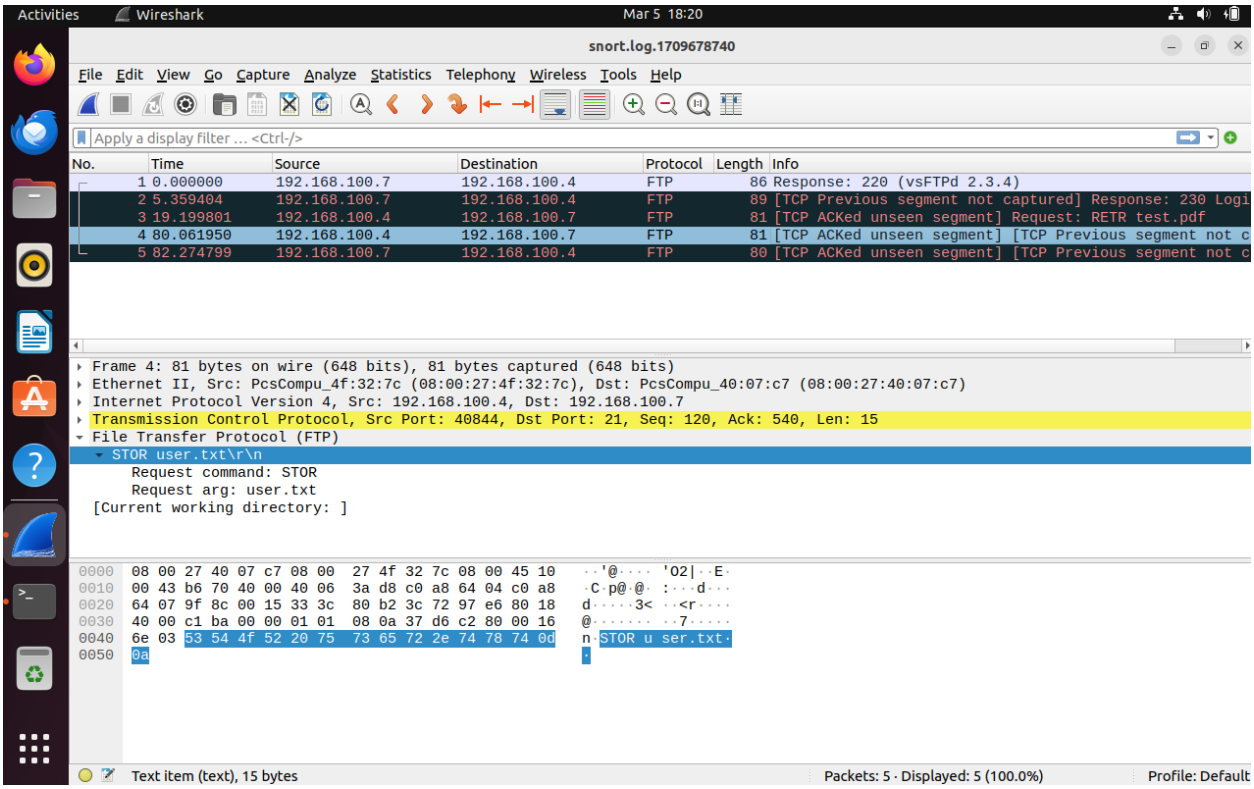


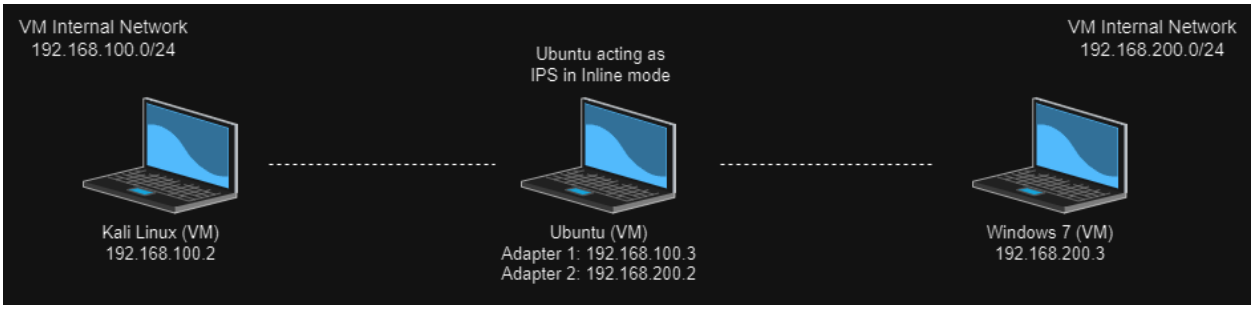
Figure 27 FTP upload

In the above screenshot, we can see that through FTP user.txt file was uploaded. Not only this Wireshark also gives us a lot of important information and it is very easy to analyze any pcap files. So, using an unsecured port is not a good practice because all the activities can be easily analyzed by hackers as well. If we are using a secured port like FTPS which is secured and instead of using telnet we can use SSH which would be a far better option as the messages would be encrypted.

3.7 Running Snort in IPS Mode

Running snort in IPS mode is a bit different than running in IDS. IDS is typically deployed in a passive mode where it observes the network traffic rather than being in the direct path of the flow of traffic. IPS on the other hand is deployed in line with the network traffic so that the packets generated in the network will pass through it where it can block or allow traffic based on the rule set.

3.7.1 Network Designing for IPS Mode



This is a simple setup where 3 devices are interacting with each other. The first one is the Kali Linux machine which is configured as an internal network in a virtual box with an IP of 192.168.100.2. The next one is a Windows 7 machine which is also configured as an internal network but in a different network with an IP of 192.168.200.3. The Ubuntu machine here is acting as a Snort IPS which has Snort installed and has 2 adapters, one connecting the Kali Linux machine and another connecting the Windows 7 machine. A simple IP forward is done to connect the 2 different networks through Ubuntu so that it acts as a Snort IPS in inline mode. This setup allows the traffic to pass through the Ubuntu machine. For example, if a Kali machine wants to ping a Windows 7 machine, it will pass through Ubuntu, but if the Ubuntu machine is down the connection will not be established.

3.7.2 Network configuration

For Kali Linux machine

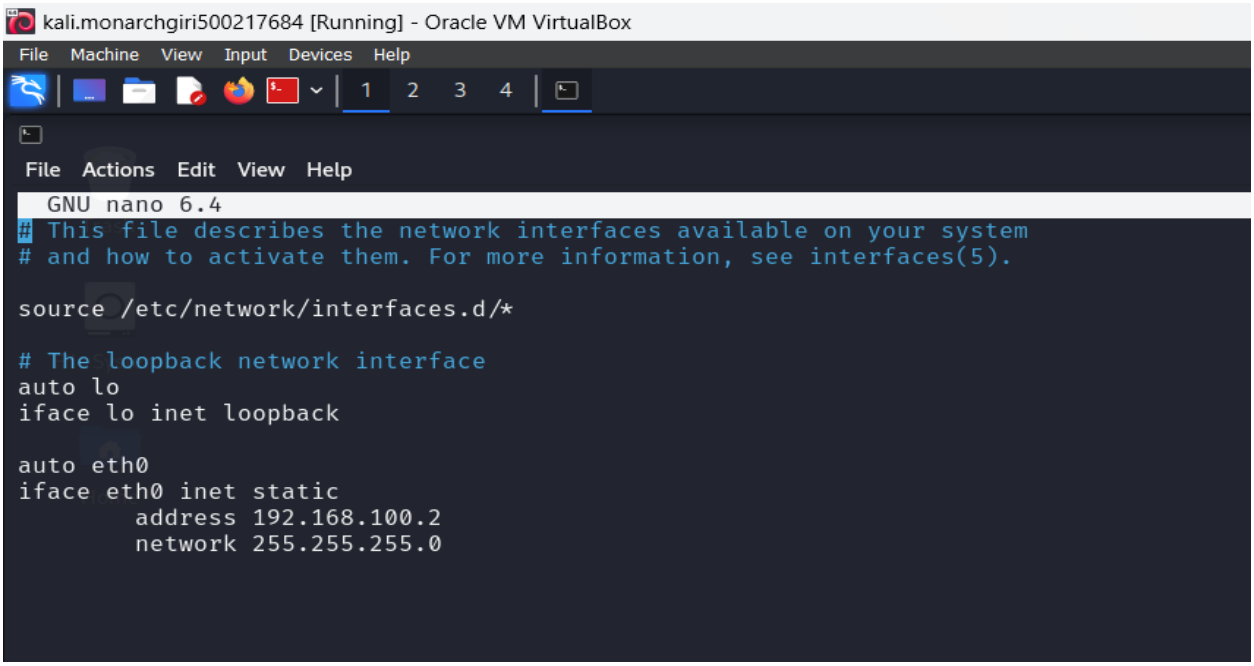


Figure 28 Static IP for kali

A static IP is configured for Kali Linux as shown in the screenshot above.

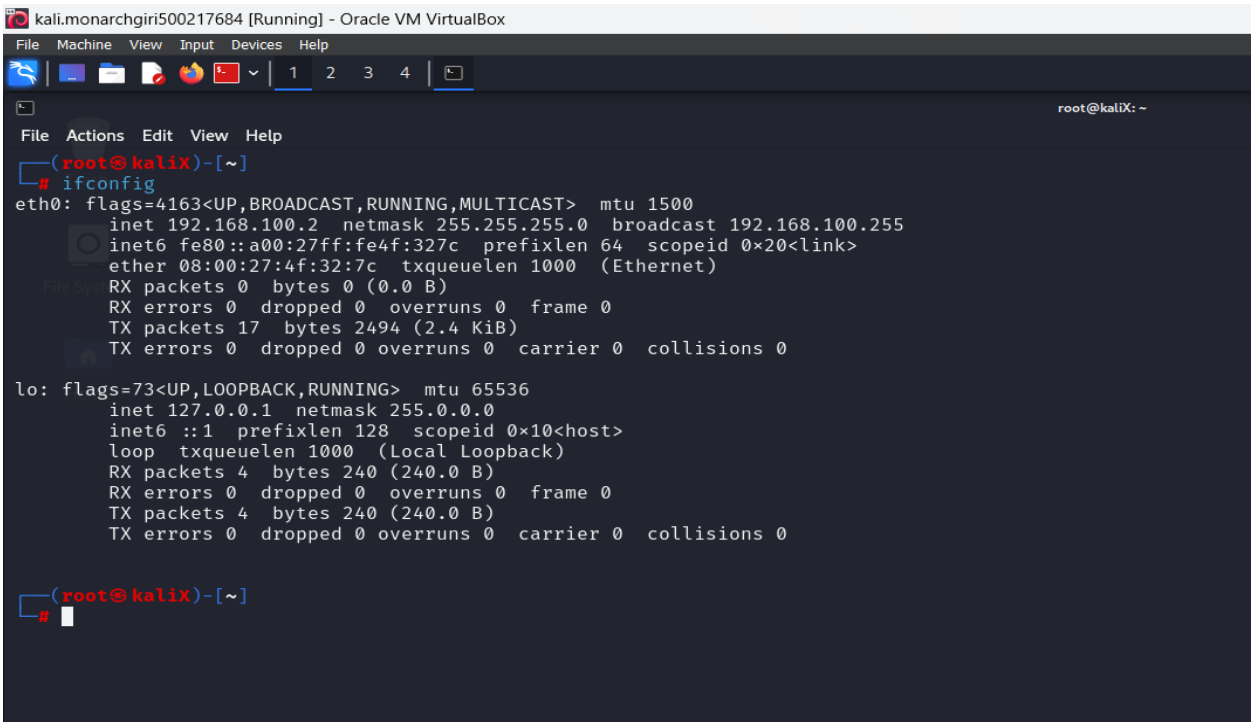


Figure 29 Kali IP

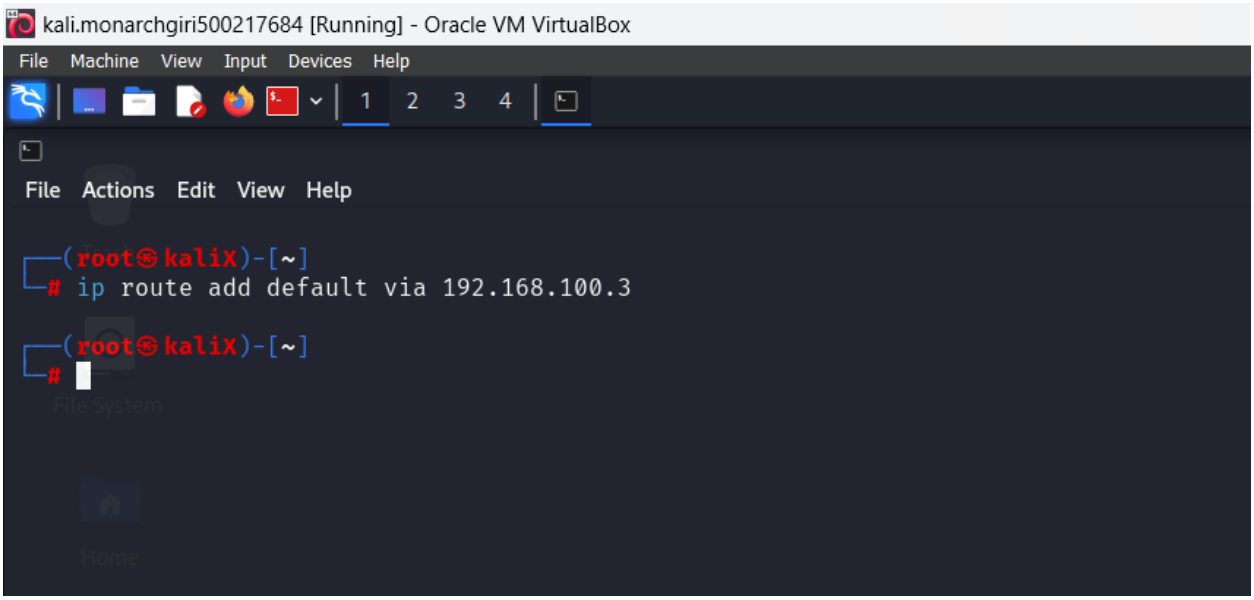


Figure 30 IP route

In the Kali Linux machine after configuring the static IP address the command shown in the screenshot is run to direct the packet via one of the Ubuntu machine interfaces (enp0s3).

For Ubuntu Machine

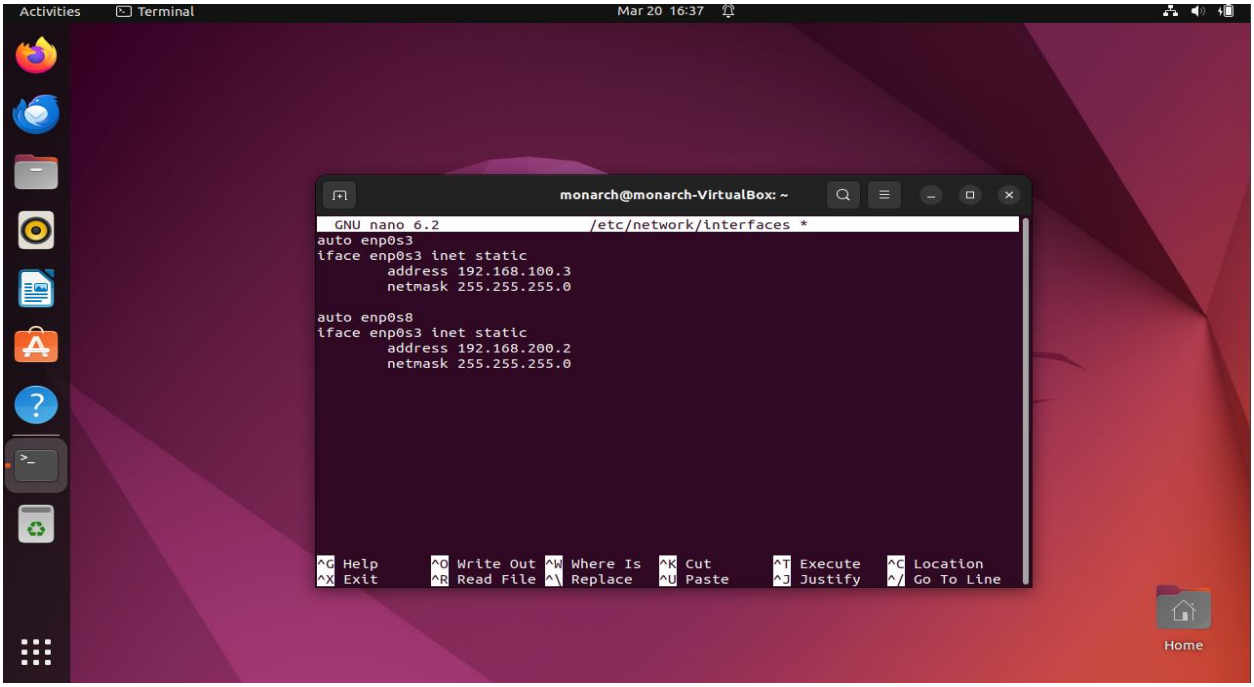


Figure 31 Adapter IP for ubuntu

Ubuntu has 2 adapters one connecting the Kali machine and one connecting the Windows 7 machine. Static IPs for both the interfaces are assigned here, enp0s3 has an IP of 192.168.100.3 which is in the same network as Kali machine, whereas enp0s8 has an IP of 192.168.200.2 which is the same network as Windows machine.

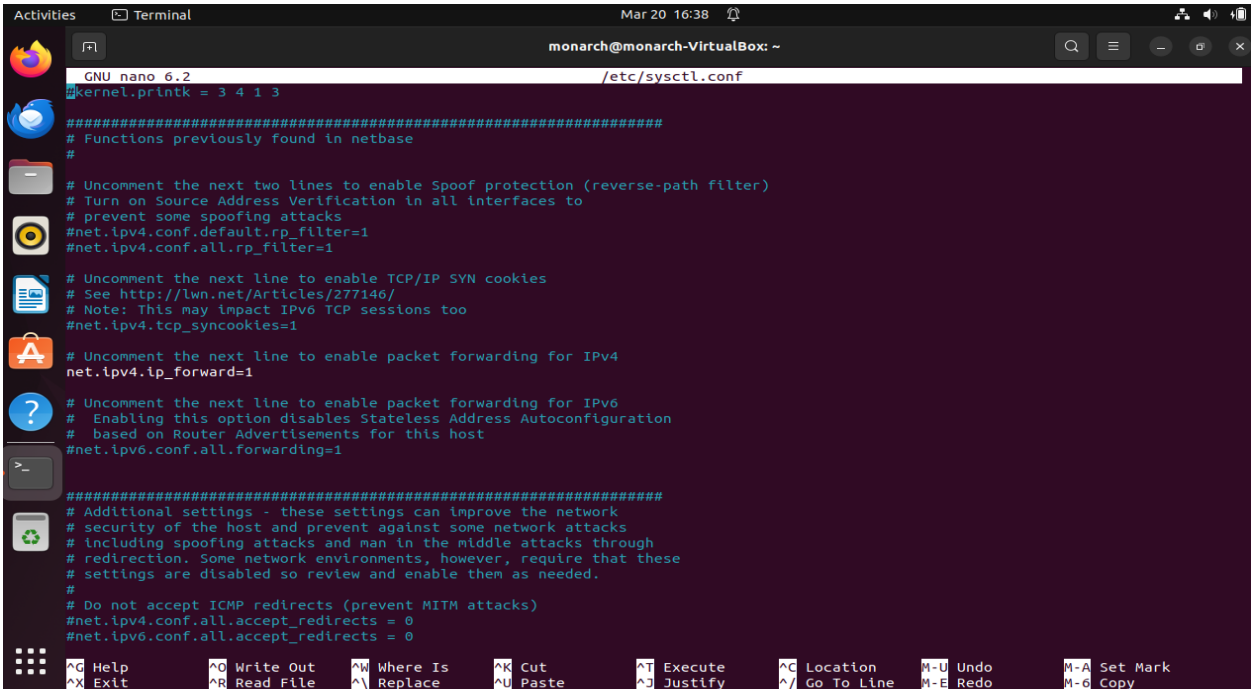


Figure 32 Enable packet forwarding

/etc/sysctl.conf file is where we can find this line, we need to uncomment to enable packet forwarding between networks.

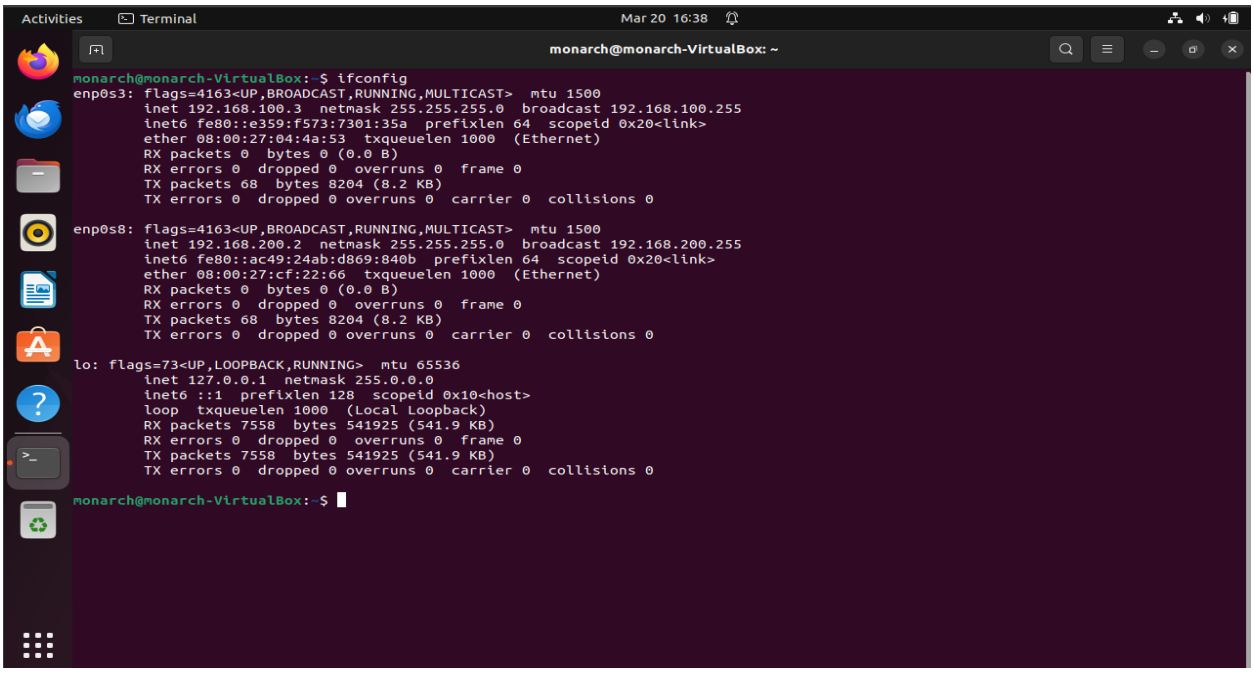


Figure 33 IP's in ubuntu

After that as we can see the two interfaces have the IP that we provided.

For Windows 7 Machine

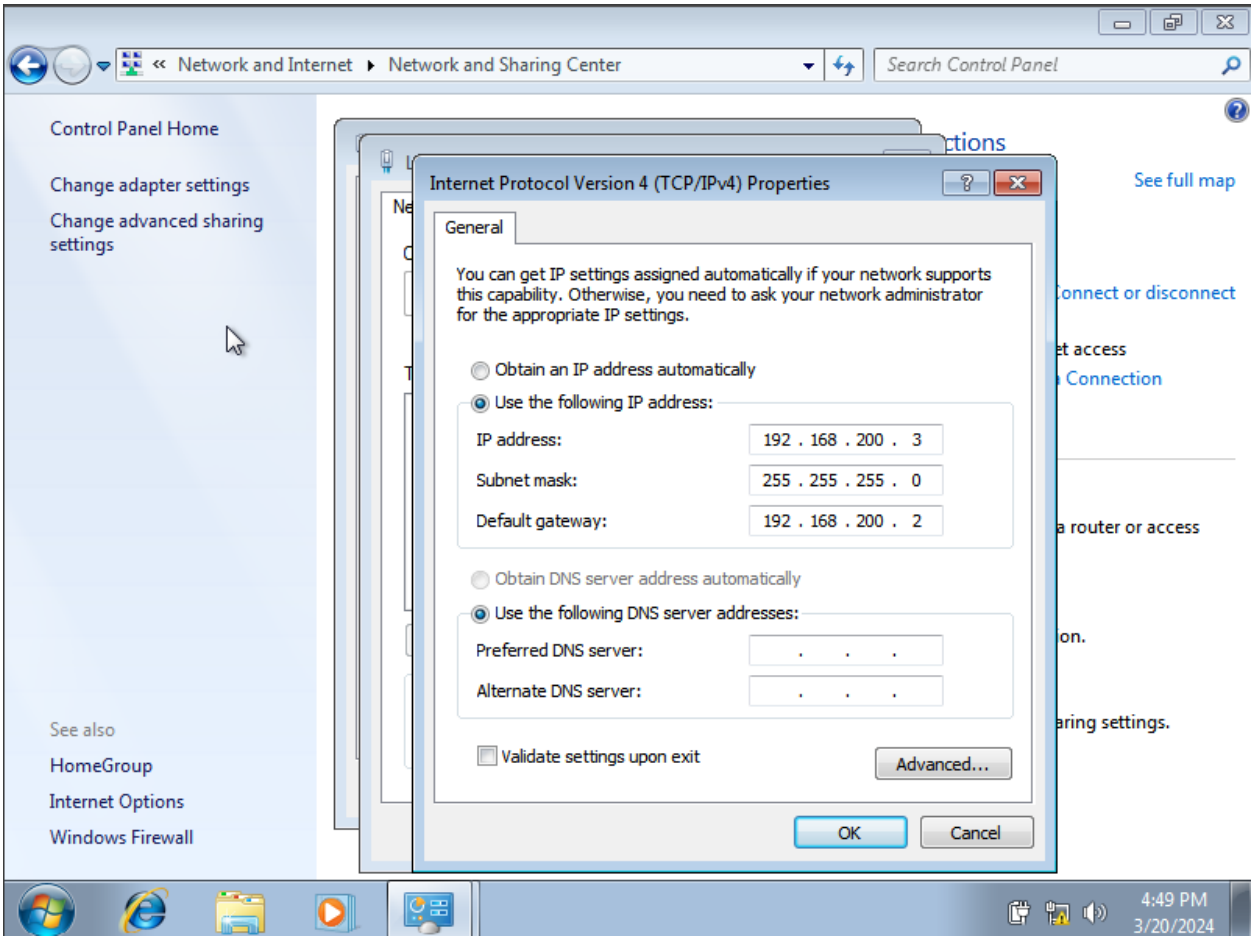


Figure 34 Static IP for Windows 7

To configure the static IP in a Windows 7 machine, it can be found in the control panel inside the network and the internet. As shown in the screenshot above, the static IP is configured, and the default gateway is set to one of the interfaces of the Ubuntu machine (enp0s8).

3.7.3 Testing the Network Configuration

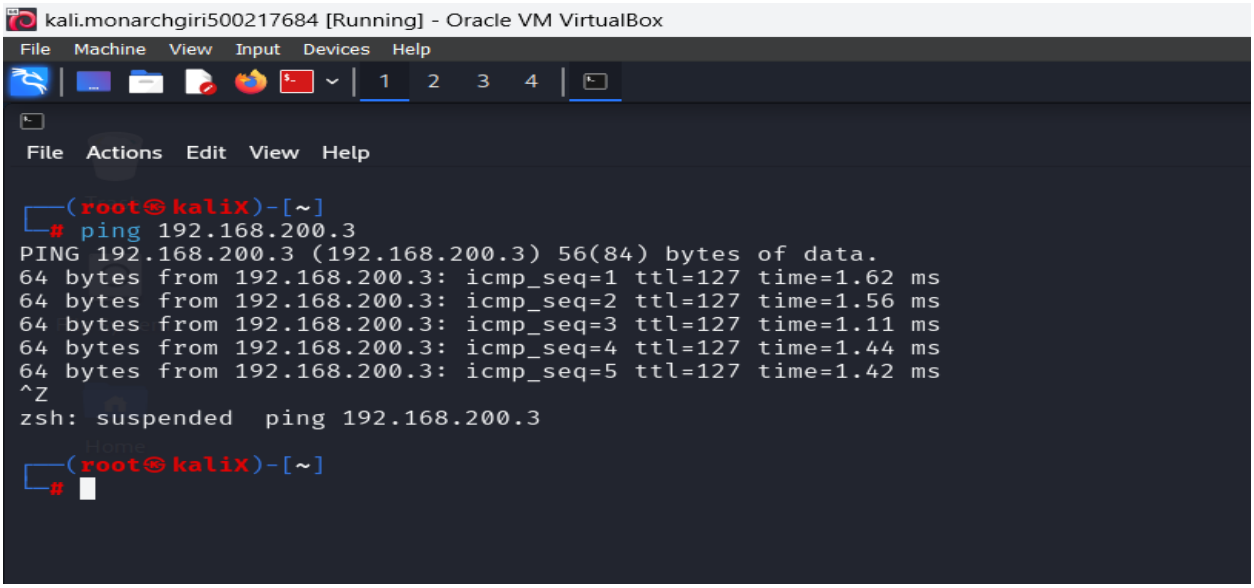


Figure 35 Ping from kali

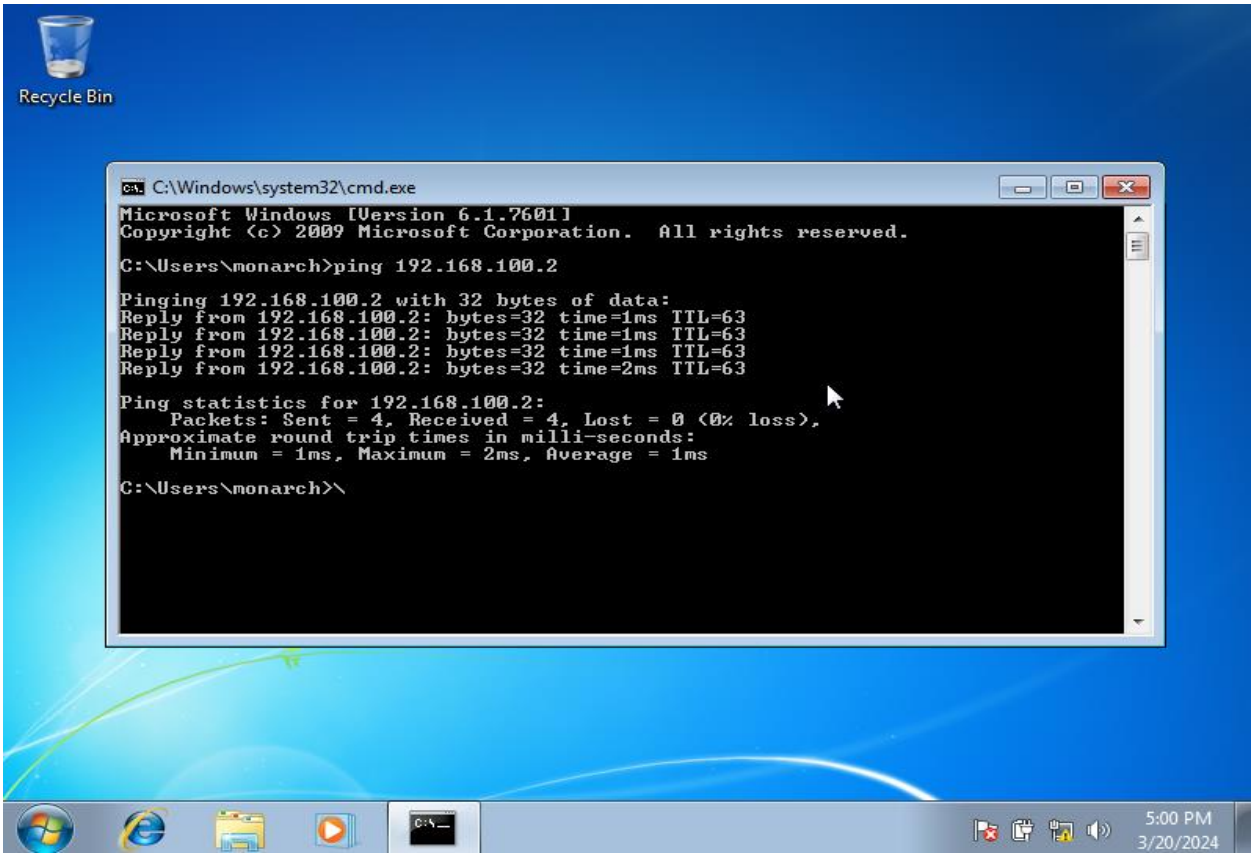


Figure 36 Ping from Windows

As we can see both machines can ping each other, which means that our configuration is properly done.

3.7.4 Snort IPS configuration

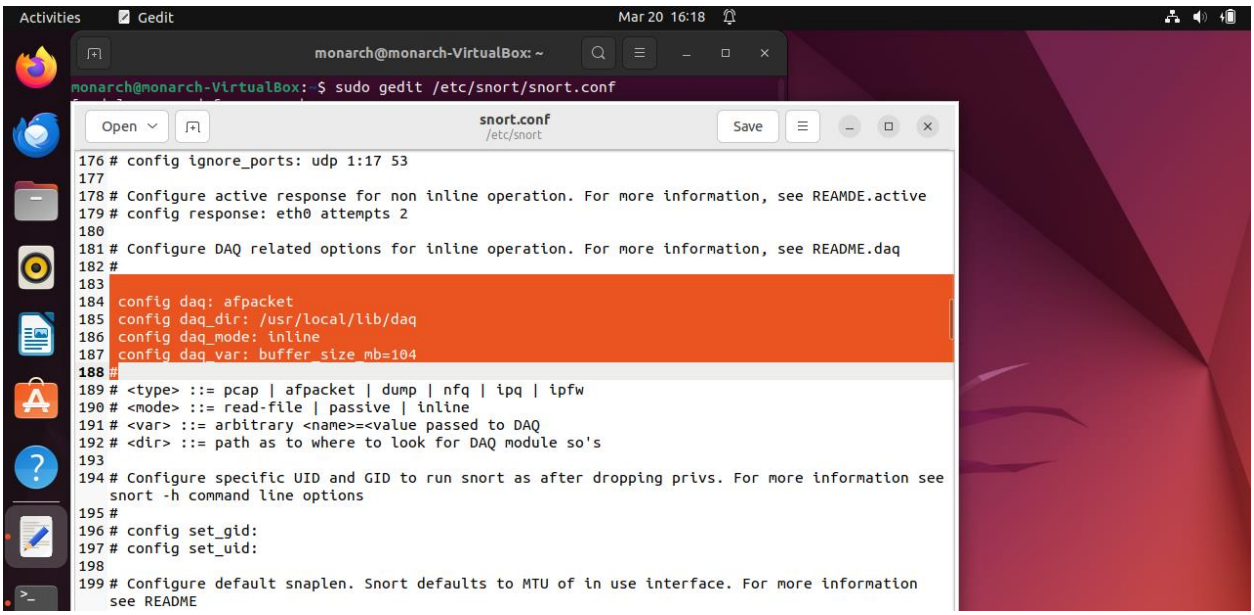


Figure 37 IPS config

The first thing to change for IPS mode is the configuration file of snort, where we need to specify the packet type and set the mode to inline. In IDS mode these lines would be commented out.

3.7.5 Snort IPS rule

IPS and IDS rules in snort are the same, just the action of the rule is changed. In IDS we say alert, but in IPS we can drop or reject the packet. Below is a simple IPS rule for blocking the ICMP ping request.

Figure 38 IPS rule

This is a simple rule to test our IPS to drop any ICMP packets that are coming from any network and any port to any destination network and port.

3.7.6 Running Snort in IPS Mode and Triggering the Rule

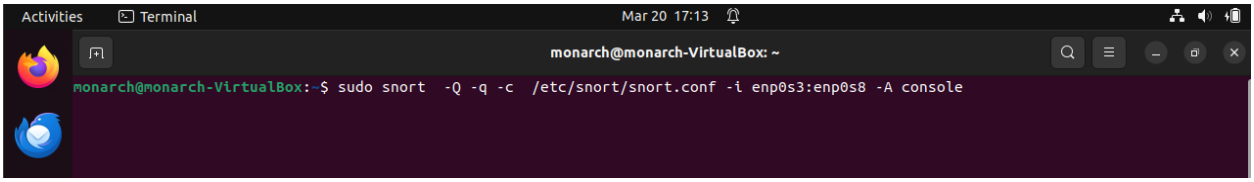


Figure 39 Snort in inline mode

Running IPS is a bit different than IDS, here -Q says snort to run it on inline mode and there are 2 interfaces which is enp0s3 and enp0s8 which are the 2 interfaces of the Ubuntu machine, compared to IDS where there would be only one interface to listen on.

Ping from Kali to Windows 7

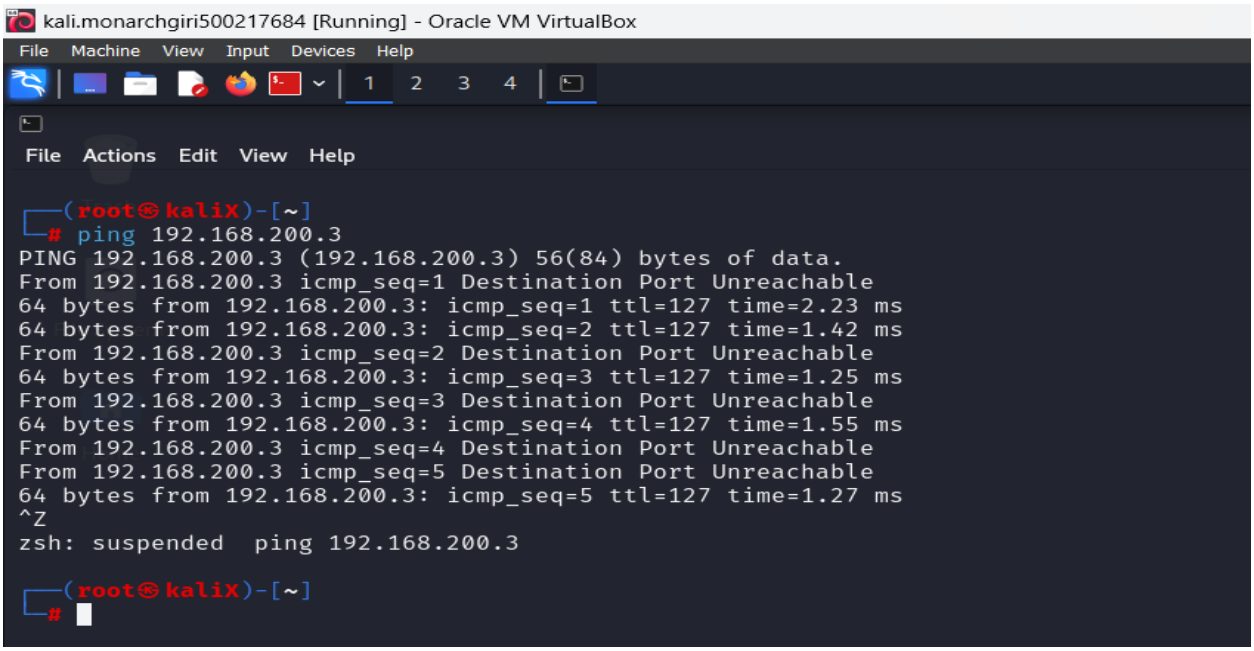


Figure 40 ICMP packet from kali

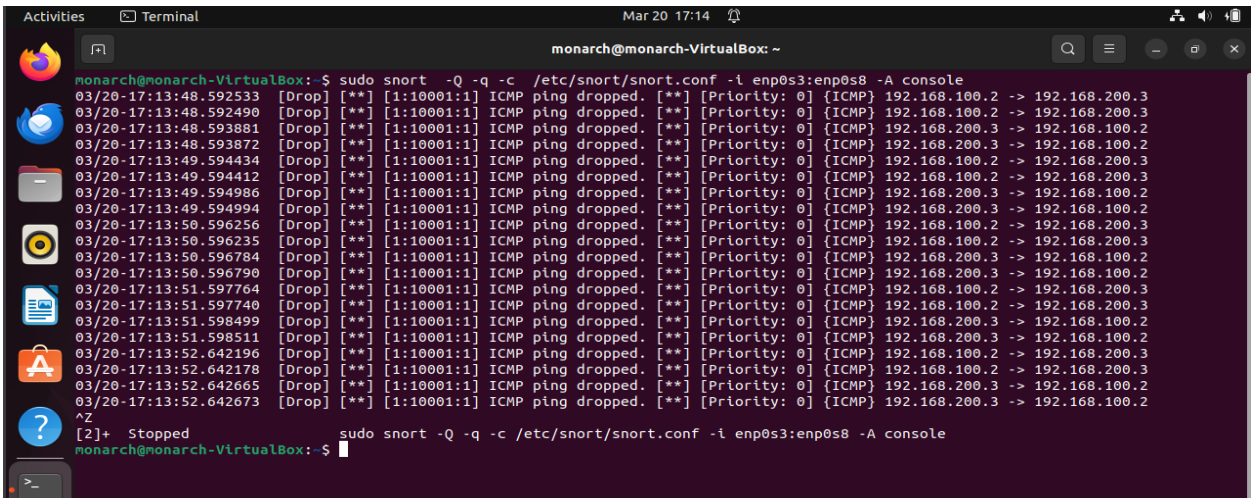


Figure 41 ICMP packet dropped

As we can see in Kali machine it says the destination port is unreachable and the snort also sends the alert in the console that the packet is dropped.

Ping from Windows 7 to Kali

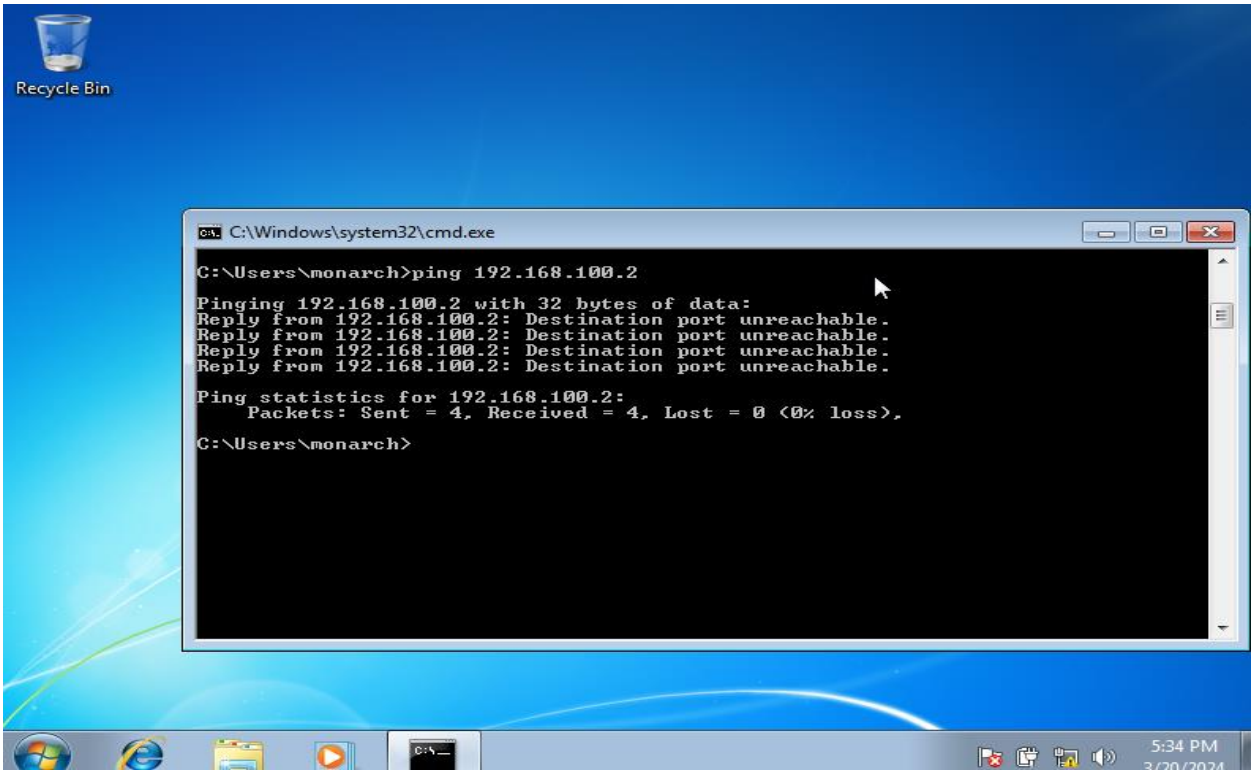


Figure 42 ICMP packets from windows

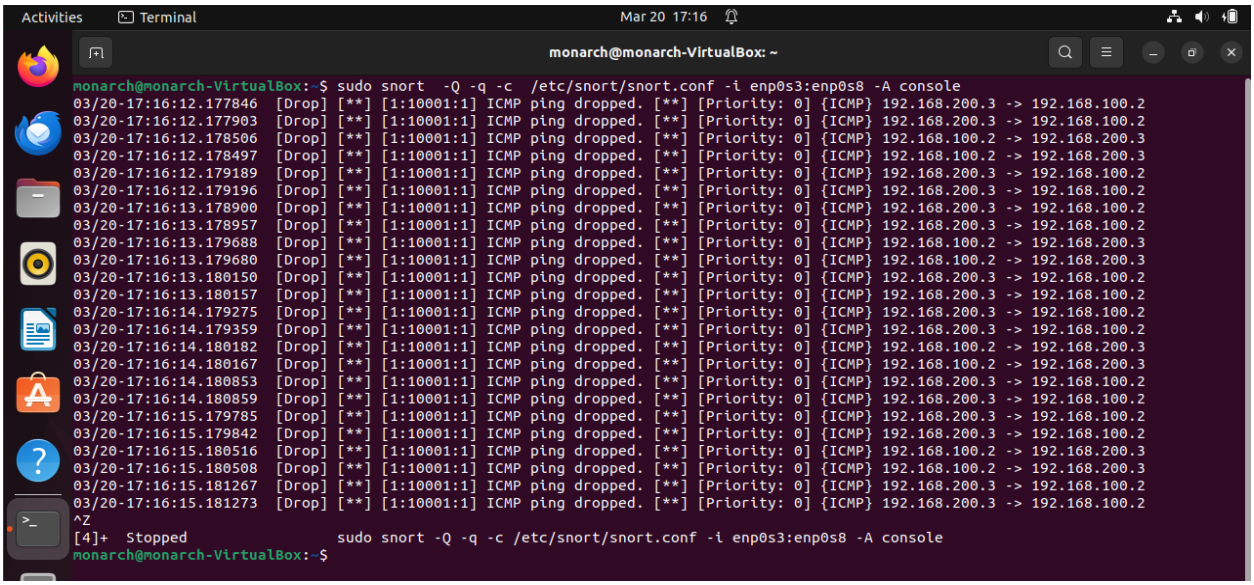


Figure 43 ICMP packet dropped

As we can see in the Windows 7 machine it says the destination port is unreachable and the snort also sends the alert in the console that the packet is dropped.

4. Splunk in Ubuntu and Try Hack Me

Splunk is one of the important tools used in the SOC environment for log analysis and knowing how to navigate through it is very important. Now let's see how we can start using Splunk for our tasks.

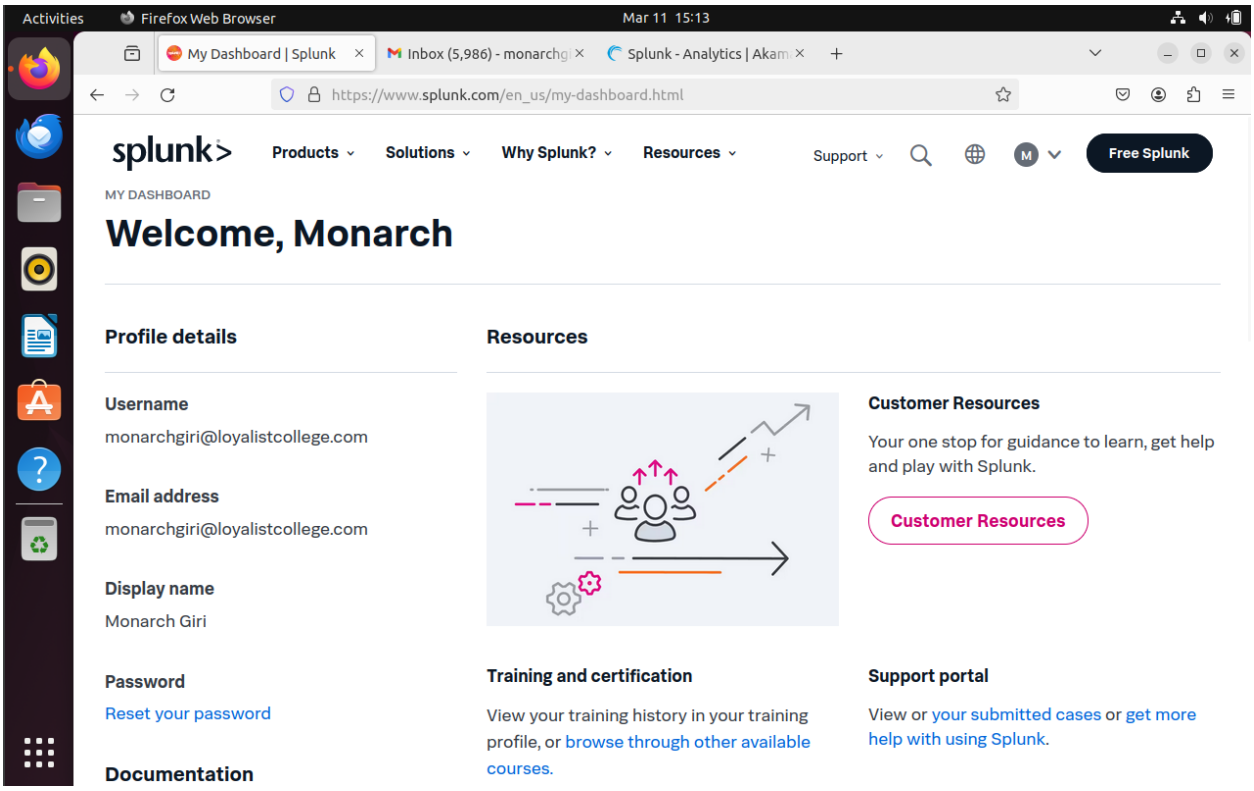


Figure 44 Splunk

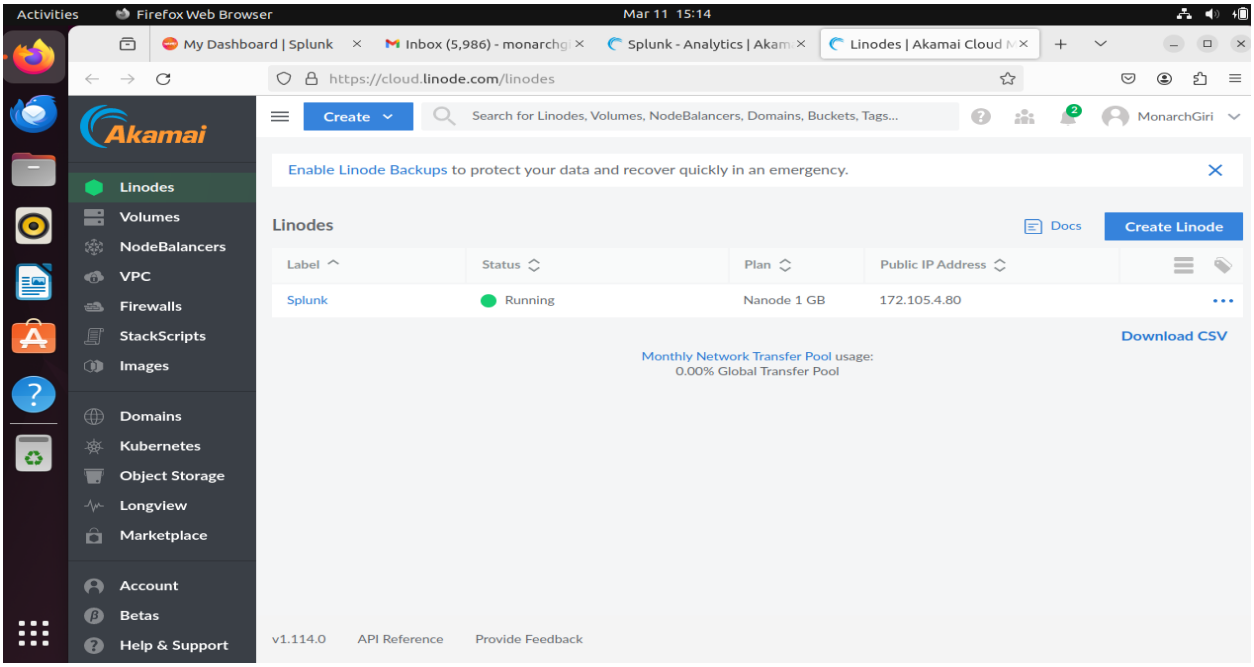


Figure 45 Akamai Linodes 1

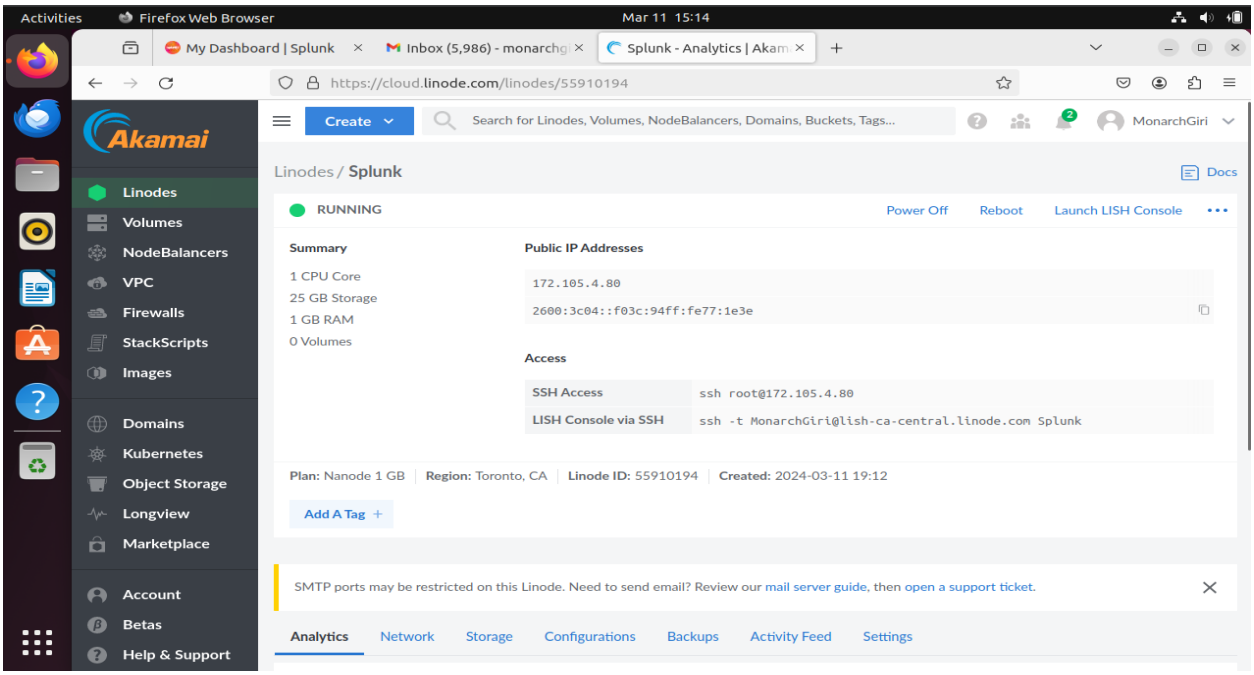


Figure 46 Akamai Linodes 2

So, these are the Akamai Linodes where we can register and use our own Splunk through the IP address displayed above. It is a great way to play around with Splunk and also forward logs through different sources in Splunk and analyzing those logs in one place is much easier.

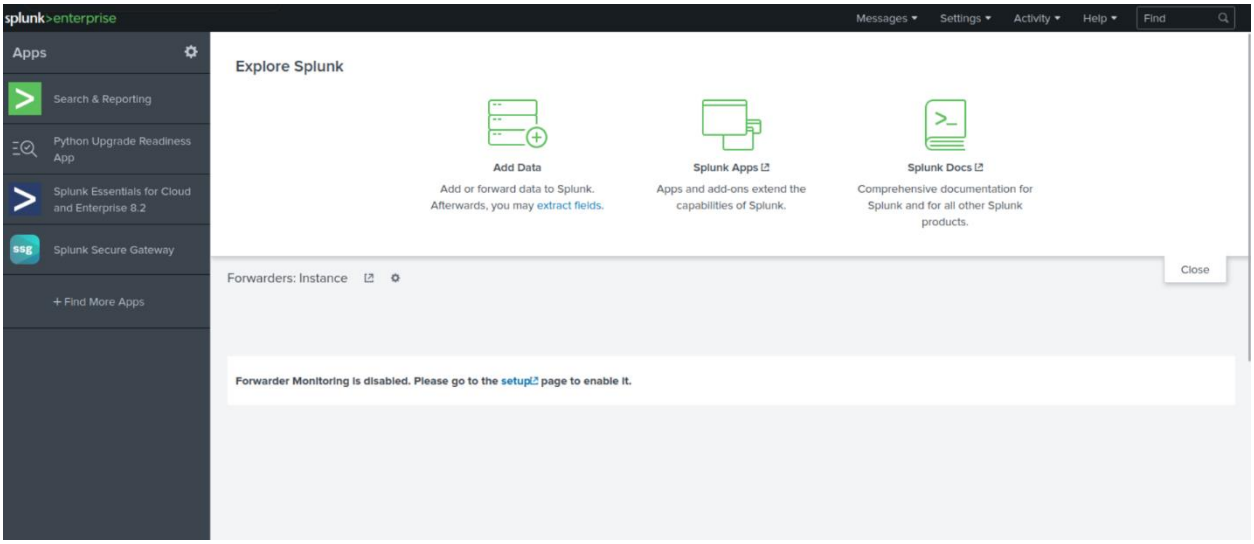


Figure 47 Splunk dashboard

This is the dashboard we can see when we first successfully log into Splunk.

4.1 Try Hack Me

THM is one of the great platforms to learn cybersecurity, I have been using THM, and it has helped me a lot, and learning and practicing skills there is very easy and fun.

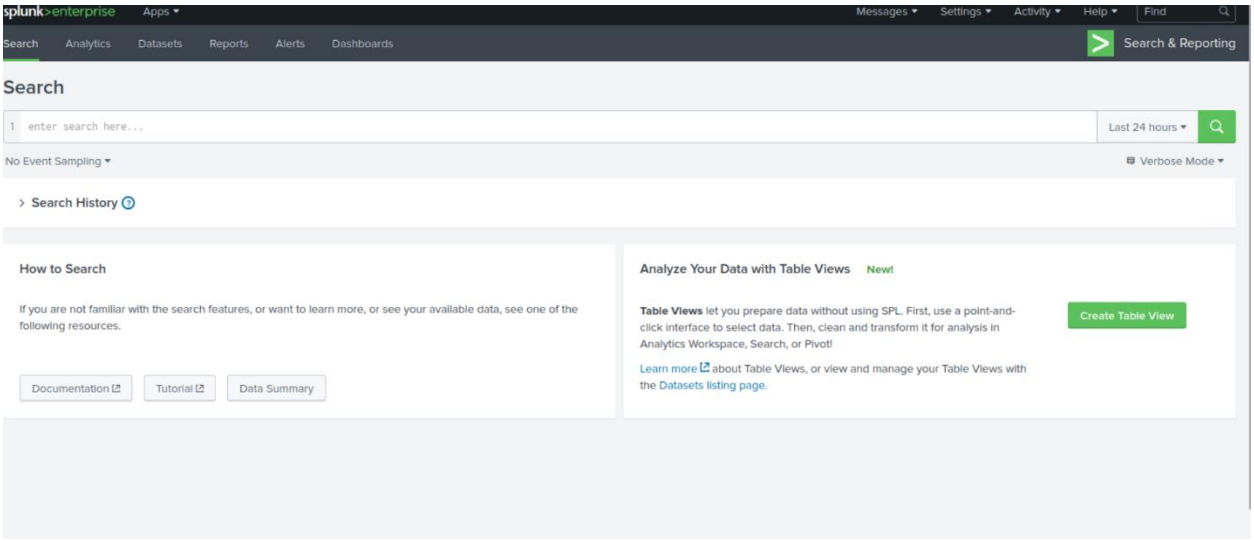
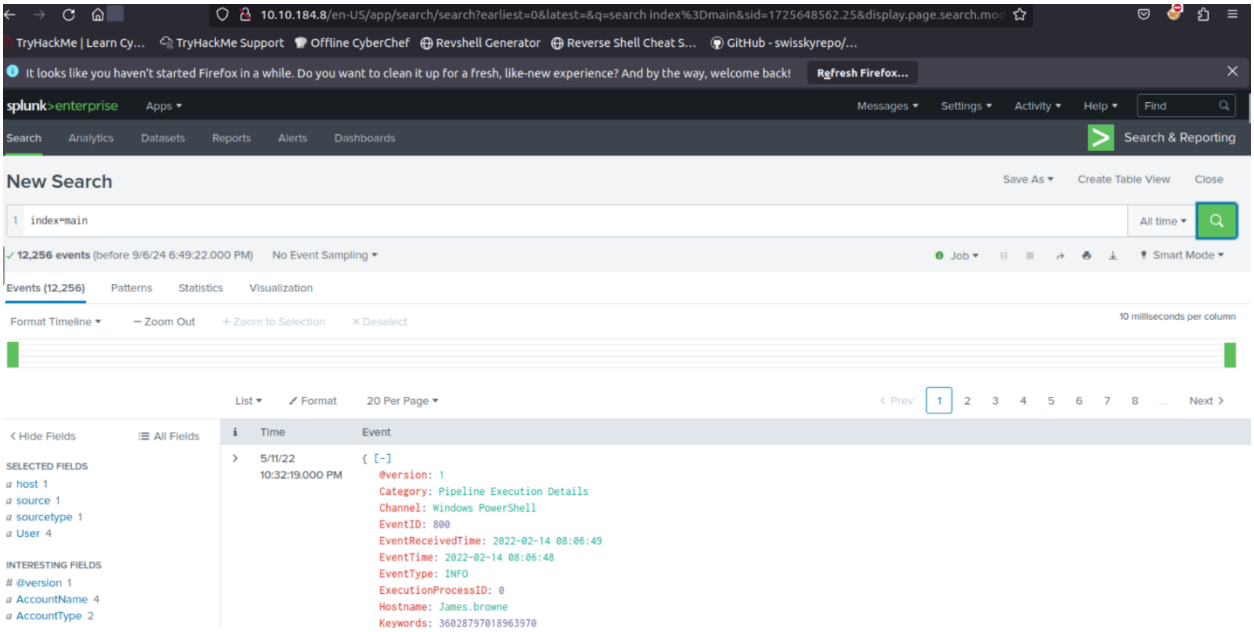
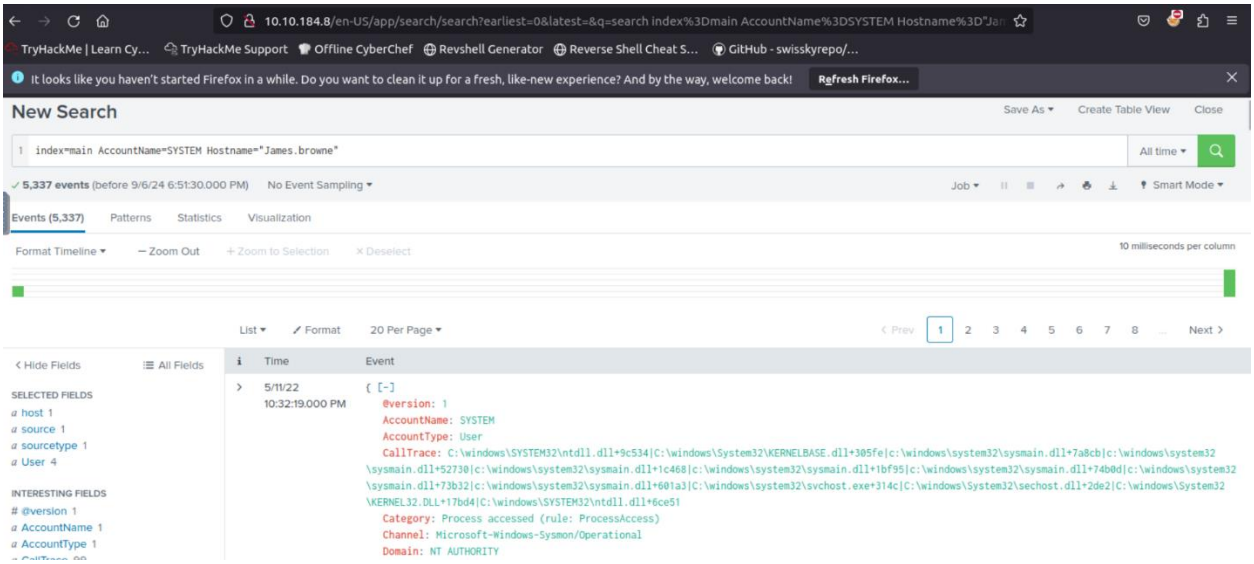


Figure 48 THM Splunk





As we can see Splunk is also in THM, there are tasks to be completed and these are some of the screenshots that I was doing in THM about Splunk and log analysis in Splunk using THM is easy if you don't want to use Linode.

5. Conclusion

In this project, we successfully showed the usage of Snort as an IDS/IPS tool with Splunk for log analysis, offering a thorough picture of network security management. By simulating multiple attacks in a controlled virtual environment, we demonstrated the importance of intrusion detection and prevention in network security. The practical application of Snort rules for identifying malicious activity, combined with Splunk log analysis, demonstrates the importance of these technologies in monitoring and defending against cyber attacks. This study underscores the significance of a layered security approach that combines detection, prevention, and analysis to improve overall cybersecurity posture. Further research might expand on these findings to investigate advanced settings, performance optimization, and the incorporation of additional security measures to improve network resilience.