

DAY 6 — Network Traffic Inspection & Packet Analysis

Today we are using two **tools**: Wireshark + tcpdump.

Mindset: "If it moves on the wire, I can see it."

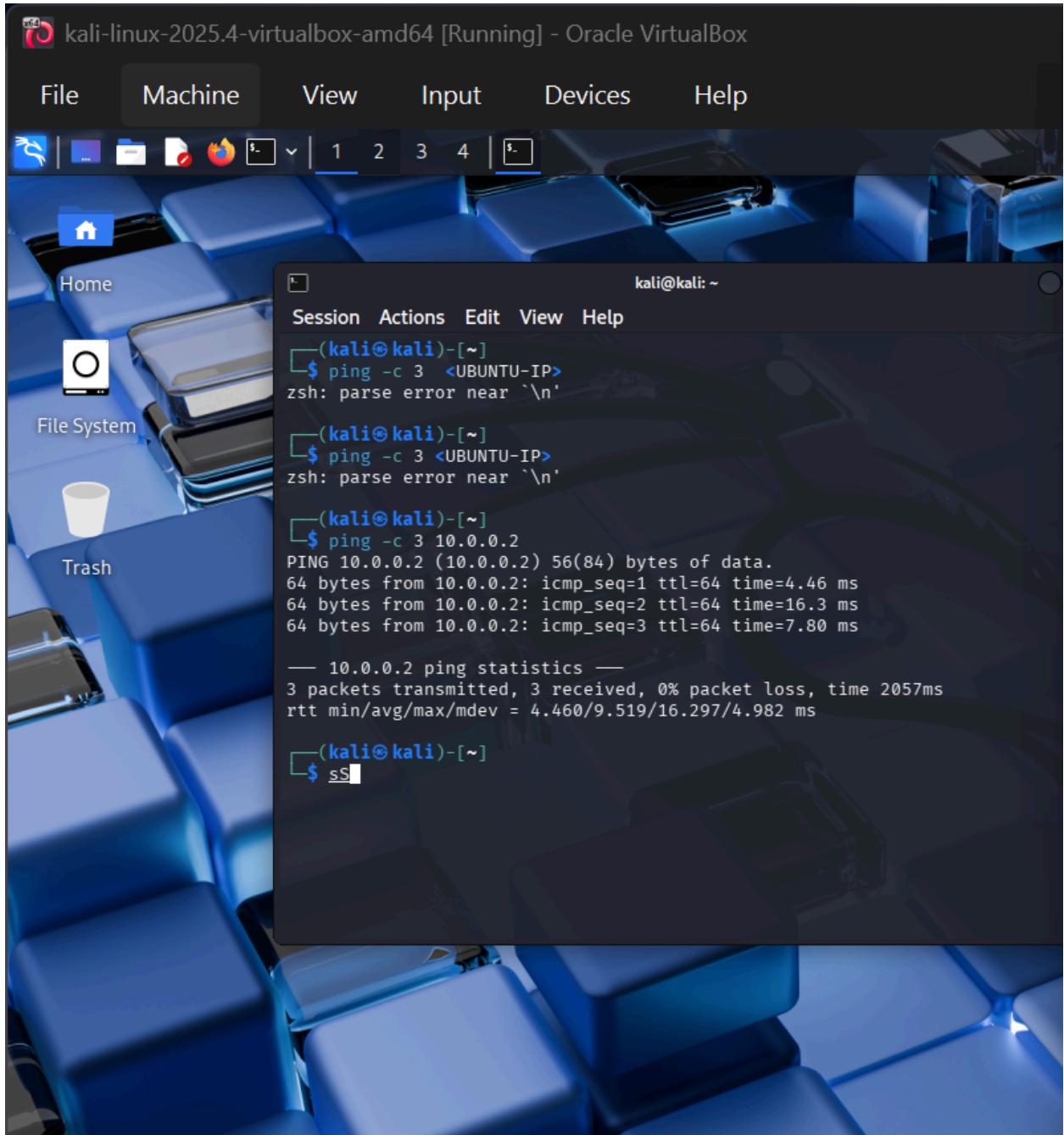
Today, you will learn to use a tool called **Wireshark**. Think of it as an X-ray machine for your network. You will see exactly what a "packet" of data looks like when you visit a website or log in via SSH.

LAB PRECHECK :

Before we start the "Digital X-ray," we must make sure our "Patient" (Ubuntu) and "Doctor" (Kali) are ready.

First of all we have to check our vms kali and ubuntu ,that both are connected to each other, and main important both would have to connect to same networks mainly (internal network)

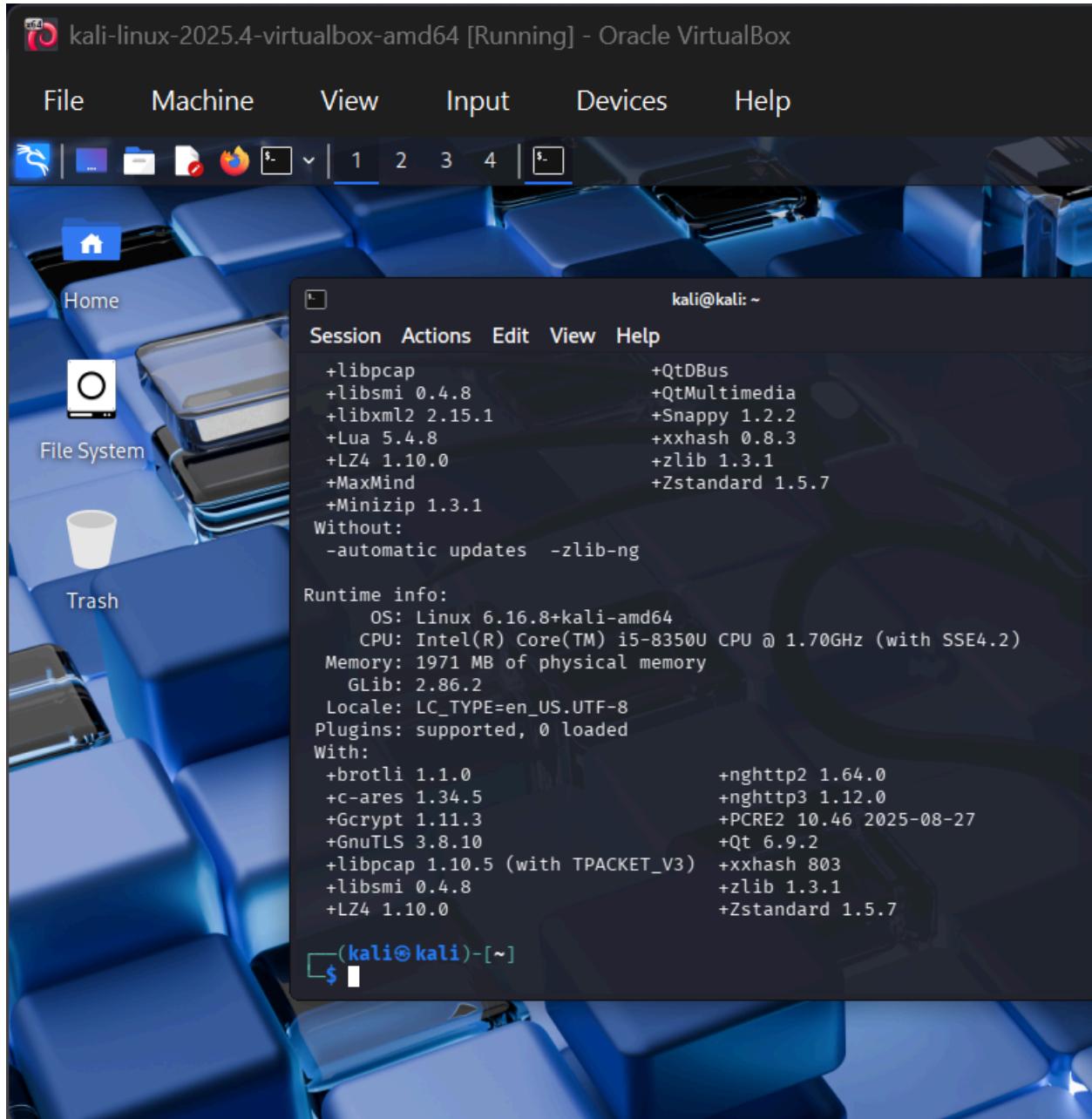
On the kali terminal we have to put this command (ping -c 3 10.0.0.2) to check both are talking to each other.



STEP 1: Install Wireshark (if not already)

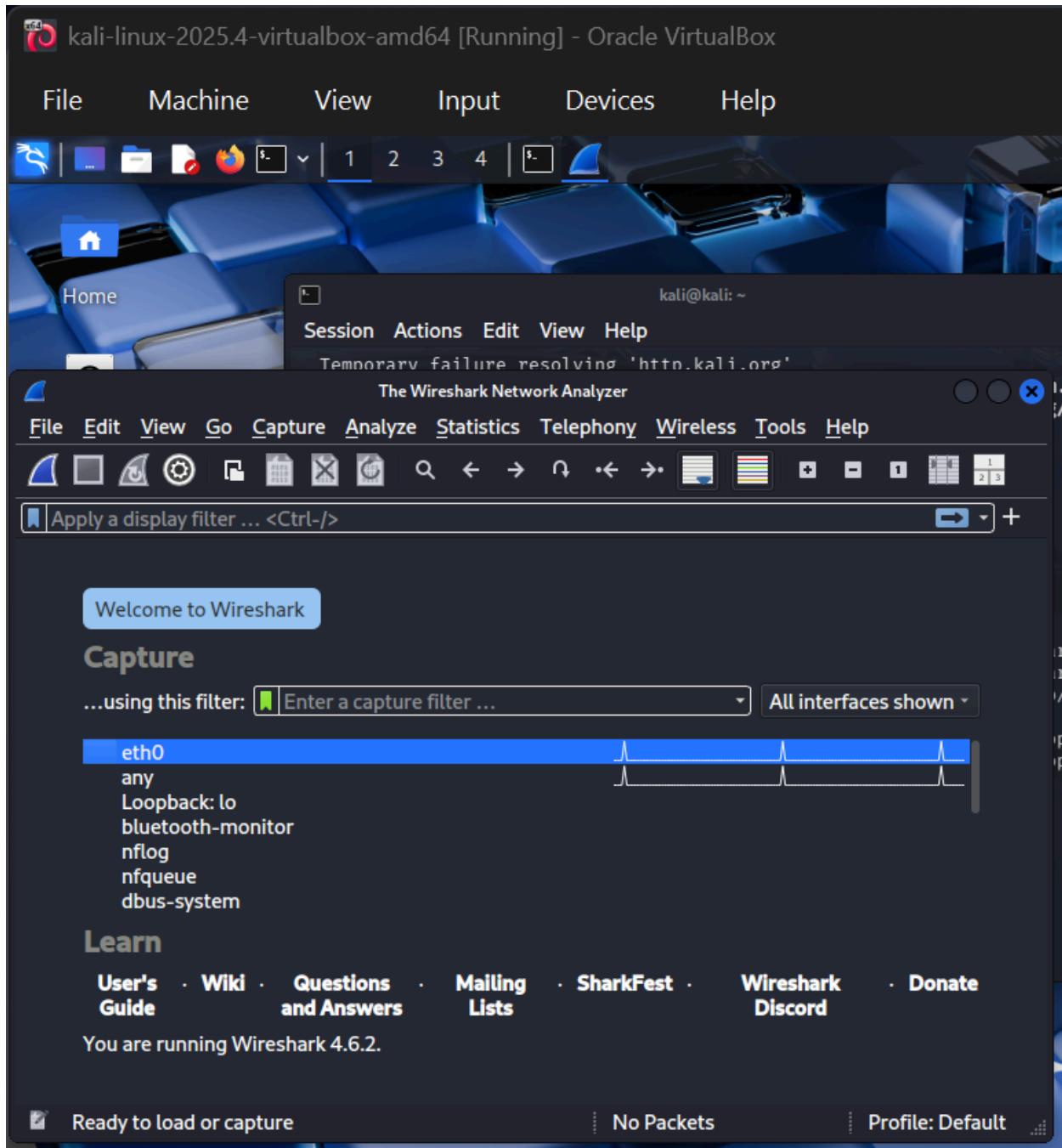
In this step we have to install wireshark on linux .first if linux is on (internal network) we have to provide linux to access of internet ,we will convert internal network to NAT Then we have to enter this command to update (sudo apt update) after that we have install wireshark using this command (sudo apt install wireshark -y) inbetween installation we have to allow to capture root by typing (Yes) if they ask only .after that we have to logout by (ctrl+c) and we will check the status by giving command

(wireshark –version) and if have to open wireshark we can use this command (wireshark&).



STEP 2: Identify the Correct Interface

After opening wireshark we have to watch active interface inform of (eth0)or (any)
Both packets would be moving like heartbeats up and down we have to amin chose eth0

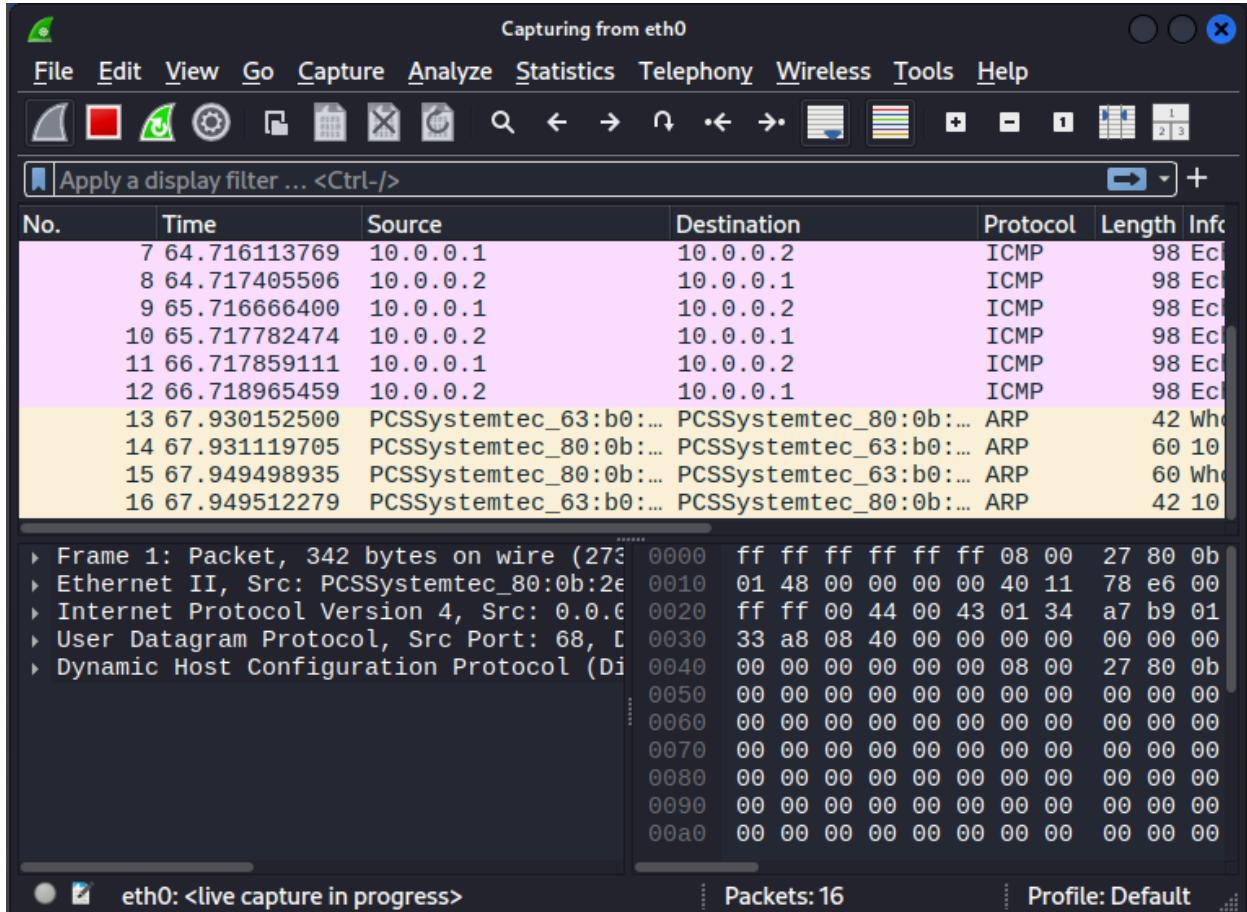


STEP 3: Capture ICMP (Ping Test)

Now we can see two terminal is open one is kali terminal and other is wireshark terminal .now we have to move to to kali terminal and type this command (ping 10.0.0.2) for ping testing

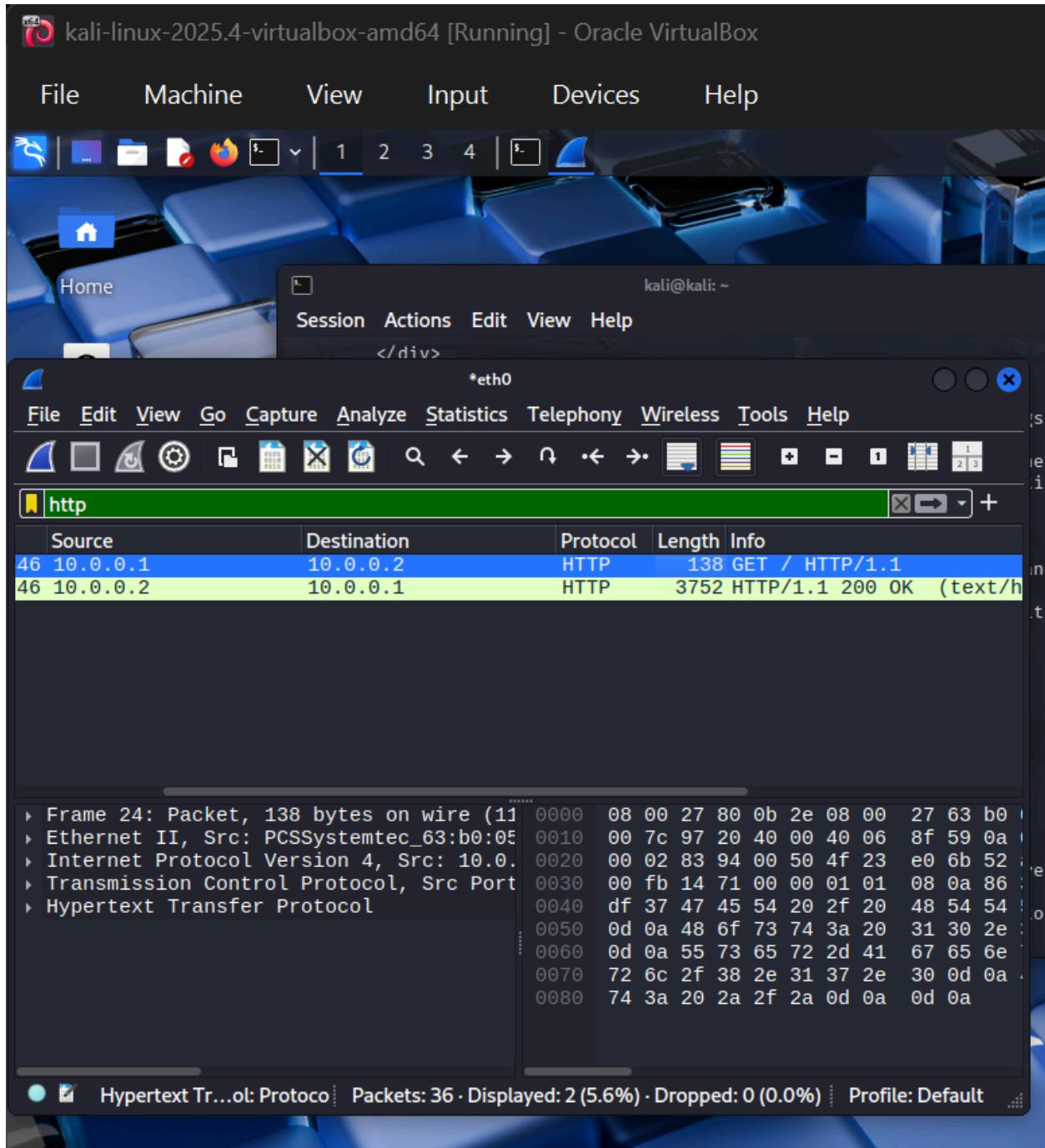
After command we have to move to wireshark terminal and on the above searchbar we have to type icmp now the result we can see in layers .In Wireshark, make sure **eth0** is highlighted.

Go Live: Click the **Blue Shark Fin** icon in the top-left corner.**Stop the Recording:** Go back to Wireshark and click the **Red Square** icon (next to the shark fin).



STEP 4: Capture HTTP Traffic (Plaintext)

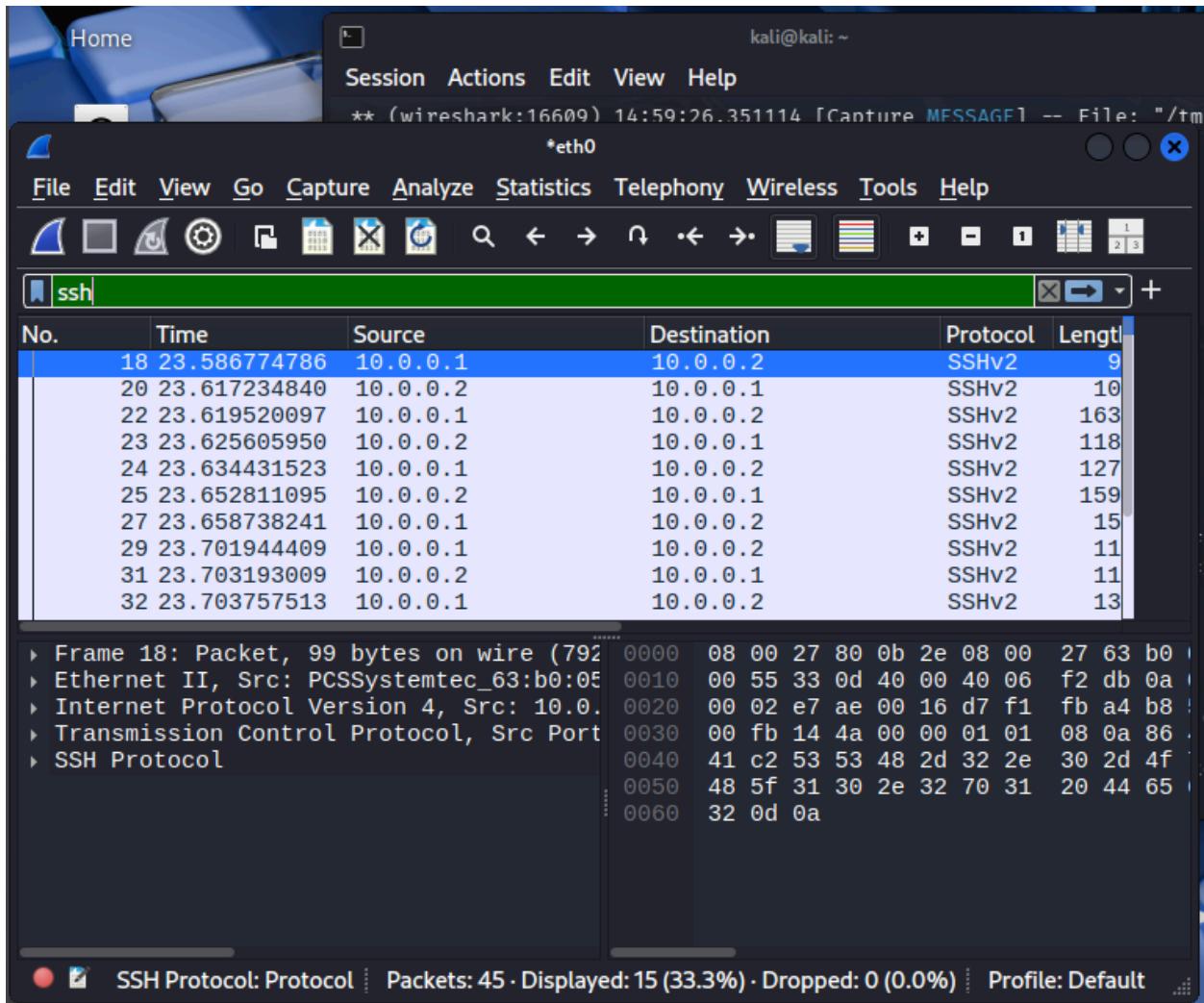
Now we have capture http(hyper text transfer protocol) traffic .on the kali terminal we have to type (curl <http://10.0.0.2>) after this command we have to go back to wireshark terminal , search http then it will show everything .**Stop the Recording:** Go back to Wireshark and click the **Red Square** icon (next to the shark fin).then type http



STEP 5: Capture SSH Traffic (Encrypted)

Now we do same thing that we have done on http but the command will be change , on the kali terminal we will put (ssh testuser@10.0.0.2) after this we have to put wrong or any password for encrypted action appears then have to go back to wireshark terminal and **Stop the**

Recording: Go back to Wireshark and click the **Red Square** icon (next to the shark fin). Then in the search bar type ssh we will see many layers of encrypted ,that is the wrong password we enterend .



STEP 6: Compare & Think (MOST IMPORTANT STEP)

Answer these in your notes:

- Why can I read HTTP but not SSH?
- What layer is encryption applied?

- Why does packet visibility still matter even when encrypted?

1ans : why i can read http(port80) but not ssh(port22) becz the first thing the command we have entered ,it is to see website html and on that time we have not entered in ubuntu terminal

2ans: in which layer I have put an incorrect or correct password that layer is encrypted.

3ans: packet still matters while encrypted ,if anyone (attacker)is doing something with our os ,server ,like attacking we can see that by the wireshark tool.|

STEP 7: (Optional but Powerful) Use tcpdump

See to install tsdpdump we have to use same procedure that we have done before for wireshark Just the command to install we will type (sudo apt install tcpdump -y) but it should be install in ubuntu terminal not kali.

Now we are doing (The Terminal Sniffer) on ubuntu ,we are going to prepare the Ubuntu "Listener" the command we are using in this is (sudo tcpdump -i any port 22 or port 80)

Explanation: -i any means listen on every wire; port 22 is SSH; port 80 is HTTP.

Press Enter. The terminal will say: `tcpdump: listening on any...` It is now waiting for a signal.Trigger from Kali (The "Sender")**Go to your Kali Linux VM.**

Trigger HTTP: Type `curl http://10.0.0.2` and press Enter.

Trigger SSH: Type `ssh 10.0.0.2` and press Enter (you don't even need to log in, just start the connection).

Go back to Ubuntu. You will see lines of text flying across the screen!

Stop it: Press Ctrl + C on your keyboard to stop the capture.

Look for the names:

- Look for lines containing `.http` or `.80`.

- Look for lines containing `.ssh` or `.22`.

```

Ubuntu Server [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

20:17:11.198972 enp0s3 Out IP target-server.http > 10.0.0.1.58606: Flags [P.], seq 7241:10927, ack 73, win 509, options [nop 5], length 3686: HTTP
20:17:11.200036 enp0s3 In IP 10.0.0.1.58606 > target-server.http: Flags [.], ack 10927, win 337, options [nop,nop,TS val 225]
20:17:11.200533 enp0s3 In IP 10.0.0.1.58606 > target-server.http: Flags [F.], seq 73, ack 10927, win 337, options [nop,nop,TS val 0]
20:17:11.201282 enp0s3 Out IP target-server.http > 10.0.0.1.58606: Flags [F.], seq 10927, ack 74, win 509, options [nop,nop,TS val 0]
20:17:11.202414 enp0s3 In IP 10.0.0.1.58606 > target-server.http: Flags [.], ack 10928, win 337, options [nop,nop,TS val 225]
20:17:17.933141 enp0s3 In IP 10.0.0.1.59310 > target-server.ssh: Flags [F.], seq 0, ack 1, win 252, options [nop,nop,TS val 0]
20:17:31.457405 enp0s3 In IP 10.0.0.1.47356 > target-server.ssh: Flags [S.], seq 3221400963, win 64240, options [mss 1460,sack 1, ts val 8], length 0
20:17:31.457523 enp0s3 Out IP target-server.ssh > 10.0.0.1.47356: Flags [S.], seq 2570461993, ack 3221400964, win 65160, options [mss 1460,sack 1, ts val 943, ecr 2254067719,nop,wscale 7], length 0
20:17:31.458405 enp0s3 In IP 10.0.0.1.47356 > target-server.ssh: Flags [.], ack 1, win 251, options [nop,nop,TS val 2254067719]
20:17:31.459462 enp0s3 In IP 10.0.0.1.47356 > target-server.ssh: Flags [P.], seq 1:34, ack 1, win 251, options [nop,nop,TS val 0]
20:17:31.459516 enp0s3 Out IP target-server.ssh > 10.0.0.1.47356: Flags [.], ack 34, win 509, options [nop,nop,TS val 1103371]
20:17:31.485133 enp0s3 Out IP target-server.ssh > 10.0.0.1.47356: Flags [P.], seq 1:44, ack 34, win 509, options [nop,nop,TS val 0]
20:17:31.489119 enp0s3 In IP 10.0.0.1.47356 > target-server.ssh: Flags [.], ack 44, win 251, options [nop,nop,TS val 2254067719]
20:17:31.489120 enp0s3 In IP 10.0.0.1.47356 > target-server.ssh: Flags [P.], seq 34:1602, ack 44, win 251, options [nop,nop,TS val 0]
20:17:31.490739 enp0s3 Out IP target-server.ssh > 10.0.0.1.47356: Flags [P.], seq 44:1164, ack 1602, win 497, options [nop,nop,TS val 0], length 1120
20:17:31.505351 enp0s3 In IP 10.0.0.1.47356 > target-server.ssh: Flags [P.], seq 1602:2810, ack 1164, win 258, options [nop,nop,TS val 0], length 1208
20:17:31.533660 enp0s3 Out IP target-server.ssh > 10.0.0.1.47356: Flags [P.], seq 1164:2696, ack 2810, win 500, options [nop,nop,TS val 0], length 1532
20:17:31.535596 enp0s3 In IP 10.0.0.1.47356 > target-server.ssh: Flags [.], ack 2696, win 253, options [nop,nop,TS val 2254067719]
20:17:31.540443 enp0s3 In IP 10.0.0.1.47356 > target-server.ssh: Flags [P.], seq 2810:2894, ack 2696, win 253, options [nop,nop,TS val 0], length 84
20:17:31.582175 enp0s3 Out IP target-server.ssh > 10.0.0.1.47356: Flags [.], ack 2894, win 500, options [nop,nop,TS val 1103371]
20:17:31.583196 enp0s3 In IP 10.0.0.1.47356 > target-server.ssh: Flags [P.], seq 2894:2938, ack 2696, win 253, options [nop,nop,TS val 0], length 44
20:17:31.583311 enp0s3 Out IP target-server.ssh > 10.0.0.1.47356: Flags [.], ack 2938, win 500, options [nop,nop,TS val 1103371]
20:17:31.583614 enp0s3 Out IP target-server.ssh > 10.0.0.1.47356: Flags [P.], seq 2696:2740, ack 2938, win 500, options [nop,nop,TS val 0], length 44
20:17:31.584710 enp0s3 In IP 10.0.0.1.47356 > target-server.ssh: Flags [P.], seq 2938:2998, ack 2740, win 253, options [nop,nop,TS val 0], length 60
20:17:31.598119 enp0s3 Out IP target-server.ssh > 10.0.0.1.47356: Flags [P.], seq 2740:3004, ack 2998, win 500, options [nop,nop,TS val 0], length 264
20:17:31.647535 enp0s3 In IP 10.0.0.1.47356 > target-server.ssh: Flags [.], ack 3004, win 252, options [nop,nop,TS val 2254067719]
20:17:46.099248 enp0s3 In IP 10.0.0.1.59310 > target-server.ssh: Flags [F.], seq 0, ack 1, win 252, options [nop,nop,TS val 0]
^C
42 packets captured
42 packets received by filter
0 packets dropped by kernel
dawood@target-server:~$
```

. Connectivity & Troubleshooting

Command	Where to Type	What it Does

<code>ping -c 4 10.0.0.2</code>	Kali	Checks if the "bridge" to Ubuntu is open.
<code>nc -zv 10.0.0.2 22</code>	Kali	Checks if the SSH "door" (Port 22) is open.
<code>sudo systemctl restart ssh</code>	Ubuntu	Wakes up the SSH service if it's sleeping.
<code>sudo ufw allow 22/tcp</code>	Ubuntu	Tells the firewall to let SSH traffic pass through.

2. Generating Traffic

Command	Where to Type	What it Does
<code>curl http://10.0.0.2</code>	Kali	Grabs the website data (creates HTTP traffic).
<code>ssh testuser@10.0.0.2</code>	Kali	Starts a secure connection (creates SSH traffic).

3. Packet Capturing (CLI)

Command	Where to Type	What it Does
sudo tcpdump -i any	Ubuntu	Starts sniffing all traffic on the server.
sudo tcpdump -i any port 80	Ubuntu	Sniffs ONLY website (HTTP) traffic.

"Golden" Command Sheet

Action	Command	Result
Test	ping -c 4 10.0.0.2	Confirms connection.
Sniff (GUI)	wireshark	Visualizes the packets.
Trigger HTTP	curl http://10.0.0.2	Generates insecure traffic.
Trigger SSH	ssh 10.0.0.2	Generates secure traffic.
Sniff (CLI)	sudo tcpdump -i any	Professional server-side capture.

