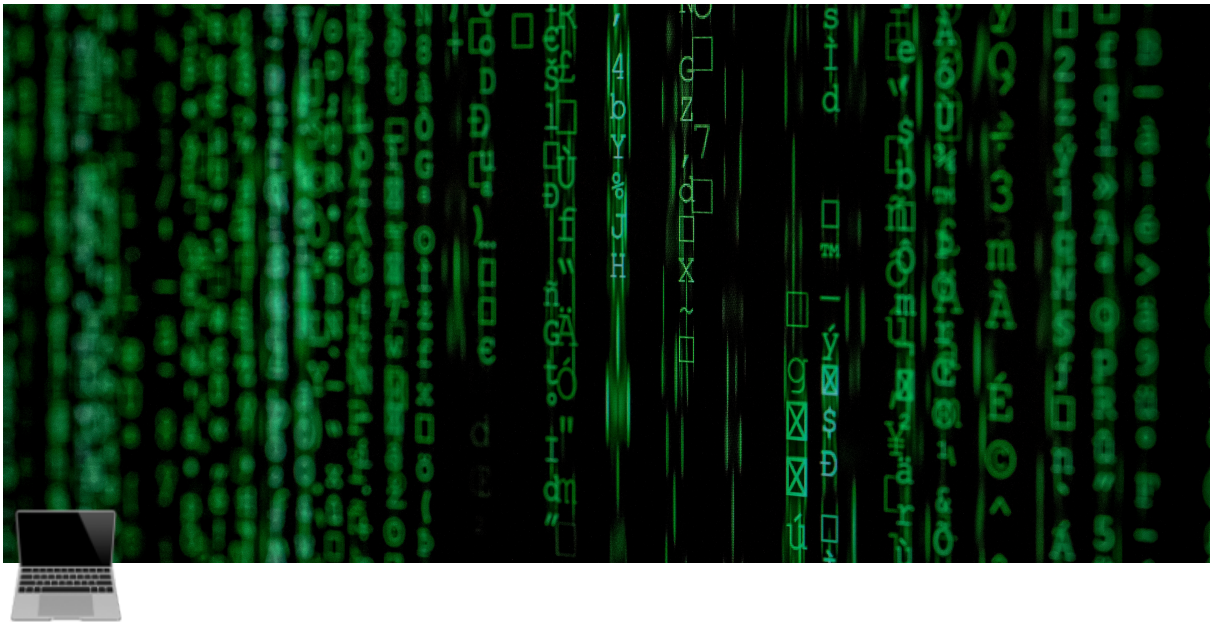


Introduction - Cybersecurity

TP1 - Traffic Sniffing with Wireshark

Objectives

- Learn how to configure a virtual network.
- Learn how to filter data in Wireshark.
- Discover vulnerabilities in Telnet and HTTP services.

---

1. Virtualization Environment Setup

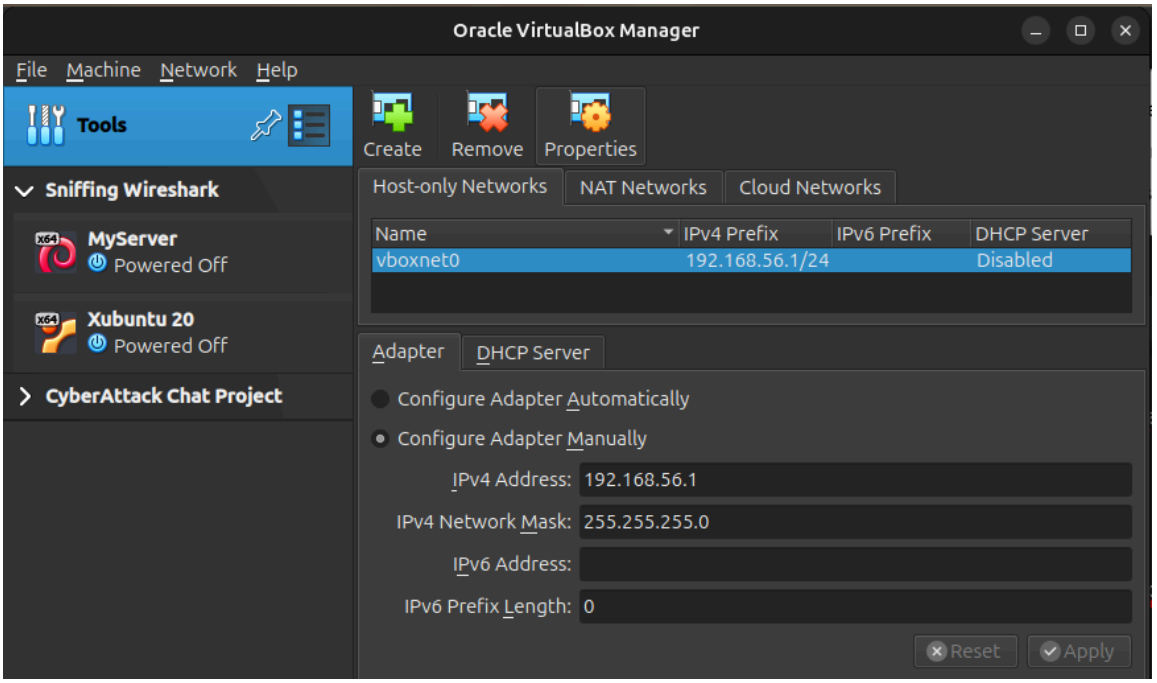
1. Download and install the latest version of **VirtualBox**.
2. Download and install **Oracle VM VirtualBox Extension Pack**.

---

2. Virtual Machine Preparation

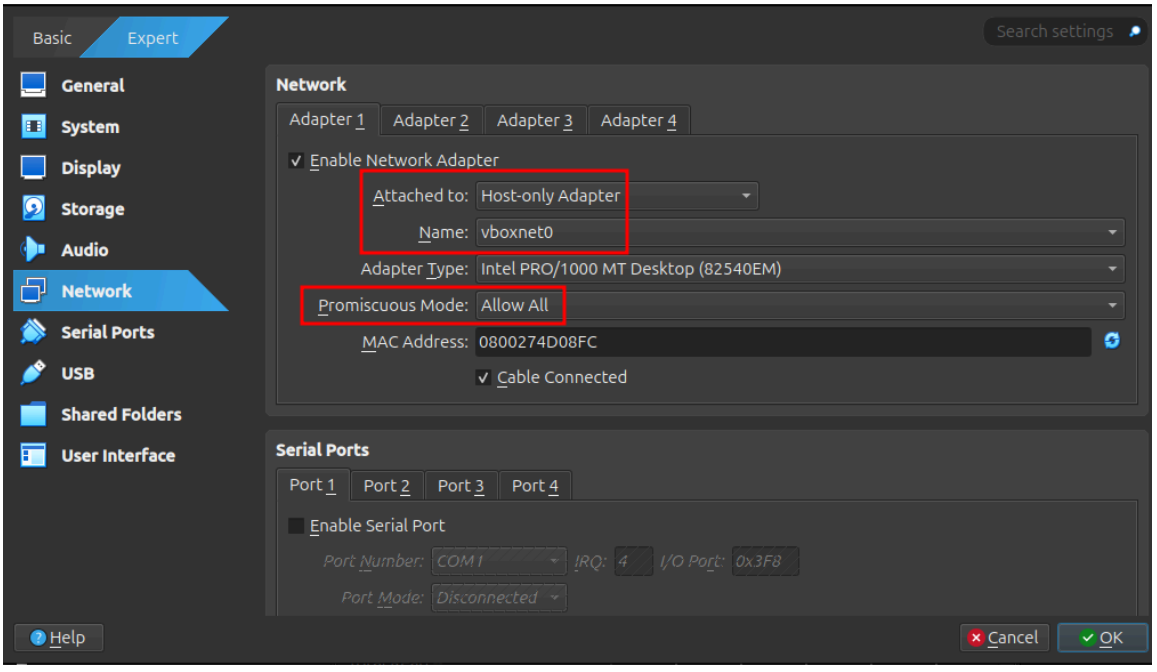
1. Download the two virtual machines **"Student.ova"** and **"MyServer.ova"** from this platform.
2. Import these virtual machines into VirtualBox.
3. Discuss with your instructor how to set up the virtual network.

File > Tools > Network Manager:



{VM name} > Settings > Network:

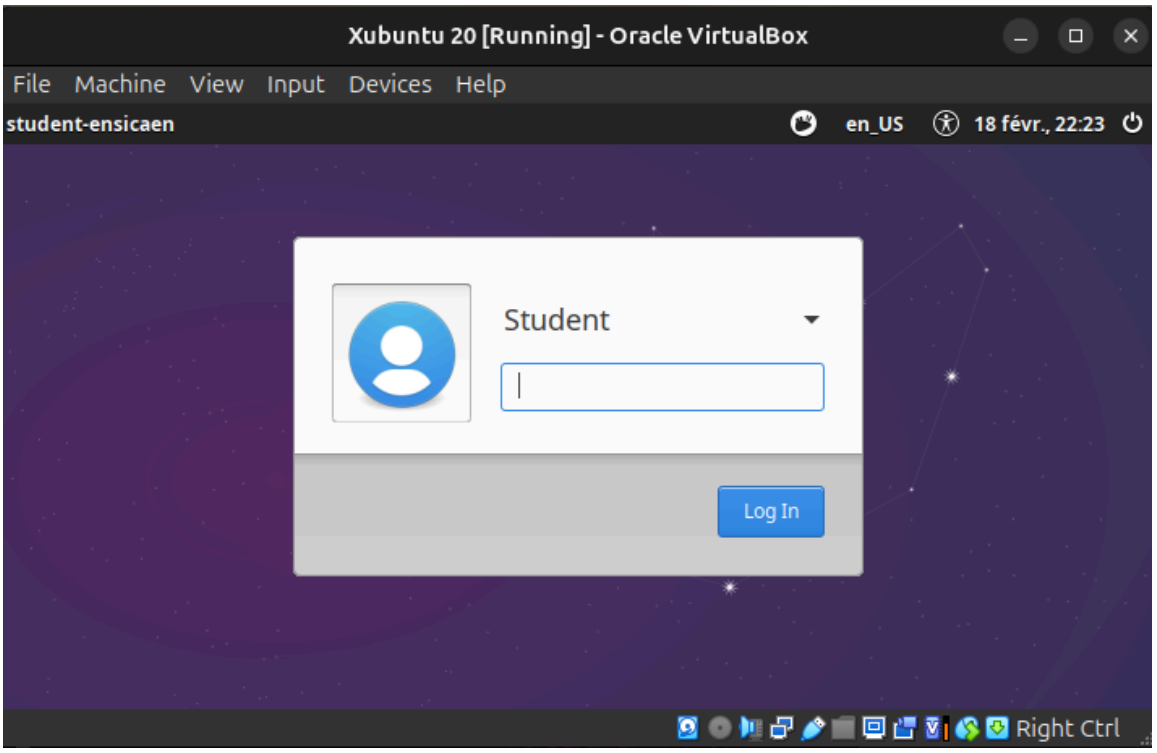
Do these configurations on both of the machines. This allow us to to use the network we created which includes the host, making it possible the sniffing by the host.



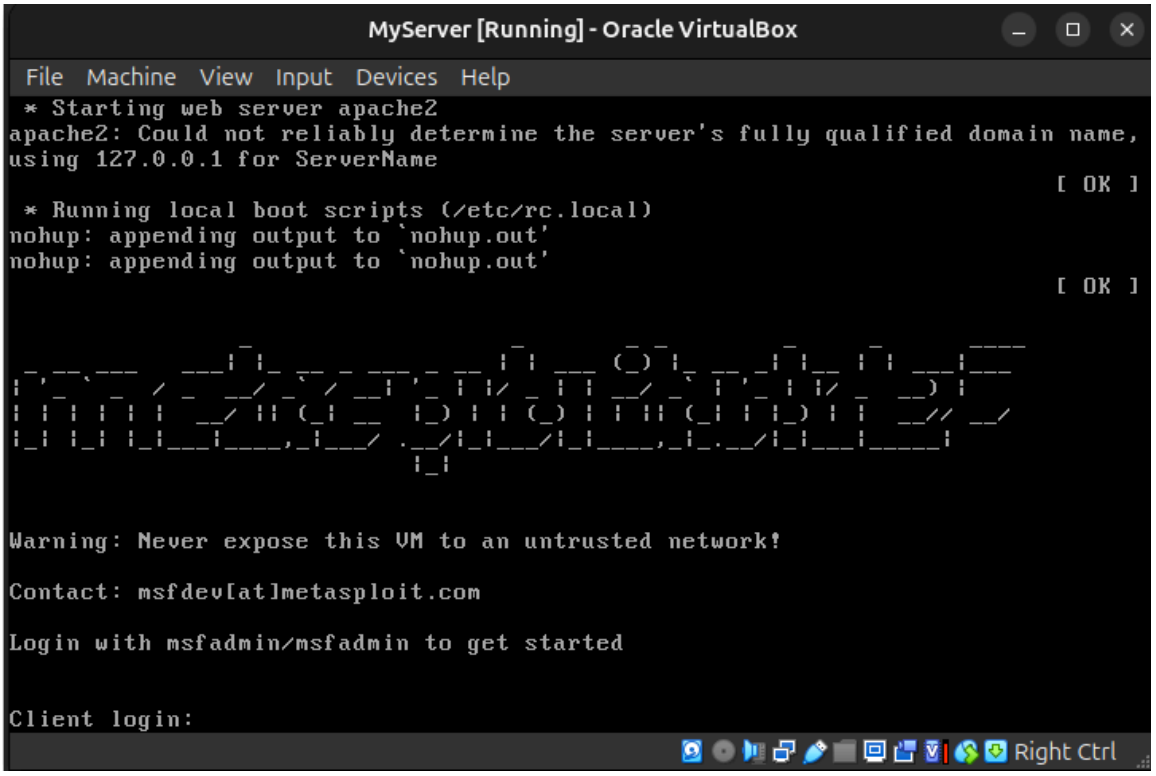
Host’s terminal:

```
sudo ip link set vboxnet0 promisc on # before starting VMs
```

4. Start the **"Student" VM** and log in with the credentials: **student | student**.



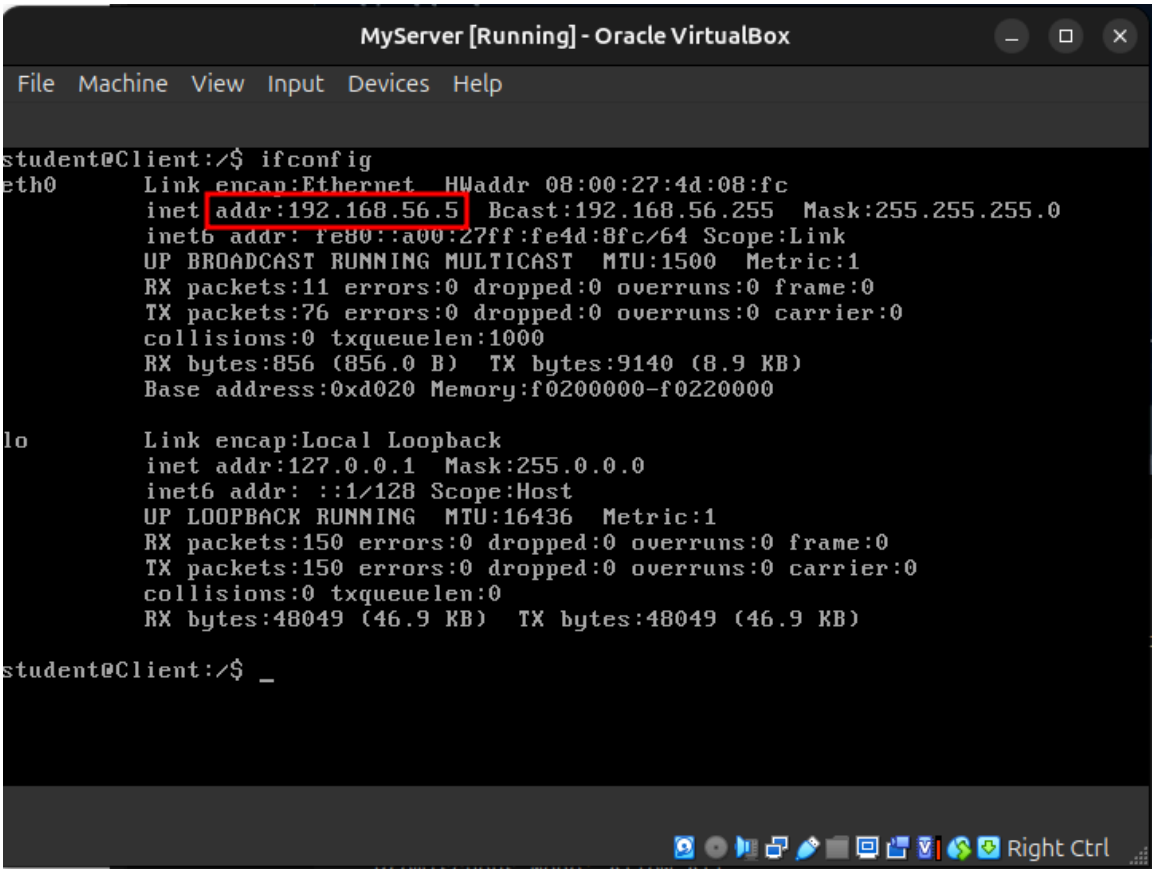
5. Start the **"MyServer" VM** and log in with: **student | ensicaen**.



6. Assign an **IP address** to each virtual machine.

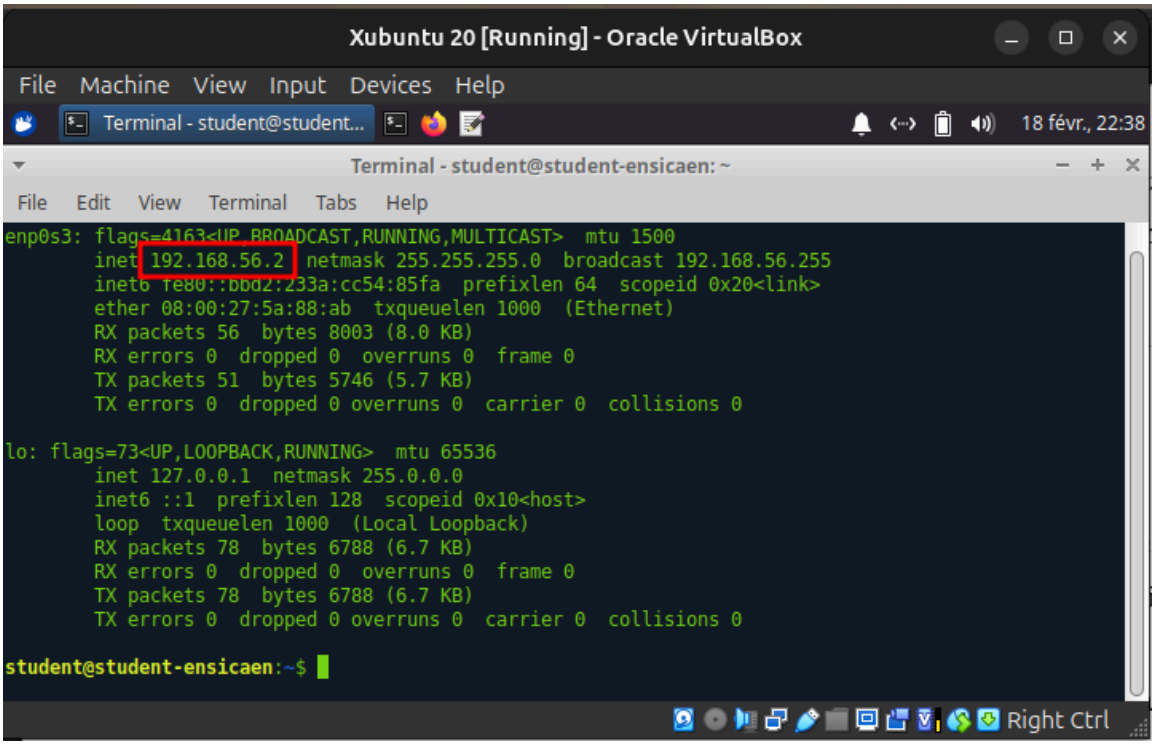
Server's terminal:

```
sudo nano /etc/network/interfaces # add new ip based on our network
```



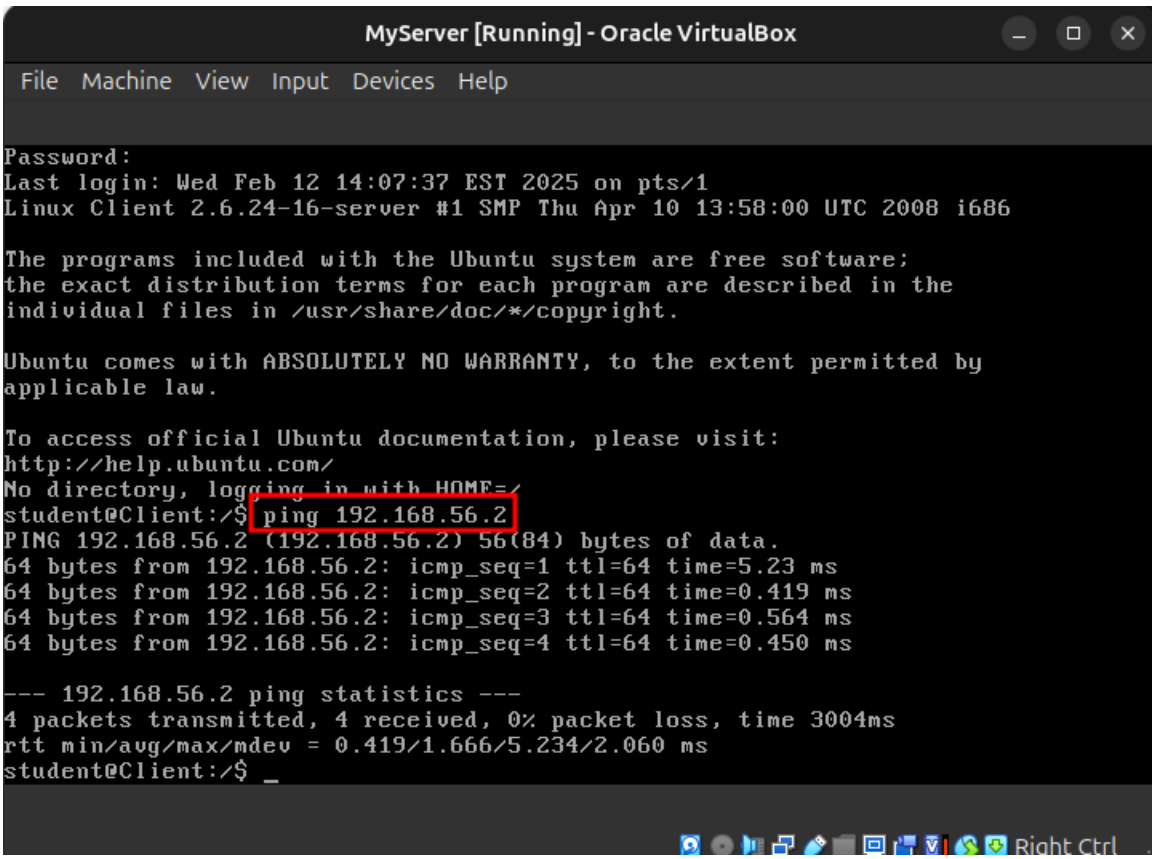
Student's terminal:

```
sudo ip addr flush dev enp0s3 # clean up old ip
sudo ip addr add 192.168.56.2/24 dev enp0s3 # adding new ip/netmask ba
```



7. Verify connectivity between the two VMs.
   - The **MyServer VM** should **not** have access to the internet.

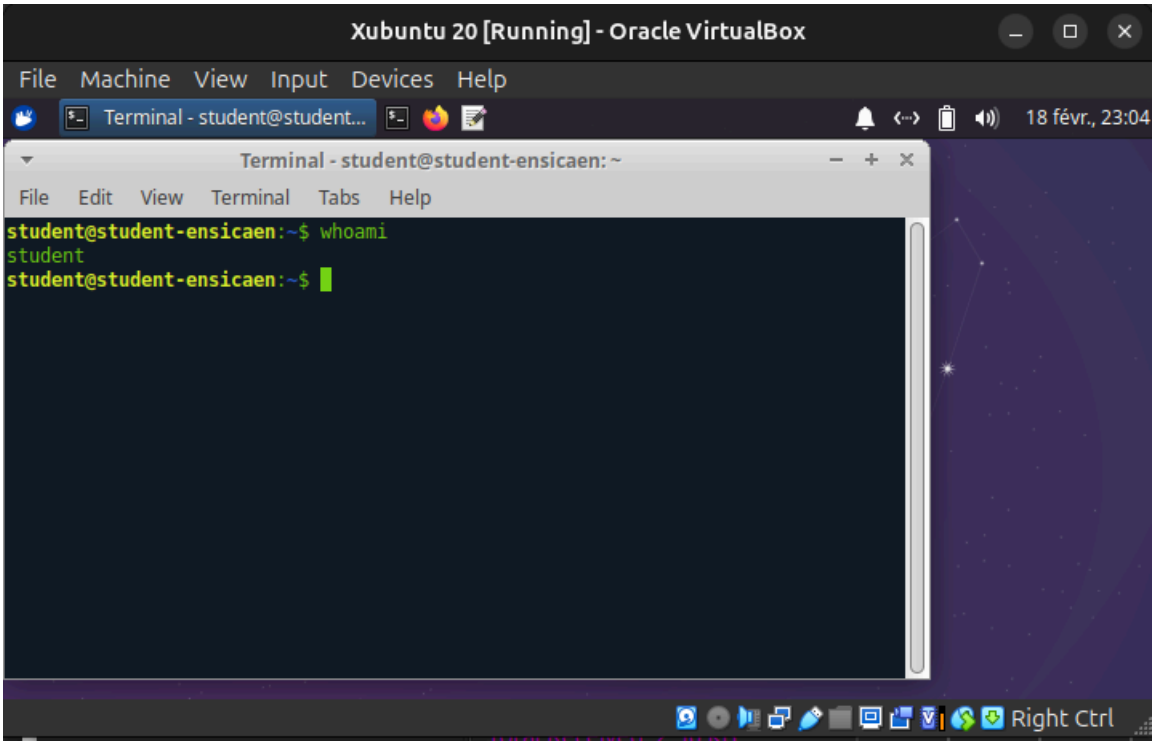
Doing ping from server to student ✅



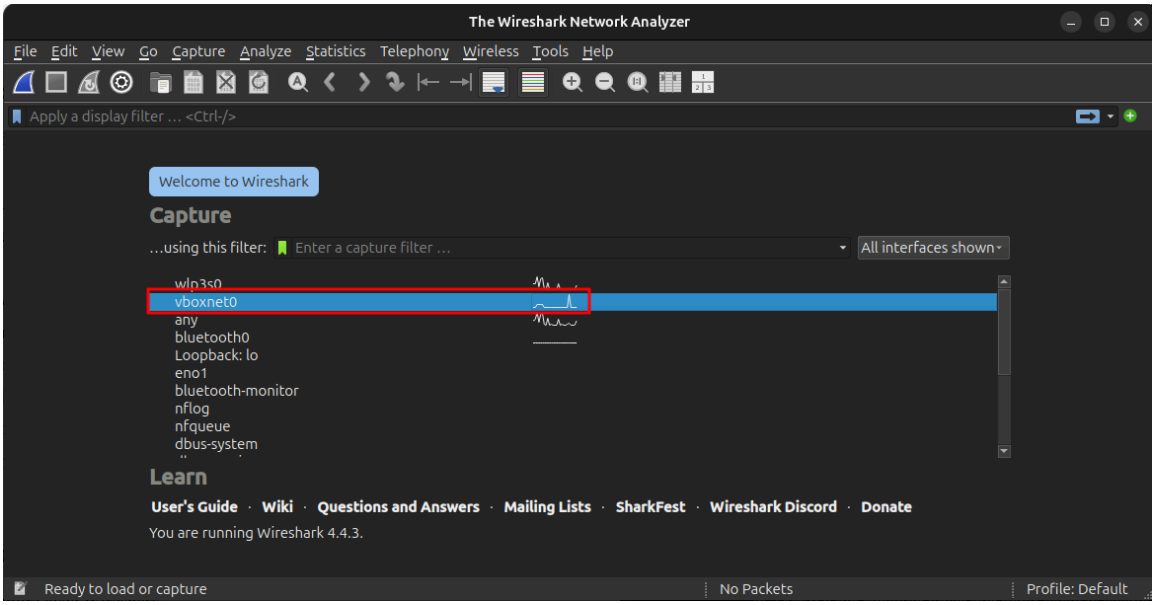
3. Capturing Telnet Passwords

1. On the **Student** machine, open a terminal and run the command:

```
whoami
```

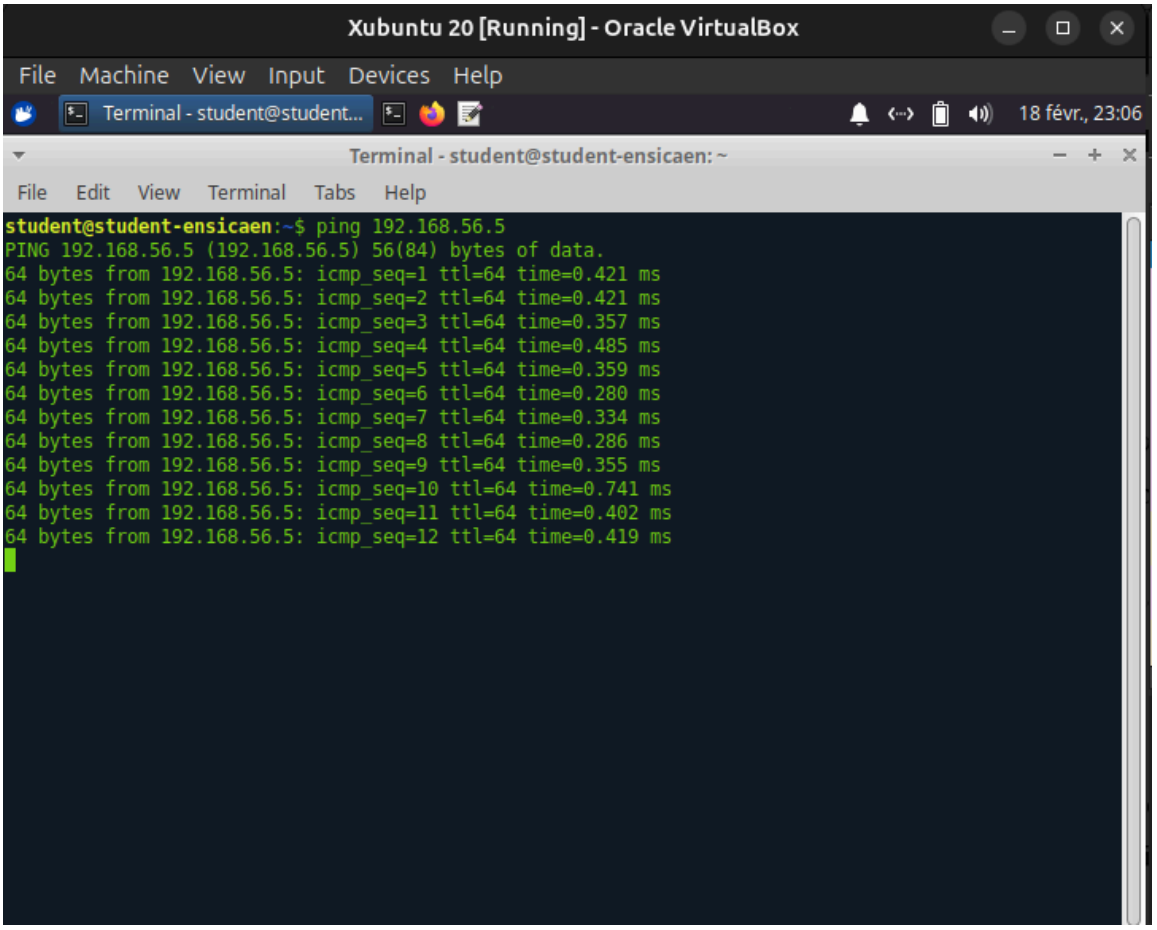


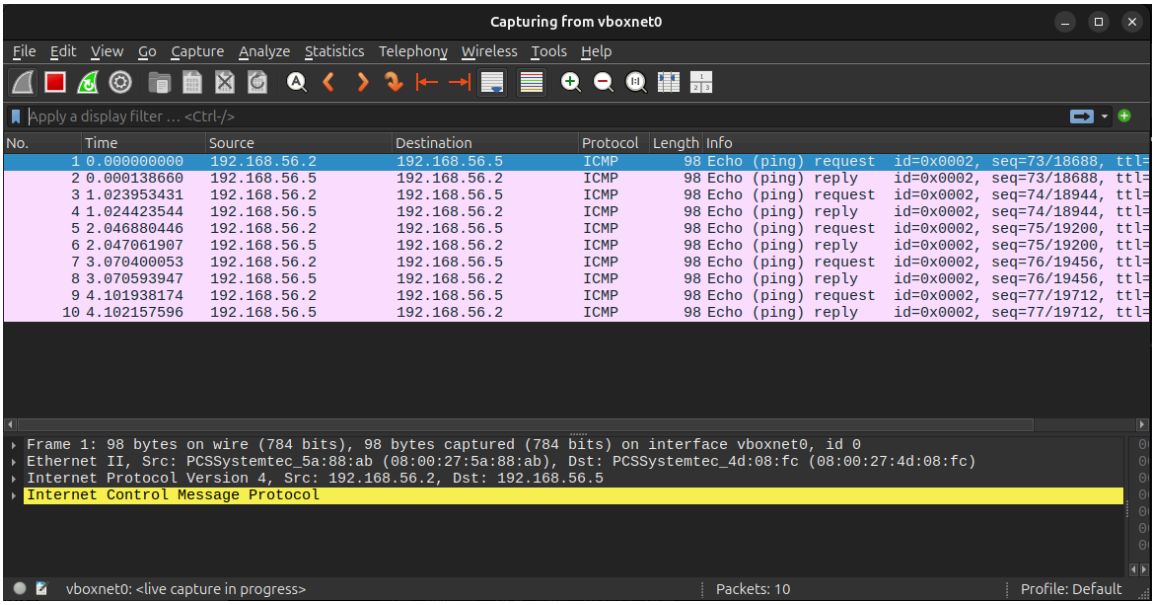
2. Open **Wireshark** and select the correct network interface for traffic capture.



3. Start packet capture in Wireshark.

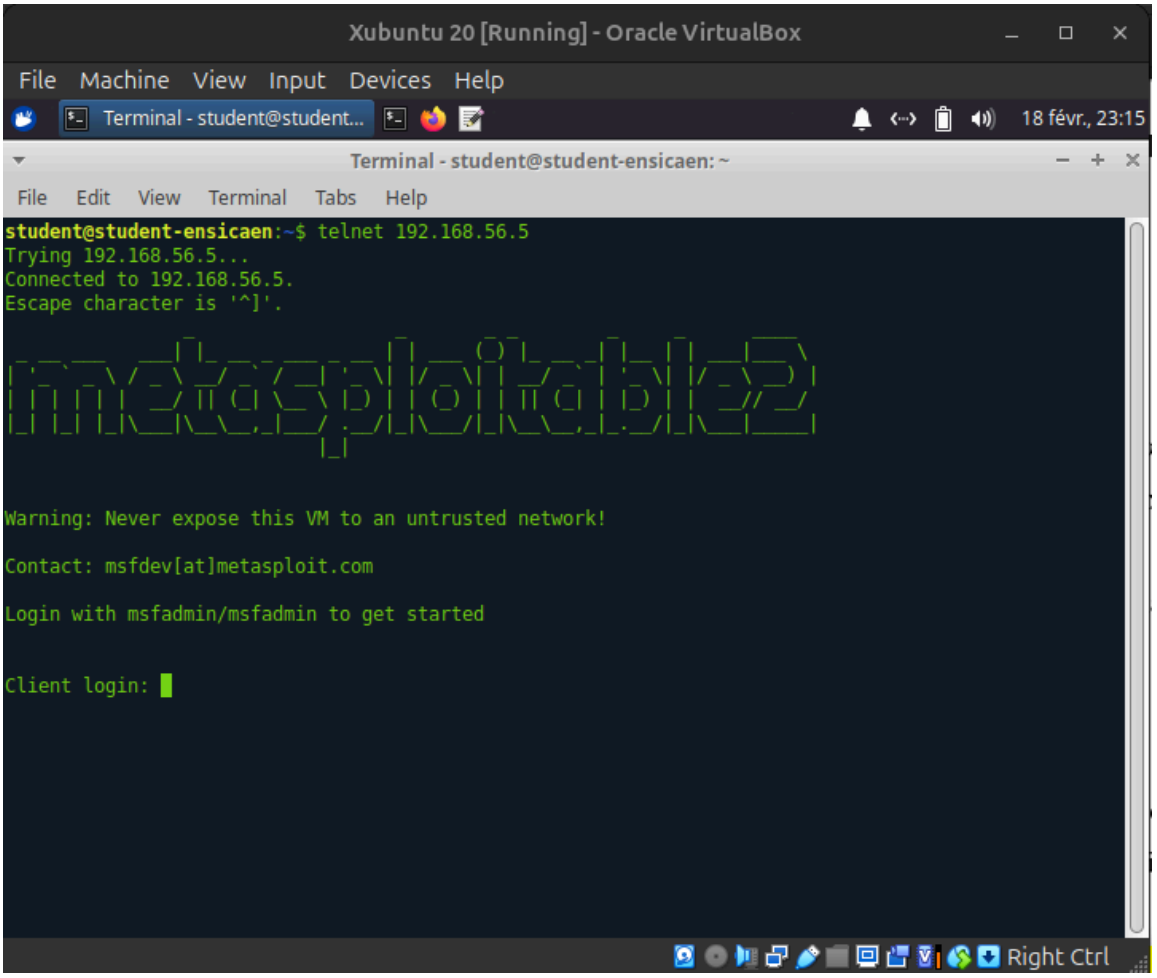
Testing packet capture with our network ✅





4. Open a terminal and run:

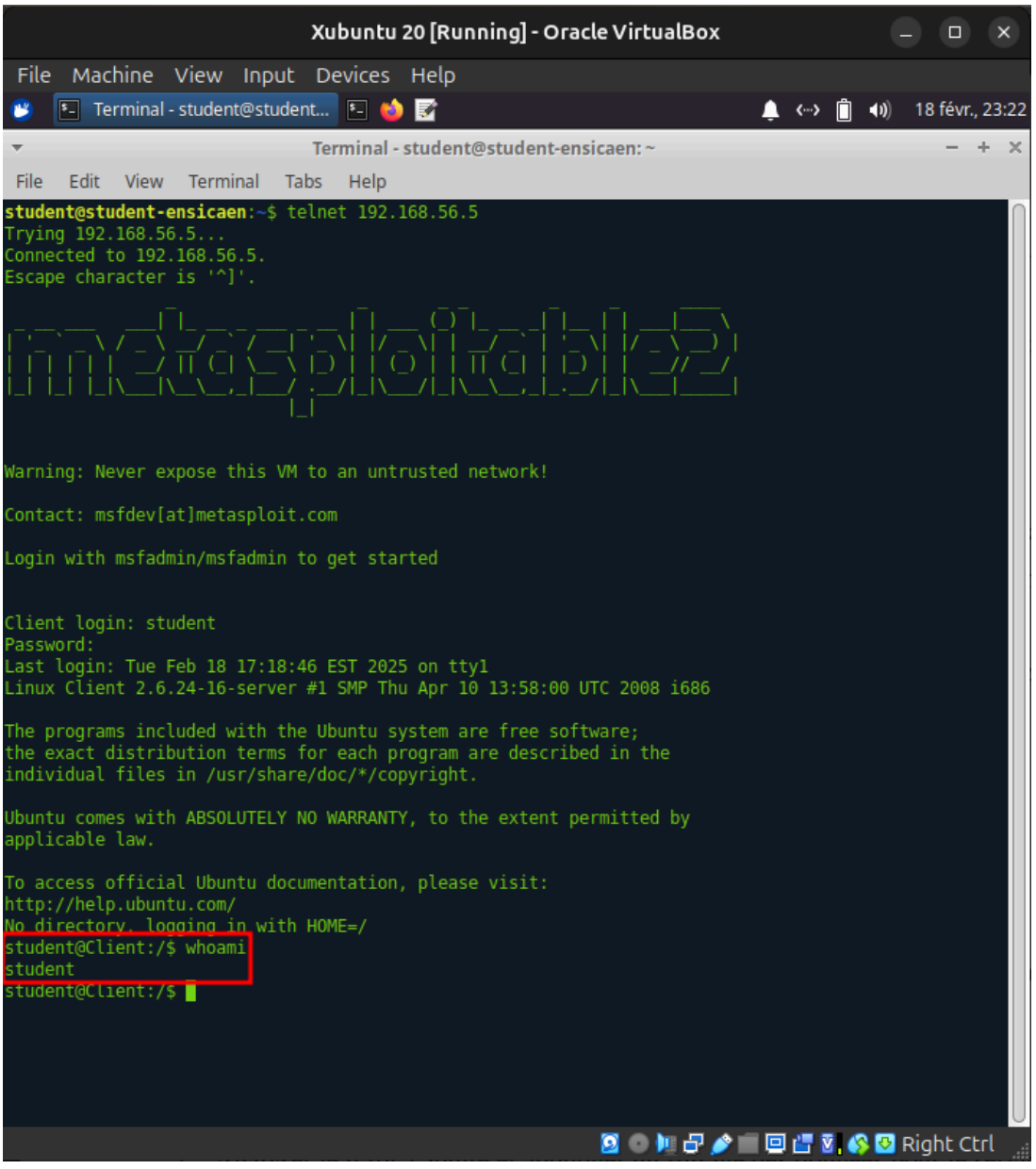
```
telnet <MyServer IP>
```



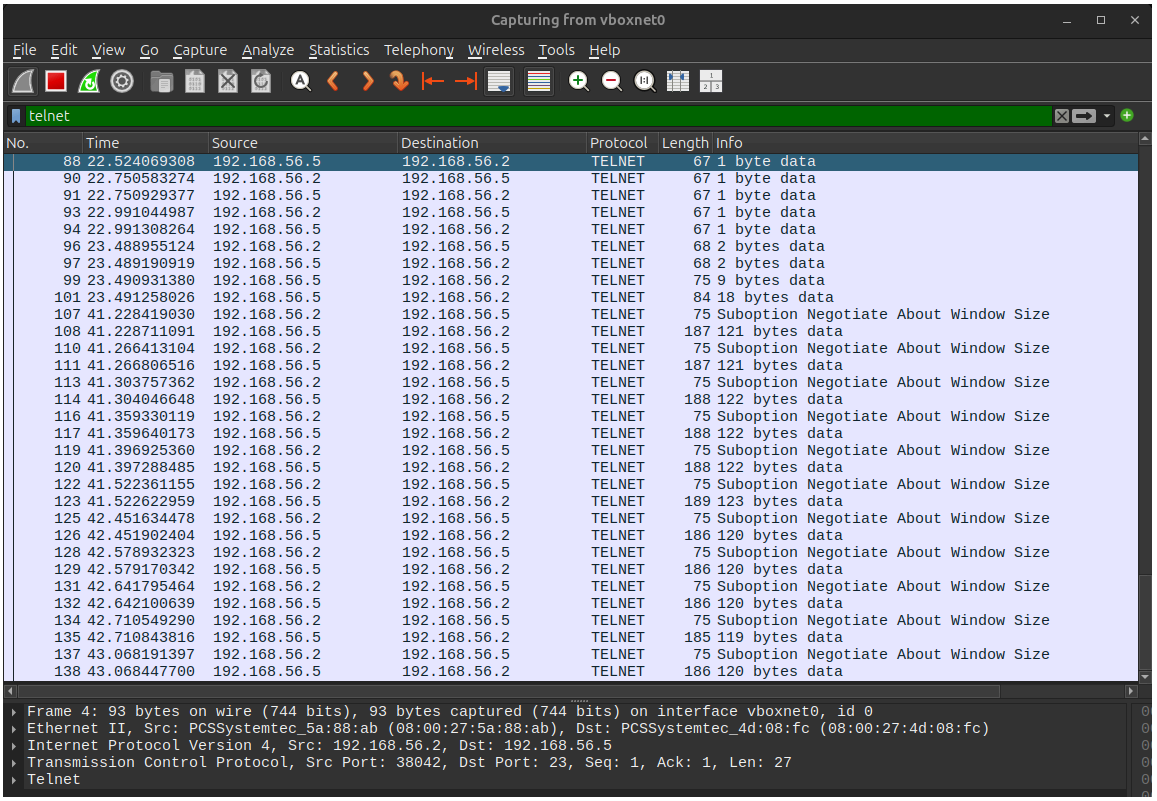
5. Log in using the **MyServer** credentials.
6. Run `whoami` again and explain the output.

   After connecting to the **MyServer** machine via **Telnet** and running the `whoami` command, the terminal will display the authenticated user on MyServer.

   - **If the login is successful**, the output will be:

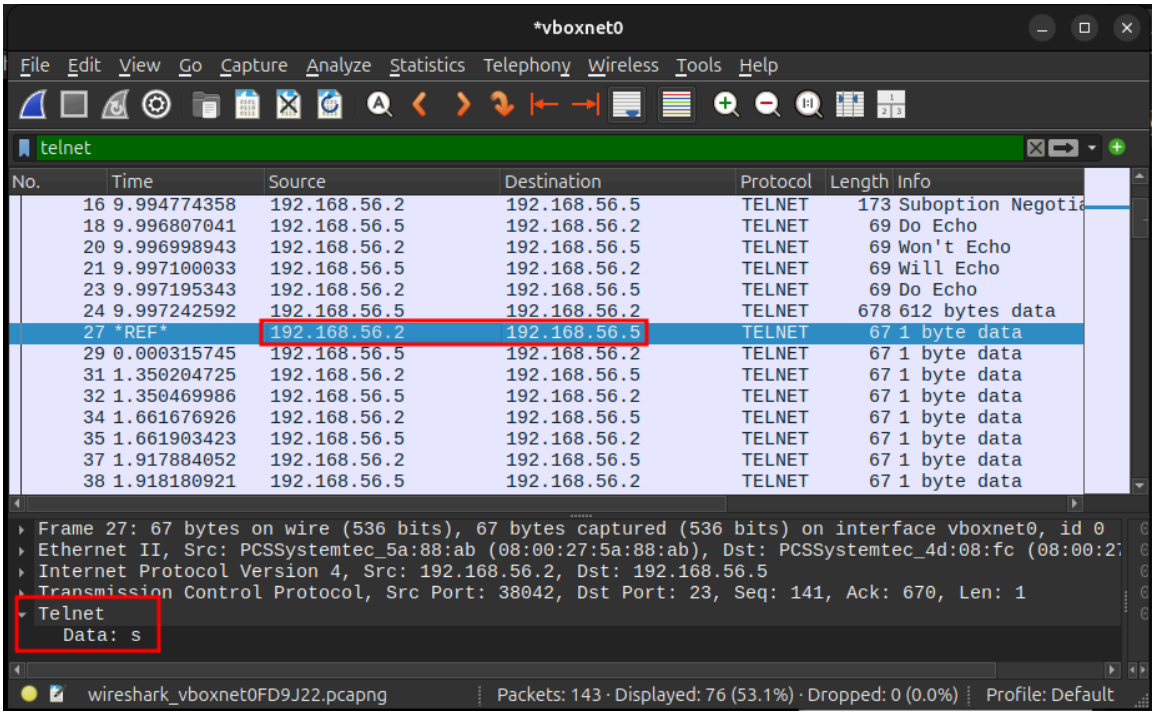


   This confirms that authentication was successful and that the user "student" has access to the remote system.
7. Stop the traffic capture in Wireshark.



8. Analyze the captured traffic and apply filters to isolate **Telnet** data.
9. Identify and extract the **username and password** exchanged during the session.

Login ✅



```
Telnet
    Data: t
```

```
Telnet
    Data: u
```

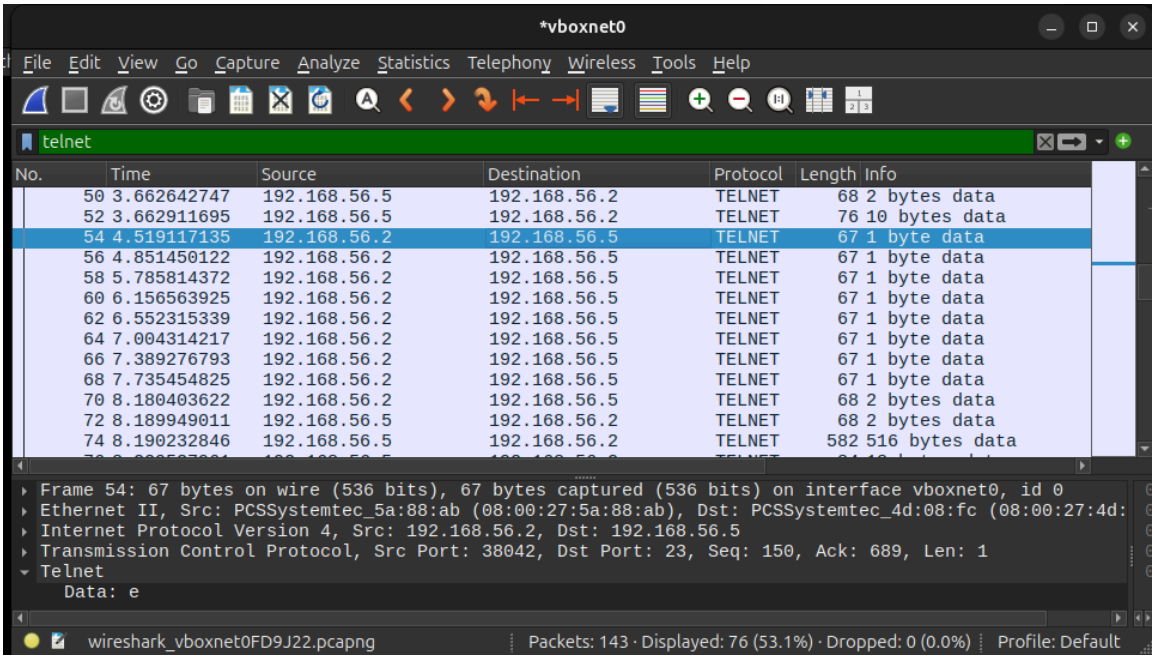
```
Telnet
    Data: d
```

```
Telnet
    Data: e
```

```
Telnet
    Data: n
```

```
Telnet
    Data: t
```

Password ✅




```
Telnet
    Data: n
```

```
Telnet
    Data: s
```

```
Telnet
    Data: i
```

```
Telnet
    Data: c
```

10. Type **"quit"** to exit the Telnet session.
11. What solution can be implemented to **secure authentication data**?

    The **Telnet** protocol does not provide encryption, making it highly vulnerable to sniffing attacks.

    Replace Telnet with SSH:

    - **SSH (Secure Shell)** encrypts communications, preventing credential capture.
    - SSH supports **public key authentication**, eliminating the need to send passwords over the network.

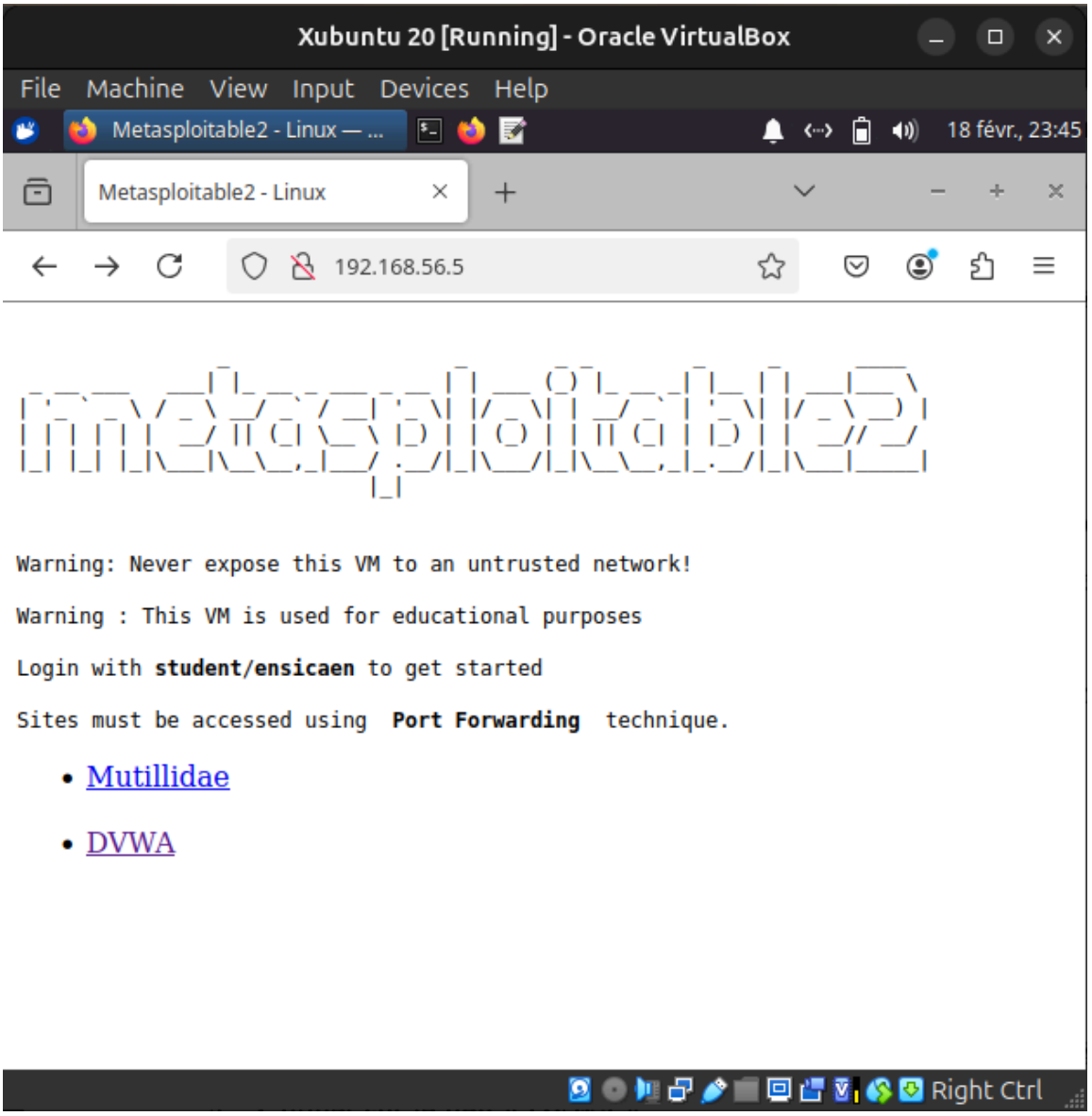
---

4. Capturing HTTP Data

1. Start a **new capture** in Wireshark.
2. Open **Firefox** browser.
3. In the address bar, enter:

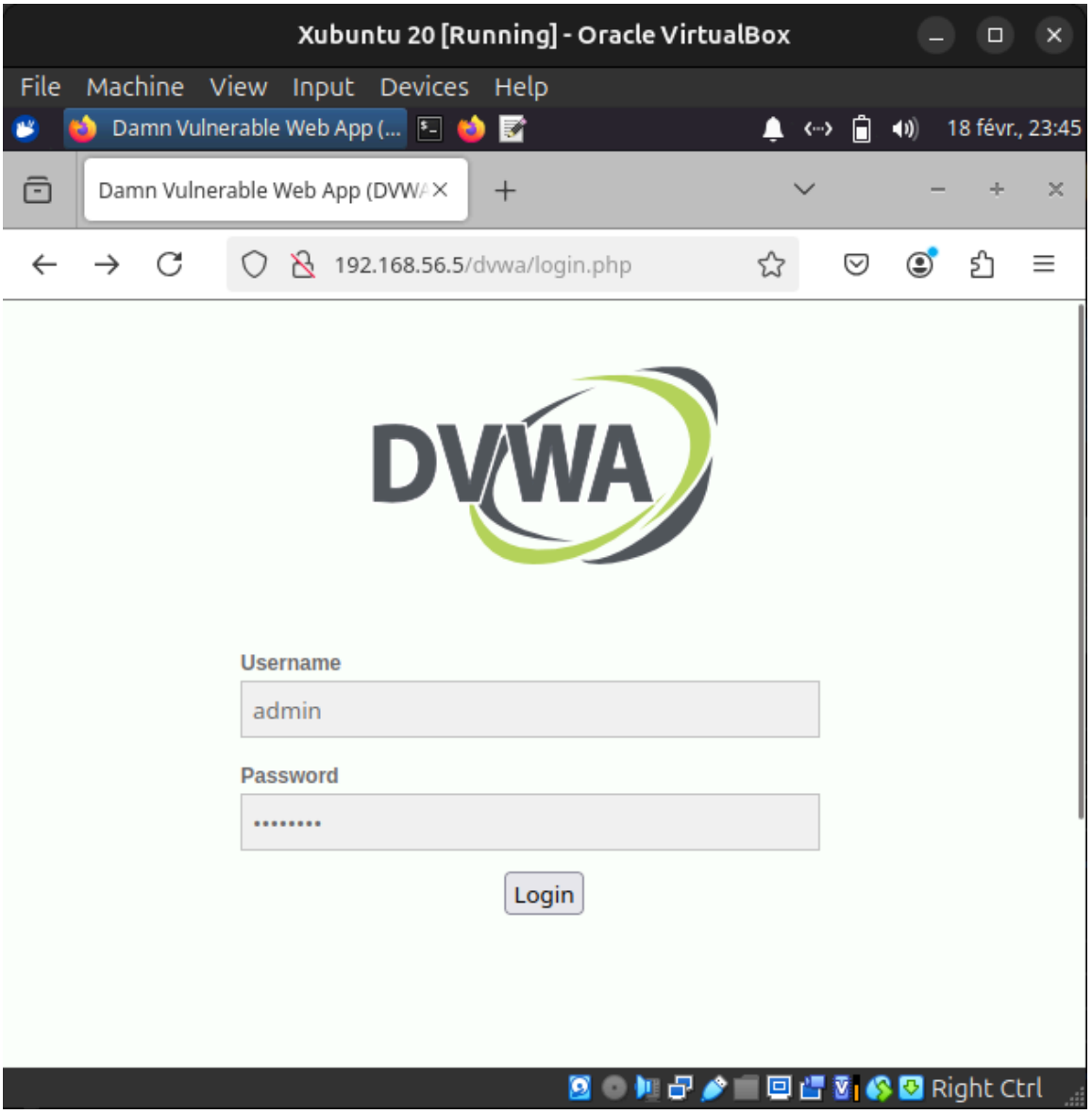
   ```
   http://<MyServer IP>
   ```

4. Click on the **"DVWA"** link.

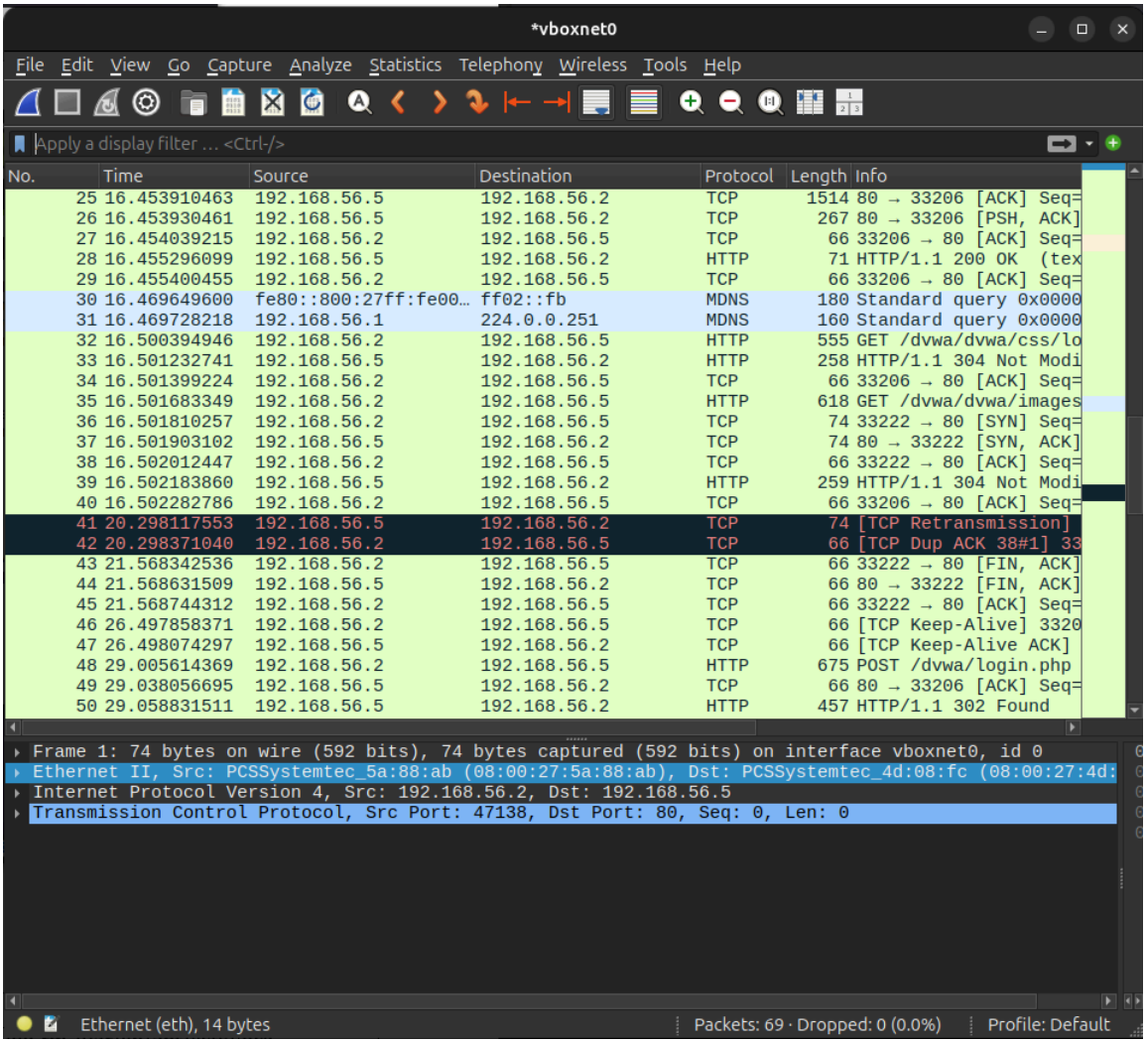


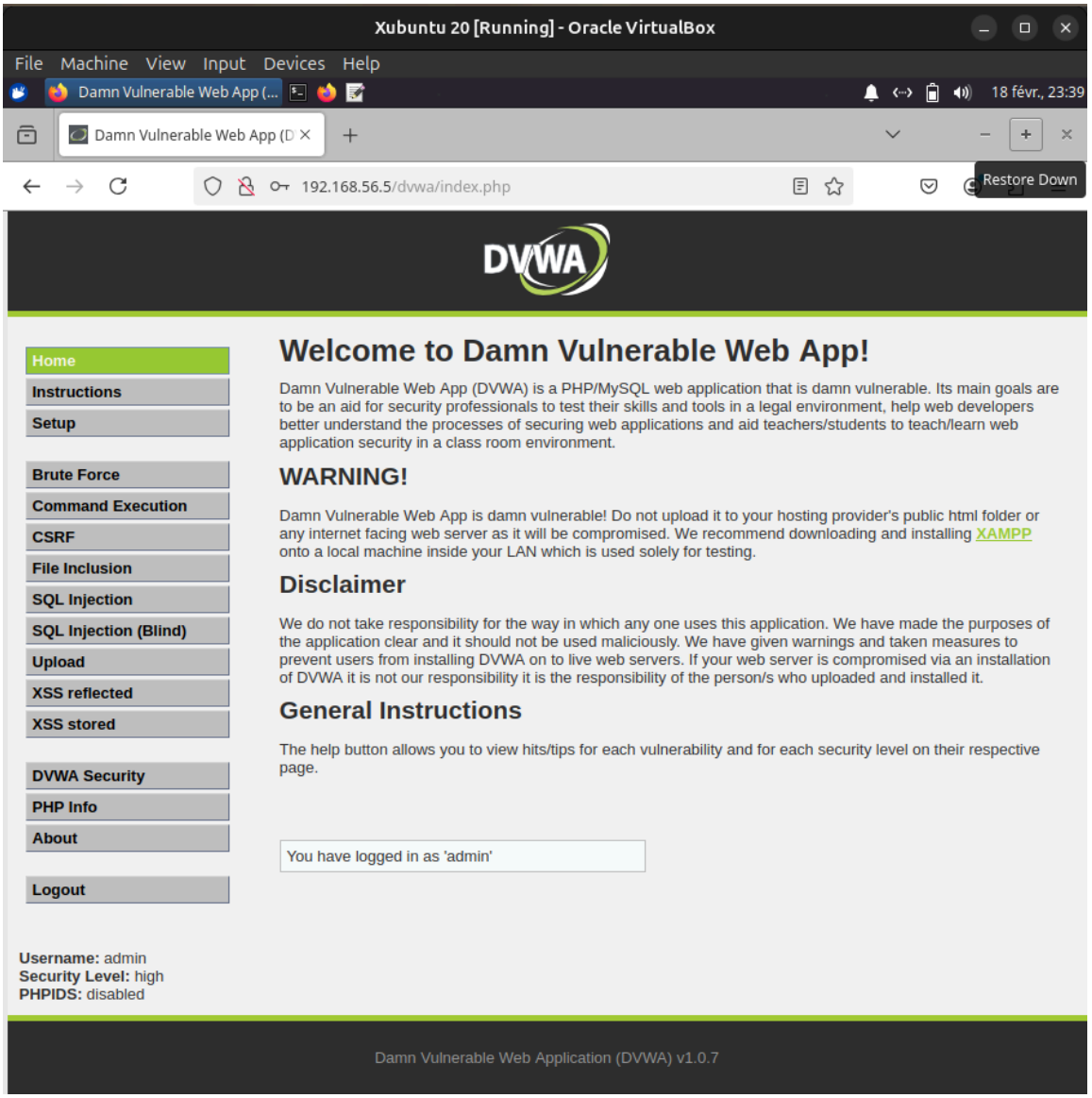
5. On the login page, enter the credentials:

admin | password

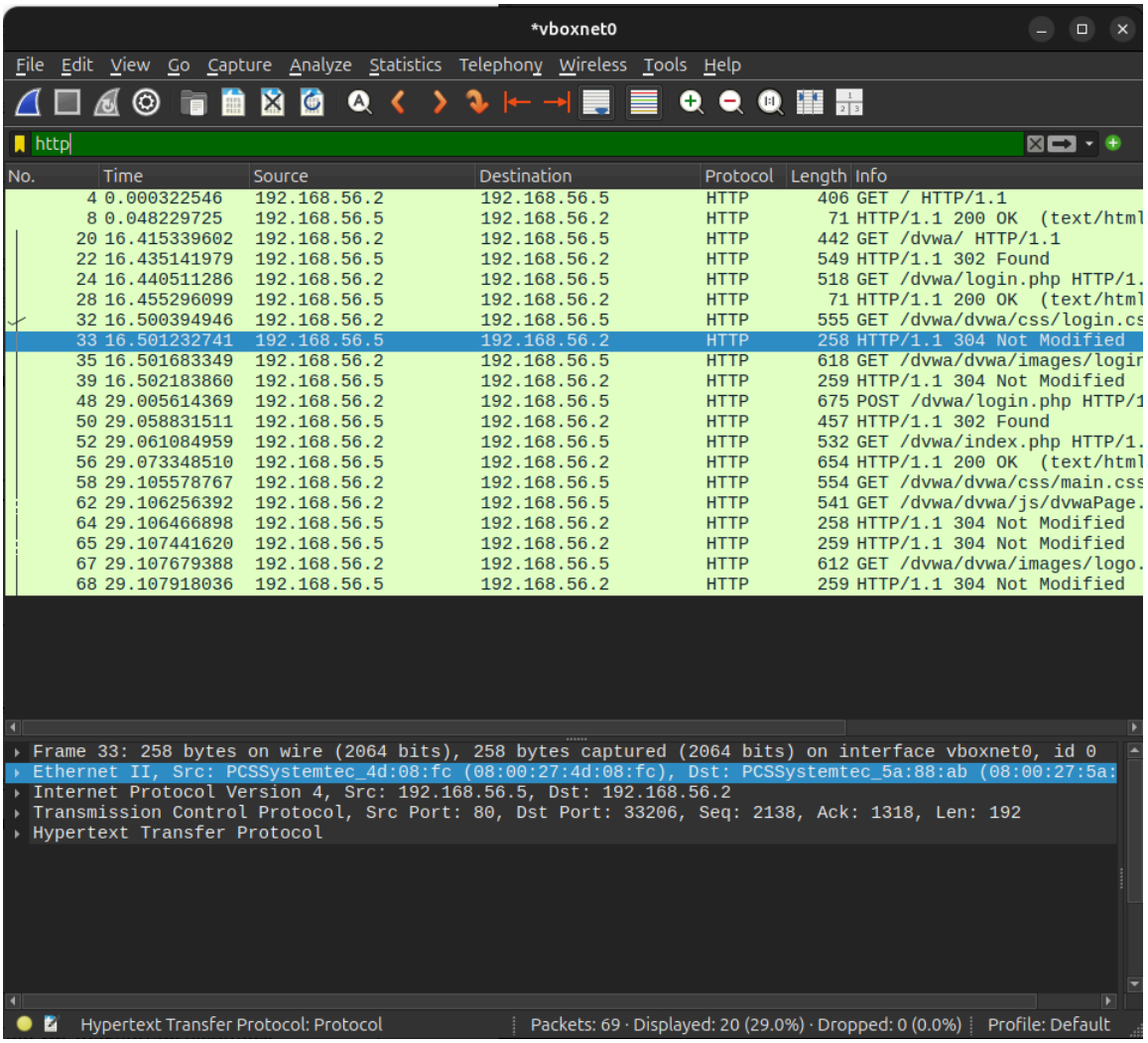


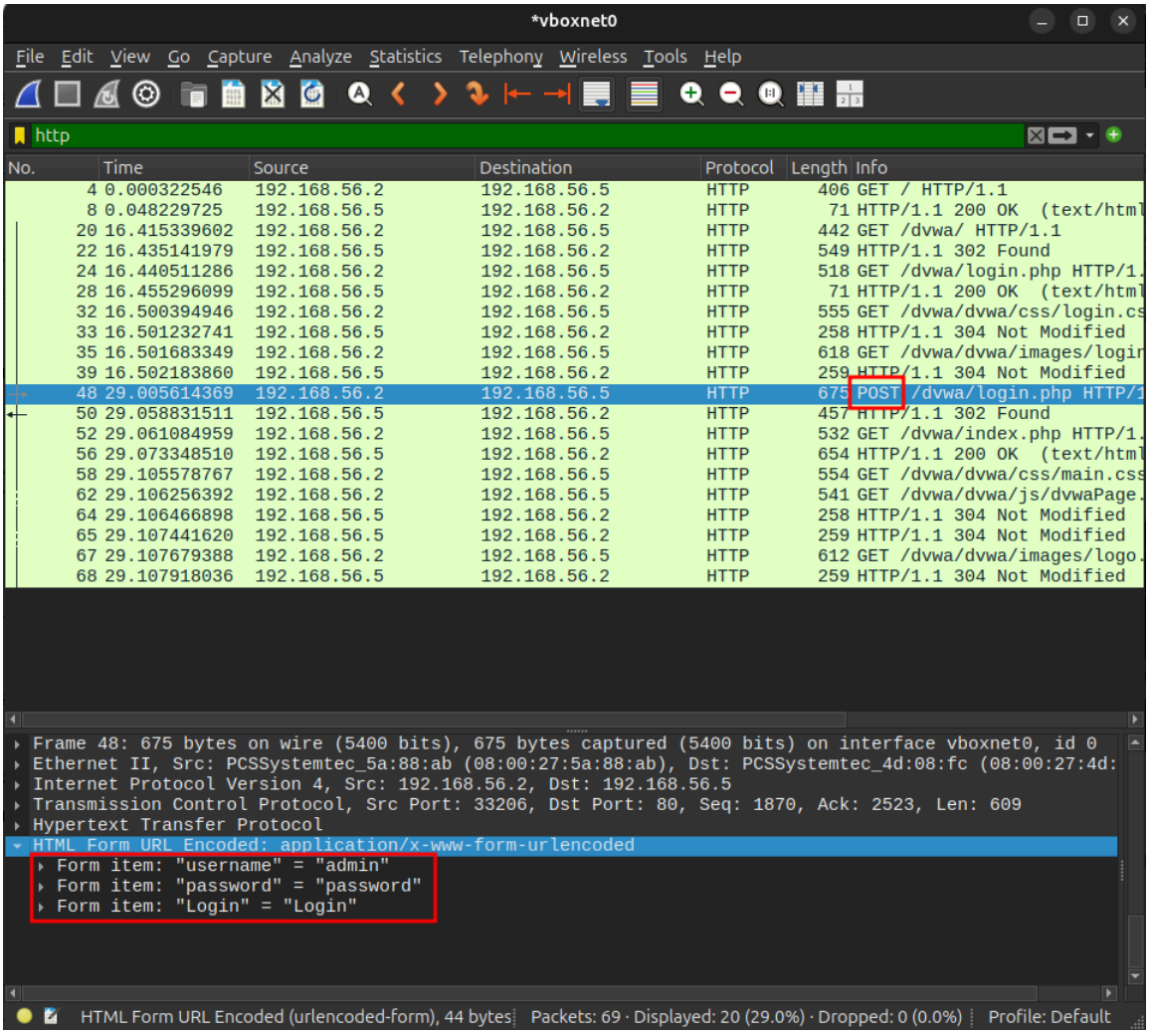
6. Once logged in, stop the packet capture.


*vboxnet0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
No. Time Source Destination Protocol Length Info
25 16.453910463 192.168.56.5 192.168.56.2 TCP 1514 80 → 33206 [ACK] Seq=
26 16.453930461 192.168.56.5 192.168.56.2 TCP 267 80 → 33206 [PSH, ACK]
27 16.454039215 192.168.56.2 192.168.56.5 TCP 66 33206 → 80 [ACK] Seq=
28 16.455296099 192.168.56.5 192.168.56.2 HTTP 71 HTTP/1.1 200 OK (tex
29 16.455400455 192.168.56.2 192.168.56.5 TCP 66 33206 → 80 [ACK] Seq=
30 16.469649600 fe80::800:27ff:fe00… ff02::fb MDNS 180 Standard query 0x0000
31 16.469728218 192.168.56.1 224.0.0.251 MDNS 160 Standard query 0x0000
32 16.500394946 192.168.56.2 192.168.56.5 HTTP 555 GET /dvwa/dvwa/css/lo
33 16.501232741 192.168.56.5 192.168.56.2 HTTP 258 HTTP/1.1 304 Not Modi
34 16.501399224 192.168.56.2 192.168.56.5 TCP 66 33206 → 80 [ACK] Seq=
35 16.501683349 192.168.56.2 192.168.56.5 HTTP 618 GET /dvwa/dvwa/images
36 16.501810257 192.168.56.2 192.168.56.5 TCP 74 33222 → 80 [SYN] Seq=
37 16.501903102 192.168.56.5 192.168.56.2 TCP 74 80 → 33222 [SYN, ACK]
38 16.502012447 192.168.56.2 192.168.56.5 TCP 66 33222 → 80 [ACK] Seq=
39 16.502183860 192.168.56.5 192.168.56.2 HTTP 259 HTTP/1.1 304 Not Modi
40 16.502282786 192.168.56.2 192.168.56.5 TCP 66 33206 → 80 [ACK] Seq=
41 20.298117553 192.168.56.5 192.168.56.2 TCP 74 [TCP Retransmission]
42 20.298371040 192.168.56.2 192.168.56.5 TCP 66 [TCP Dup ACK 38#1] 33
43 21.568342536 192.168.56.2 192.168.56.5 TCP 66 33222 → 80 [FIN, ACK]
44 21.568631509 192.168.56.5 192.168.56.2 TCP 66 80 → 33222 [FIN, ACK]
45 21.568744312 192.168.56.2 192.168.56.5 TCP 66 33222 → 80 [ACK] Seq=
46 26.497858371 192.168.56.2 192.168.56.5 TCP 66 [TCP Keep-Alive] 3320
47 26.498074297 192.168.56.5 192.168.56.2 TCP 66 [TCP Keep-Alive ACK]
48 29.005614369 192.168.56.2 192.168.56.5 HTTP 675 POST /dvwa/login.php
49 29.038056695 192.168.56.5 192.168.56.2 TCP 66 80 → 33206 [ACK] Seq=
50 29.058831511 192.168.56.5 192.168.56.2 HTTP 457 HTTP/1.1 302 Found
Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface vboxnet0, id 0
Ethernet II, Src: PCSSystemtec_5a:88:ab (08:00:27:5a:88:ab), Dst: PCSSystemtec_4d:08:fc (08:00:27:4d:
Internet Protocol Version 4, Src: 192.168.56.2, Dst: 192.168.56.5
Transmission Control Protocol, Src Port: 47138, Dst Port: 80, Seq: 0, Len: 0
Ethernet (eth), 14 bytes
Packets: 69 · Dropped: 0 (0.0%)
Profile: Default



7. Analyze the captured traffic and apply filters for **HTTP traffic**.


*vboxnet0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
No. Time Source Destination Protocol Length Info
4 0.000322546 192.168.56.2 192.168.56.5 HTTP 406 GET / HTTP/1.1
8 0.048229725 192.168.56.5 192.168.56.2 HTTP 71 HTTP/1.1 200 OK (text/html
20 16.415339602 192.168.56.2 192.168.56.5 HTTP 442 GET /dvwa/ HTTP/1.1
22 16.435141979 192.168.56.5 192.168.56.2 HTTP 549 HTTP/1.1 302 Found
24 16.440511286 192.168.56.2 192.168.56.5 HTTP 518 GET /dvwa/login.php HTTP/1.
28 16.455296099 192.168.56.5 192.168.56.2 HTTP 71 HTTP/1.1 200 OK (text/html
32 16.500394946 192.168.56.2 192.168.56.5 HTTP 555 GET /dvwa/dvwa/css/login.cs
33 16.501232741 192.168.56.5 192.168.56.2 HTTP 258 HTTP/1.1 304 Not Modified
35 16.501683349 192.168.56.2 192.168.56.5 HTTP 618 GET /dvwa/dvwa/images/logir
39 16.502183860 192.168.56.5 192.168.56.2 HTTP 259 HTTP/1.1 304 Not Modified
48 29.005614369 192.168.56.2 192.168.56.5 HTTP 675 POST /dvwa/login.php HTTP/1
50 29.058831511 192.168.56.5 192.168.56.2 HTTP 457 HTTP/1.1 302 Found
52 29.061084959 192.168.56.2 192.168.56.5 HTTP 532 GET /dvwa/index.php HTTP/1.
56 29.073348510 192.168.56.5 192.168.56.2 HTTP 654 HTTP/1.1 200 OK (text/html
58 29.105578767 192.168.56.2 192.168.56.5 HTTP 554 GET /dvwa/dvwa/css/main.css
62 29.106256392 192.168.56.2 192.168.56.5 HTTP 541 GET /dvwa/dvwa/js/dvwaPage.
64 29.106466898 192.168.56.5 192.168.56.2 HTTP 258 HTTP/1.1 304 Not Modified
65 29.107441620 192.168.56.5 192.168.56.2 HTTP 259 HTTP/1.1 304 Not Modified
67 29.107679388 192.168.56.2 192.168.56.5 HTTP 612 GET /dvwa/dvwa/images/logo.
68 29.107918036 192.168.56.5 192.168.56.2 HTTP 259 HTTP/1.1 304 Not Modified
Frame 33: 258 bytes on wire (2064 bits), 258 bytes captured (2064 bits) on interface vboxnet0, id 0
Ethernet II, Src: PCSSystemtec_4d:08:fc (08:00:27:4d:08:fc), Dst: PCSSystemtec_5a:88:ab (08:00:27:5a:
Internet Protocol Version 4, Src: 192.168.56.5, Dst: 192.168.56.2
Transmission Control Protocol, Src Port: 80, Dst Port: 33206, Seq: 2138, Ack: 1318, Len: 192
Hypertext Transfer Protocol
Hypertext Transfer Protocol: Protocol
Packets: 69 · Displayed: 20 (29.0%) · Dropped: 0 (0.0%)
Profile: Default



8. What solution can be implemented to **secure authentication data**?

   The **HTTP** protocol also transmits data in **plaintext**, meaning credentials can be easily intercepted. Solutions to protect authentication data include:

   1 - Use HTTPS (HyperText Transfer Protocol Secure)

   - HTTPS uses **TLS/SSL** to encrypt data between the client and the server.

   2 - Implement token-based authentication

   - Methods like **OAuth, JWT (JSON Web Token)**, or **encrypted session cookies** prevent credentials from being repeatedly sent.

   3 - Enable HSTS (HTTP Strict Transport Security)

   - HSTS forces browsers to always use HTTPS when communicating with the server.

Troubleshooting Errors

If you encounter errors such as:

```
E: Could not get lock /var/lib/dpkg/lock-frontend - open (11: Resource temp
orarily unavailable)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), is
another process using it?
```

Try running the following commands:

```
sudo rm /var/lib/apt/lists/lock
sudo rm /var/cache/apt/archives/lock
sudo rm /var/lib/dpkg/lock*
sudo dpkg --configure -a
sudo apt update
```