Lab - Examining Telnet and SSH in Wireshark

ISC6020

Humphrey Chivini

Group 1

Objectives

- **Part 1**: I examined a Telnet session with Wireshark
- **Part 2**: I examined an SSH session with Wireshark

Background / Scenario

In this lab, I configured a router to accept SSH connectivity and used Wireshark to capture and view both Telnet and SSH sessions. This demonstrated the importance of encryption when using SSH.

Required Resources

- CyberOps Workstation virtual machine

Part 1: Examining a Telnet Session with Wireshark

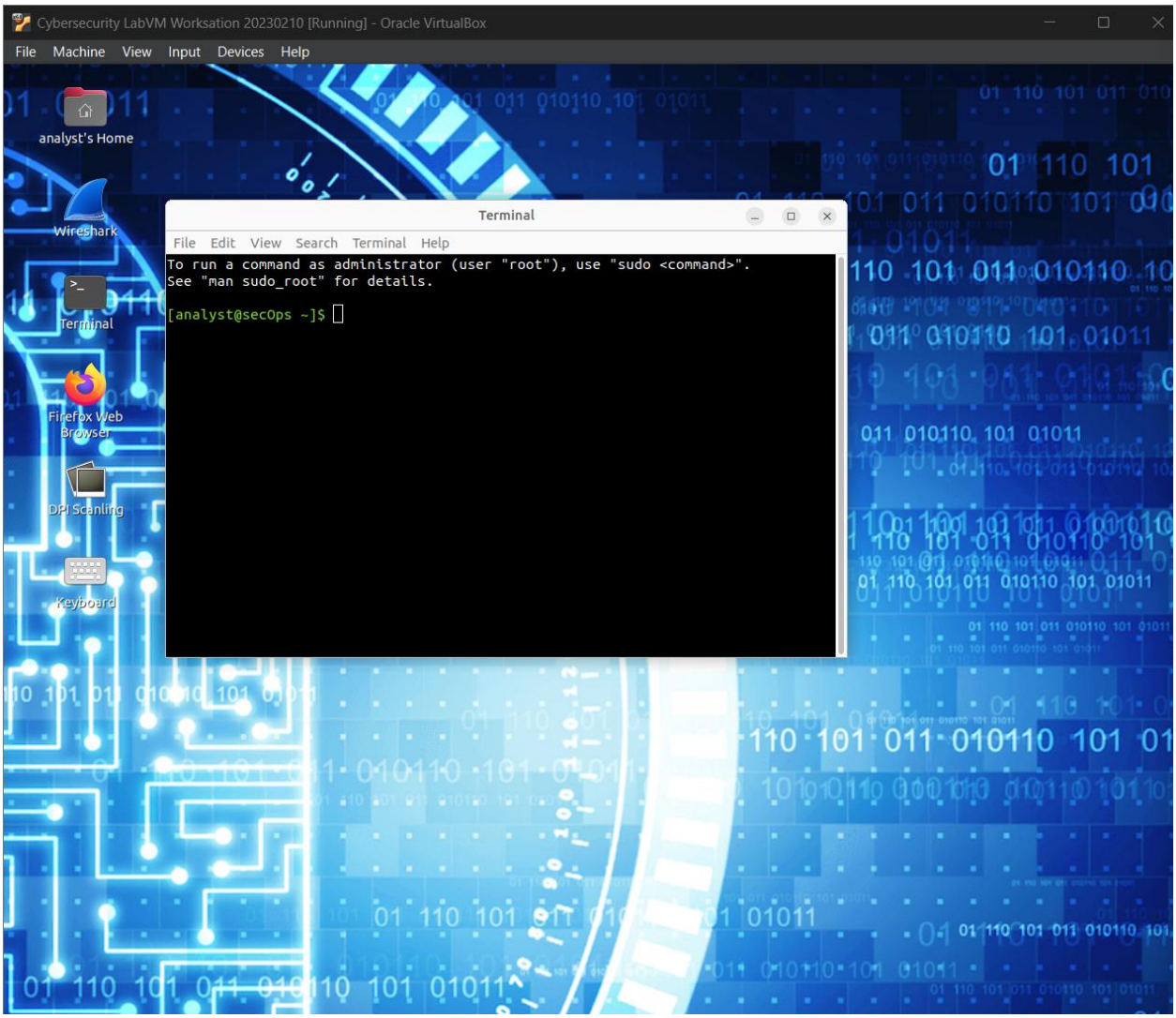
Step 1: I Captured Data

a. I started the CyberOps Workstation VM and logged in using the username `analyst` and password `cyberops`.

b. I opened a terminal window and started Wireshark:

```
bash
CopyEdit
[analyst@secOps ~]$ wireshark &
```

c. I started a Wireshark capture on lothe `Loopback: lo` interface.



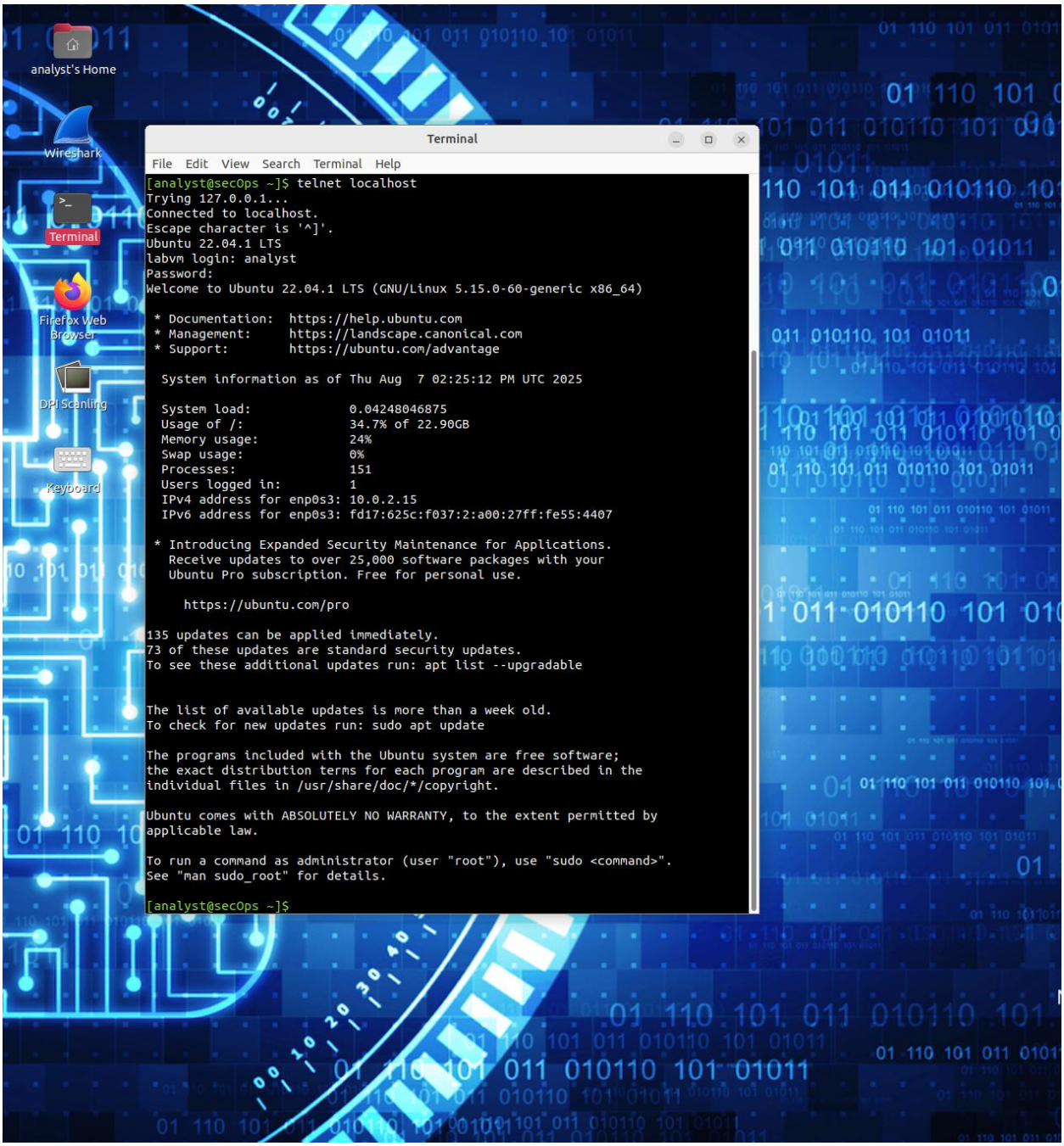
d. I opened another terminal window and started a Telnet session to the localhost. I entered the username `analyst` and password `cyberops` when prompted. It took a few minutes for the "connected to localhost" and login prompt to appear.

```
bash
CopyEdit
[analyst@secOps ~]$ c
Trying ::1...
Connected to localhost.
Escape character is '^]'.

Linux 4.10.10-1-ARCH (unallocated.barefruit.co.uk) (pts/12)
```

```
secOps login: analyst
Password:
Last login: Fri Apr 28 10:50:52 from localhost.localdomain
[analyst@secOps ~]$
```

e. I stopped the Wireshark capture after providing my user credentials.



Step 2: I Examined the Telnet Session

a. I applied a filter that only displayed Telnet-related traffic by typing `telnet` in the filter field and clicking **Apply**.

b. I right-clicked one of the Telnet lines in the **Packet List** section of Wireshark, then selected **Follow > TCP Stream** from the dropdown.

c. The **Follow TCP Stream** window showed the data from my Telnet session with the CyberOps Workstation VM. The entire session was displayed in plaintext, including my password. I noticed that the username I entered appeared with duplicate characters, which was due to Telnet's echo setting that lets me see what I typed on the screen.

d. After reviewing the Telnet session, I clicked **Close**.

e. I typed `exit` at the terminal to close the Telnet session.

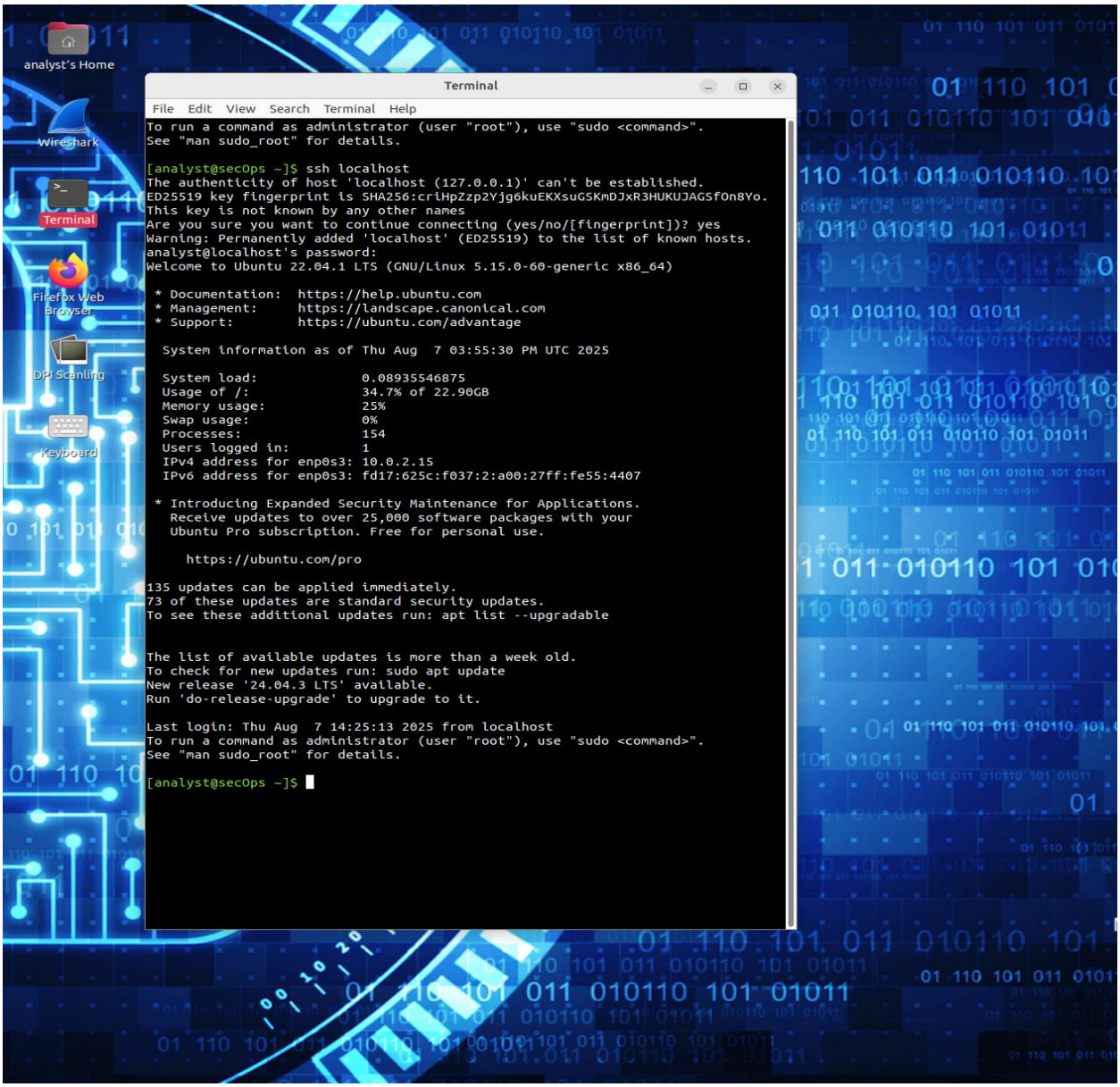
```
bash
CopyEdit
[analyst@secOps ~]$ exit
```

Part 2: Examining an SSH Session with Wireshark

a. I started a new Wireshark capture using the `Loopback: lo` interface.

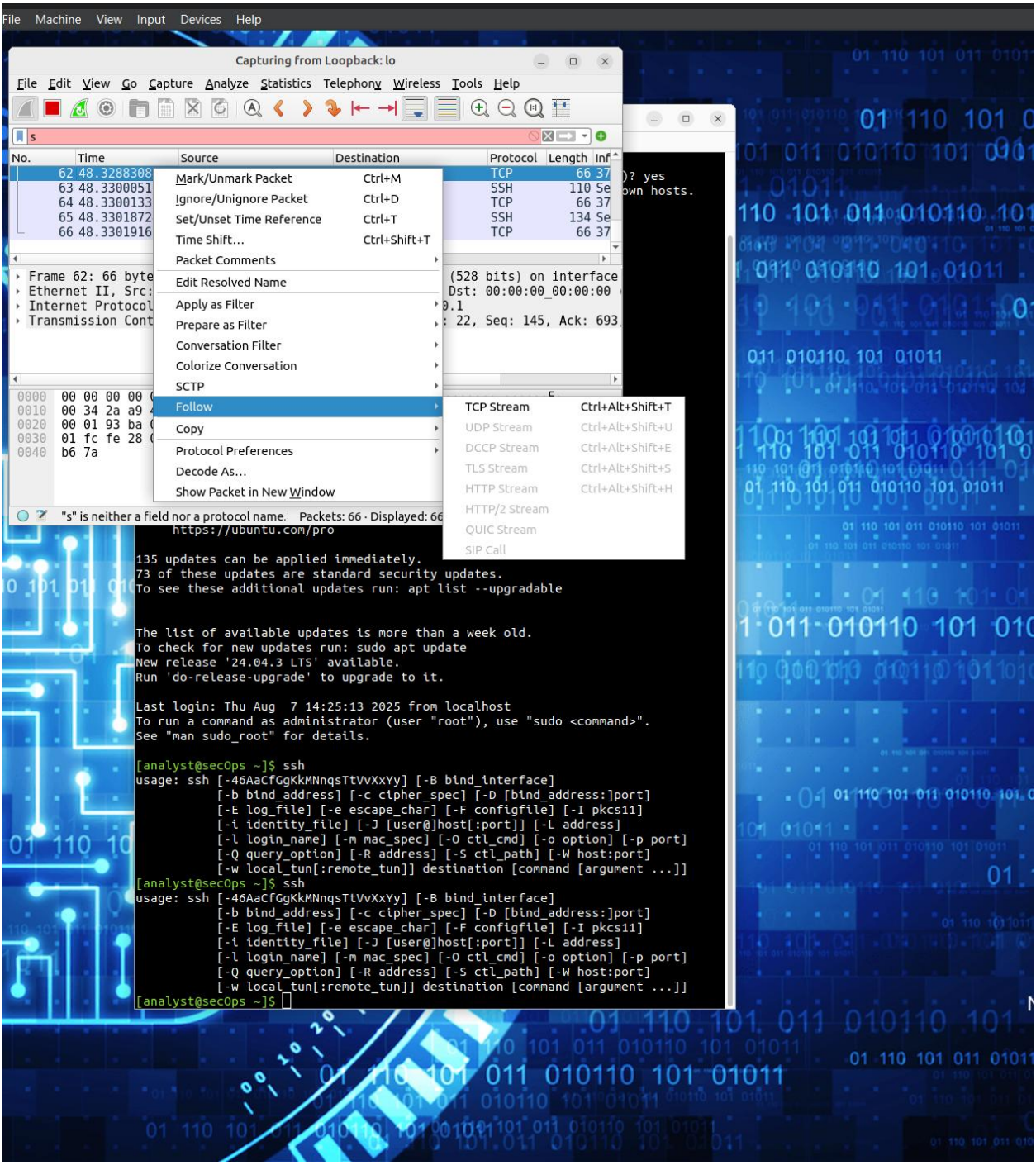
b. I established an SSH session with the localhost by entering `ssh localhost` at the terminal prompt. I entered `yes` to continue connecting and then typed the password `cyberops`.

```
bash
CopyEdit
[analyst@secOps ~]$ ssh localhost
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:1xZuV8NMeVsNQPRrzVf9nXHzdUP+EtgVouZVbWH80XA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
analyst@localhost's password:
Last login: Sat May 23 10:18:47 2020
```



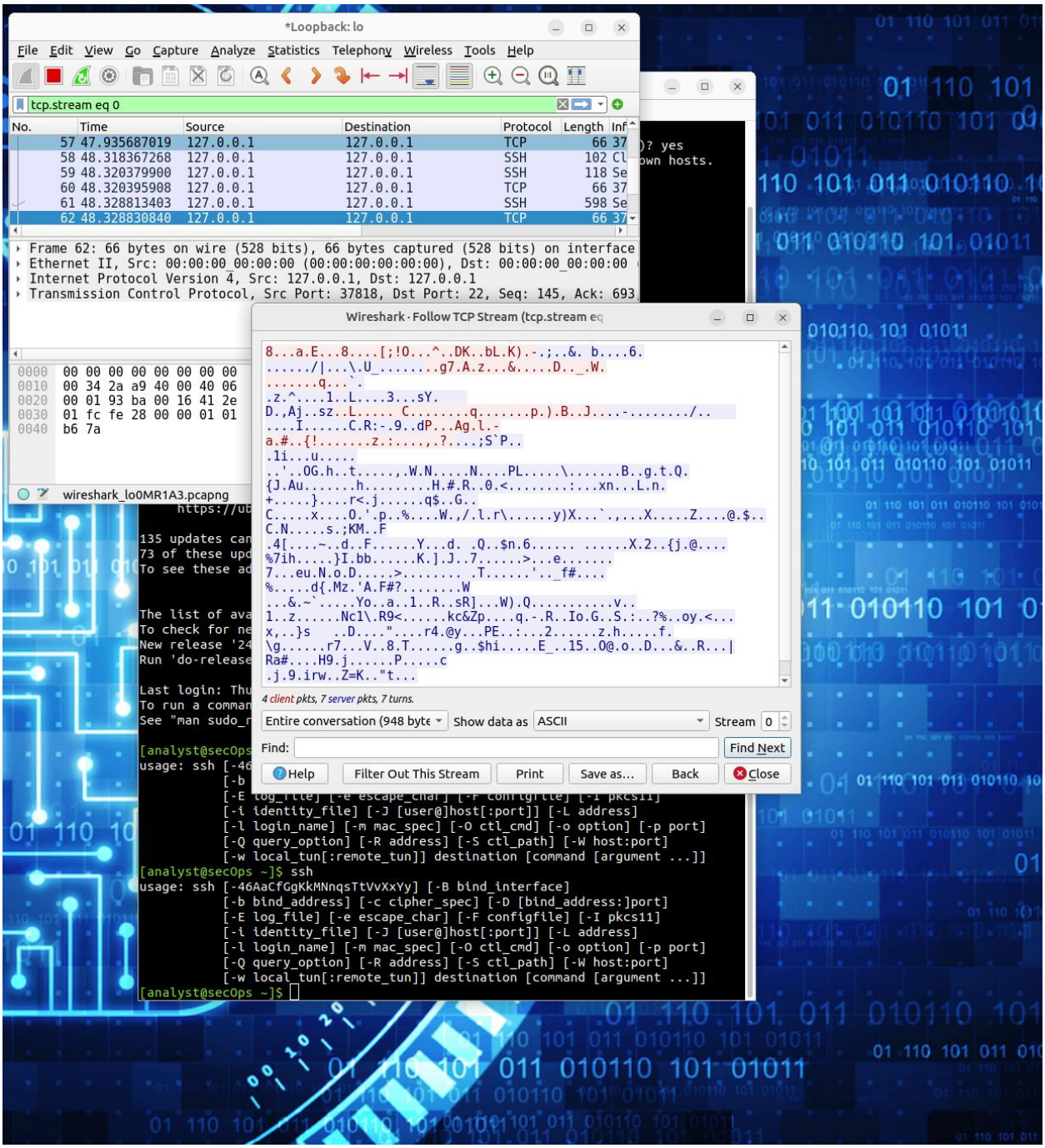
d. I applied an SSH filter on the Wireshark capture data by typing `ssh` in the filter field and clicking **Apply**.

e. I right-clicked one of the SSHv2 lines in the **Packet List** section of Wireshark, and selected **Follow > TCP Stream** from the dropdown.



f. I examined the **Follow TCP Stream** window for the SSH session. The data was encrypted and unreadable, unlike the Telnet session which showed everything in plaintext.

g. After reviewing the SSH session, I clicked **Close**.



h. I closed Wireshark.

Reflection Question

Why is SSH preferred over Telnet for remote connections?

SSH is preferred over Telnet because it encrypts all communication, protecting sensitive data like usernames and passwords from being viewed by unauthorized individuals. In contrast, Telnet sends information in plaintext, which poses a serious security risk.