TELNET VS SSH

"A PRACTICAL DEMONSTRATION USING WIRESHARK IN KALI LINUX"

NAME-NIKHIL KUMAR
ERP ID -6606652

INTRODUCTION

Remote access protocols allow users to connect to and control a computer system over a network. Among these protocols, Telnet and SSH (Secure Shell) are widely known. Telnet was one of the earliest remote login protocols but lacks security features. SSH was later developed to overcome the security limitations of Telnet by providing encrypted communication.

This project focuses on a theoretical comparison of Telnet and SSH to highlight why SSH is preferred in modern secure networks.

TELNET PROTOCOL⏩

Telnet is a client–server protocol used for remote login into systems over a network.

KEY CHARACTERISTICS OF TELNET

- 🔹Uses TCP port 23
- 🔹Transfers data in plain text
- 🔹Does not support encryption
- 🔹Username, password, and commands are transmitted openly

Vulnerable to packet sniffing and Man-in-the-Middle (MITM) attacks

Security Weakness of Telnet

Because Telnet does not encrypt data, any attacker monitoring the network can easily read sensitive information such as login credentials and executed commands. This makes Telnet unsuitable for use in secure or public networks.

SSH PROTOCOL ⏩

SSH (Secure Shell) is a secure remote access protocol designed to replace insecure protocols like Telnet.

Key Characteristics of SSH

- Uses TCP port 22
- Encrypts all transmitted data
- Supports secure authentication
- Protects confidentiality and integrity of data
- Resistant to network sniffing attacks

Security Strength of SSH

SSH uses cryptographic techniques to encrypt communication between client and server. Even if network traffic is captured, the data remains unreadable, ensuring secure remote access.

ENCRYPTION concept in SSH

Encryption is the process of converting readable data into an unreadable format. SSH uses:

- Symmetric encryption for data transfer
- Asymmetric encryption for key exchange
- Hashing for data integrity

These mechanisms ensure that only authorized users can access the transmitted information.

COMPARATIVE SECURITY ANALYSIS OF TELNET AND SSH

Step 1:Prepare the lab Environment ❗

Requirements...

- 🔹 Kali Linux (server)
- 🔹 KALI LINUX VIRTUAL BOX
- 🔹 Wireshark
- 🔹 TELNET SERVER
- 🔹 SSH

Step 2:Change Hostname

(Change the hostname to your name for project clarity).

Step 3:Install Telnet service

1.Open terminal in Kali Linux

2.Install Telnet server:

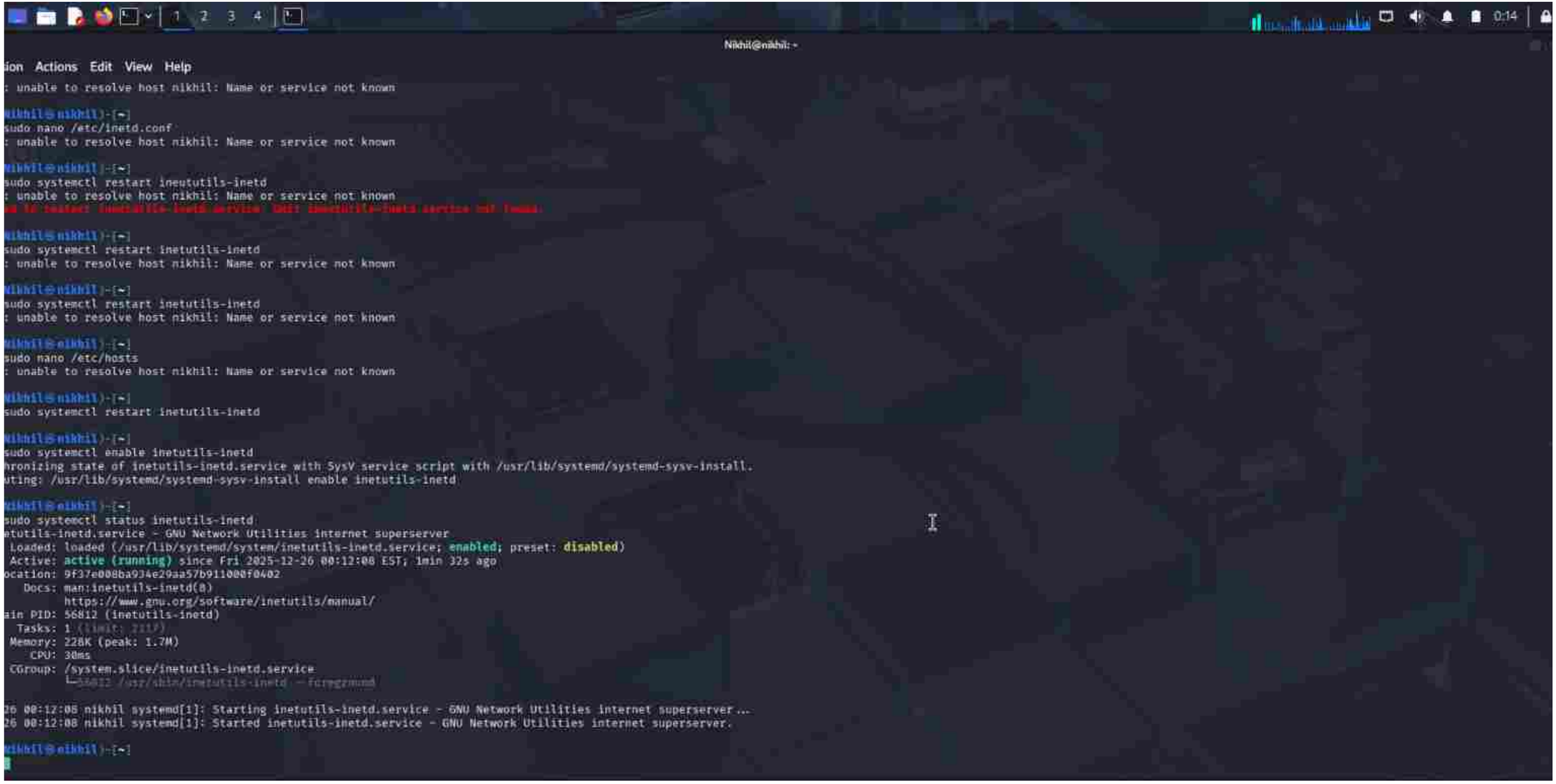
```
sudo apt update
sudo apt install inetutils-telnetd
```

3.Start Telnet service

```
sudo service inetutils-telnetd start
```

4.Check service status:

```
sudo service inetutils-telnetd status
```


Nikhil@nikhil: ~
ion Actions Edit View Help
: unable to resolve host nikhil: Name or service not known
sudo nano /etc/inetd.conf
: unable to resolve host nikhil: Name or service not known
sudo systemctl restart ineututils-inetd
: unable to resolve host nikhil: Name or service not known
sudo systemctl restart inetutils-inetd
: unable to resolve host nikhil: Name or service not known
sudo systemctl restart inetutils-inetd
: unable to resolve host nikhil: Name or service not known
sudo nano /etc/hosts
: unable to resolve host nikhil: Name or service not known
sudo systemctl restart inetutils-inetd
sudo systemctl enable inetutils-inetd
hronizing state of inetutils-inetd.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
uting: /usr/lib/systemd/systemd-sysv-install enable inetutils-inetd
sudo systemctl status inetutils-inetd
etutils-inetd.service - GNU Network Utilities internet superserver
Loaded: loaded (/usr/lib/systemd/system/inetutils-inetd.service; enabled; preset: disabled)
Active: active (running) since Fri 2025-12-26 00:12:08 EST; 1min 32s ago
ocation: 9f37e008ba934e20aa57b91100f0402
Docs: man:inetutils-inetd(8)
https://www.gnu.org/software/inetutils/manual/
ain PID: 56812 (inetutils-inetd)
Tasks: 1
Memory: 228K (peak: 1.7M)
CPU: 30ms
CGroup: /system.slice/inetutils-inetd.service
26 00:12:08 nikhil systemd[1]: Starting inetutils-inetd.service - GNU Network Utilities internet superserver...
26 00:12:08 nikhil systemd[1]: Started inetutils-inetd.service - GNU Network Utilities internet superserver.

Step 4:Start Wireshark for Telnet capture

1.Open Wireshark
2.Select active network interface(eth0/elan0)
3.Click Start Capture

Step 5: Connect Using Telnet

1.Open Telnet Client
2.Connect to Kali Linux:

```
telnet <Kali-IP>
```

3.Enter username and password
4.Execute Commands like:

```
ls
whoami
ifconfig
```

Step 6: Analyse Telnet Traffic

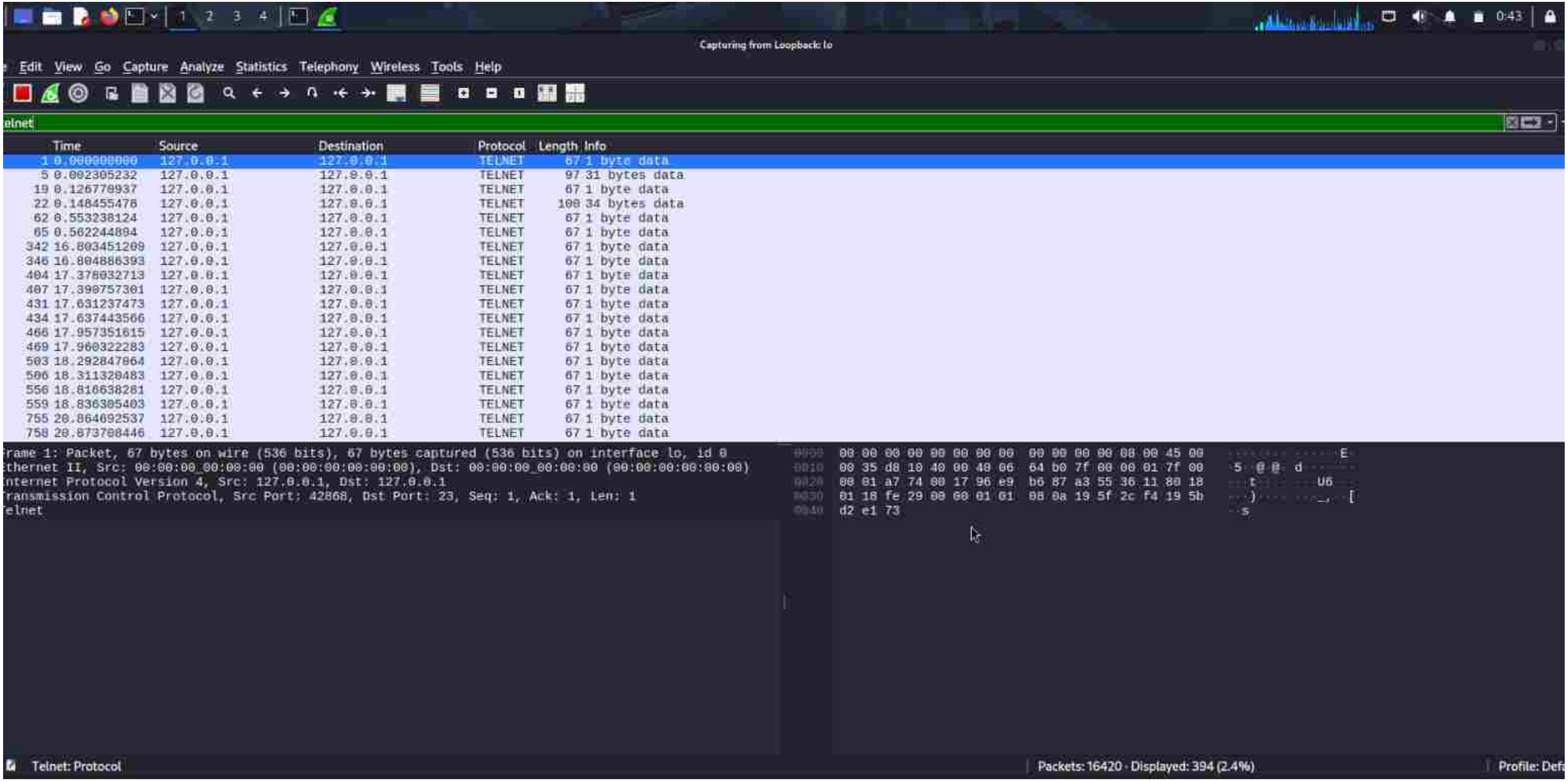
1.Stop Wireshark capture
2.Apply filter:

telnet

3.Click packets ➡️ Observe Readable text

4.Observation:

- Username visible
- Password visible
- Commands visible


Capturing from Loopback: lo
Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
elnet
Time Source Destination Protocol Length Info
1 0.000000000 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
5 0.002305232 127.0.0.1 127.0.0.1 TELNET 97 31 bytes data
19 0.126770937 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
22 0.148455478 127.0.0.1 127.0.0.1 TELNET 100 34 bytes data
62 0.553238124 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
65 0.562244894 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
342 16.803451209 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
345 16.804886393 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
404 17.378032713 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
407 17.390757301 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
431 17.631237473 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
434 17.637443566 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
466 17.957351615 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
469 17.960322283 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
503 18.292847864 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
506 18.311320483 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
556 18.816638281 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
559 18.836305403 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
755 20.864692537 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
758 20.873708446 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
rame 1: Packet, 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface lo, id 0
thernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
nternet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
ransmission Control Protocol, Src Port: 42868, Dst Port: 23, Seq: 1, Ack: 1, Len: 1
elnet
0000 00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00 E
0010 00 35 d8 10 40 00 40 06 64 b0 7f 00 00 01 7f 00 5 @ @ d
0020 00 01 a7 74 00 17 96 e9 b6 87 a3 55 36 11 80 18 t U6
0030 01 18 fe 29 00 00 01 01 08 0a 19 5f 2c f4 19 5b) _, [
0040 d2 e1 73 s
Telnet: Protocol
Packets: 16420 · Displayed: 394 (2.4%)
Profile: Def

Step 7:Install and start SSH Service

1.Install SSH:

```
sudo apt install openssh-server
```

2.Start SSH service:

```
sudo service ssh start
```

3.Check Status:

```
sudo service ssh status
```

```
Nikhil@nikhil: ~
Session  Actions  Edit  View  Help
openssh-server is already the newest version (1:10.2p1-3).
wireshark is already the newest version (4.6.0-1).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 9

┌──(Nikhil㉿nikhil)-[~]
└─$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh

┌──(Nikhil㉿nikhil)-[~]
└─$ sudo systemctl start ssh

┌──(Nikhil㉿nikhil)-[~]
└─$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: disabled)
     Active: active (running) since Thu 2025-12-25 23:23:18 EST; 1h 0min ago
 Invocation: 47e04f94ad054f51bb7401b137f425fc
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 612 (sshd)
      Tasks: 1 (limit: 211)
     Memory: 1.5M (peak: 2.9M, swap: 4K, swap peak: 4K)
        CPU: 31ms
     CGroup: /system.slice/ssh.service
             └─612 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Dec 25 23:23:18 nikhil systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Dec 25 23:23:18 nikhil sshd[612]: Server listening on 0.0.0.0 port 22.
Dec 25 23:23:18 nikhil sshd[612]: Server listening on :: port 22.
Dec 25 23:23:18 nikhil systemd[1]: Started ssh.service - OpenBSD Secure Shell server.

┌──(Nikhil㉿nikhil)-[~]
└─$ ssh localhost
The authenticity of host 'localhost (::1)' can't be established.
ED25519 key fingerprint is: SHA256:l9M5lpgcl3BrgwuxvMJJVcrfTdH50pWvekilKYbOrf8
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
Nikhil@localhost's password:
Linux nikhil 6.17.10+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.17.10-1kali1 (2025-12-08) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
┌──(Nikhil㉿nikhil)-[~]
└─$
```

Step 8: Start Wireshark for SSH Capture

1.Clear old captures

2.Start new capture in Wireshark

3.Select same network interface

Step 9: Connect using SSH

1.From client PC

```
ssh username@<Kali-IP>
```

2.Enter password

3.Run commands:

```
ls
pwd
whoami
```

Step 10: Analyse SSH Traffic

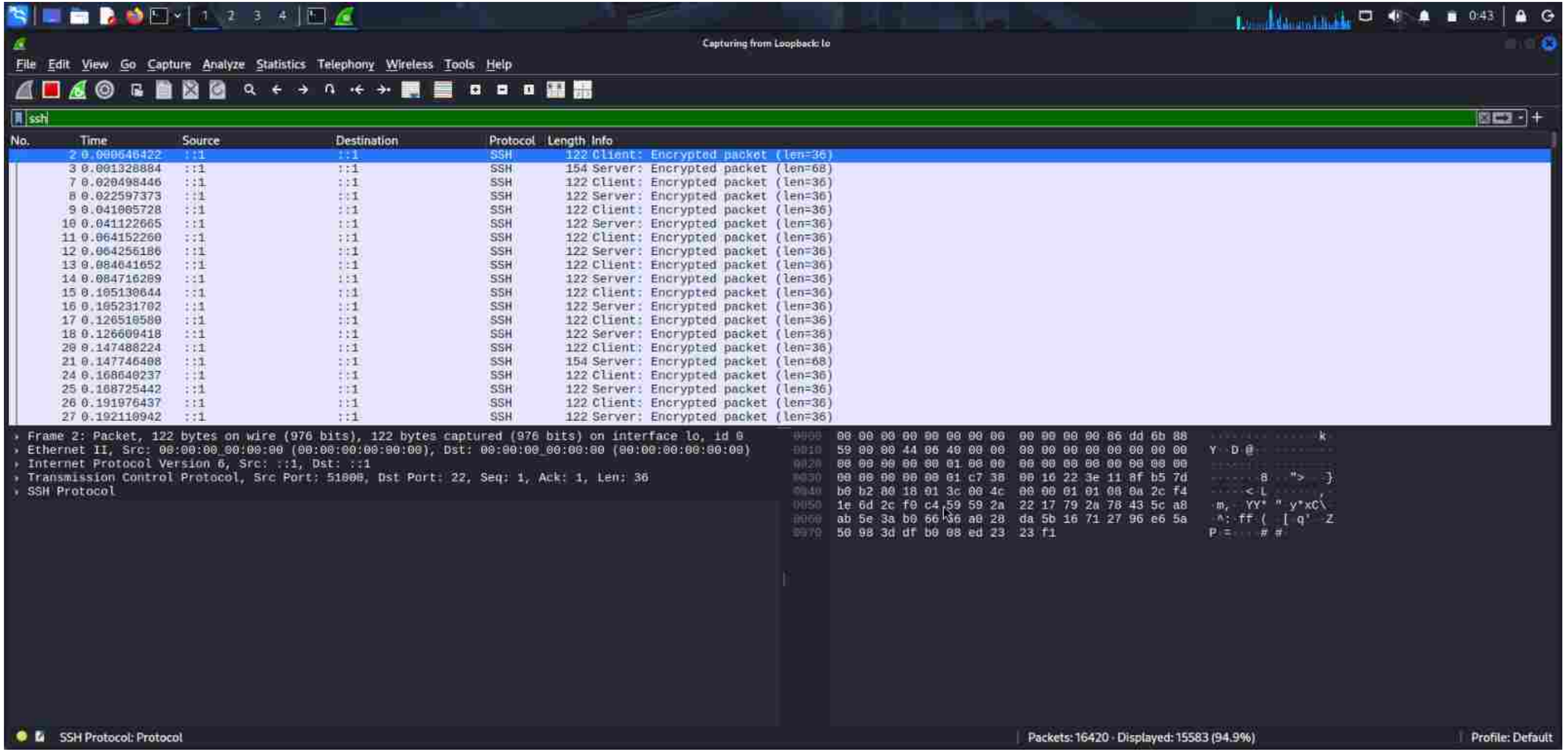
1. Stop Wireshark capture
2. Apply filter:

```
ssh
```

3. Open packets

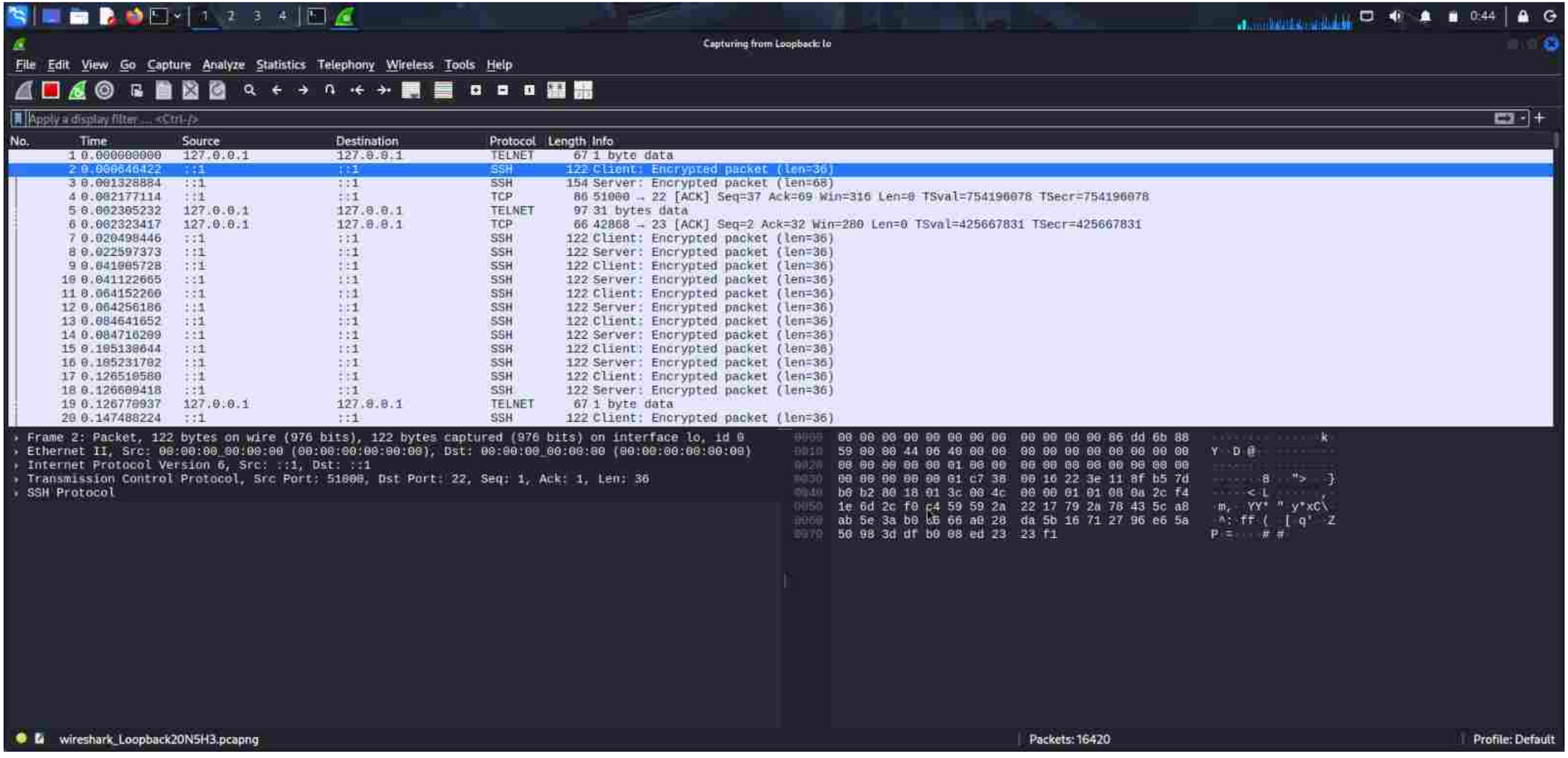
Observation:

- Data is encrypted
- No readable username/password
- Payload unreadable


Capturing from Loopback: lo
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ssh
No. Time Source Destination Protocol Length Info
2 0.000646422 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
3 0.001328884 ::1 ::1 SSH 154 Server: Encrypted packet (len=68)
7 0.020498446 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
8 0.022597373 ::1 ::1 SSH 122 Server: Encrypted packet (len=36)
9 0.041005728 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
10 0.041122665 ::1 ::1 SSH 122 Server: Encrypted packet (len=36)
11 0.064152260 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
12 0.064256186 ::1 ::1 SSH 122 Server: Encrypted packet (len=36)
13 0.084641652 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
14 0.084716269 ::1 ::1 SSH 122 Server: Encrypted packet (len=36)
15 0.105130644 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
16 0.105231702 ::1 ::1 SSH 122 Server: Encrypted packet (len=36)
17 0.126516580 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
18 0.126609418 ::1 ::1 SSH 122 Server: Encrypted packet (len=36)
20 0.147488224 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
21 0.147746408 ::1 ::1 SSH 154 Server: Encrypted packet (len=68)
24 0.168640237 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
25 0.168725442 ::1 ::1 SSH 122 Server: Encrypted packet (len=36)
26 0.191976437 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
27 0.192110942 ::1 ::1 SSH 122 Server: Encrypted packet (len=36)
Frame 2: Packet, 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 6, Src: ::1, Dst: ::1
Transmission Control Protocol, Src Port: 51808, Dst Port: 22, Seq: 1, Ack: 1, Len: 36
SSH Protocol
0000 00 00 00 00 00 00 00 00 00 00 00 00 86 dd 6b 88 ..... k
0010 59 00 00 44 06 40 00 00 00 00 00 00 00 00 00 00 Y D @ .....
0020 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 .....
0030 00 00 00 00 00 01 c7 38 00 16 22 3e 11 8f b5 7d ..... 8 ">
0040 b0 b2 80 18 01 3c 00 4c 00 00 01 01 08 0a 2c f4 ..... < L ..... ,
0050 1e 6d 2c f0 c4 59 59 2a 22 17 79 2a 78 43 5c a8 .m, YY* " y*xC\
0060 ab 5e 3a b0 66 36 a0 28 da 5b 16 71 27 96 e6 5a ^: f6 ([q' Z
0070 50 98 3d df b0 08 ed 23 23 f1 P = ... # #
SSH Protocol: Protocol
Packets: 16420 · Displayed: 15583 (94.9%)
Profile: Default

Step 11: Comparison Observation Table

Features	Telnet	SSH
Port	23	22
Encryption	No	Yes
Password visible	Yes	No
Security Level	Low	High


Capturing from Loopback: lo
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 0.000000000 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
2 0.000646422 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
3 0.001328884 ::1 ::1 SSH 154 Server: Encrypted packet (len=68)
4 0.002177114 ::1 ::1 TCP 86 51000 → 22 [ACK] Seq=37 Ack=69 Win=316 Len=0 TSval=754190078 TSecr=754190078
5 0.002305232 127.0.0.1 127.0.0.1 TELNET 97 31 bytes data
6 0.002323417 127.0.0.1 127.0.0.1 TCP 66 42868 → 23 [ACK] Seq=2 Ack=32 Win=280 Len=0 TSval=425667831 TSecr=425667831
7 0.020498446 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
8 0.022597373 ::1 ::1 SSH 122 Server: Encrypted packet (len=36)
9 0.041005728 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
10 0.041122665 ::1 ::1 SSH 122 Server: Encrypted packet (len=36)
11 0.064152260 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
12 0.064256186 ::1 ::1 SSH 122 Server: Encrypted packet (len=36)
13 0.084641652 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
14 0.084716209 ::1 ::1 SSH 122 Server: Encrypted packet (len=36)
15 0.105138644 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
16 0.105231702 ::1 ::1 SSH 122 Server: Encrypted packet (len=36)
17 0.126510580 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
18 0.126609418 ::1 ::1 SSH 122 Server: Encrypted packet (len=36)
19 0.126770937 127.0.0.1 127.0.0.1 TELNET 67 1 byte data
20 0.147488224 ::1 ::1 SSH 122 Client: Encrypted packet (len=36)
Frame 2: Packet, 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 6, Src: ::1, Dst: ::1
Transmission Control Protocol, Src Port: 51000, Dst Port: 22, Seq: 1, Ack: 1, Len: 36
SSH Protocol
0000 00 00 00 00 00 00 00 00 00 00 00 00 86 dd 6b 88 ..... k
0010 59 00 00 44 06 40 00 00 00 00 00 00 00 00 00 00 Y D @ .....
0020 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 .....
0030 00 00 00 00 00 01 c7 38 00 16 22 3e 11 8f b5 7d ..... 8 ">
0040 b0 b2 80 18 01 3c 00 4c 00 00 01 01 08 0a 2c f4 ..... < L ..... ,
0050 1e 6d 2c f0 c4 59 59 2a 22 17 79 2a 78 43 5c a8 .m, YY* " y*xC\
0060 ab 5e 3a b0 b6 66 a0 28 da 5b 16 71 27 96 e6 5a ^: f ([q' Z
0070 50 98 3d df b0 08 ed 23 23 f1 P = # #
wireshark_Loopback20N5H3.pcapng
Packets: 16420
Profile: Default

Step 12: Result

- Telnet sends data in plain text
- SSH encrypts all data
- Telnet is unsafe
- SSH is secure and recommended

Step 13: Conclusion

This experiment proves that Telnet is insecure and vulnerable to sniffing attacks, while SSH provides strong encryption and secure communication. Hence, SSH should always be used in real-world networks.


THANK YOU