

Project-2: Comparative Analysis of Telnet and SSH Using Wireshark -

1. Kali Linux Hostname / User Configuration:

A new user (shivam) was created and user switching was tested to simulate real-world non-root access.

```
(kali㉿kali)-[/home]
└─$ sudo adduser shivam
[sudo] password for kali:
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for shivam
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]

(kali㉿kali)-[/home]
└─$ sudo su shivam
(shivam㉿kali)-[/home]
└─$ whoami
shivam
```

2. Telnet Service Configuration and Activation:

Telnet service was enabled using openbsd-inetd to demonstrate insecure plaintext communication.

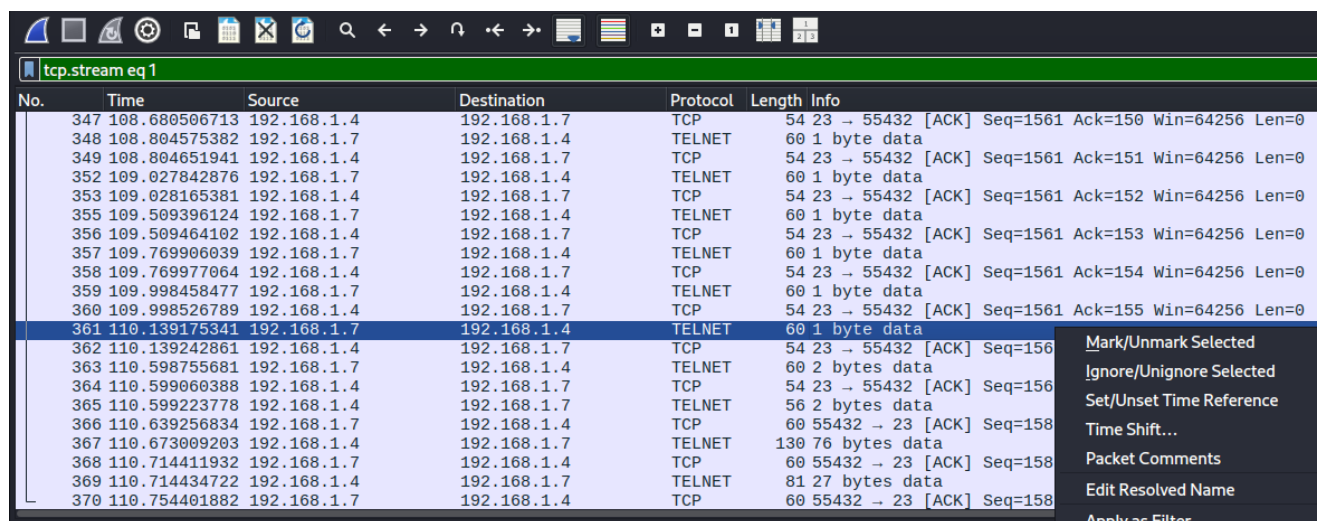
```
(shivam@kali)-[/home]
$ sudo systemctl start openbsd-inetd

(shivam@kali)-[/home]
$ telnet localhost
Trying ::1 ...
Connected to localhost.
Escape character is '^]'.

Linux 6.12.38+kali-amd64 (kali) (pts/2)
kali login: █
```

3. Wireshark Capturing Telnet Traffic:

Wireshark was used on Kali Linux to capture Telnet packets on TCP port 23.



No.	Time	Source	Destination	Protocol	Length	Info
347	108.680506713	192.168.1.4	192.168.1.7	TCP	54	23 → 55432 [ACK] Seq=1561 Ack=150 Win=64256 Len=0
348	108.804575382	192.168.1.7	192.168.1.4	TELNET	60	1 byte data
349	108.804651941	192.168.1.4	192.168.1.7	TCP	54	23 → 55432 [ACK] Seq=1561 Ack=151 Win=64256 Len=0
352	109.027842876	192.168.1.7	192.168.1.4	TELNET	60	1 byte data
353	109.028165381	192.168.1.4	192.168.1.7	TCP	54	23 → 55432 [ACK] Seq=1561 Ack=152 Win=64256 Len=0
355	109.509396124	192.168.1.7	192.168.1.4	TELNET	60	1 byte data
356	109.509464102	192.168.1.4	192.168.1.7	TCP	54	23 → 55432 [ACK] Seq=1561 Ack=153 Win=64256 Len=0
357	109.769906039	192.168.1.7	192.168.1.4	TELNET	60	1 byte data
358	109.769977064	192.168.1.4	192.168.1.7	TCP	54	23 → 55432 [ACK] Seq=1561 Ack=154 Win=64256 Len=0
359	109.998458477	192.168.1.7	192.168.1.4	TELNET	60	1 byte data
360	109.998526789	192.168.1.4	192.168.1.7	TCP	54	23 → 55432 [ACK] Seq=1561 Ack=155 Win=64256 Len=0
361	110.139175341	192.168.1.7	192.168.1.4	TELNET	60	1 byte data
362	110.139242861	192.168.1.4	192.168.1.7	TCP	54	23 → 55432 [ACK] Seq=1561 Ack=156 Win=64256 Len=0
363	110.598755681	192.168.1.7	192.168.1.4	TELNET	60	2 bytes data
364	110.599060388	192.168.1.4	192.168.1.7	TCP	54	23 → 55432 [ACK] Seq=1561 Ack=157 Win=64256 Len=0
365	110.599223778	192.168.1.7	192.168.1.4	TELNET	56	2 bytes data
366	110.639256834	192.168.1.4	192.168.1.7	TCP	60	55432 → 23 [ACK] Seq=1581 Ack=158 Win=64256 Len=0
367	110.673009203	192.168.1.7	192.168.1.4	TELNET	130	76 bytes data
368	110.714411932	192.168.1.4	192.168.1.7	TCP	60	55432 → 23 [ACK] Seq=1581 Ack=159 Win=64256 Len=0
369	110.714434722	192.168.1.7	192.168.1.4	TELNET	81	27 bytes data
370	110.754401882	192.168.1.4	192.168.1.7	TCP	60	55432 → 23 [ACK] Seq=1581 Ack=160 Win=64256 Len=0

4. Telnet Login and Command Execution:

Telnet login was performed from Windows and commands were executed.

```
Telnet 192.168.1.2

Linux 6.12.38+kali-amd64 (kali) (pts/3)

kali login: shivam

Password:
ΓöîΓöÇΓöÇ(shivamπë_7kali)-[~]
ΓööΓöÇ$ ls

ΓöîΓöÇΓöÇ(shivamπë_7kali)-[~]
ΓööΓöÇ$

ΓöîΓöÇΓöÇ(shivamπë_7kali)-[~]
ΓööΓöÇ$ cd ..

ΓöîΓöÇΓöÇ(shivamπë_7kali)-[/home]
ΓööΓöÇ$

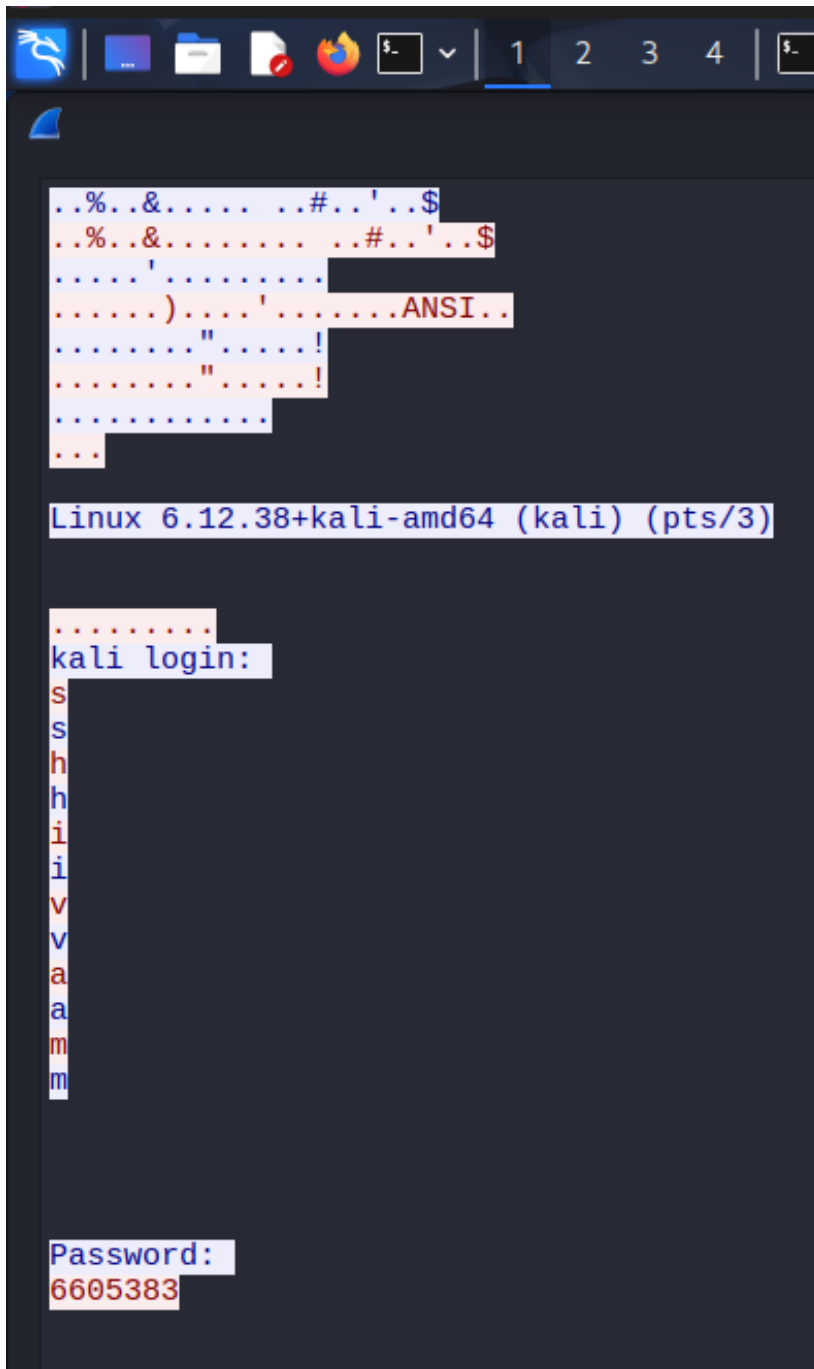
ΓöîΓöÇΓöÇ(shivamπë_7kali)-[/home]
ΓööΓöÇ$ ls
kali practice1 shivam Test_folder zsh zshes

ΓöîΓöÇΓöÇ(shivamπë_7kali)-[/home]
ΓööΓöÇ$

ΓöîΓöÇΓöÇ(shivamπë_7kali)-[/home]
ΓööΓöÇ$ sudo su practice1
[sudo] password for shivam:
ΓöîΓöÇΓöÇ(practice1πë_7kali)-[/home]
ΓööΓöÇ$ ls
kali practice1 shivam Test_folder zsh zshes
```

5. Plain-text Credentials Visible in Packets:

Telnet TCP stream shows username and password in plaintext.

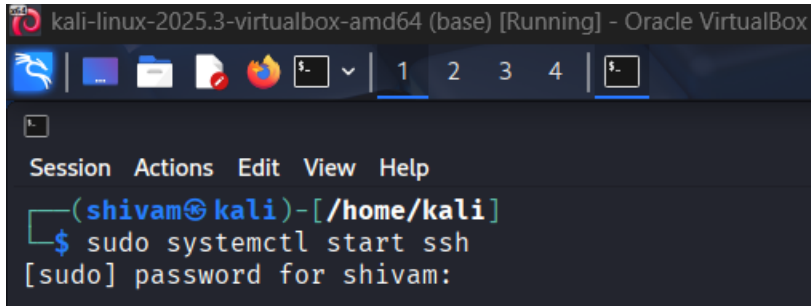


The image shows a Wireshark packet capture of a Telnet session. The top toolbar includes icons for the network analyzer, packet list, packet details, packet bytes, and packet capture status. The packet list pane shows several packets, with the first packet selected. The packet details pane shows the structure of the selected packet, including the Ethernet II header, Internet Protocol Version 4 header, and Transmission Control Protocol header. The packet bytes pane shows the raw data of the selected packet, which is a Telnet login sequence. The data is displayed in a hex dump format, with the ASCII column showing the plaintext credentials. The first line of the ASCII column shows the prompt 'Linux 6.12.38+kali-amd64 (kali) (pts/3)'. The second line shows the prompt 'kali login:'. The third line shows the username 'sshhhhiiiiivvaamm'. The fourth line shows the prompt 'Password:'. The fifth line shows the password '6605383'.

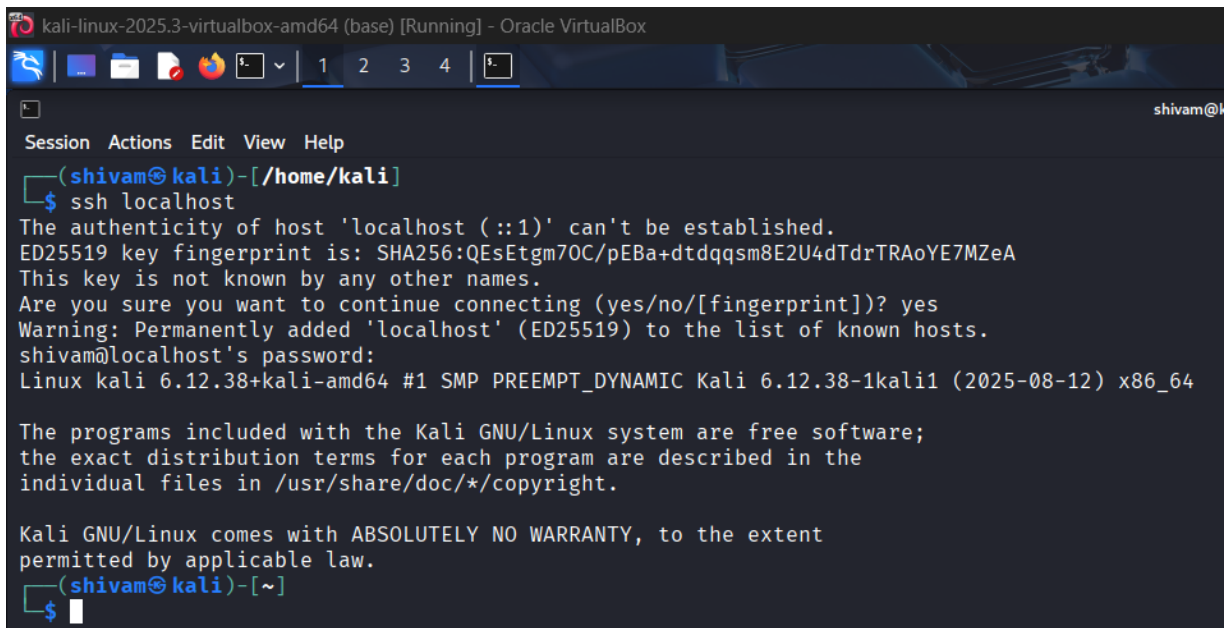
```
Linux 6.12.38+kali-amd64 (kali) (pts/3)
kali login:
sshhhhiiiiivvaamm
Password:
6605383
```

6. SSH Service Configuration and Activation:

SSH service was verified and started on Kali Linux.

A terminal window titled 'kali-linux-2025.3-virtualbox-amd64 (base) [Running] - Oracle VirtualBox'. The terminal shows a user 'shivam' at 'kali' in the directory '/home/kali'. They run the command 'sudo systemctl start ssh'. The prompt changes to '[sudo] password for shivam:'.

```
kali-linux-2025.3-virtualbox-amd64 (base) [Running] - Oracle VirtualBox
(shivam@kali)-[/home/kali]
$ sudo systemctl start ssh
[sudo] password for shivam:
```

A terminal window titled 'kali-linux-2025.3-virtualbox-amd64 (base) [Running] - Oracle VirtualBox'. The user 'shivam' at 'kali' runs 'ssh localhost'. The terminal displays a warning about the authenticity of the host 'localhost (:::1)' and its SHA256 fingerprint. The user confirms the connection. The terminal then shows the user's password prompt and the system version information: 'Linux kali 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1kali1 (2025-08-12) x86_64'. It also displays the Kali GNU/Linux license information.

```
kali-linux-2025.3-virtualbox-amd64 (base) [Running] - Oracle VirtualBox
(shivam@kali)-[/home/kali]
$ ssh localhost
The authenticity of host 'localhost (:::1)' can't be established.
ED25519 key fingerprint is: SHA256:QEsEtgm70C/pEBa+dttdqsm8E2U4dTdrTRAoYE7MZEA
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.
shivam@localhost's password:
Linux kali 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1kali1 (2025-08-12) 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.
(shivam@kali)-[~]
$
```

7. Wireshark Capturing SSH Traffic:

Wireshark captured SSH traffic on TCP port 22.

ssh						
No.	Time	Source	Destination	Protocol	Length	Info
17	40.441976821	192.168.1.7	192.168.1.4	SSHv2	87	Client: Protocol (SSH-2.0-OpenSSH_for_Windows_9.5)
19	40.450356727	192.168.1.4	192.168.1.7	SSHv2	87	Server: Protocol (SSH-2.0-OpenSSH_10.2p1 Debian-3)
20	40.459104103	192.168.1.4	192.168.1.7	SSHv2	1094	Server: Key Exchange Init
22	40.460988745	192.168.1.7	192.168.1.4	SSHv2	1486	Client: Key Exchange Init
24	40.508767061	192.168.1.7	192.168.1.4	SSHv2	102	Client: Elliptic Curve Diffie-Hellman Key Exchange Init
26	40.512470544	192.168.1.4	192.168.1.7	SSHv2	546	Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys,
27	40.516487848	192.168.1.7	192.168.1.4	SSHv2	70	Client: New Keys
31	40.557678975	192.168.1.7	192.168.1.4	SSHv2	98	Client: Encrypted packet (len=44)
33	40.557875460	192.168.1.4	192.168.1.7	SSHv2	98	Server: Encrypted packet (len=44)
34	40.558318225	192.168.1.7	192.168.1.4	SSHv2	122	Client: Encrypted packet (len=68)
35	40.561657766	192.168.1.4	192.168.1.7	SSHv2	106	Server: Encrypted packet (len=52)
42	51.278751164	192.168.1.7	192.168.1.4	SSHv2	202	Client: Encrypted packet (len=148)
45	53.396968954	192.168.1.4	192.168.1.7	SSHv2	106	Server: Encrypted packet (len=52)
67	128.225295854	192.168.1.7	192.168.1.4	SSHv2	87	Client: Protocol (SSH-2.0-OpenSSH_for_Windows_9.5)
69	128.233289006	192.168.1.4	192.168.1.7	SSHv2	87	Server: Protocol (SSH-2.0-OpenSSH_10.2p1 Debian-3)
70	128.236554932	192.168.1.7	192.168.1.4	SSHv2	1486	Client: Key Exchange Init
71	128.241848773	192.168.1.4	192.168.1.7	SSHv2	1094	Server: Key Exchange Init
72	128.243368667	192.168.1.7	192.168.1.4	SSHv2	102	Client: Elliptic Curve Diffie-Hellman Key Exchange Init
73	128.247259023	192.168.1.4	192.168.1.7	SSHv2	546	Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys,
74	128.251282195	192.168.1.7	192.168.1.4	SSHv2	70	Client: New Keys
76	128.292729784	192.168.1.7	192.168.1.4	SSHv2	98	Client: Encrypted packet (len=44)
78	128.293060623	192.168.1.4	192.168.1.7	SSHv2	98	Server: Encrypted packet (len=44)
79	128.293350724	192.168.1.7	192.168.1.4	SSHv2	122	Client: Encrypted packet (len=68)
80	128.294508691	192.168.1.4	192.168.1.7	SSHv2	106	Server: Encrypted packet (len=52)
83	132.533014028	192.168.1.7	192.168.1.4	SSHv2	202	Client: Encrypted packet (len=148)
85	132.612166408	192.168.1.4	192.168.1.7	SSHv2	82	Server: Encrypted packet (len=52)
86	132.617637815	192.168.1.7	192.168.1.4	SSHv2	166	Client: Encrypted packet (len=148)
88	132.661448887	192.168.1.4	192.168.1.7	SSHv2	682	Server: Encrypted packet (len=52)
89	132.662248800	192.168.1.4	192.168.1.7	SSHv2	98	Server: Encrypted packet (len=44)
91	132.665195477	192.168.1.7	192.168.1.4	SSHv2	106	Client: Encrypted packet (len=52)
93	132.666172179	192.168.1.4	192.168.1.7	SSHv2	126	Server: Encrypted packet (len=68)
94	132.669717799	192.168.1.4	192.168.1.7	SSHv2	130	Server: Encrypted packet (len=70)
95	132.669987826	192.168.1.4	192.168.1.7	SSHv2	158	Server: Encrypted packet (len=102)
97	132.670298771	192.168.1.4	192.168.1.7	SSHv2	126	Server: Encrypted packet (len=68)

Mark/Unmark Selected	Ctrl+M
Ignore/Unignore Selected	Ctrl+D
Set/Unset Time Reference	Ctrl+T
Time Shift...	Ctrl+Shift+T
Packet Comments	
Edit Resolved Name	
Apply as Filter	
Prepare as Filter	
Conversation Filter	
Colorize Conversation	
SCTP	
Follow	TCP Stream Ct

8. SSH Login and Command Execution:

SSH login from Windows to Kali was successful.

practice1@kali: /home

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements!

PS C:\Users\nucle> ssh shivam@192.168.1.4 24

shivam@192.168.1.4's password:

bash: line 1: 24: command not found

PS C:\Users\nucle> ssh shivam@192.168.1.4

shivam@192.168.1.4's password:

Linux kali 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38

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.

Last login: Sat Dec 20 04:13:02 2025 from ::1

(shivam@kali)~

\$ ls

(shivam@kali)~

\$ ls

(shivam@kali)~

\$ cd ..

(shivam@kali)~/home

\$ ls

kali practice1 shivam Test_folder zsh zshes

(shivam@kali)~/home

\$ echo SHIVAM USER

SHIVAM USER

(shivam@kali)~/home

\$ sudo su practice1

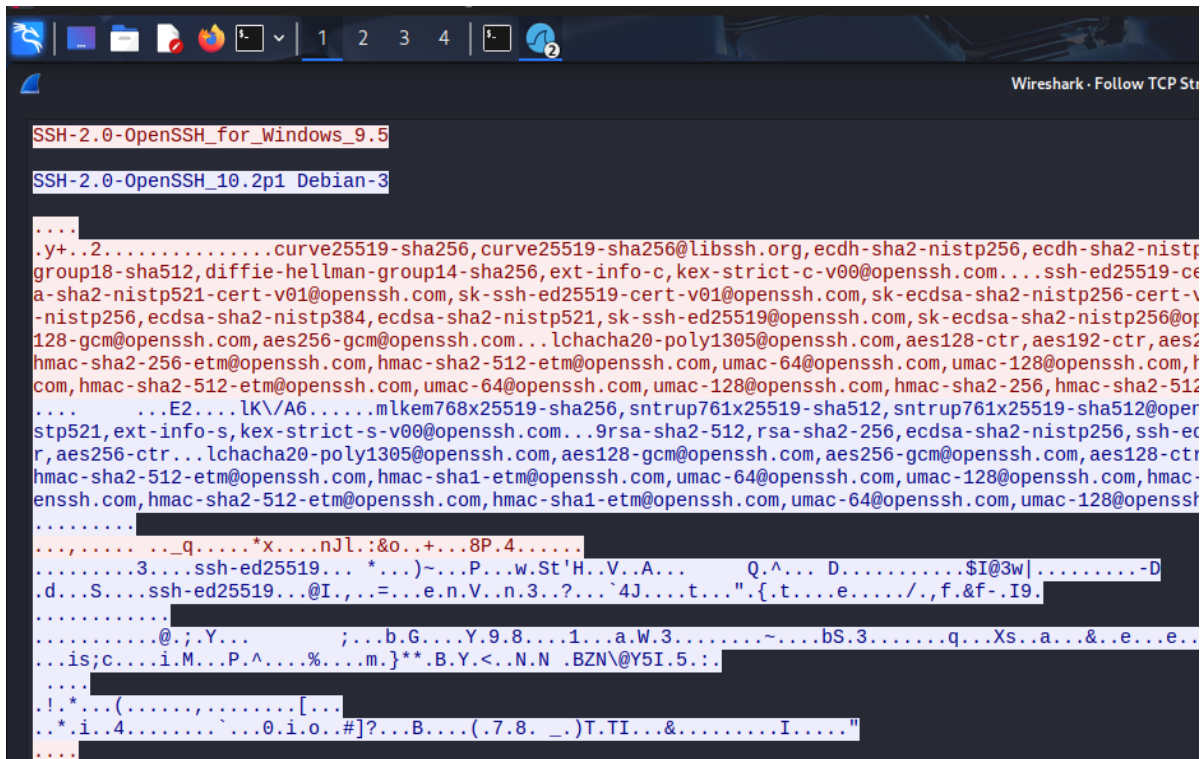
[sudo] password for shivam:

(practice1@kali)~/home

\$

9. Encrypted SSH Packets (Unreadable Payload)

SSH TCP stream shows encrypted unreadable payload.



Final Comparison Summary

Telnet transmits credentials in plaintext and is insecure, whereas SSH encrypts all communication, making it safe against packet sniffing attacks.

Telnet runs over TCP port 23 and does not encrypt data. Enabling it intentionally creates an insecure service to demonstrate real-world risks in legacy systems.

Wireshark captures packets at the network interface level. By filtering Telnet traffic, we can clearly observe how data travels without encryption.

User credentials and commands are transmitted directly over the network. This simulates how attackers can sniff sensitive data on unsecured networks.

The visibility of usernames and passwords proves Telnet is insecure. This is a critical vulnerability that violates basic security principles.

SSH replaces Telnet by providing encrypted remote access. It is the industry standard for secure system administration.

Although packets are captured, their content is unreadable. This demonstrates the effectiveness of cryptographic protection.

Even login credentials and commands remain encrypted. This protects users against packet sniffing and man-in-the-middle attacks.

The payload appears as random data due to encryption algorithms. This ensures confidentiality even if traffic is intercepted.

Telnet should never be used in modern networks. SSH is secure, reliable, and resistant to passive network attacks.