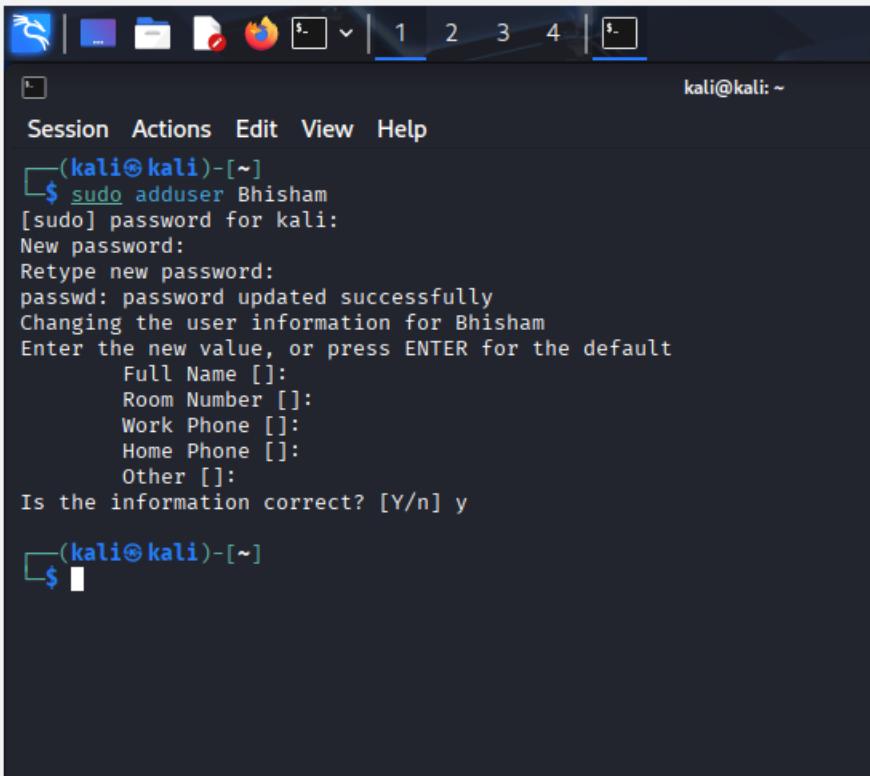


Project-2: Comparative Analysis of Telnet and SSH Using Wireshark –

By: Bhisham Sahu

1. Kali Linux Hostname / User Configuration:

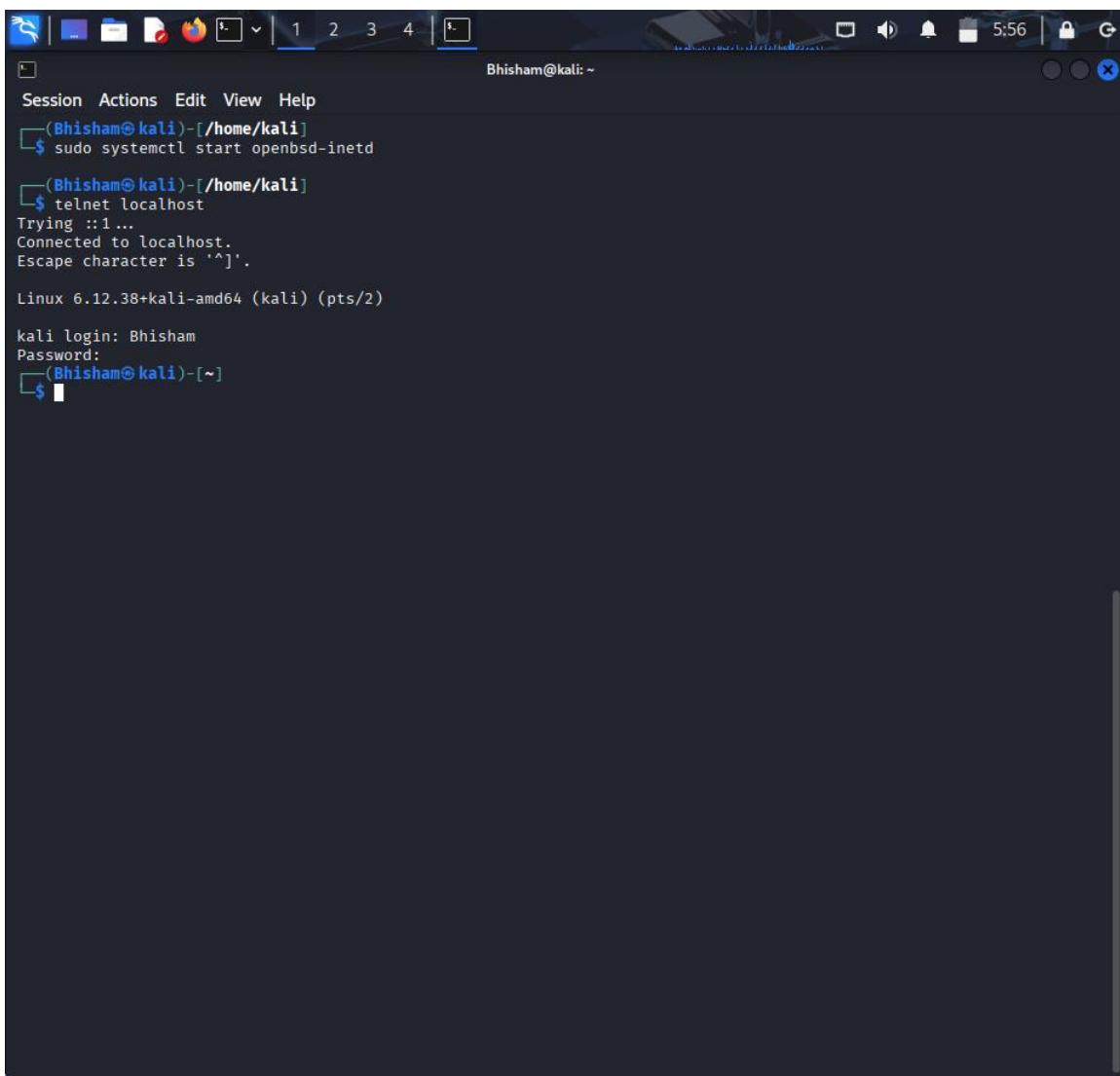
A non-root user (Bhisham) was created on Kali Linux and user switching was successfully performed to simulate a real-world multi-user operating environment.



```
(kali㉿kali)-[~]
$ sudo adduser Bhisham
[sudo] password for kali:
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for Bhisham
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] y
(kali㉿kali)-[~]
$
```

2. Telnet Service Configuration and Activation:

The Telnet service was enabled using `openbsd-inetd`, allowing plaintext communication for demonstrating protocol-level security weaknesses.



A screenshot of a terminal window titled "Bhisham@kali: ~". The terminal shows the following session:

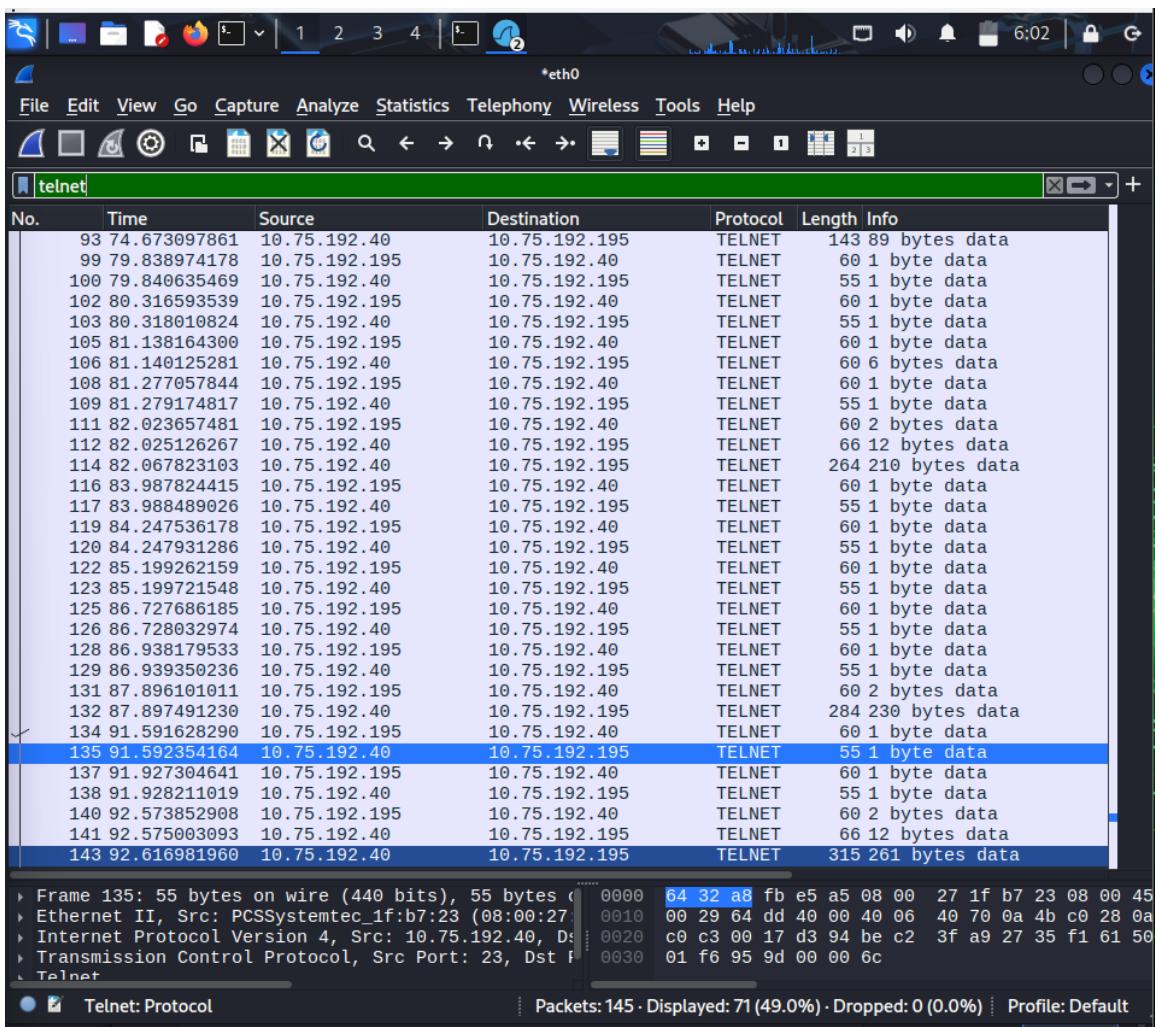
```
Session Actions Edit View Help
(Bhisham@kali)-[/home/kali]
$ sudo systemctl start openbsd-inetd
(Bhisham@kali)-[/home/kali]
$ telnet localhost
Trying ::1 ...
Connected to localhost.
Escape character is '^]'.

Linux 6.12.38+kali-amd64 (kali) (pts/2)

kali login: Bhisham
Password:
(Bhisham@kali)-[~]
```

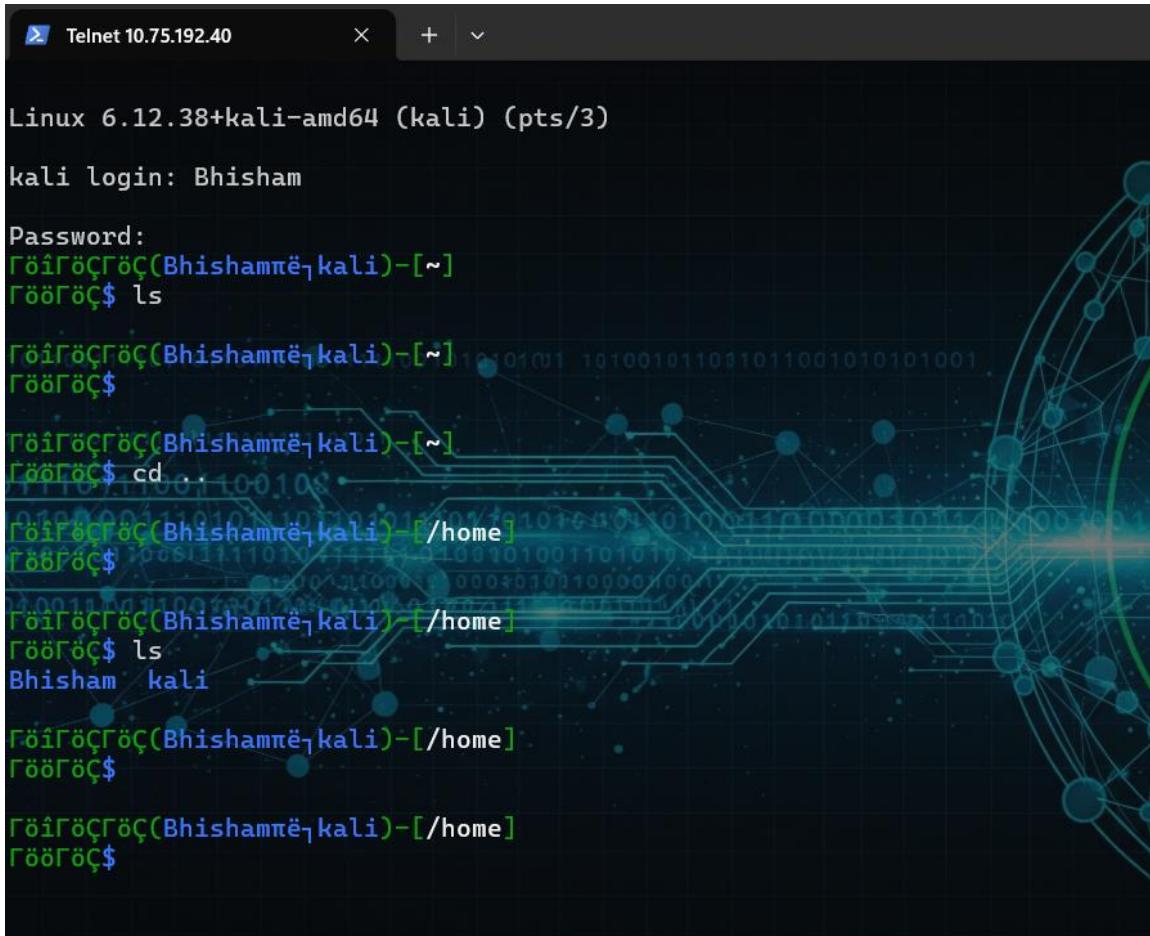
3. Wireshark Capturing Telnet Traffic:

Wireshark was used to capture Telnet network traffic on TCP port 23, enabling packet-level inspection of unencrypted data.



4. Telnet Login and Command Execution:

A successful Telnet login was performed from a Windows client, and commands were executed on the Kali Linux server.



```
Telnet 10.75.192.40
Linux 6.12.38+kali-amd64 (kali) (pts/3)

kali login: Bhisham

Password:
ΓöïΓöçΓöç(Bhishamπë_ë_kali)-[~]
ΓööΓöç$ ls

ΓöïΓöçΓöç(Bhishamπë_ë_kali)-[~]
ΓööΓöç$ cd ..

ΓöïΓöçΓöç(Bhishamπë_ë_kali)-[/home]
ΓööΓöç$ 

ΓöïΓöçΓöç(Bhishamπë_ë_kali)-[/home]
ΓööΓöç$ ls
Bhisham kali

ΓöïΓöçΓöç(Bhishamπë_ë_kali)-[/home]
ΓööΓöç$ 

ΓöïΓöçΓöç(Bhishamπë_ë_kali)-[/home]
ΓööΓöç$ 
```

5. Plain-text Credentials Visible in Packets:

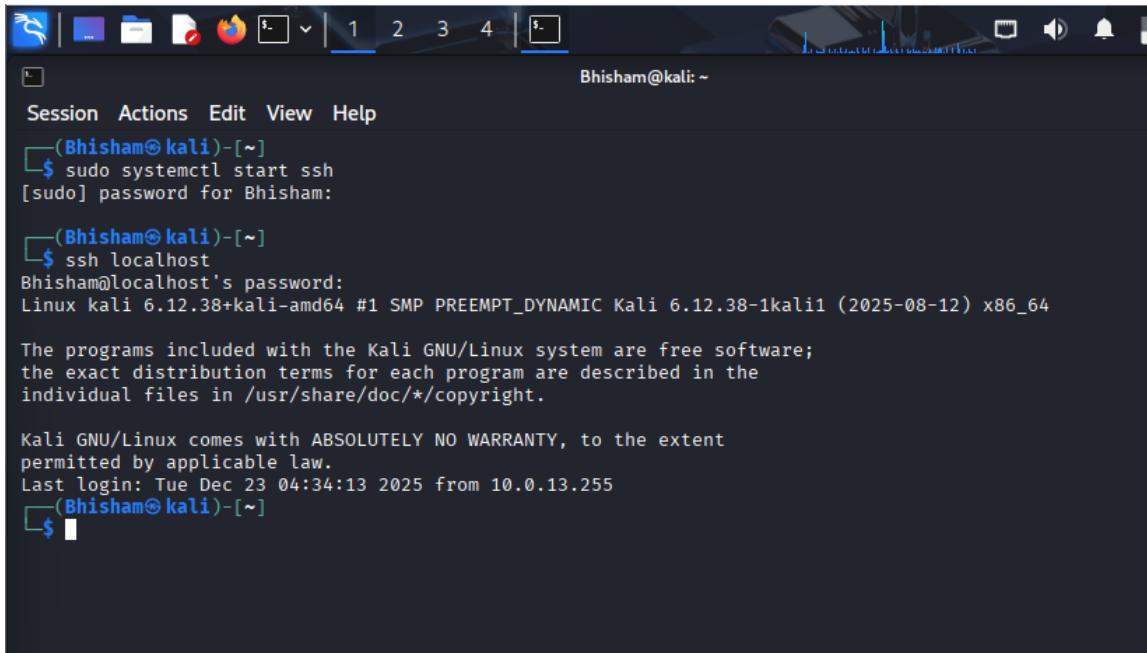
Usernames, passwords, and commands were observed in readable plaintext within the Telnet TCP stream, confirming the insecure nature of Telnet.

The screenshot shows a Wireshark window titled "Wireshark · Follow TCP Stream (tcp.stream eq 0) · eth0". The ASCII dump pane displays an SSH session. The client (Bhisham) has sent a login request to the server (kali). The server has responded with a password prompt, asking for the user's password. The client has entered the password "6605322". The session is currently at 54 turns.

```
..%..&.....#.'$  
..%..&.....#.'$  
.....'  
....x.....'....ANSI..  
....."  
....."  
.....  
Linux 6.12.38+kali-amd64 (kali) (pts/3)  
kali login:  
.....B  
B  
h  
h  
i  
i  
s  
s  
h  
h  
a  
a  
m  
m  
  
Password:  
6605322  
  
. [?2004h. [;32m.....(.[1;34mBhisham...kali.[;32m)-[.[0;1m~.[;32m]  
. [;32m.....[1;34m$. [0m  
l  
l  
a  
a  
38 client pkts, 33 server pkts, 54 turns.  
Entire conversation (1,092 bytes) Show as ASCII No delta times Stream 0  
Find: Filter Out This Stream Print Save as... Back × Close Help
```

6. SSH Service Configuration and Activation:

The SSH service was verified to be active, enabling secure and encrypted remote access.



A screenshot of a terminal window on a Kali Linux system. The terminal shows a user named Bhisham logging in via SSH. The session starts with:

```
Bhisham@kali:~
```

Session Actions Edit View Help

```
(Bhisham@kali)-[~]
$ sudo systemctl start ssh
[sudo] password for Bhisham:
```

The user enters their sudo password. After logging in, the user runs the command:

```
$ ssh localhost
Bhisham@localhost's password:
```

The user enters their local password. The terminal then displays the system information:

```
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.
Last login: Tue Dec 23 04:34:13 2025 from 10.0.13.255
(Bhisham@kali)-[~]
```

The user then types a command starting with '\$'.

7. Wireshark Capturing SSH Traffic:

SSH network traffic was captured on TCP port 22, allowing analysis of encrypted communication.

The screenshot shows a Wireshark capture window titled "ssh". The packet list pane displays 113 SSHv2 protocol frames. The columns in the table are: No., Time, Source, Destination, Protocol, Length, and Info. The "Info" column provides detailed descriptions of each frame's content, such as "Client: Protocol (SSH-", "Server: Protocol (SSH-", "Key Exchange I", "Elliptic Curve", "New Keys", and various "Encrypted pack" types. The details pane at the bottom shows the raw hex and ASCII data for frame 69, which is the first encrypted packet. The status bar at the bottom right indicates "Packets: 901 · Displayed: 789 (87.6%) · Dropped: 0 (0.0%) · Profile: Default".

No.	Time	Source	Destination	Protocol	Length	Info
69	55.9967428093	10.75.192.195	10.75.192.40	SSHv2	87	Client: Protocol (SSH-
71	55.995164594	10.75.192.40	10.75.192.195	SSHv2	87	Server: Protocol (SSH-
72	55.997939847	10.75.192.195	10.75.192.40	SSHv2	1486	Client: Key Exchange I
73	56.024059814	10.75.192.40	10.75.192.195	SSHv2	1094	Server: Key Exchange I
74	56.027570849	10.75.192.195	10.75.192.40	SSHv2	102	Client: Elliptic Curve
75	56.038018363	10.75.192.40	10.75.192.195	SSHv2	546	Server: Elliptic Curve
79	61.610012624	10.75.192.195	10.75.192.40	SSHv2	70	Client: New Keys
81	61.654753468	10.75.192.195	10.75.192.40	SSHv2	98	Client: Encrypted pack
83	61.655408863	10.75.192.40	10.75.192.195	SSHv2	98	Server: Encrypted pack
84	61.656487827	10.75.192.195	10.75.192.40	SSHv2	122	Client: Encrypted pack
85	61.660807000	10.75.192.40	10.75.192.195	SSHv2	106	Server: Encrypted pack
87	73.229932126	10.75.192.195	10.75.192.40	SSHv2	202	Client: Encrypted pack
89	73.376290294	10.75.192.40	10.75.192.195	SSHv2	82	Server: Encrypted pack
90	73.396494740	10.75.192.195	10.75.192.40	SSHv2	166	Client: Encrypted pack
92	73.470656639	10.75.192.40	10.75.192.195	SSHv2	682	Server: Encrypted pack
94	73.513659940	10.75.192.40	10.75.192.195	SSHv2	98	Server: Encrypted pack
95	73.515081119	10.75.192.195	10.75.192.40	SSHv2	190	Client: Encrypted pack
97	73.547984166	10.75.192.40	10.75.192.195	SSHv2	162	Server: Encrypted pack
98	73.550371210	10.75.192.40	10.75.192.195	SSHv2	522	Server: Encrypted pack
100	73.821193587	10.75.192.40	10.75.192.195	SSHv2	98	Server: Encrypted pack
101	73.822570941	10.75.192.40	10.75.192.195	SSHv2	202	Server: Encrypted pack
104	77.580105844	10.75.192.195	10.75.192.40	SSHv2	90	Client: Encrypted pack
105	77.583182823	10.75.192.40	10.75.192.195	SSHv2	90	Server: Encrypted pack
106	77.599603115	10.75.192.195	10.75.192.40	SSHv2	90	Client: Encrypted pack
107	77.599935933	10.75.192.40	10.75.192.195	SSHv2	90	Server: Encrypted pack
108	77.623339372	10.75.192.195	10.75.192.40	SSHv2	90	Client: Encrypted pack
109	77.623744771	10.75.192.40	10.75.192.195	SSHv2	90	Server: Encrypted pack
110	77.655347223	10.75.192.195	10.75.192.40	SSHv2	90	Client: Encrypted pack
111	77.6556333627	10.75.192.40	10.75.192.195	SSHv2	90	Server: Encrypted pack
112	77.686596703	10.75.192.195	10.75.192.40	SSHv2	90	Client: Encrypted pack
113	77.686869495	10.75.192.40	10.75.192.195	SSHv2	90	Server: Encrypted pack

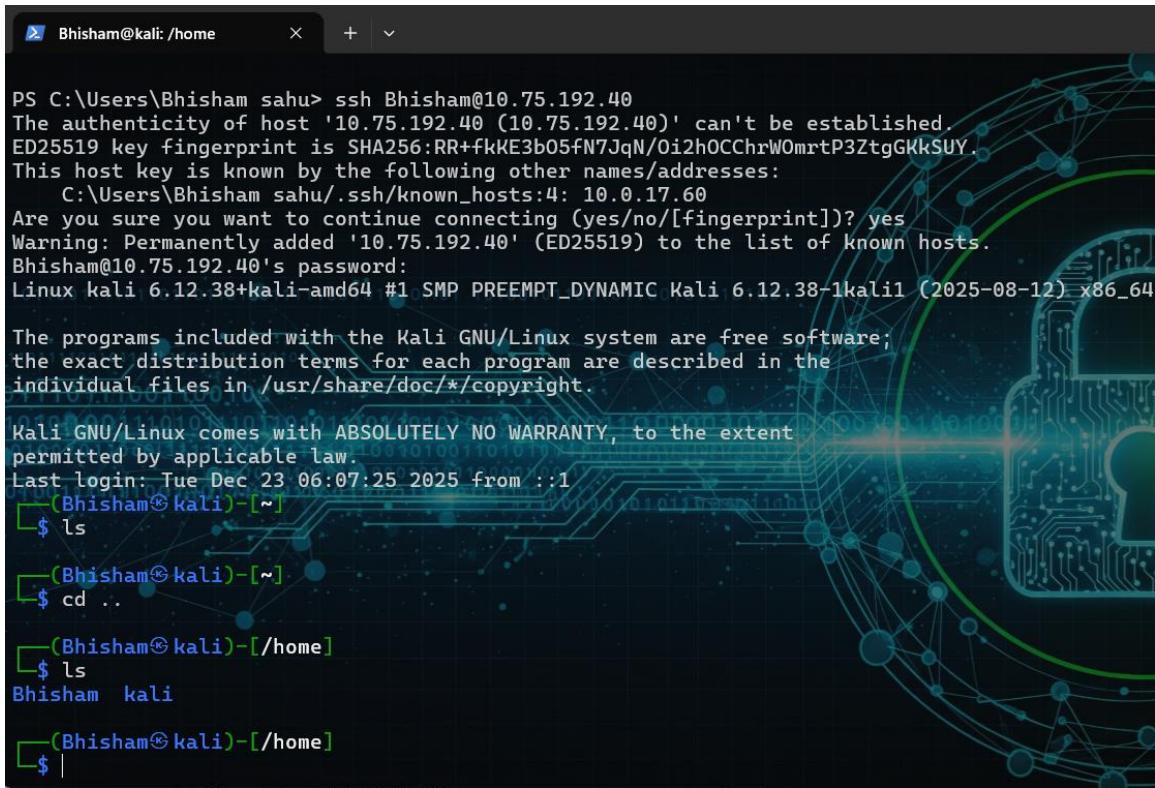
Frame 69: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface eth0
 Ethernet II, Src: Intel_fb:e5:a5 (64:32:a8:fb:e5:a5), Dst: (08:00:27:1f:b7:23)
 Internet Protocol Version 4, Src: 10.75.192.195, Dst: 10.75.192.40
 Transmission Control Protocol, Src Port: 65521, Dst Port: 22
 SSH Protocol

SSH Protocol: Protocol

Packets: 901 · Displayed: 789 (87.6%) · Dropped: 0 (0.0%) · Profile: Default

8. SSH Login and Command Execution:

Secure authentication and remote command execution were successfully performed using the SSH protocol.



```
Bhisham@kali: /home      X + ^

PS C:\Users\Bhisham sahu> ssh Bhisham@10.75.192.40
The authenticity of host '10.75.192.40 (10.75.192.40)' can't be established.
ED25519 key fingerprint is SHA256:RR+fkKE3b05fN7JqN/Oi2h0CChrW0mrtP3ZtgGKkSUY.
This host key is known by the following other names/addresses:
  C:\Users\Bhisham sahu/.ssh/known_hosts:4: 10.0.17.60
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.75.192.40' (ED25519) to the list of known hosts.
Bhisham@10.75.192.40'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.
Last login: Tue Dec 23 06:07:25 2025 from ::1
[Bhisham@kali] ~]
$ ls
[Bhisham@kali] ~]
$ cd ..
[Bhisham@kali] /home]
$ ls
Bhisham  kali
[Bhisham@kali] /home]
$ |
```

9. Encrypted SSH Packets (Unreadable Payload):

The SSH TCP stream payload appeared encrypted and unreadable, demonstrating effective cryptographic protection of data.

Wireshark - Follow TCP Stream (tcp.stream eq 1) - eth0

SSH-2.0-OpenSSH_for_Windows_9.5

SSH-2.0-OpenSSH_10.0p2 Debian-8

....C...K.*.Ag....curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-hellman-group14-sha256,ext-info-c,kex-strict-c-v00@openssh.com....ssh-ed25519-cert-v01@openssh.com,ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-sha2-nistp521-cert-v01@openssh.com,sk-ssh-ed25519-cert-v01@openssh.com,rsa-sha2-512-cert-v01@openssh.com,rsa-sha2-256-cert-v01@openssh.com,ssh-ed25519,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,sk-ssh-ed25519@openssh.com,sk-ecdsa-sha2-nistp256@openssh.com,rsa-sha2-512,rsa-sha2-256...lchacha20-poly1305@openssh.com,aes128ctr,aes192ctr,aes256ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com...lchacha20-poly1305@openssh.com,aes128ctr,aes192ctr,aes256ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com....umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512...umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512....none,zlib@openssh.com,zlib....none,zlib@openssh.com,zlib.....

....V.[...r.S^....mlkem768x25519-sha256,sstrup761x25519-sha512,sstrup761x25519-sha512@openssh.com,curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,ext-info-s,kex-strict-s-v00@openssh.com...9rsa-sha2-512,rsa-sha2-256,ecdsa-sha2-nistp256,ssh-ed25519...lchacha20-poly1305@openssh.com,aes128-gcm@openssh.com,aes256-gcm@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr...lchacha20-poly1305@openssh.com,aes128-gcm@openssh.com,aes256-gcm@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr...umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512...umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512....none,zlib@openssh.com....none,zlib@openssh.com.....

....Z.b..?VN..j9+.{.q+&..u.US+). .{.3....G....3V.Df?.3....W....d....N....Y(Q...~A:. .w.+....P.x....S....ssh-ed25519...@...\$.!....;y.o....mTI..#0D.jr..G5..Z6.0Xb....n.K..{}....1....Ww..Oz.H....d-v2....^....~..X..`bC.)Z-B..!.gM....\:\..1.a....xr#.D2>..r.b.q.[..cj5..)p.;....}.-w.).1)J@....d....i7..E..<....l....:\`g+a....E'9n.5..S~ s ..m.]

387 client pkts, 402 served pkts, 771 turns.

Entire conversation (33 kB) Show as ASCII No delta times Stream 1

Find: Case sensitive Find Next

10. Final Comparison Summary:

This project presents a comparative security analysis of Telnet and SSH using real-time network traffic capture with Wireshark. Telnet was intentionally configured to demonstrate its insecure design, as it transmits usernames, passwords, and commands in plaintext, making it highly vulnerable to packet sniffing attacks. In

contrast, SSH was implemented to provide encrypted communication, ensuring confidentiality and integrity of data even when network traffic is intercepted. The captured packets clearly highlight the difference between unencrypted and encrypted protocols.

Based on the observed results, the project concludes that Telnet should not be used in modern networks, whereas SSH is the secure and industry-standard protocol for remote system administration.