

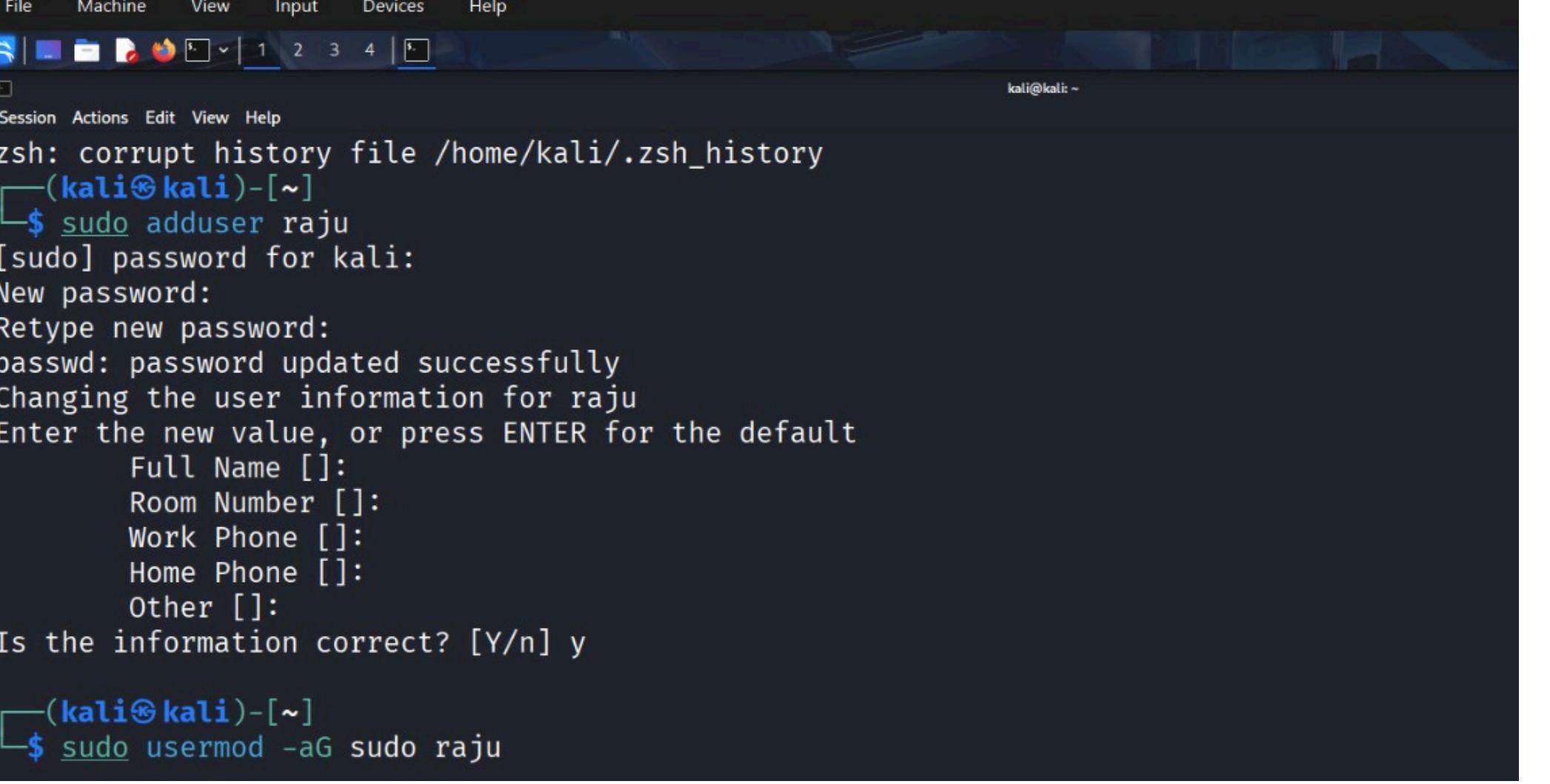
# Creating User name

**run : sudo adduser raju**

this will create user with raju name

**Run : sudo usermod -aG sudo raju**

this will add this user to sudo file



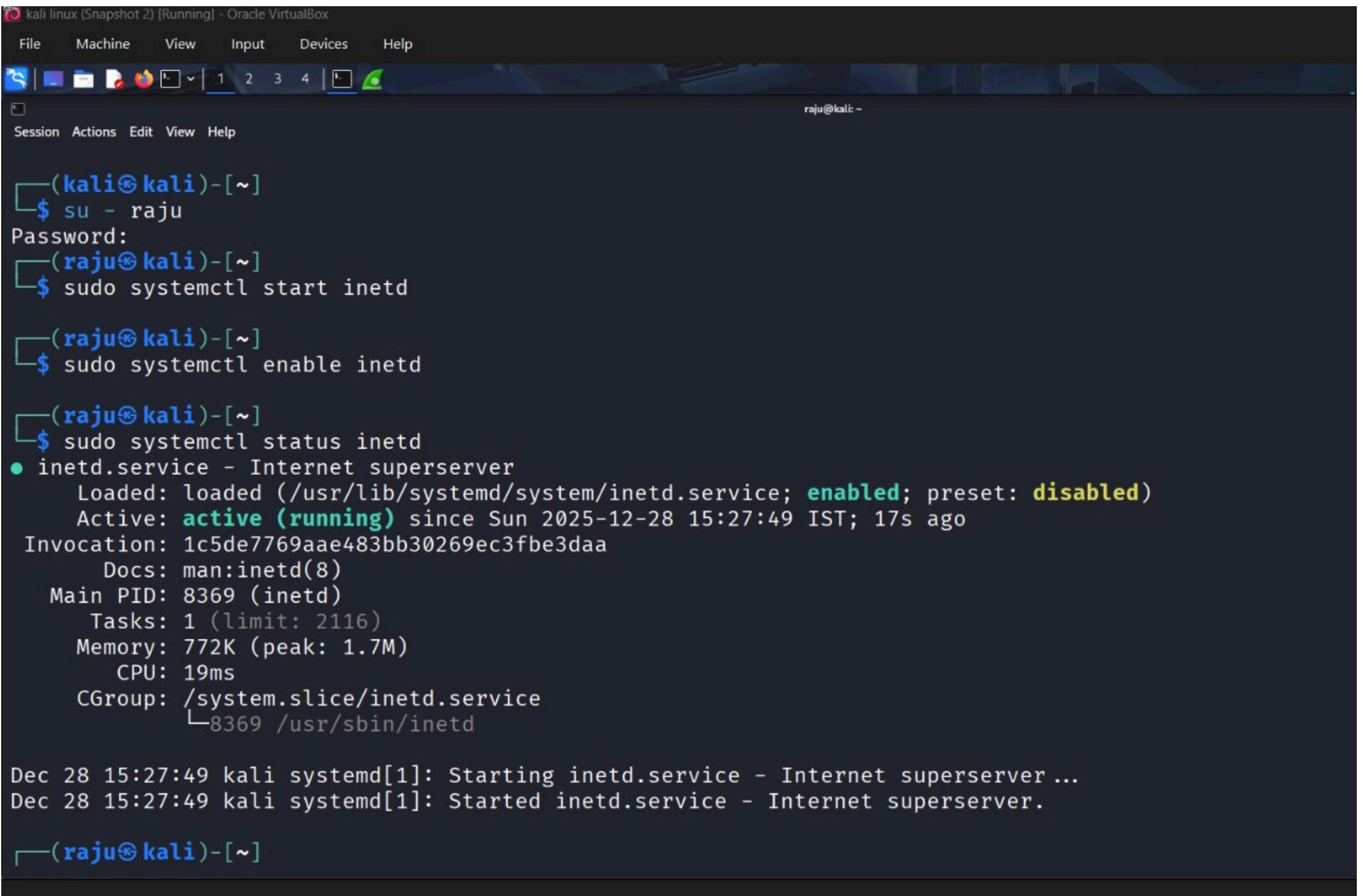
The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal title bar says "Session Actions Edit View Help". The window title is "zsh: corrupt history file /home/kali/.zsh\_history". The terminal prompt is "(kali㉿kali)-[~]". The user runs the command "\$ sudo adduser raju". A password prompt "[sudo] password for kali:" appears, followed by "New password:" and "Retype new password:". The password is successfully updated, and the message "passwd: password updated successfully" is displayed. Then, the user runs the command "\$ sudo usermod -aG sudo raju". The terminal asks for user information: "Changing the user information for raju", "Enter the new value, or press ENTER for the default", and lists fields for "Full Name []:", "Room Number []:", "Work Phone []:", "Home Phone []:", and "Other []:". Finally, the user is prompted "Is the information correct? [Y/n] y". The terminal ends with the prompt "(kali㉿kali)-[~]".

```
zsh: corrupt history file /home/kali/.zsh_history
[(kali㉿kali)-[~]]$ sudo adduser raju
[sudo] password for kali:
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for raju
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)-[~]]$ sudo usermod -aG sudo raju
```

# Checking status running or not

Run : su- raju

Run : sudo systemctl start  
inetd  
to start server.



The screenshot shows a terminal window titled "kali linux [Snapshot 2] [Running] - Oracle VirtualBox". The terminal session starts with the command "su - raju", followed by a password entry. Then, the user runs "sudo systemctl start inetd" and "sudo systemctl enable inetd". Finally, the user checks the status of the "inetd.service" with "sudo systemctl status inetd", which shows it is active (running) since the previous day. The terminal ends with a message from the kernel about starting the service and its successful start.

```
(kali㉿kali)-[~]
$ su - raju
Password:
(raju㉿kali)-[~]
$ sudo systemctl start inetd

(raju㉿kali)-[~]
$ sudo systemctl enable inetd

(raju㉿kali)-[~]
$ sudo systemctl status inetd
● inetd.service - Internet superserver
    Loaded: loaded (/usr/lib/systemd/system/inetd.service; enabled; preset: disabled)
    Active: active (running) since Sun 2025-12-28 15:27:49 IST; 17s ago
      Invocation: 1c5de7769aae483bb30269ec3fbe3daa
        Docs: man:inetd(8)
     Main PID: 8369 (inetd)
       Tasks: 1 (limit: 2116)
      Memory: 772K (peak: 1.7M)
        CPU: 19ms
       CGroup: /system.slice/inetd.service
               └─8369 /usr/sbin/inetd

Dec 28 15:27:49 kali systemd[1]: Starting inetd.service - Internet superserver ...
Dec 28 15:27:49 kali systemd[1]: Started inetd.service - Internet superserver.

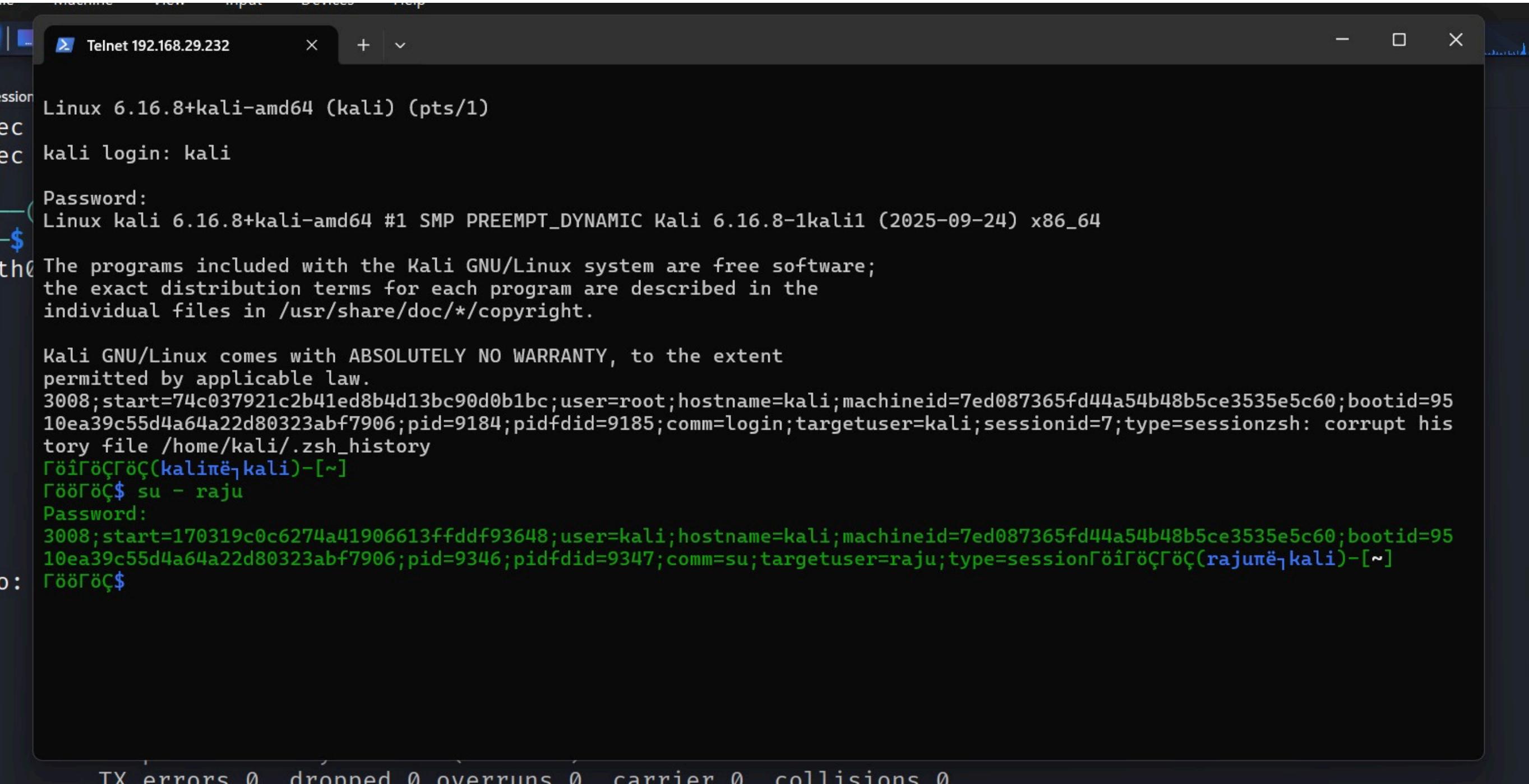
(raju㉿kali)-[~]
```

# Getting ip address

```
(raju㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.29.232 netmask 255.255.255.0 broadcast 192.168.29.255
        inet6 2405:201:3006:58d7:a00:27ff:fe1d:27 prefixlen 64 scopeid 0x0<global>
        inet6 fe80::a00:27ff:fe1d:27 prefixlen 64 scopeid 0x20<link>
        inet6 2405:201:3006:58d7:c7a3:b975:4cb7:4b94 prefixlen 64 scopeid 0x0<global>
        ether 08:00:27:1d:00:27 txqueuelen 1000 (Ethernet)
        RX packets 1226 bytes 97939 (95.6 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 304 bytes 34442 (33.6 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 8 bytes 480 (480.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 8 bytes 480 (480.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

# Getting access of victim os



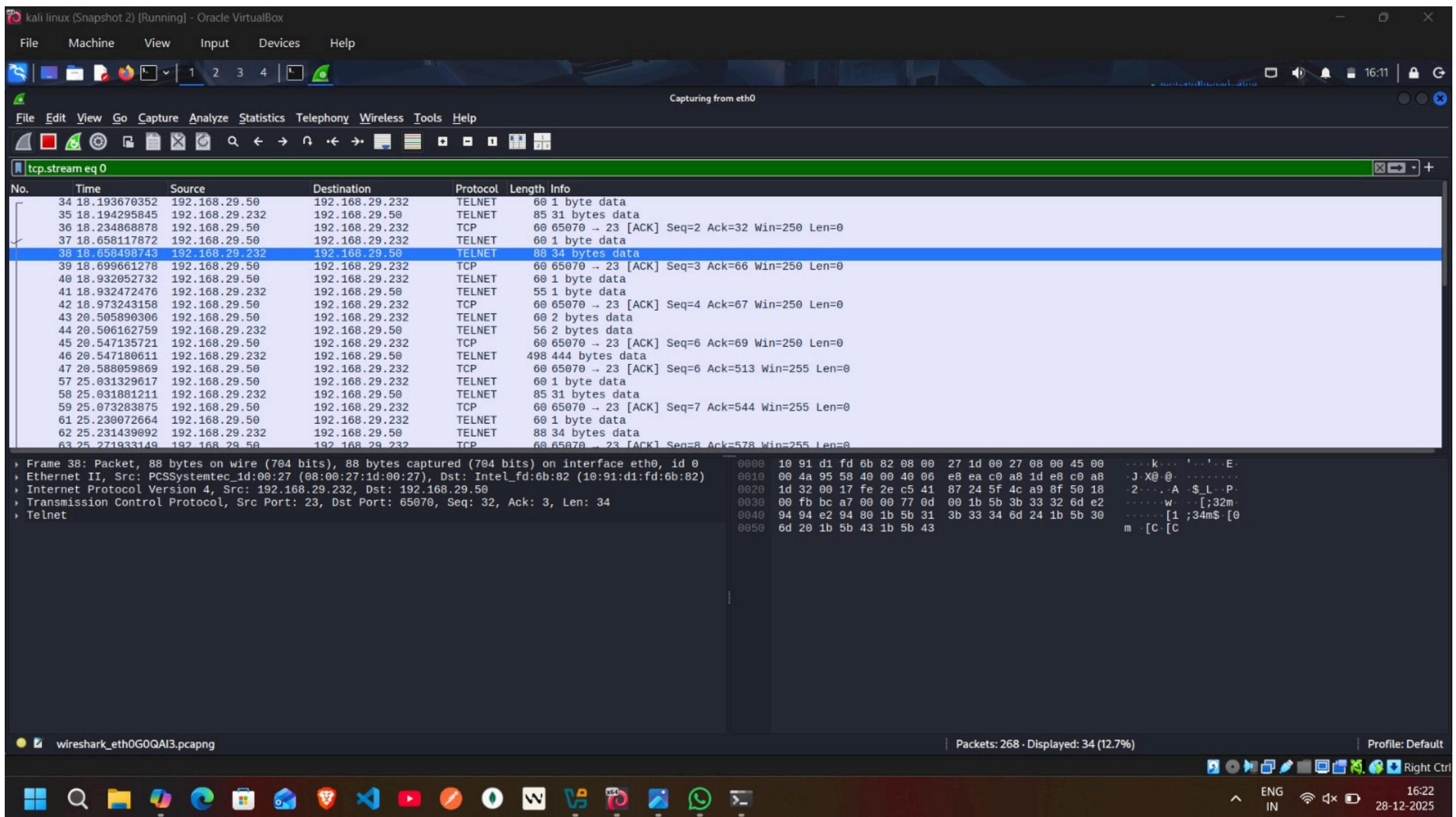
The screenshot shows a Windows Telnet window titled "Telnet 192.168.29.232". The session content is as follows:

```
session
ec
ec kali login: kali
Password:
Linux kali 6.16.8+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.16.8-1kali1 (2025-09-24) 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.

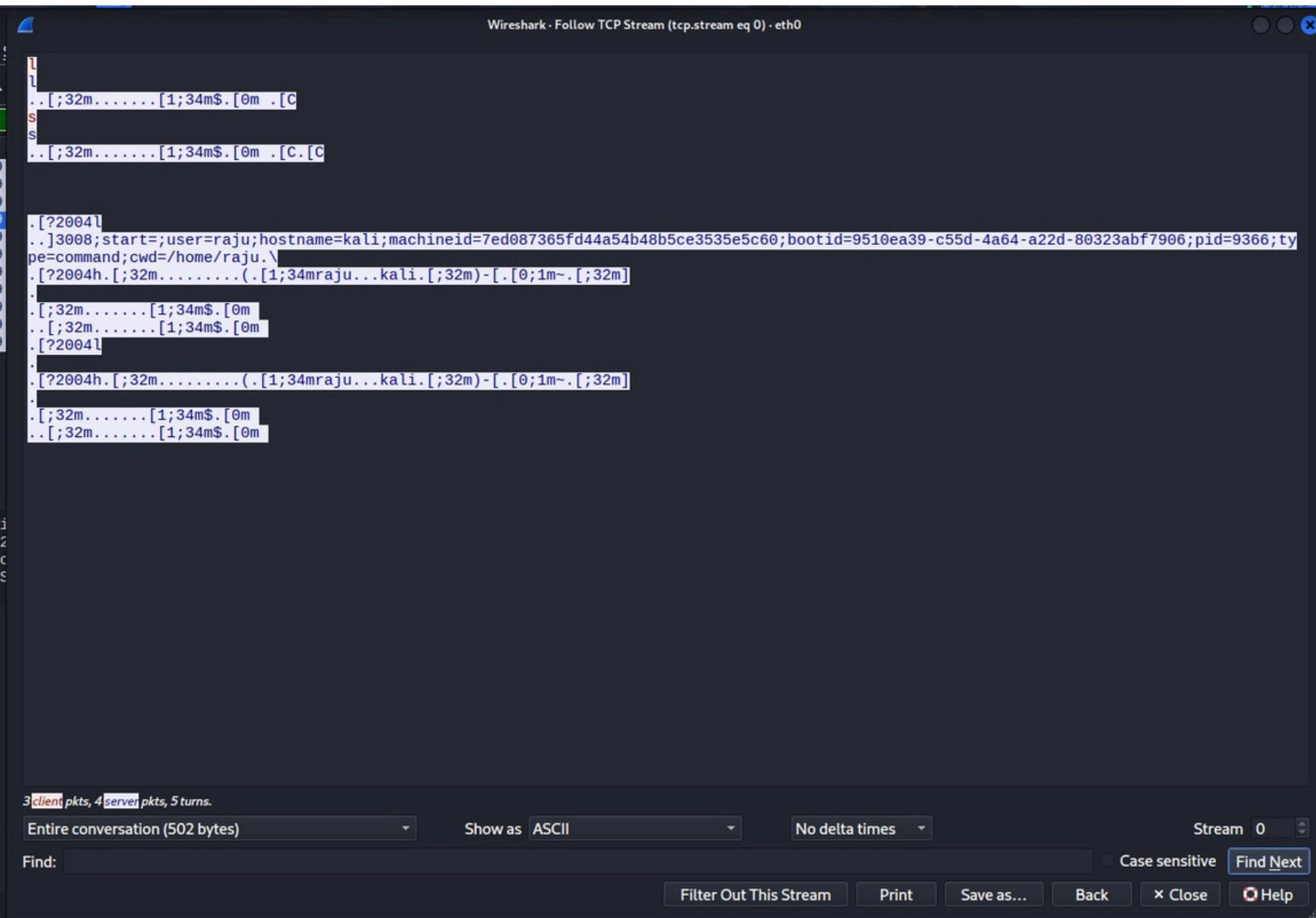
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
3008;start=74c037921c2b41ed8b4d13bc90d0b1bc;user=root;hostname=kali;machineid=7ed087365fd44a54b48b5ce3535e5c60;bootid=95
10ea39c55d4a64a22d80323abf7906;pid=9184;pidfdid=9185;comm=login;targetuser=kali;sessionid=7;type=sessionzsh: corrupt his
tory file /home/kali/.zsh_history
ΓöîΓöçΓöç(kaliπë_ñkali)-[~]
ΓööΓöç$ su - raju
Password:
3008;start=170319c0c6274a41906613ffddf93648;user=kali;hostname=kali;machineid=7ed087365fd44a54b48b5ce3535e5c60;bootid=95
10ea39c55d4a64a22d80323abf7906;pid=9346;pidfdid=9347;comm=su;targetuser=raju;type=sessionΓöîΓöçΓöç(rajuπë_ñkali)-[~]
o: ΓööΓöç$
```

At the bottom of the window, network statistics are displayed: TX errors 0, dropped 0, overruns 0, carrier 0, collisions 0.

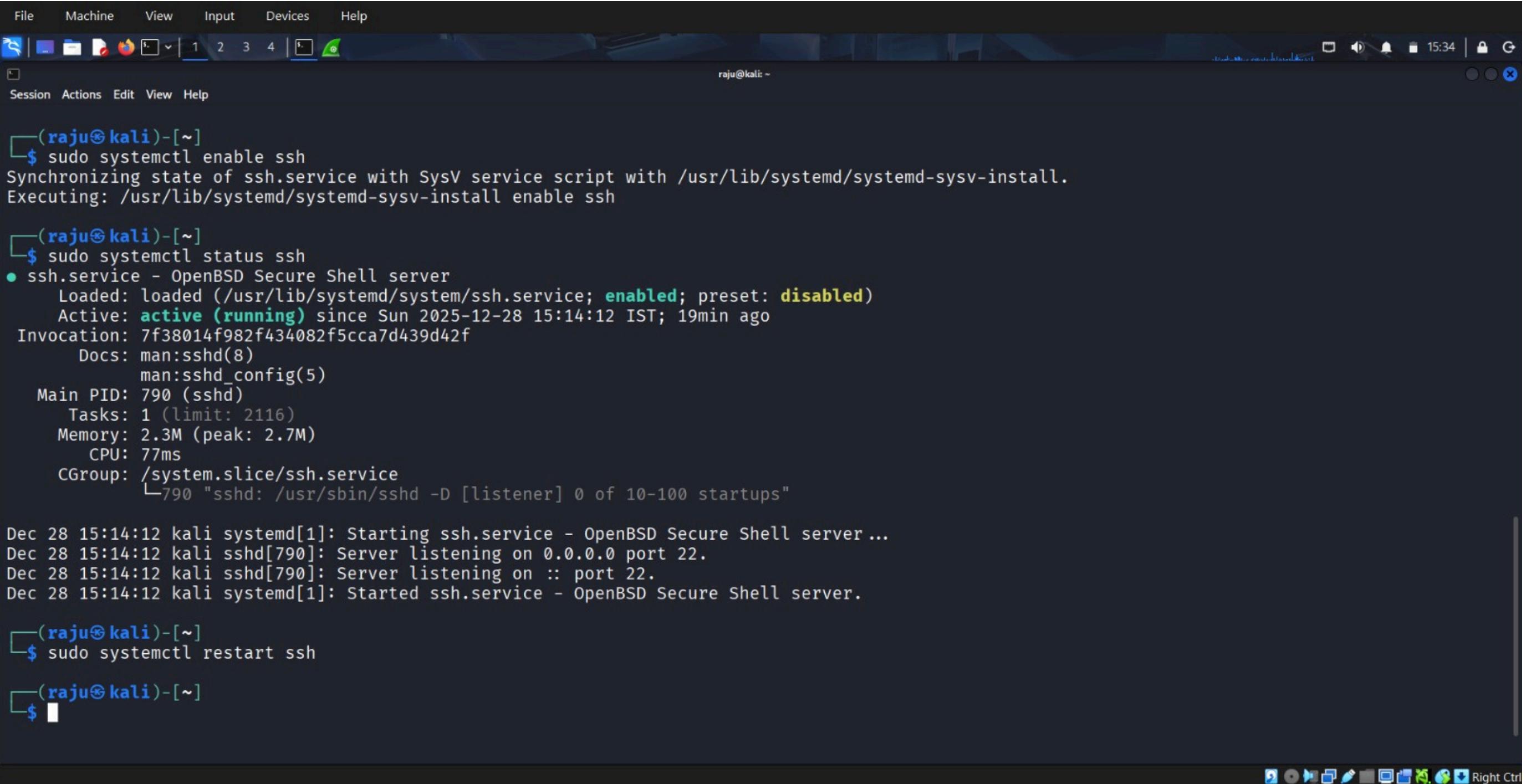
# Observing telnet through wireshark



# All file in readable format



# Starting SSH server and Getting it running



The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal session is as follows:

```
(raju㉿kali)-[~]
$ 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

(raju㉿kali)-[~]
$ 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 Sun 2025-12-28 15:14:12 IST; 19min ago
    Invocation: 7f38014f982f434082f5cca7d439d42f
      Docs: man:sshd(8)
             man:sshd_config(5)
    Main PID: 790 (sshd)
      Tasks: 1 (limit: 2116)
     Memory: 2.3M (peak: 2.7M)
        CPU: 77ms
       CGroup: /system.slice/ssh.service
               └─790 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Dec 28 15:14:12 kali systemd[1]: Starting ssh.service - OpenBSD Secure Shell server ...
Dec 28 15:14:12 kali sshd[790]: Server listening on 0.0.0.0 port 22.
Dec 28 15:14:12 kali sshd[790]: Server listening on :: port 22.
Dec 28 15:14:12 kali systemd[1]: Started ssh.service - OpenBSD Secure Shell server.

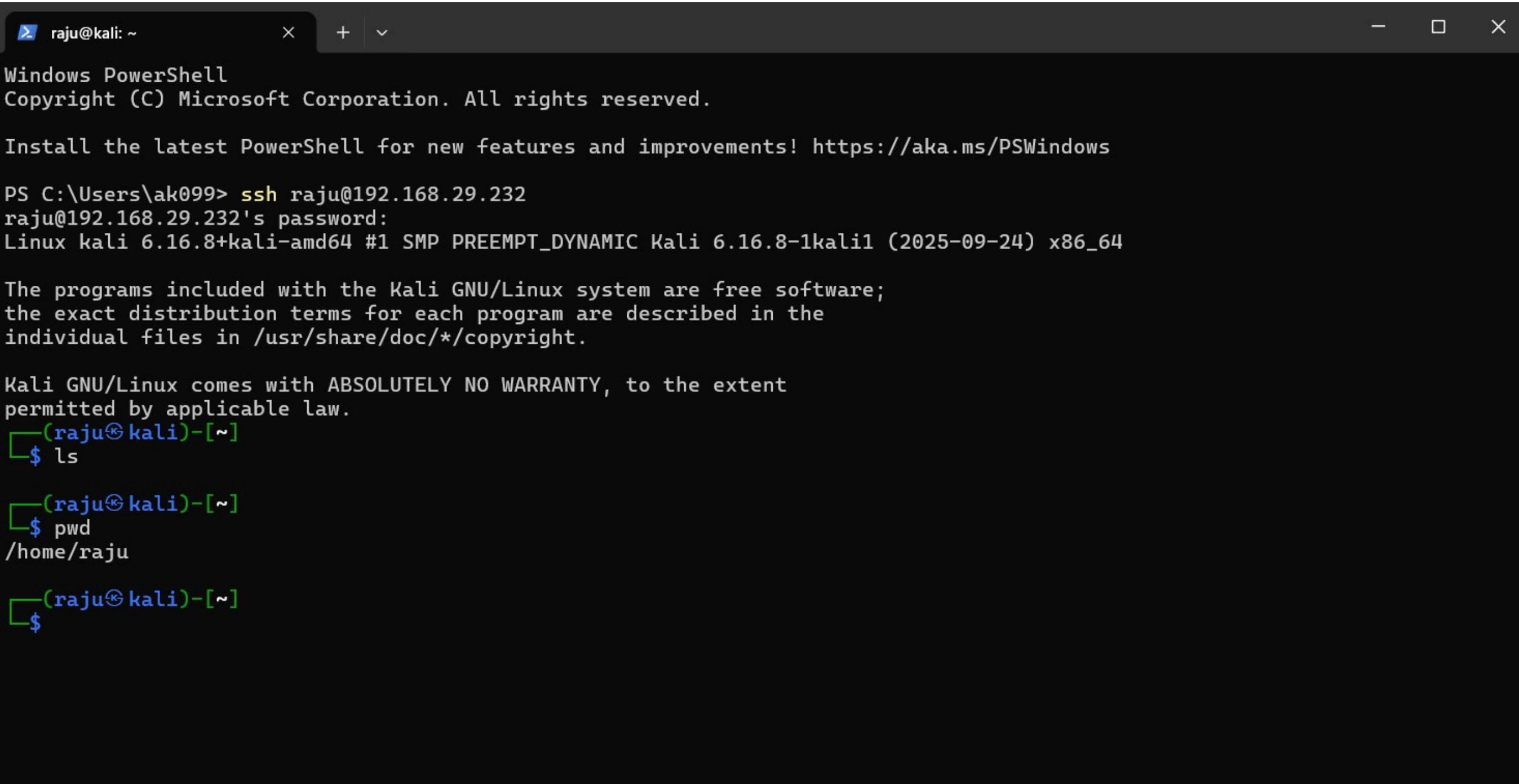
(raju㉿kali)-[~]
$ sudo systemctl restart ssh
(raju㉿kali)-[~]
$
```

# Getting ip Address

```
(raju㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.29.232 netmask 255.255.255.0 broadcast 192.168.29.255
        inet6 2405:201:3006:58d7:a00:27ff:fe1d:27 prefixlen 64 scopeid 0x0<global>
        inet6 fe80::a00:27ff:fe1d:27 prefixlen 64 scopeid 0x20<link>
        inet6 2405:201:3006:58d7:c7a3:b975:4cb7:4b94 prefixlen 64 scopeid 0x0<global>
        ether 08:00:27:1d:00:27 txqueuelen 1000 (Ethernet)
        RX packets 1226 bytes 97939 (95.6 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 304 bytes 34442 (33.6 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 8 bytes 480 (480.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 8 bytes 480 (480.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

# Getting access of Victim machine



The screenshot shows a Windows PowerShell window titled "raju@kali: ~". The session starts with the PowerShell welcome message and copyright notice. It then shows the user attempting to ssh into a Kali Linux machine at 192.168.29.232, entering the password, and displaying the Kali Linux system information. The user then runs ls, pwd, and ends the session.

```
raju@kali: ~
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ak099> ssh raju@192.168.29.232
raju@192.168.29.232's password:
Linux kali 6.16.8+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.16.8-1kali1 (2025-09-24) 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.

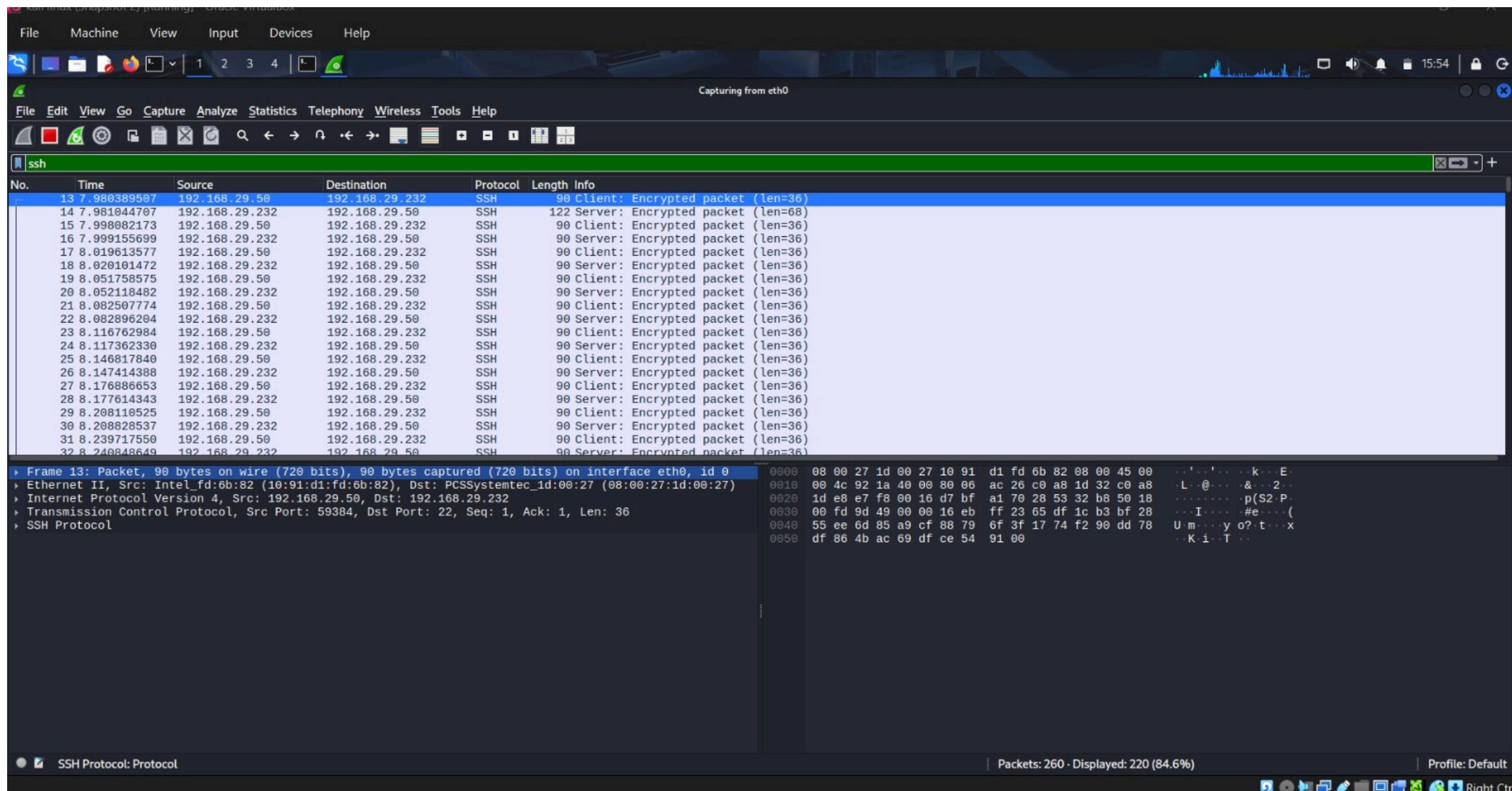
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

[raju@kali:~]
$ ls

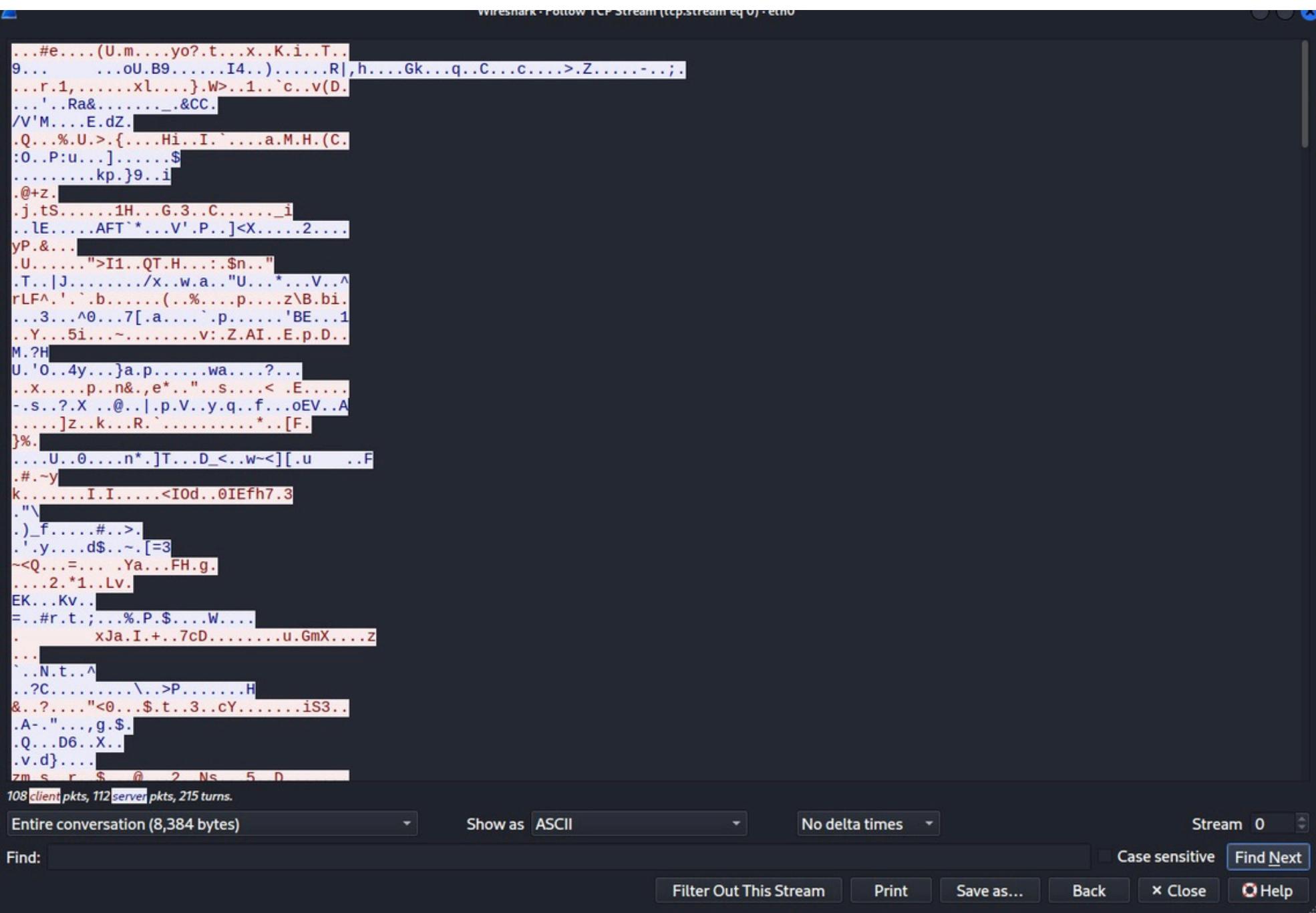
[raju@kali:~]
$ pwd
/home/raju

[raju@kali:~]
$
```

# Observing ssh through wireshark



# All data in encrypted form



## **SSH and Telnet Explained**

- **SSH hides your messages so no one can see them.**
- **Telnet shows your messages to anyone watching.**
- **SSH asks for safe keys or passwords before opening a connection.**
- **Telnet only uses a simple password that can be stolen easily.**
- **SSH works well on modern systems and keeps things private.**
- **Telnet is only used for simple testing because it is not safe.**

## **Secure SSH vs Open Telnet**

- **SSH keeps your login private using strong protection.**
- **Telnet sends your login openly across the network.**
- **SSH stops others from changing your messages.**
- **Telnet lets attackers read or change your messages.**
- **SSH supports safe remote work.**
- **Telnet is unsafe for real-world use.**

## **SSH vs Telnet**

- **SSH keeps your data safe by locking everything with strong protection.**
- **Telnet sends everything in plain form, so anyone can read it.**
- **SSH checks who you are before letting you in, which makes it safer.**
- **Telnet has no safety checks, so it's easy to break into.**
- **SSH is used today for servers because it protects both login and data.**
- **Telnet is mostly gone now because it cannot keep data private.**