

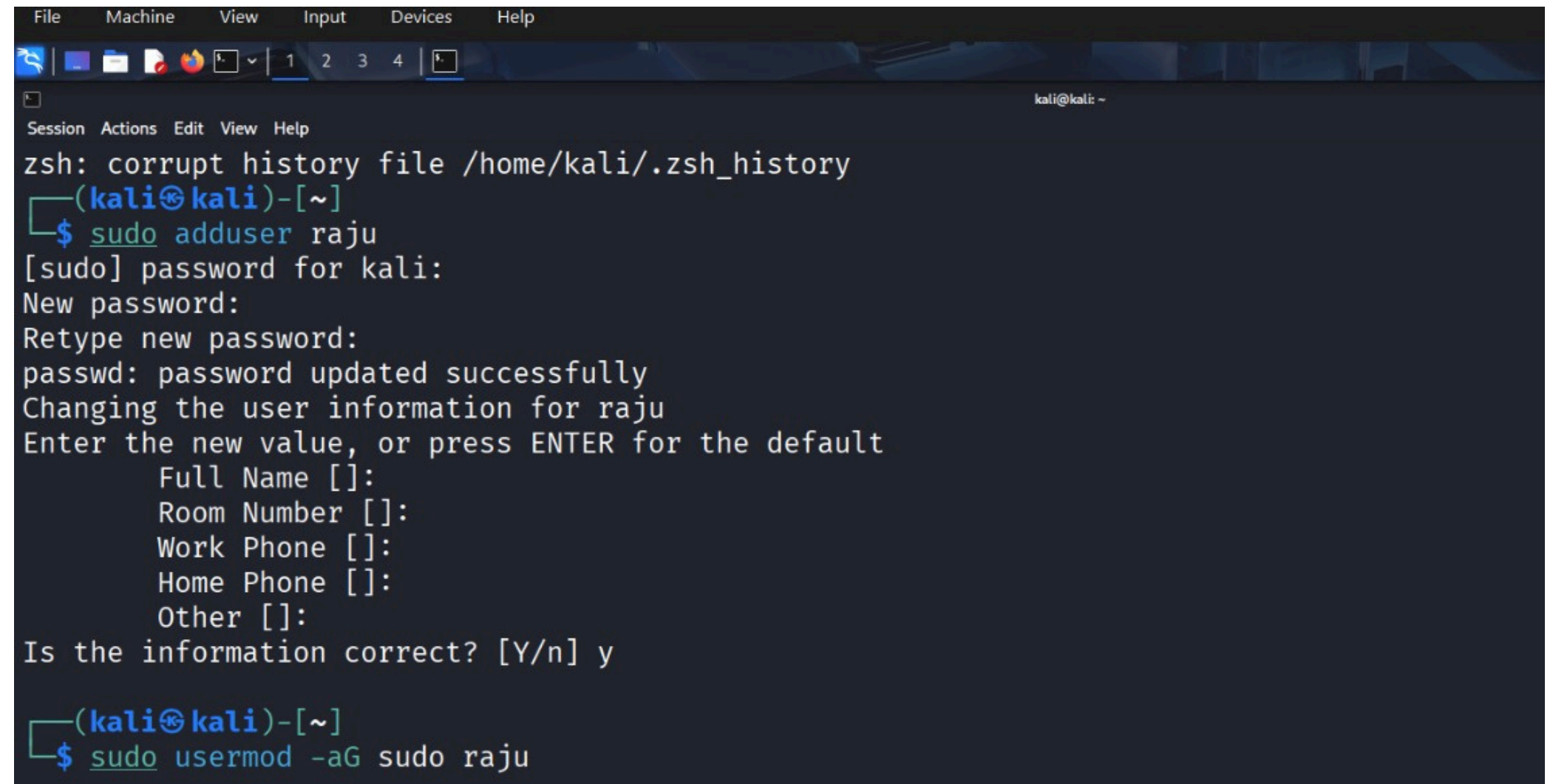
# Creating User name

**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

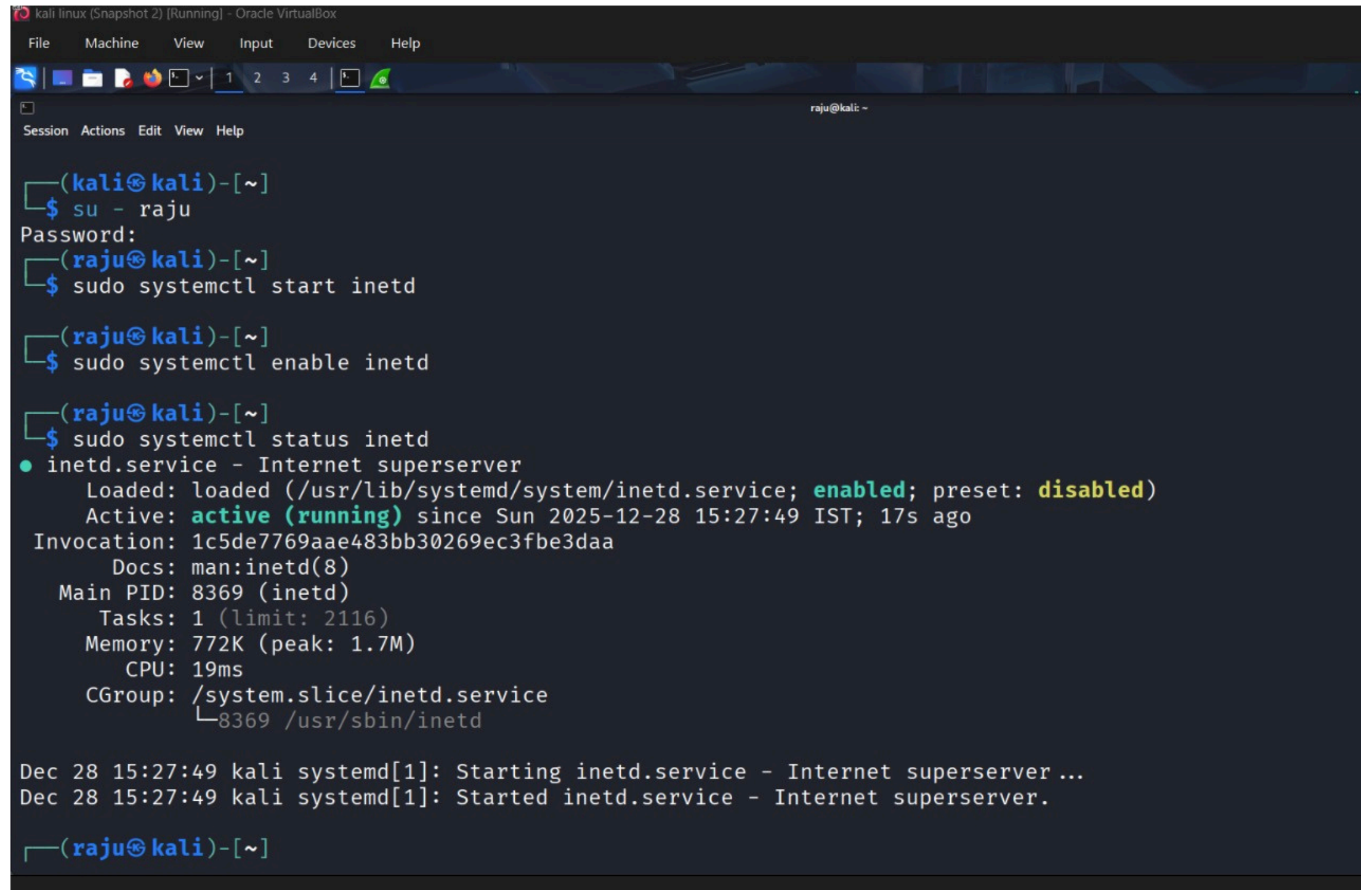


```
File Machine View Input Devices Help
zsh: corrupt history file /home/kali/.zsh_history
(kali㉿kali)-[~]
$ sudo adduser raju
[sudo] password for kali:
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for raju
Enter the new value, or press ENTER 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.



```
kali linux (Snapshot 2) [Running] - Oracle VirtualBox
File Machine View Input Devices Help

raj@kali: ~
Session Actions Edit View Help

(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.

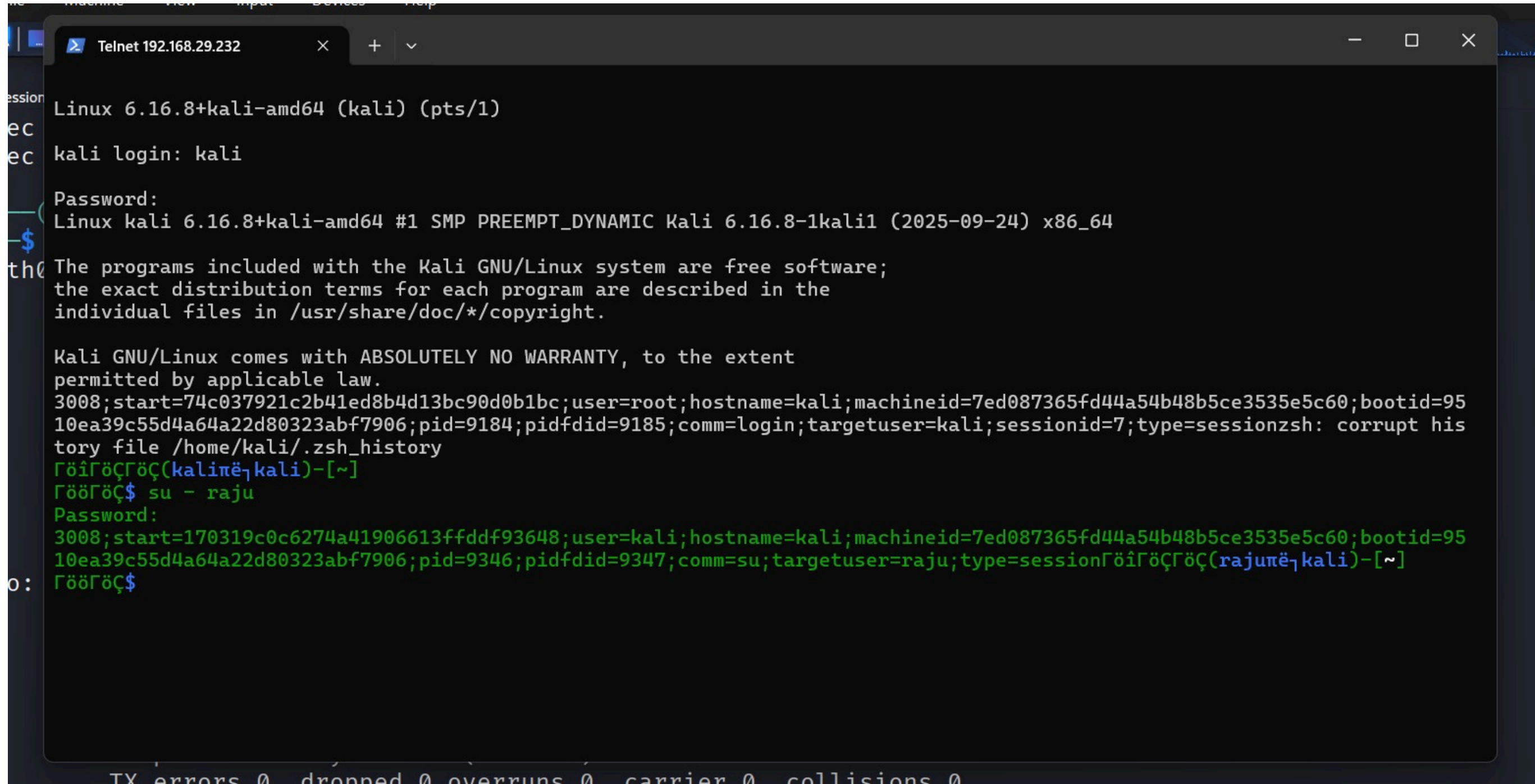
(raju@kali)-[~]
```



# 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 image shows a terminal window titled "Telnet 192.168.29.232". The session begins with a login prompt for a Kali Linux system. The user enters "kali" as the username and a password. The system then displays the Kali GNU/Linux version and architecture. A message about the GNU/Linux system's warranty and distribution terms is shown. The user then enters the command "su - raju" to switch to the user "raju". The system prompts for a password, and the user enters it. The session then shows the user "raju" at the prompt. The terminal window also displays network statistics at the bottom: "TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0".

```
Telnet 192.168.29.232

Linux 6.16.8+kali-amd64 (kali) (pts/1)
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(kali~kali)-[~]
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(rajum~kali)-[~]
ΓööΓöÇ$

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



# Observing telnet through wireshark

The screenshot shows a Kali Linux virtual machine running Oracle VM VirtualBox. The Wireshark network protocol analyzer is open, capturing traffic on the eth0 interface. The packet list pane shows a series of Telnet and TCP packets. Packet 38 is selected, showing its details in the packet details pane and its raw data in the packet bytes pane.

**Packet List:**

No.	Time	Source	Destination	Protocol	Length	Info
34	18.193670352	192.168.29.50	192.168.29.232	TELNET	60	1 byte data
35	18.194295845	192.168.29.232	192.168.29.50	TELNET	85	31 bytes data
36	18.234868878	192.168.29.50	192.168.29.232	TCP	60	65070 → 23 [ACK] Seq=2 Ack=32 Win=250 Len=0
37	18.658117872	192.168.29.50	192.168.29.232	TELNET	60	1 byte data
38	18.658498743	192.168.29.232	192.168.29.50	TELNET	88	34 bytes data
39	18.699661278	192.168.29.50	192.168.29.232	TCP	60	65070 → 23 [ACK] Seq=3 Ack=66 Win=250 Len=0
40	18.932052732	192.168.29.50	192.168.29.232	TELNET	60	1 byte data
41	18.932472476	192.168.29.232	192.168.29.50	TELNET	55	1 byte data
42	18.973243158	192.168.29.50	192.168.29.232	TCP	60	65070 → 23 [ACK] Seq=4 Ack=67 Win=250 Len=0
43	20.505890306	192.168.29.50	192.168.29.232	TELNET	60	2 bytes data
44	20.506162759	192.168.29.232	192.168.29.50	TELNET	56	2 bytes data
45	20.547135721	192.168.29.50	192.168.29.232	TCP	60	65070 → 23 [ACK] Seq=6 Ack=69 Win=250 Len=0
46	20.547180611	192.168.29.232	192.168.29.50	TELNET	498	444 bytes data
47	20.588059869	192.168.29.50	192.168.29.232	TCP	60	65070 → 23 [ACK] Seq=6 Ack=513 Win=255 Len=0
57	25.031329617	192.168.29.50	192.168.29.232	TELNET	60	1 byte data
58	25.031881211	192.168.29.232	192.168.29.50	TELNET	85	31 bytes data
59	25.073283875	192.168.29.50	192.168.29.232	TCP	60	65070 → 23 [ACK] Seq=7 Ack=544 Win=255 Len=0
61	25.230072664	192.168.29.50	192.168.29.232	TELNET	60	1 byte data
62	25.231439092	192.168.29.232	192.168.29.50	TELNET	88	34 bytes data
63	25.271933149	192.168.29.50	192.168.29.232	TCP	60	65070 → 23 [ACK] Seq=8 Ack=578 Win=255 Len=0

**Packet Details (Frame 38):**

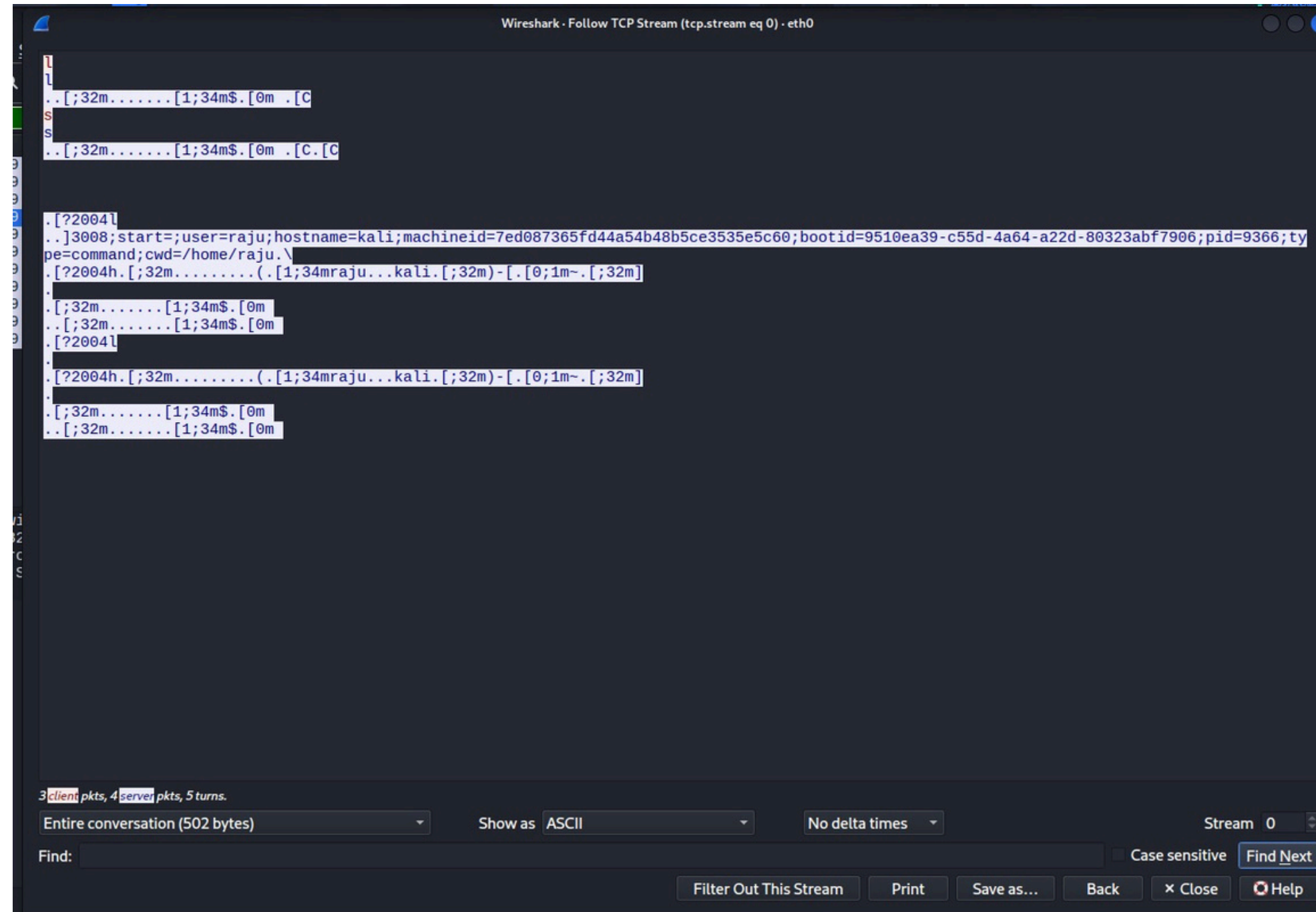
- Frame 38: Packet, 88 bytes on wire (704 bits), 88 bytes captured (704 bits) on interface eth0, id 0
- Ethernet II, Src: PCSSystemtec\_1d:00:27 (08:00:27:1d:00:27), Dst: Intel\_fd:6b:82 (10:91:d1:fd:6b:82)
- Internet Protocol Version 4, Src: 192.168.29.232, Dst: 192.168.29.50
- Transmission Control Protocol, Src Port: 23, Dst Port: 65070, Seq: 32, Ack: 3, Len: 34
- Telnet

**Packet Bytes:**

```
0000  10 91 d1 fd 6b 82 08 00 27 1d 00 27 08 00 45 00  ...k... '...'E.
0010  00 4a 95 58 40 00 40 06 e8 ea c0 a8 1d e8 c0 a8  .J.X@.@. ....
0020  1d 32 00 17 fe 2e c5 41 87 24 5f 4c a9 8f 50 18  .2...A$.L..P.
0030  00 fb bc a7 00 00 77 0d 00 1b 5b 3b 33 32 6d e2  ....w...[;32m
0040  94 94 e2 94 80 1b 5b 31 3b 33 34 6d 24 1b 5b 30  ....[1;34m$.[0
0050  6d 20 1b 5b 43 1b 5b 43  ....m.[C.[C
```

Wireshark Status: Packets: 268 · Displayed: 34 (12.7%)

# All file in redable format



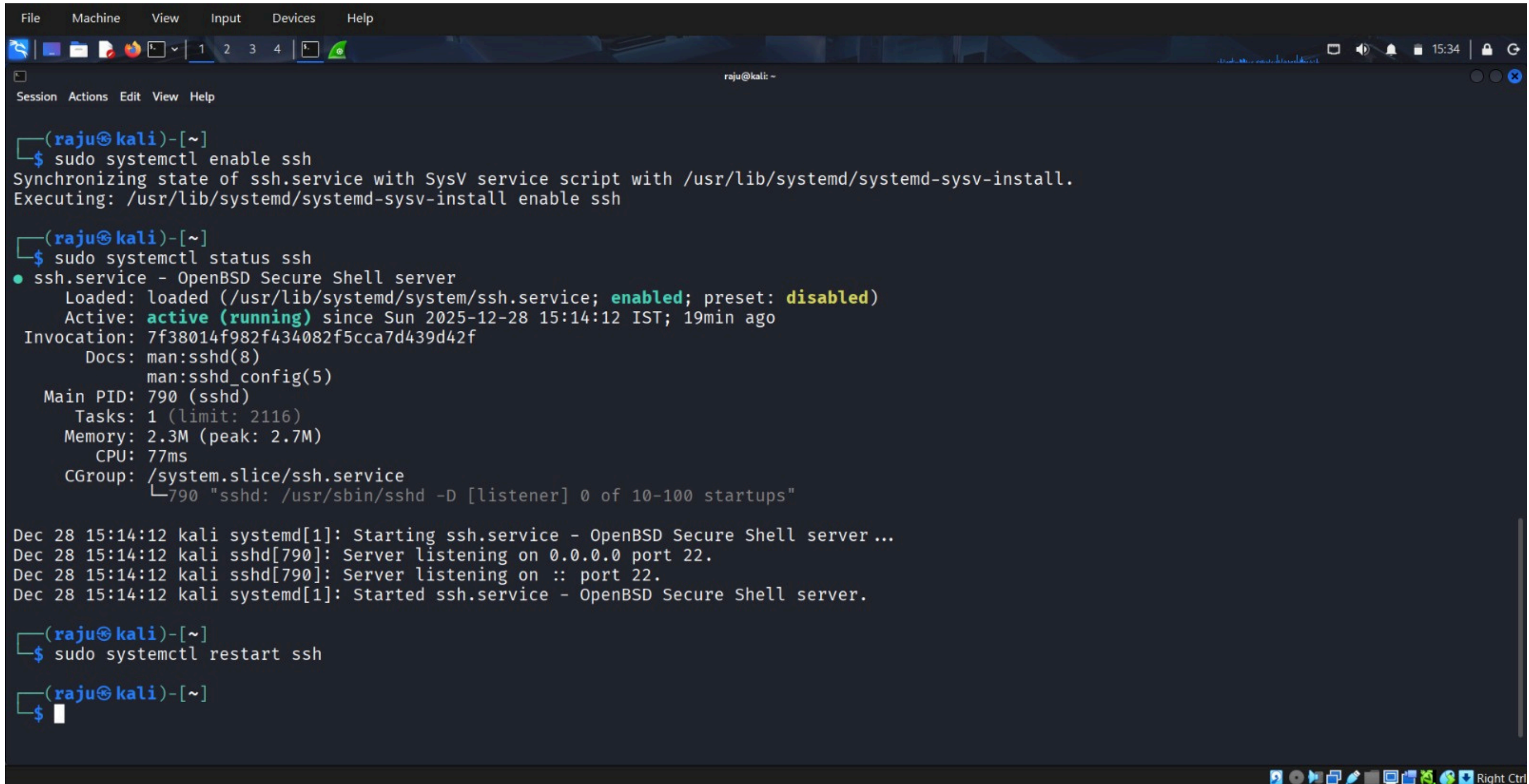
The image shows a Wireshark 'Follow TCP Stream' window for a TCP stream on interface eth0. The stream contains a shell session. The text is displayed in a dark-themed editor with syntax highlighting. The session starts with a prompt, followed by a command that sets environment variables and runs a command. The output shows the user 'raju' on host 'kali' with a specific machine ID and boot ID. The session continues with a prompt and a command that sets the current directory to the user's home directory. The session ends with a prompt and a command that sets the current directory to the user's home directory.

```
[?2004l  
..]3008;start=;user=raju;hostname=kali;machineid=7ed087365fd44a54b48b5ce3535e5c60;bootid=9510ea39-c55d-4a64-a22d-80323abf7906;pid=9366;ty  
pe=command;cwd=/home/raju.  
[?2004h.[;32m.....(. [1;34mraju...kali.[;32m)-[. [0;1m~.[;32m]  
.  
[;32m..... [1;34m$. [0m  
.. [;32m..... [1;34m$. [0m  
.[?2004l  
.  
[?2004h.[;32m.....(. [1;34mraju...kali.[;32m)-[. [0;1m~.[;32m]  
.  
[;32m..... [1;34m$. [0m  
.. [;32m..... [1;34m$. [0m
```

3 client pkts, 4 server pkts, 5 turns.  
Entire conversation (502 bytes) Show as ASCII No delta times Stream 0  
Find: Case sensitive Find Next  
Filter Out This Stream Print Save as... Back Close Help



# Starting SSH server and Getting it running



```
File Machine View Input Devices Help
raju@kali: ~
Session Actions Edit View Help

(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

```
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

The image shows a Wireshark network traffic capture of an SSH session. The top menu bar includes File, Machine, View, Input, Devices, and Help. Below the menu is a toolbar with various icons for file operations and analysis. The main window displays a list of captured packets, with the 'ssh' filter applied. The packet list shows a series of encrypted packets between two hosts.

No.	Time	Source	Destination	Protocol	Length	Info
13	7.980389507	192.168.29.50	192.168.29.232	SSH	90	Client: Encrypted packet (len=36)
14	7.981044707	192.168.29.232	192.168.29.50	SSH	122	Server: Encrypted packet (len=68)
15	7.998002173	192.168.29.50	192.168.29.232	SSH	90	Client: Encrypted packet (len=36)
16	7.999155699	192.168.29.232	192.168.29.50	SSH	90	Server: Encrypted packet (len=36)
17	8.019613577	192.168.29.50	192.168.29.232	SSH	90	Client: Encrypted packet (len=36)
18	8.020101472	192.168.29.232	192.168.29.50	SSH	90	Server: Encrypted packet (len=36)
19	8.051758575	192.168.29.50	192.168.29.232	SSH	90	Client: Encrypted packet (len=36)
20	8.052118482	192.168.29.232	192.168.29.50	SSH	90	Server: Encrypted packet (len=36)
21	8.082507774	192.168.29.50	192.168.29.232	SSH	90	Client: Encrypted packet (len=36)
22	8.082896204	192.168.29.232	192.168.29.50	SSH	90	Server: Encrypted packet (len=36)
23	8.116762984	192.168.29.50	192.168.29.232	SSH	90	Client: Encrypted packet (len=36)
24	8.117362330	192.168.29.232	192.168.29.50	SSH	90	Server: Encrypted packet (len=36)
25	8.146817840	192.168.29.50	192.168.29.232	SSH	90	Client: Encrypted packet (len=36)
26	8.147414388	192.168.29.232	192.168.29.50	SSH	90	Server: Encrypted packet (len=36)
27	8.176886653	192.168.29.50	192.168.29.232	SSH	90	Client: Encrypted packet (len=36)
28	8.177614343	192.168.29.232	192.168.29.50	SSH	90	Server: Encrypted packet (len=36)
29	8.208110525	192.168.29.50	192.168.29.232	SSH	90	Client: Encrypted packet (len=36)
30	8.208828537	192.168.29.232	192.168.29.50	SSH	90	Server: Encrypted packet (len=36)
31	8.239717550	192.168.29.50	192.168.29.232	SSH	90	Client: Encrypted packet (len=36)
32	8.240848649	192.168.29.232	192.168.29.50	SSH	90	Server: Encrypted packet (len=36)

The detailed view of packet 13 shows the following structure:

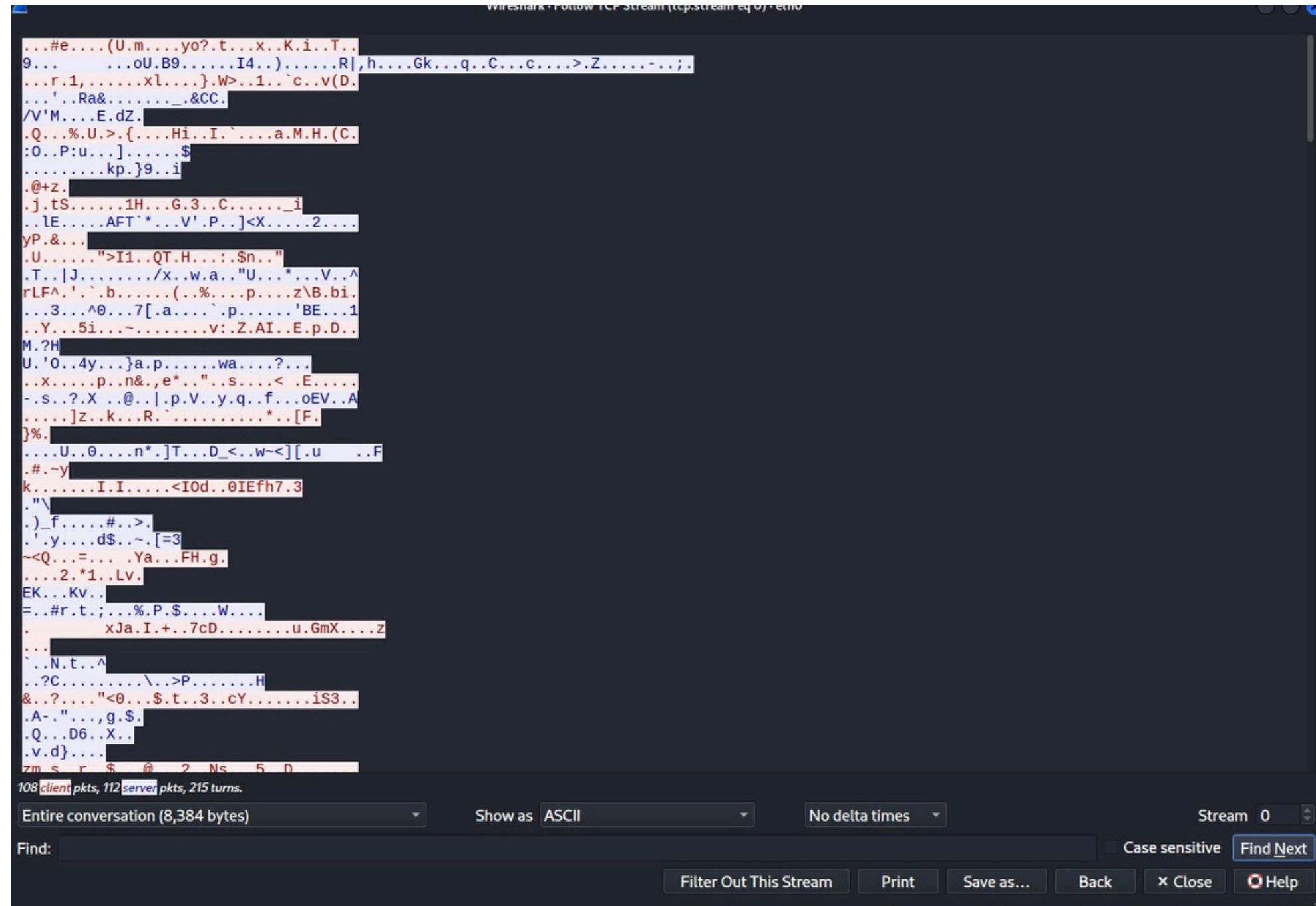
- Frame 13: Packet, 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface eth0, id 0
- Ethernet II, Src: Intel\_fd:6b:82 (10:91:d1:fd:6b:82), Dst: PCSSystemtec\_id:00:27 (08:00:27:1d:00:27)
- Internet Protocol Version 4, Src: 192.168.29.50, Dst: 192.168.29.232
- Transmission Control Protocol, Src Port: 59384, Dst Port: 22, Seq: 1, Ack: 1, Len: 36
- SSH Protocol

The packet bytes are displayed in hexadecimal and ASCII format. The ASCII column shows the following text: "...k...E...L...@...&...2...p(S2.P...I...#e...(U.m...y o?t...x...K.i.T...

The status bar at the bottom indicates "SSH Protocol: Protocol" and "Packets: 260 · Displayed: 220 (84.6%)".



# All data in encrypted form



The image shows a Wireshark packet capture window with the title bar "Wireshark - Follow TCP Stream (tcp.stream eq 0) - eth0". The main display area shows a list of packets in the left pane and a detailed view of a selected packet in the right pane. The detailed view shows the packet data in ASCII format, which is entirely encrypted and appears as a series of non-sensical characters and symbols. The bottom status bar indicates "108 client pkts, 112 server pkts, 215 turns." and the bottom toolbar includes buttons for "Filter Out This Stream", "Print", "Save as...", "Back", "Close", and "Help".

```
...#e...(U.m...yo?.t...x..K.i..T..
9... ..oU.B9.....I4..).....R],h...Gk...q..C...c....>.Z.....-.;.
...r.1,.....xl.....}.W>..1..`c..v(D.
...'.Ra&....._&CC.
/V'M....E.dZ.
.Q...%.U.>.{...Hi..I..`....a.M.H.(C.
:O..P:u...]......$
.....kp.)9..i
.@+z.
.j.tS.....1H...G.3..C....._i
..lE....AFT`*...V'.P..]<X.....2....
yP.&...
.U.....">I1..QT.H....$n.."
.T..|J...../x..w.a.."U...*...V..^
rLFA^.'..b.....(..%....p....z\B.bi.
...3...^0...7[.a....`p.....'BE...1
..Y...5i...~.....v:..Z.AI..E.p.D..
M.?H
U.'0..4y...}a.p.....wa....?...
..x....p..n&,e*..."..s...<..E.....
-.s..?.X..@..|p.V..y.q..f...oEV..A
.....]z..k...R..`.....*[F.
}%..
...U..0....n*.]T...D_<..w~<][.u...F
.#..~y
k.....I.I.....<I0d..0IEfh7.3
."
.)_f.....#..>.
.'y....d$.~. [=3
~<Q...=....Ya...FH.g.
....2.*1..LV.
EK...Kv..
=..#r.t.;...%.P.$...W....
..xJa.I.+..7cD.....u.GmX....Z
...
`.N.t..^
..?C.....\..>P.....H
&..?...."<0...$.t..3..cY.....iS3..
.A~"...g.$.
.Q...D6..X..
.v.d}....
zm s r $ @ 2 Ns 5 D
```

## **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.**