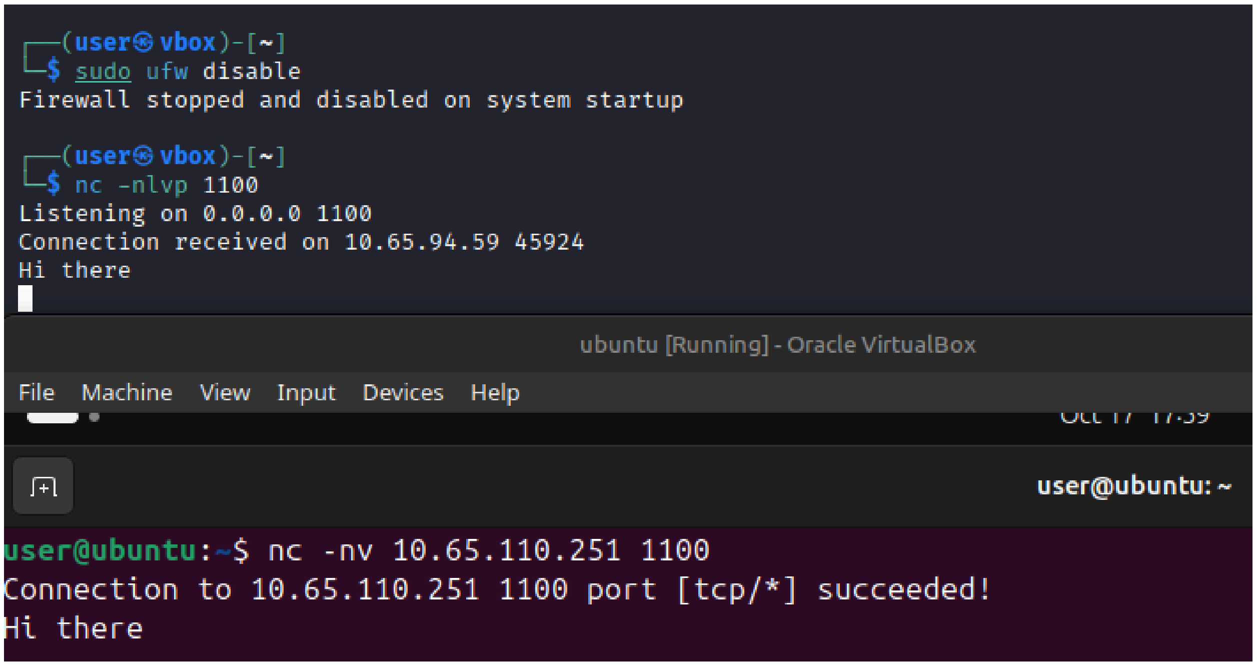
Part 1

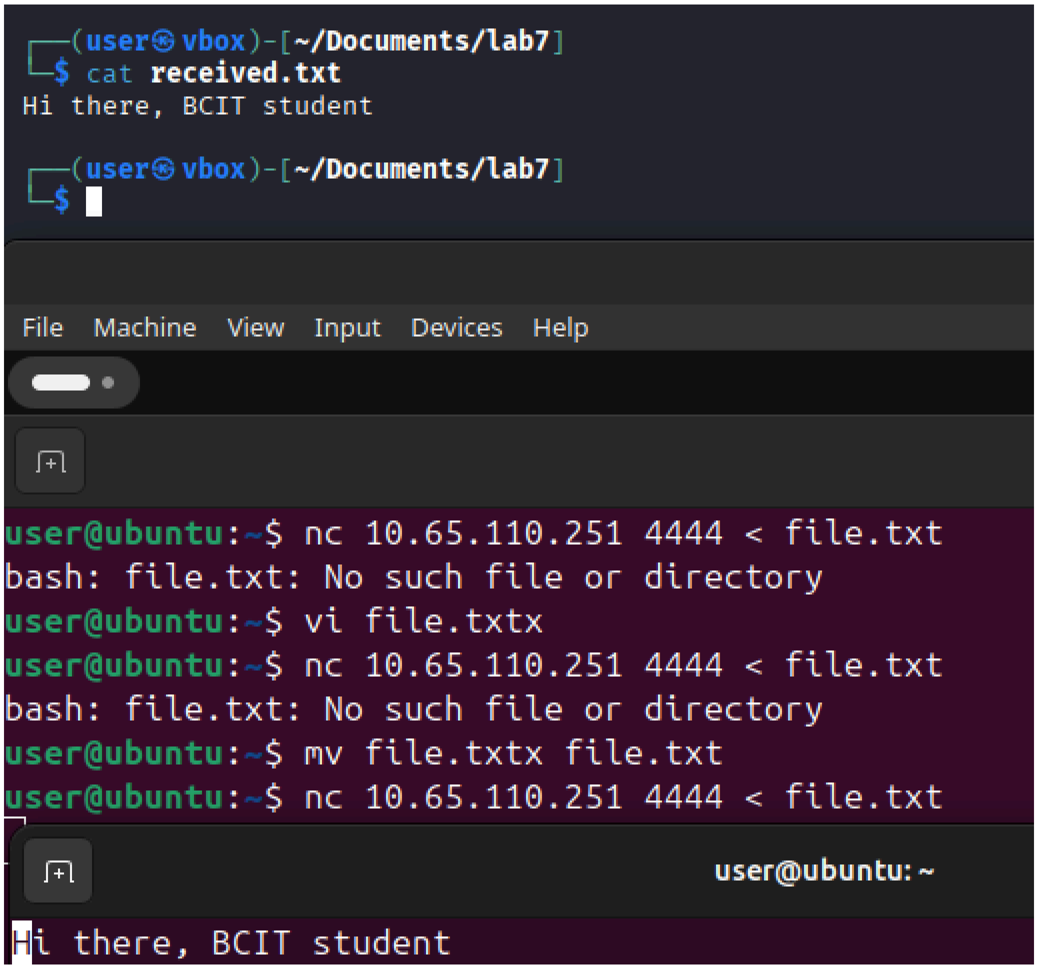


```
┌──(user㉿vbox)-[~]
└─$ sudo ufw disable
Firewall stopped and disabled on system startup

┌──(user㉿vbox)-[~]
└─$ nc -nlvp 1100
Listening on 0.0.0.0 1100
Connection received on 10.65.94.59 45924
Hi there
```

ubuntu [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Oct 17 17:39
user@ubuntu: ~

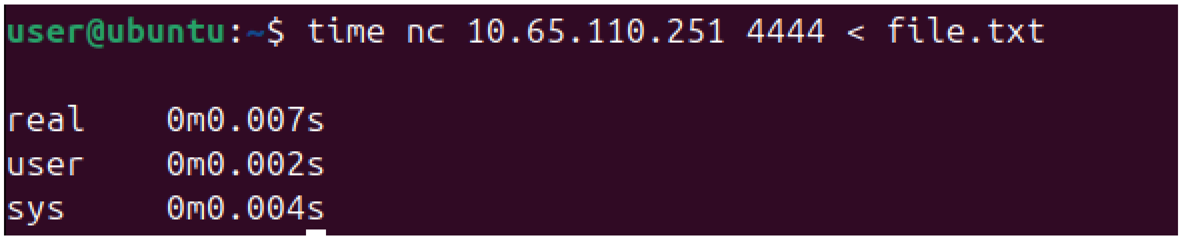
```
user@ubuntu:~$ nc -nv 10.65.110.251 1100
Connection to 10.65.110.251 1100 port [tcp/*] succeeded!
Hi there
```

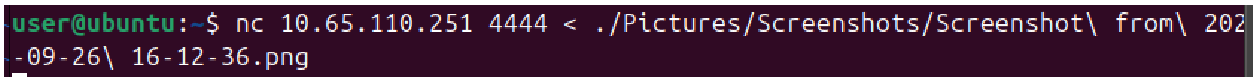
1. The connection to the server is closed, no further messages can be send
2. No, netcat connect and establishes connection with the first client that connects to the server, if a second clients tries to connect to the server while the first client is still connected, the TCP SYN queue is never complete so it can never connect until first client is disconnected from the server
3. TCP is connection-oriented, meaning that sender and receiver firstly need to establish a connection based on agreed parameters. They do this through as 3-way handshake procedure. The server must be listening for connection requests from clients before a connection is established.

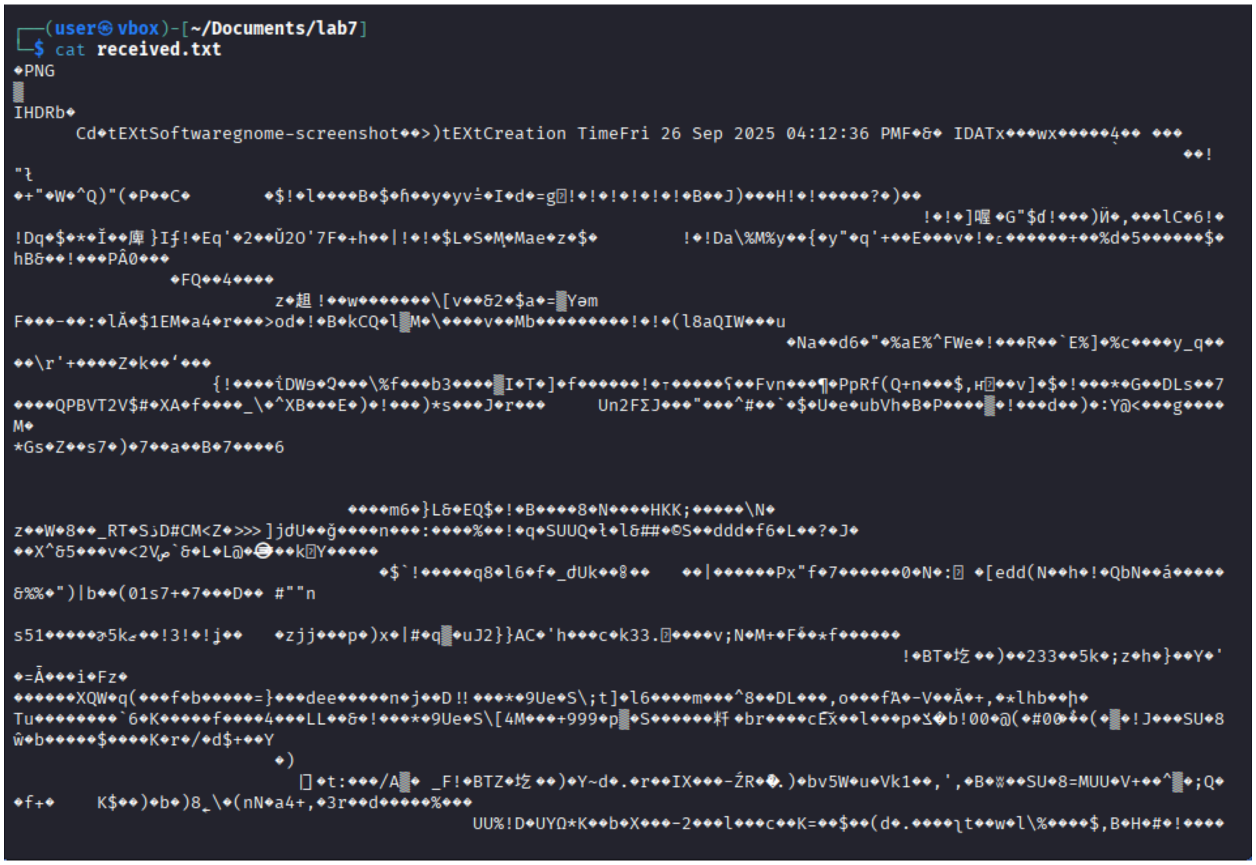

```
┌──(user㉿vbox)-[~/Documents/lab7]
└─$ cat received.txt
Hi there, BCIT student

┌──(user㉿vbox)-[~/Documents/lab7]
└─$
```

```
user@ubuntu:~$ nc 10.65.110.251 4444 < file.txt
bash: file.txt: No such file or directory
user@ubuntu:~$ vi file.txtx
user@ubuntu:~$ nc 10.65.110.251 4444 < file.txt
bash: file.txt: No such file or directory
user@ubuntu:~$ mv file.txtx file.txt
user@ubuntu:~$ nc 10.65.110.251 4444 < file.txt
```

```
Hi there, BCIT student
```

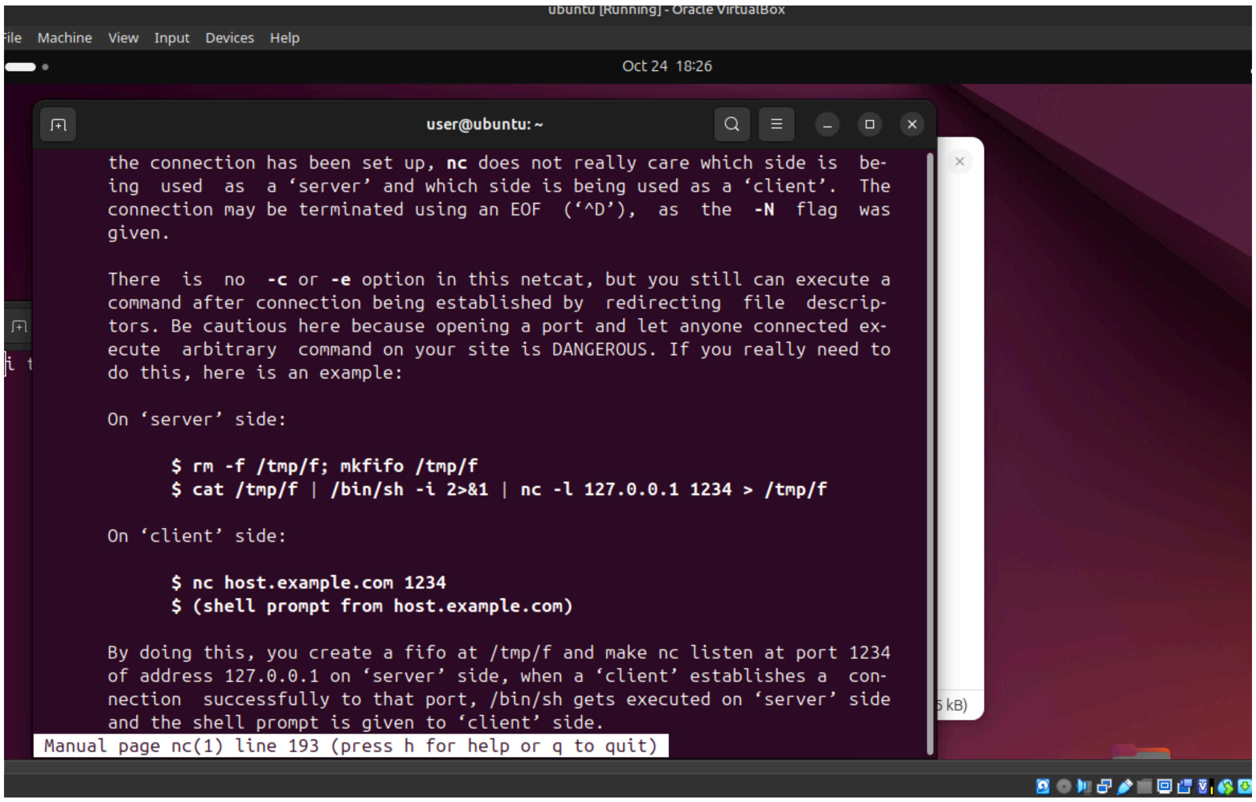
4. 
```
user@ubuntu:~$ time nc 10.65.110.251 4444 < file.txt

real	0m0.007s
user	0m0.002s
sys	0m0.004s
```

5. 
```
user@ubuntu:~$ nc 10.65.110.251 4444 < ./Pictures/Screenshots/Screenshot\ from\ 202
-09-26\ 16-12-36.png
```



```
┌──(user㉿vbox)-[~/Documents/lab7]
└─$ cat received.txt
◆PNG
▒
IHDRb◆
      Cd◆tEXtSoftwaregnome-screenshot◆◆>)tEXtCreation TimeFri 26 Sep 2025 04:12:36 PMF◆&◆ IDATx◆◆◆wx◆◆◆◆◆4◆◆ ◆◆◆
                                                                                                                   ◆◆!
"ƚ
◆+"◆W◆^Q)"(◆P◆◆C◆       ◆$!◆l◆◆◆◆B◆$◆ɦ◆◆y◆yv≐◆I◆d◆=g⍰!◆!◆!◆!◆!◆!◆B◆◆J)◆◆◆H!◆!◆◆◆◆◆?◆)◆◆
                                                                                         !◆!◆]喔◆G"$ď!◆◆◆)Ӥ◆,◆◆◆lC◆6!◆
!Dq◆$◆*◆Ĭ◆◆庳 }Iƒ!◆Eq'◆2◆◆Ŭ2O'7F◆+h◆◆|!◆!◆$L◆S◆M◆Mae◆z◆$◆        !◆!Da\%M%y◆◆{◆y"◆q'+◆◆E◆◆◆v◆!◆꜀◆◆◆◆◆◆+◆◆%d◆5◆◆◆◆◆◆$◆
hB&◆◆!◆◆◆PÂ0◆◆◆
               ◆FQ◆◆4◆◆◆◆
                          z◆超 !◆◆w◆◆◆◆◆◆◆◆\[v◆◆&2◆$a◆=▒Yəm
F◆◆◆-◆◆:◆lĂ◆$1EM◆a4◆r◆◆◆>od◆!◆B◆kCQ◆l▒M◆\◆◆◆◆v◆◆Mb◆◆◆◆◆◆◆◆◆◆!◆!◆(l8aQIW◆◆◆u
                                                                                ◆Na◆◆d6◆"◆%aE%^FWe◆!◆◆◆R◆◆`E%]◆%c◆◆◆◆y_q◆◆
◆◆\r'+◆◆◆◆Z◆k◆◆ʻ◆◆◆
                   {!◆◆◆◆îDWɘ◆Ϙ◆◆◆\%f◆◆◆b3◆◆◆◆▒I◆T◆]◆f◆◆◆◆◆◆!◆ᴛ◆◆◆◆◆ʕ◆◆Fvn◆◆◆¶◆PpRf(Q+n◆◆◆$,ʜ⍰◆◆v]◆$◆!◆◆◆*◆G◆◆DLs◆◆7
◆◆◆◆QPBVT2V$#◆XA◆f◆◆◆◆_\◆^XB◆◆◆E◆)◆!◆◆◆)*s◆◆◆J◆r◆◆◆     Un2FΣJ◆◆◆"◆◆◆^#◆◆`◆$◆U◆e◆ubVh◆B◆P◆◆◆◆▒◆!◆◆◆d◆◆)◆:Y@<◆◆◆g◆◆◆◆
M◆
*Gs◆Z◆◆s7◆)◆7◆◆a◆◆B◆7◆◆◆◆6



                                ◆◆◆◆m6◆}L&◆EQ$◆!◆B◆◆◆◆8◆N◆◆◆◆HKK;◆◆◆◆◆\N◆
z◆◆W◆8◆◆_RT◆SادD#CM<Z◆>>>]jđU◆◆ğ◆◆◆◆n◆◆◆:◆◆◆◆%◆◆!◆q◆SUUQ◆ɫ◆l&##◆©S◆◆ddd◆f6◆L◆◆?◆J◆
◆◆X^&5◆◆◆v◆<2Vص`&◆L◆L@◆㊖◆◆k⍰Y◆◆◆◆◆
                                   ◆$`!◆◆◆◆◆q8◆l6◆f◆_đUk◆◆8◆◆   ◆◆|◆◆◆◆◆◆Px"f◆7◆◆◆◆◆◆◆0◆N◆:⍰ ◆[edd(N◆◆h◆!◆QbN◆◆á◆◆◆◆◆
&%%◆")|b◆◆(01s7+◆7◆◆◆D◆◆ #""n

s51◆◆◆◆◆◆ə5k꜖◆◆!3!◆!ʝ◆◆   ◆zjj◆◆◆p◆)x◆|#◆q▒◆uJ2}}AC◆'h◆◆◆c◆k33.⍰◆◆◆◆v;N◆M+◆F̊◆◆*f◆◆◆◆◆◆
                                                                                   !◆BT◆圪◆◆)◆◆233◆◆5k◆;z◆h◆}◆◆Y◆'
◆=Ā◆◆◆i◆Fz◆
◆◆◆◆◆◆XQW◆q(◆◆◆f◆b◆◆◆◆◆=}◆◆◆dee◆◆◆◆◆n◆j◆◆D!!◆◆◆*◆9Ue◆S\;t]◆l6◆◆◆◆m◆◆◆^8◆◆DL◆◆◆,o◆◆◆fA̋◆-V◆◆Ă◆+,◆*lhb◆◆ɧ◆
Tu◆◆◆◆◆◆◆◆◆`6◆K◆◆◆◆◆f◆◆◆◆4◆◆◆LL◆◆&◆!◆◆◆*◆9Ue◆S\[4M◆◆◆+999◆p▒◆S◆◆◆◆◆◆籿◆br◆◆◆◆◆cE͠x◆◆l◆◆◆p◆ɮ�b!00◆@(◆#00◆̊◆(◆▒◆!J◆◆◆SU◆8
ŵ◆b◆◆◆◆◆$◆◆◆◆K◆r◆/◆d$+◆◆Y
                          ◆)
                            []◆t:◆◆◆/A▒◆ _F!◆BTZ◆圪◆◆)◆Y~d◆.◆r◆◆IX◆◆◆-ŹR◆�.)◆bv5W◆u◆Vk1◆◆,',◆B◆※◆◆SU◆8=MUU◆V+◆◆^▒◆;Q◆
◆f+◆    K$◆◆)◆b◆)8˛\◆(nN◆a4+,◆3r◆◆d◆◆◆◆◆%◆◆◆
                                              UU%!D◆UYΩ*K◆◆b◆X◆◆◆-2◆◆◆l◆◆◆c◆◆K=◆◆$◆◆(d◆.◆◆◆◆ʅt◆◆w◆l\%◆◆◆◆$,B◆H◆#◆!◆◆◆◆
```

TCP transmits raw bytes not texts so it will transfer the PNG successfully


ubuntu [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Oct 24 18:26
user@ubuntu: ~

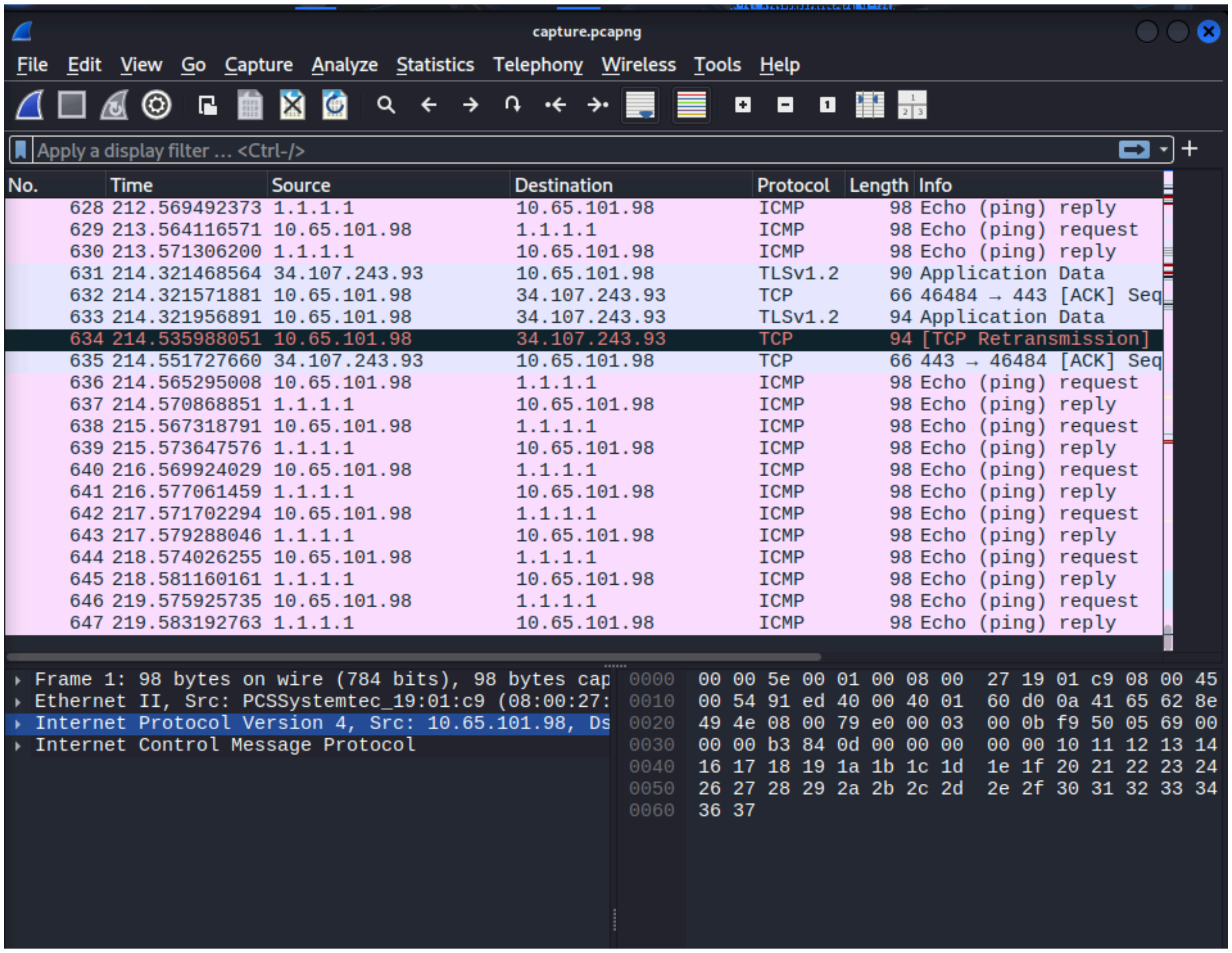
```
       the connection has been set up, nc does not really care which side is  be‐
       ing  used  as  a ‘server’ and which side is being used as a ‘client’.  The
       connection may be terminated using an EOF  (‘^D’),  as  the  -N  flag  was
       given.

       There  is  no  -c or -e option in this netcat, but you still can execute a
       command after connection being established by  redirecting  file  descrip‐
       tors. Be cautious here because opening a port and let anyone connected ex‐
       ecute  arbitrary  command on your site is DANGEROUS. If you really need to
       do this, here is an example:

       On ‘server’ side:

             $ rm -f /tmp/f; mkfifo /tmp/f
             $ cat /tmp/f | /bin/sh -i 2>&1 | nc -l 127.0.0.1 1234 > /tmp/f

       On ‘client’ side:

             $ nc host.example.com 1234
             $ (shell prompt from host.example.com)

       By doing this, you create a fifo at /tmp/f and make nc listen at port 1234
       of address 127.0.0.1 on ‘server’ side, when a ‘client’ establishes a  con‐
       nection  successfully to that port, /bin/sh gets executed on ‘server’ side
       and the shell prompt is given to ‘client’ side.
 Manual page nc(1) line 193 (press h for help or q to quit)
```

There is no -e option to remotely execute commands on victim’s machine, alternatively can use ssh for a trusted secure connection.

6. Unathorized access, privilege escalation, full compromise of the machines, dara extraction
7. Enable ufw firewall, only allowing or open necessary ports, Setup logging tools or network traffic inspection tools.
8. SSH is safeer because it provide authentication, encryption, auditing and control

Part 2

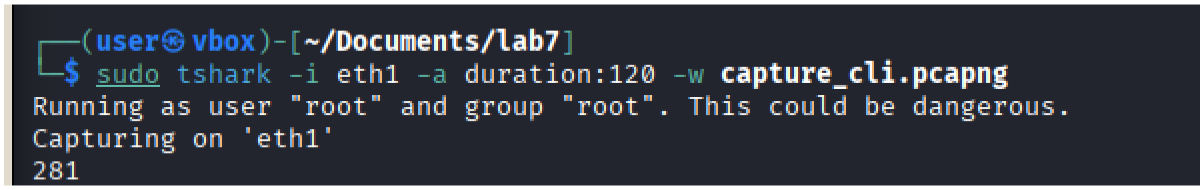
Exercise 2.1


capture.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.   Time           Source         Destination    Protocol  Length Info
  628 212.569492373 1.1.1.1        10.65.101.98   ICMP        98 Echo (ping) reply
  629 213.564116571 10.65.101.98   1.1.1.1        ICMP        98 Echo (ping) request
  630 213.571306200 1.1.1.1        10.65.101.98   ICMP        98 Echo (ping) reply
  631 214.321468564 34.107.243.93  10.65.101.98   TLSv1.2     90 Application Data
  632 214.321571881 10.65.101.98   34.107.243.93  TCP         66 46484 → 443 [ACK] Seq
  633 214.321956891 10.65.101.98   34.107.243.93  TLSv1.2     94 Application Data
  634 214.535988051 10.65.101.98   34.107.243.93  TCP         94 [TCP Retransmission]
  635 214.551727660 34.107.243.93  10.65.101.98   TCP         66 443 → 46484 [ACK] Seq
  636 214.565295008 10.65.101.98   1.1.1.1        ICMP        98 Echo (ping) request
  637 214.570868851 1.1.1.1        10.65.101.98   ICMP        98 Echo (ping) reply
  638 215.567318791 10.65.101.98   1.1.1.1        ICMP        98 Echo (ping) request
  639 215.573647576 1.1.1.1        10.65.101.98   ICMP        98 Echo (ping) reply
  640 216.569924029 10.65.101.98   1.1.1.1        ICMP        98 Echo (ping) request
  641 216.577061459 1.1.1.1        10.65.101.98   ICMP        98 Echo (ping) reply
  642 217.571702294 10.65.101.98   1.1.1.1        ICMP        98 Echo (ping) request
  643 217.579288046 1.1.1.1        10.65.101.98   ICMP        98 Echo (ping) reply
  644 218.574026255 10.65.101.98   1.1.1.1        ICMP        98 Echo (ping) request
  645 218.581160161 1.1.1.1        10.65.101.98   ICMP        98 Echo (ping) reply
  646 219.575925735 10.65.101.98   1.1.1.1        ICMP        98 Echo (ping) request
  647 219.583192763 1.1.1.1        10.65.101.98   ICMP        98 Echo (ping) reply
```

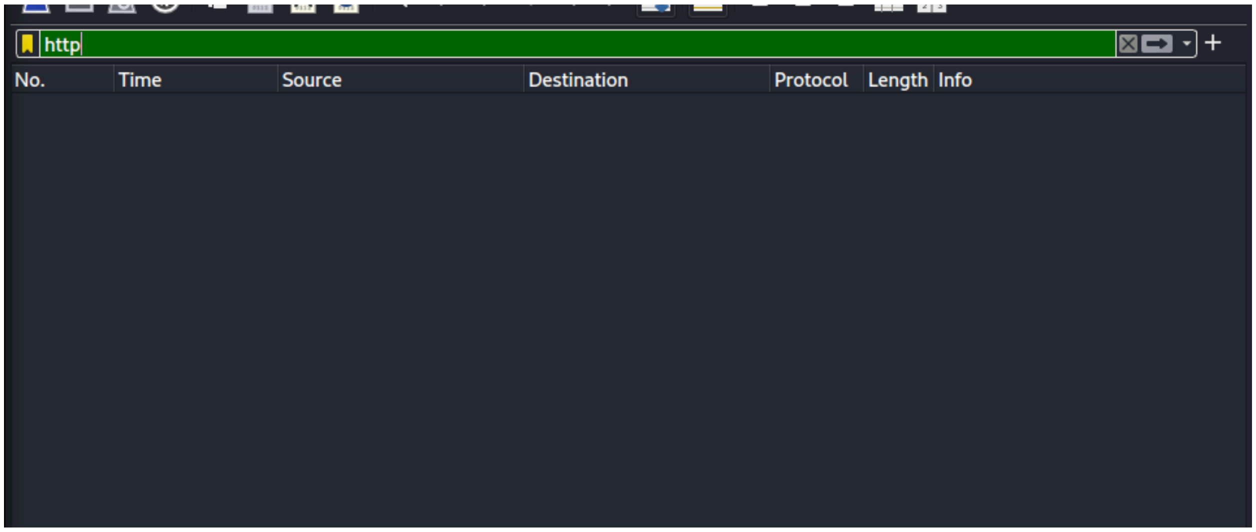
```
› Frame 1: 98 bytes on wire (784 bits), 98 bytes cap
› Ethernet II, Src: PCSSystemtec_19:01:c9 (08:00:27:
› Internet Protocol Version 4, Src: 10.65.101.98, Ds
› Internet Control Message Protocol
```

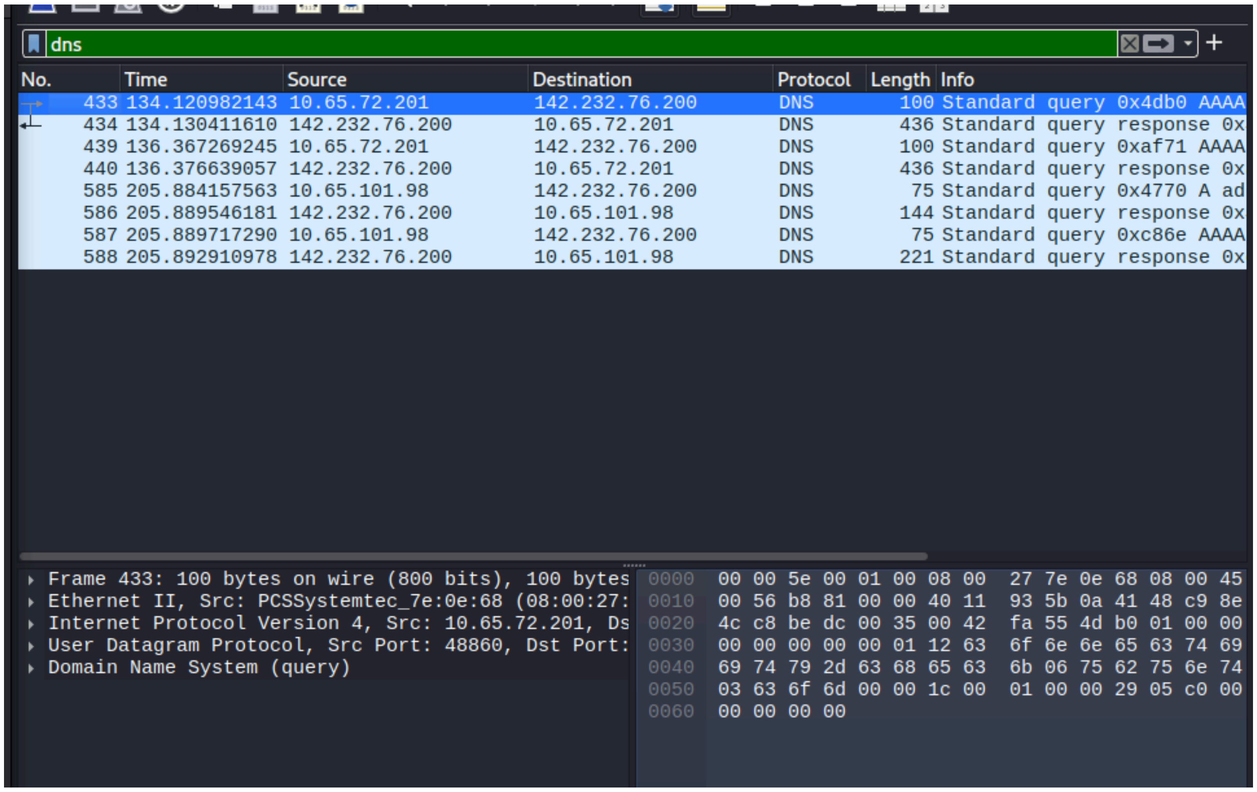
```
0000  00 00 5e 00 01 00 08 00  27 19 01 c9 08 00 45
0010  00 54 91 ed 40 00 40 01  60 d0 0a 41 65 62 8e
0020  49 4e 08 00 79 e0 00 03  00 0b f9 50 05 69 00
0030  00 00 b3 84 0d 00 00 00  00 00 10 11 12 13 14
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34
0060  36 37
```



```
┌──(user㉿vbox)-[~/Documents/lab7]
└─$ sudo tshark -i eth1 -a duration:120 -w capture_cli.pcapng
Running as user "root" and group "root". This could be dangerous.
Capturing on 'eth1'
281
```

Exercise 2.2

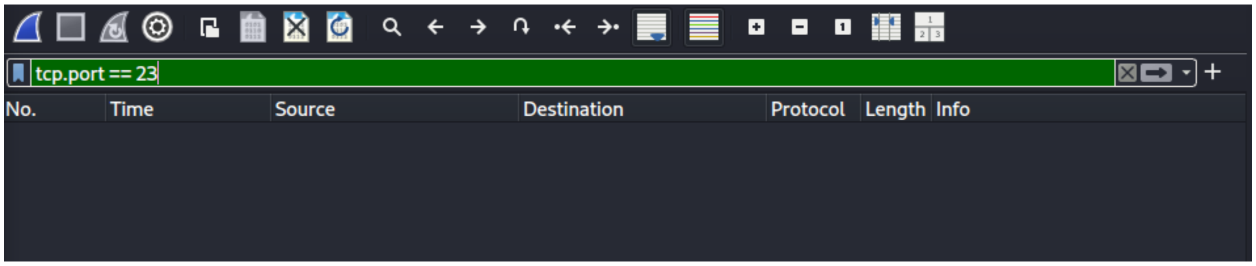




```
No.  Time           Source          Destination     Protocol  Length Info
 433 134.120982143 10.65.72.201    142.232.76.200   DNS          100 Standard query 0x4db0 AAAA
 434 134.130411610 142.232.76.200  10.65.72.201     DNS          436 Standard query response 0x
 439 136.367269245 10.65.72.201    142.232.76.200   DNS          100 Standard query 0xaf71 AAAA
 440 136.376639057 142.232.76.200  10.65.72.201     DNS          436 Standard query response 0x
 585 205.884157563 10.65.101.98    142.232.76.200   DNS           75 Standard query 0x4770 A ad
 586 205.889546181 142.232.76.200  10.65.101.98     DNS          144 Standard query response 0x
 587 205.889717290 10.65.101.98    142.232.76.200   DNS           75 Standard query 0xc86e AAAA
 588 205.892910978 142.232.76.200  10.65.101.98     DNS          221 Standard query response 0x
```

```
› Frame 433: 100 bytes on wire (800 bits), 100 bytes
› Ethernet II, Src: PCSSystemtec_7e:0e:68 (08:00:27:
› Internet Protocol Version 4, Src: 10.65.72.201, Ds
› User Datagram Protocol, Src Port: 48860, Dst Port:
› Domain Name System (query)
```

```
0000  00 00 5e 00 01 00 08 00  27 7e 0e 68 08 00 45
0010  00 56 b8 81 00 00 40 11  93 5b 0a 41 48 c9 8e
0020  4c c8 be dc 00 35 00 42  fa 55 4d b0 01 00 00
0030  00 00 00 00 00 01 12 63  6f 6e 6e 65 63 74 69
0040  69 74 79 2d 63 68 65 63  6b 06 75 62 75 6e 74
0050  03 63 6f 6d 00 00 1c 00  01 00 00 29 05 c0 00
0060  00 00 00 00
```

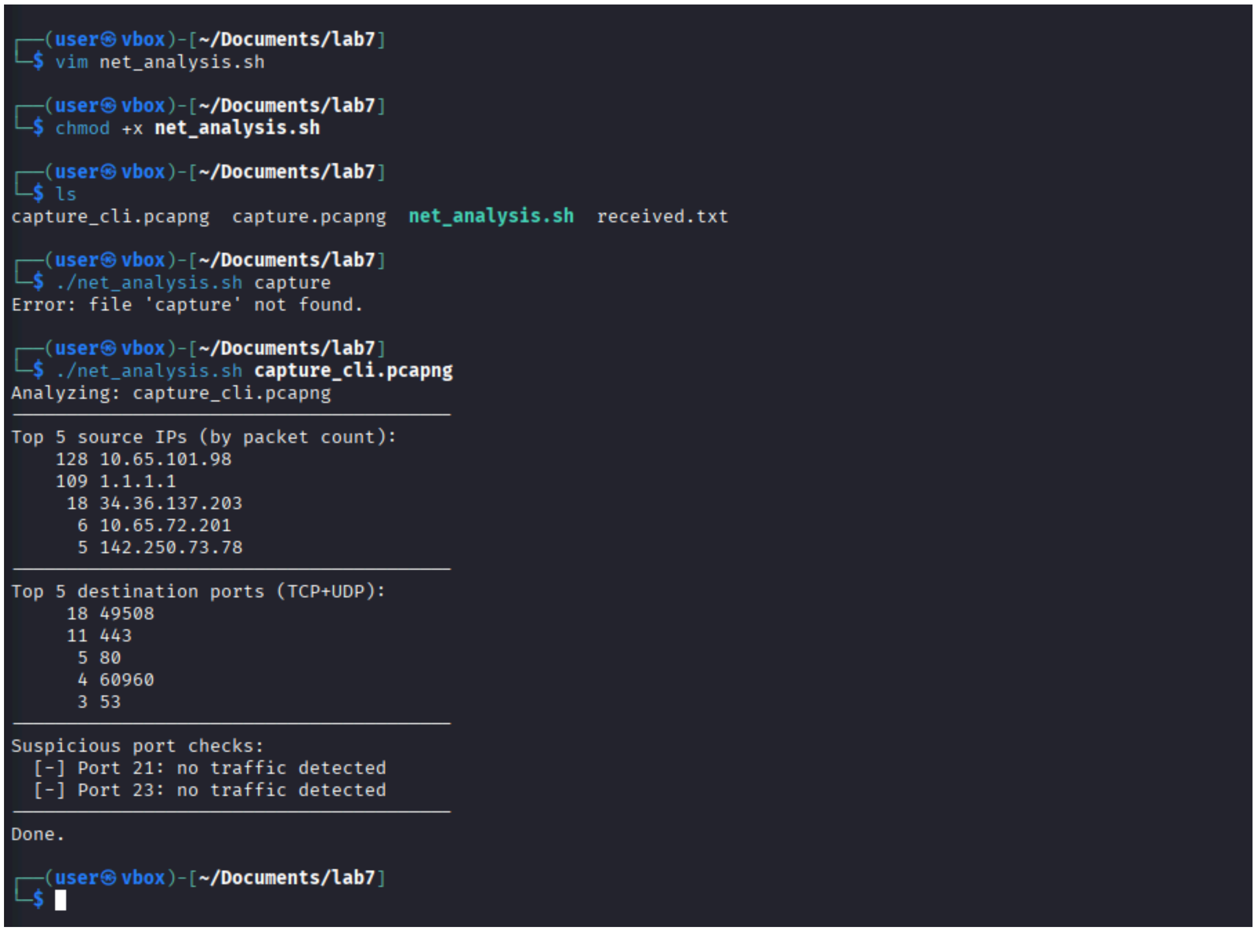


No signs of malicious activity, all traffic is from pinging google.com and 1.1.1.1

Exercise 2.3

```
#!/usr/bin/env bash
# net_analysis.sh - simple traffic analysis helper
# Usage: ./net_analysis.sh capture.pcapng

set -euo pipefail
IFS=$'\n\t'

PCAP="${1:-}"

if [[ -z "$PCAP" ]]; then
  echo "Usage: $0 <capture-file.pcapng|pcap>"
  exit 2
fi

if [[ ! -f "$PCAP" ]]; then
  echo "Error: file '$PCAP' not found."
  exit 3
fi

# Check tshark
if ! command -v tshark >/dev/null 2>&1; then
  echo "Error: tshark not found. Install tshark (Wireshark CLI) and retry."
  exit 4
fi

echo "Analyzing: $PCAP"
echo "----------------------------------------"

# Top 5 source IPs
echo "Top 5 source IPs (by packet count):"
tshark -r "$PCAP" -T fields -e ip.src 2>/dev/null \
  | grep -v '^$' \
  | sort \
  | uniq -c \
  | sort -rn \
  | head -n 5 \
  || echo "  (no IPv4 source addresses found)"

echo "----------------------------------------"

# Top 5 destination ports (handle tcp and udp)
echo "Top 5 destination ports (TCP+UDP):"
```

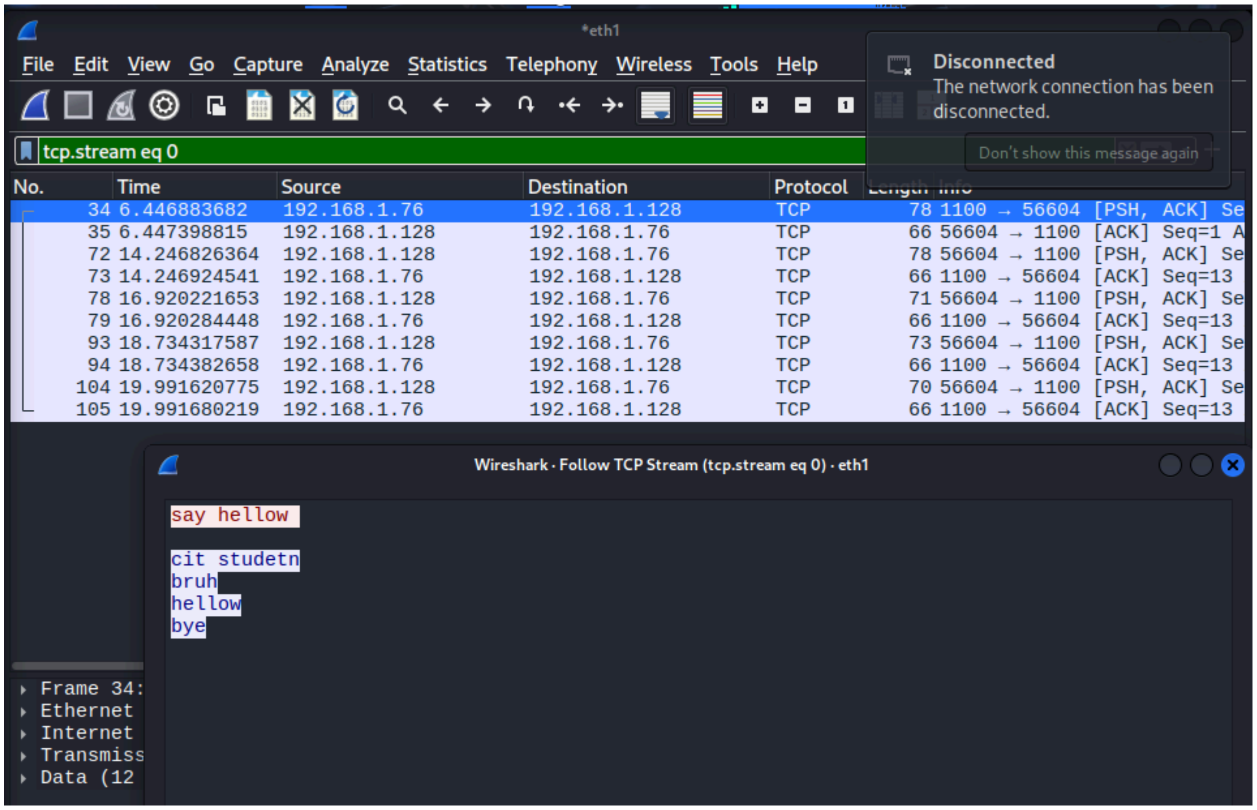
```
# Extract tcp.dstport and udp.dstport columns, pick whichever is present per line
tshark -r "$PCAP" -Y "tcp or udp" -T fields -e tcp.dstport -e udp.dstport 2>/dev/null \
  | awk -F'\t' '{ if ($1 != "") print $1; else if ($2 != "") print $2 }' \
  | grep -v '^$' \
  | sort -n \
  | uniq -c \
  | sort -rn \
  | head -n 5 \
  || echo "  (no TCP/UDP dst ports found)"

echo "----------------------------------------"

# Flag suspicious ports
SUSPICIOUS_PORTS=(21 23)
echo "Suspicious port checks:"
for p in "${SUSPICIOUS_PORTS[@]}"; do
  # Count occurrences where either tcp or udp destination is the port
  count=$(tshark -r "$PCAP" -Y "tcp.dstport == ${p} or udp.dstport == ${p} or
tcp.port == ${p} or udp.port == ${p}" -T fields -e frame.number 2>/dev/null | wc -l)
  if [[ "$count" -gt 0 ]]; then
    case "$p" in
      21) svc="FTP (21)";;
      23) svc="Telnet (23)";;
      *) svc="Port $p";;
    esac
    echo "  [!] $svc traffic detected: $count packet(s)"
  else
    echo "  [-] Port $p: no traffic detected"
  fi
done

echo "----------------------------------------"
echo "Done."
```



```
┌──(user㉿vbox)-[~/Documents/lab7]
└─$ vim net_analysis.sh

┌──(user㉿vbox)-[~/Documents/lab7]
└─$ chmod +x net_analysis.sh

┌──(user㉿vbox)-[~/Documents/lab7]
└─$ ls
capture_cli.pcapng  capture.pcapng  net_analysis.sh  received.txt

┌──(user㉿vbox)-[~/Documents/lab7]
└─$ ./net_analysis.sh capture
Error: file 'capture' not found.

┌──(user㉿vbox)-[~/Documents/lab7]
└─$ ./net_analysis.sh capture_cli.pcapng
Analyzing: capture_cli.pcapng
────────────────────────────────────────
Top 5 source IPs (by packet count):
    128 10.65.101.98
    109 1.1.1.1
     18 34.36.137.203
      6 10.65.72.201
      5 142.250.73.78
────────────────────────────────────────
Top 5 destination ports (TCP+UDP):
     18 49508
     11 443
      5 80
      4 60960
      3 53
────────────────────────────────────────
Suspicious port checks:
  [-] Port 21: no traffic detected
  [-] Port 23: no traffic detected
────────────────────────────────────────
Done.

┌──(user㉿vbox)-[~/Documents/lab7]
└─$ 
```

9. 10.65.101.98
10. No
11. Automation can improve real-world incident detection by quickly analyzing large amounts of network or system data, identifying suspicious patterns in real time, and alerting defenders faster than manual monitoring — reducing response time and human error.

Part 3



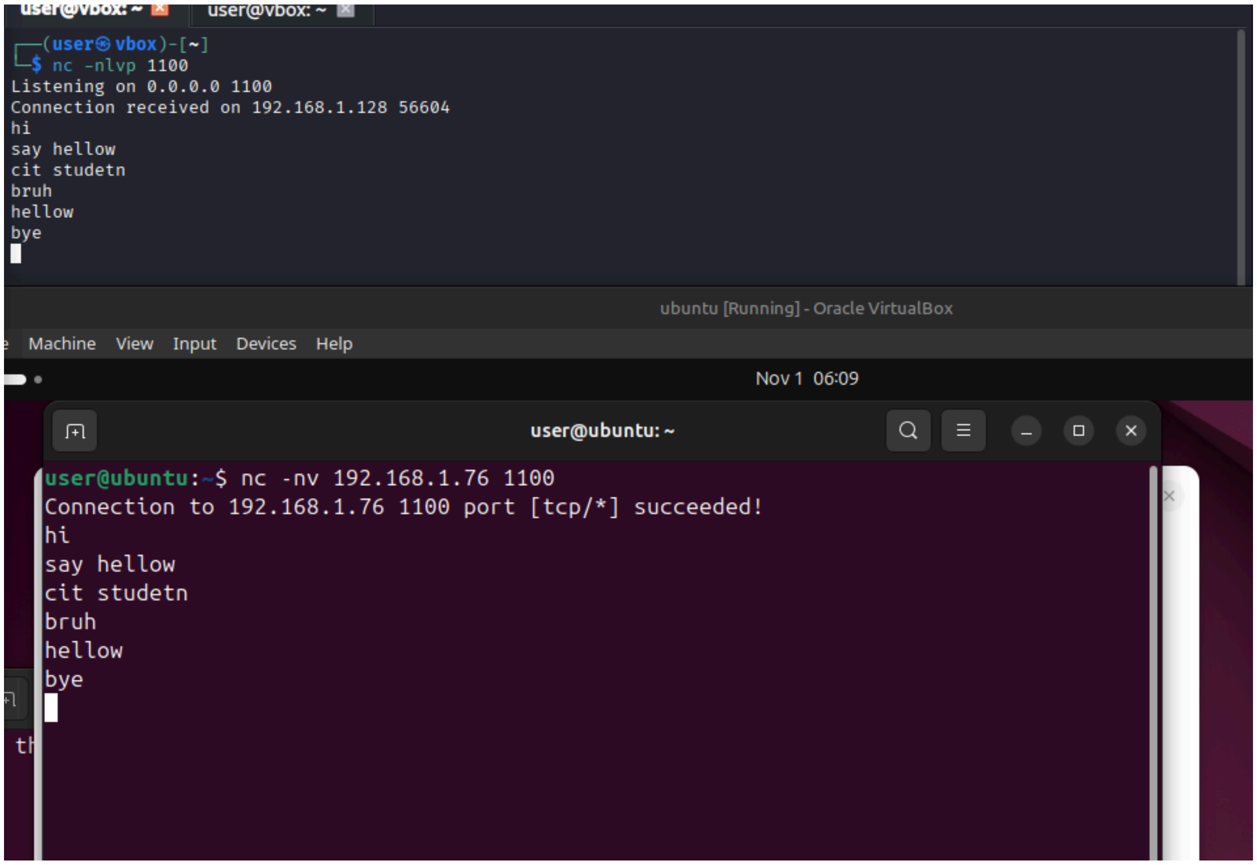
*eth1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Disconnected
The network connection has been disconnected.
Don't show this message again
Wireshark · Follow TCP Stream (tcp.stream eq 0) · eth1

```
tcp.stream eq 0
```

```
No.   Time            Source          Destination     Protocol Length Info
   34 6.446883682     192.168.1.76    192.168.1.128   TCP        78 1100 → 56604 [PSH, ACK] Se
   35 6.447398815     192.168.1.128   192.168.1.76    TCP        66 56604 → 1100 [ACK] Seq=1 A
   72 14.246826364    192.168.1.128   192.168.1.76    TCP        78 56604 → 1100 [PSH, ACK] Se
   73 14.246924541    192.168.1.76    192.168.1.128   TCP        66 1100 → 56604 [ACK] Seq=13
   78 16.920221653    192.168.1.128   192.168.1.76    TCP        71 56604 → 1100 [PSH, ACK] Se
   79 16.920284448    192.168.1.76    192.168.1.128   TCP        66 1100 → 56604 [ACK] Seq=13
   93 18.734317587    192.168.1.128   192.168.1.76    TCP        73 56604 → 1100 [PSH, ACK] Se
   94 18.734382658    192.168.1.76    192.168.1.128   TCP        66 1100 → 56604 [ACK] Seq=13
  104 19.991620775    192.168.1.128   192.168.1.76    TCP        70 56604 → 1100 [PSH, ACK] Se
  105 19.991680219    192.168.1.76    192.168.1.128   TCP        66 1100 → 56604 [ACK] Seq=13
```

```
say hellow

cit studetn
bruh
hellow
bye
```

```
Frame 34:
Ethernet
Internet
Transmiss
Data (12
```


user@vbox: ~
user@vbox: ~
ubuntu [Running] - Oracle VirtualBox
Machine View Input Devices Help
Nov 1 06:09
user@ubuntu: ~

```
┌──(user㉿vbox)-[~]
└─$ nc -nlvp 1100
Listening on 0.0.0.0 1100
Connection received on 192.168.1.128 56604
hi
say hellow
cit studetn
bruh
hellow
bye
```

```
user@ubuntu:~$ nc -nv 192.168.1.76 1100
Connection to 192.168.1.76 1100 port [tcp/*] succeeded!
hi
say hellow
cit studetn
bruh
hellow
bye
```

12. When using netcat without encryption, every message or file appears in plaintext in Wireshark, the content itself, source and destination IPs, ports and timestamp
13. Encryption scrambles the data so the content is unreadable in wireshark. IP address, ports and size of the packet will still be available, but the content is not.
14. Confidentiality: Plaintext communication is insecure — attackers can easily intercept and read sensitive information. Encryption protects privacy by hiding data content.

    Forensics: Encrypted traffic makes it harder for investigators to see what was exchanged, so they must rely on metadata and timing analysis instead of message contents.