ECEN 665- Packet Sniffing
Assignment 1
Garuda Suma Pranavi
UIN 926009146

This document contains implementation details for the source code, screenshots of the packets captured at the output and analysis of the different protocols used for transfer of information over the network.

Makefile: g++ -std=c++0x finalcode.cpp -lpcap -o output

```
./output httpsession.pcap 1
        or
./output tfsession.pcap 2
        or
./output tfsession.pcap 3
```

Here, finalcode is my cpp code which I built on CodeBlocks- Ubuntu 16.04.03 64 bit and run on the terminal. -lpcap is to associate the library files of libpcap with our program and output is the name of the object file created.

The second line is to run the program i.e. the object file, with first argument as the file from which we need to reassemble http, telnet and ftp sessions. The second argument is for the type of packets you want to capture: 1 for http, 2 for telnet and 3 for ftp.

In this program we will be reading packets till we encounter an error message.

Program Flow/Design:

I first just implemented the packet level implementation to understand the data transfer (which was pretty easy). Then with the help of many open source projects online I was able to reassemble the packets for different sessions. (I have analyzed the screenshots for the packet level implementation in detail and posted screenshots for the application level sessions).

1. We take the filename as input from the user whose packets are to be captured, we also mention the type of packets we want to capture: http, telnet or ftp.

2. We then use pcap_open_offline to open a saved capture file for reading:

```
  pcap_t *pcap_open_offline(const char *fname, char *errbuf);
```

3. We know which packets to look for and we search for these packets in a loop till we reach the cnt limit specified or till infinity. The first argument is our session handle, next is cnt (if set to -1 or 0, the processing goes on until an error condition occurs), then is the name of the callback funtcion, the last argument is just in case we have some additional arguments to send.

```
  int pcap_loop(pcap_t *p, int cnt,pcap_handler callback, u_char *user);
```

4. Now, we go to the callback function, where header contains the timestamp and length of the packet, the last argument is a pointer to the serialized version of the Ethernet, IP and TCP headers initialized before.

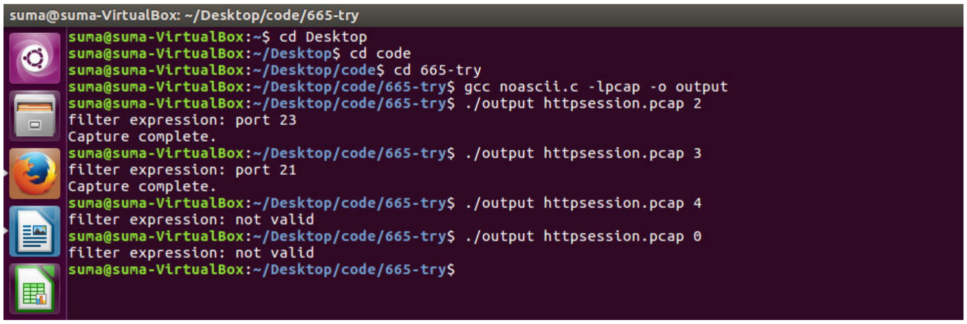
```
   void got_packet(u_char *args,const struct pcap_pkthdr *header,const u_char
*packet);
```

5. In the callback function, we do some calculations to find the exact location where the payload starts, by extracting information from the headers (Ethernet is fixed, but IP and TCP headers can have variable lengths). Then comes the difficult part, we have to reassemble the TCP, FTP and TELNET sessions. We first check if the input type of packets and port numbers is a match. If yes, we search a record with the same {ip,port} pair. If there is no match, we insert a new record, else we get the iterator and place our new record there. This process is repeated dynamically for both request and response.

6.We return back to main, iterate for the whole session and display the reassembled application level data.

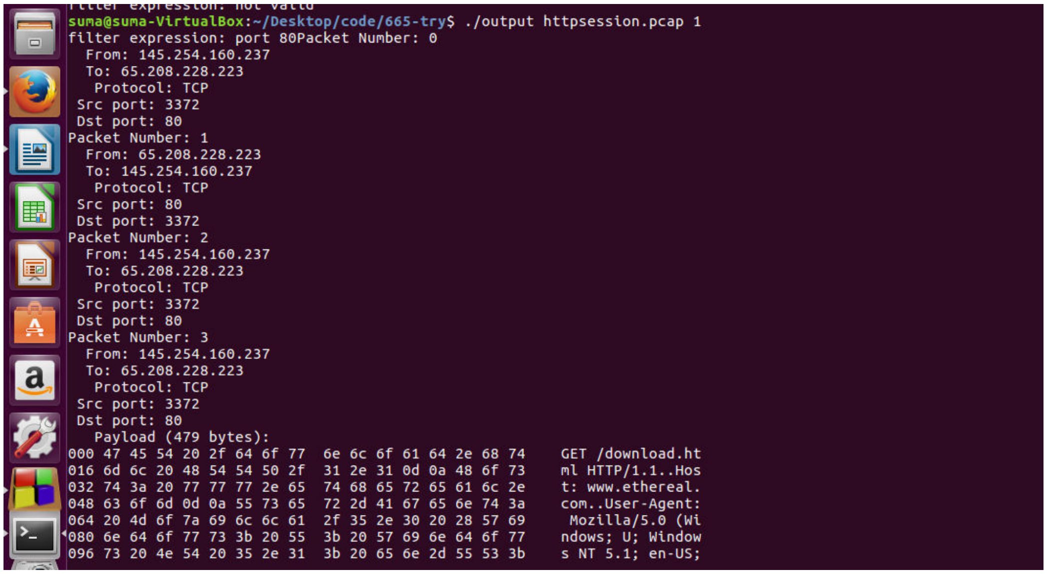
Screenshots (The italicized data is my interpretation) – Application Level data screenshots at the end.

1. Compiling the program with the given httpsession.pcap file. First we check for telnet, no packets are captured, then for ftp, no packets are captured. Then we try with some random numbers and we get the output as invalid expression. We can see that the filter is working correctly.



```
suma@suma-VirtualBox: ~/Desktop/code/665-try
suma@suma-VirtualBox:~$ cd Desktop
suma@suma-VirtualBox:~/Desktop$ cd code
suma@suma-VirtualBox:~/Desktop/code$ cd 665-try
suma@suma-VirtualBox:~/Desktop/code/665-try$ gcc noascii.c -lpcap -o output
suma@suma-VirtualBox:~/Desktop/code/665-try$ ./output httpsession.pcap 2
filter expression: port 23
Capture complete.
suma@suma-VirtualBox:~/Desktop/code/665-try$ ./output httpsession.pcap 3
filter expression: port 21
Capture complete.
suma@suma-VirtualBox:~/Desktop/code/665-try$ ./output httpsession.pcap 4
filter expression: not valid
suma@suma-VirtualBox:~/Desktop/code/665-try$ ./output httpsession.pcap 0
filter expression: not valid
suma@suma-VirtualBox:~/Desktop/code/665-try$
```

This screenshot shows the captured HTTP packets form the httpsession.pcap.



```
filter expression: not valid
suma@suma-VirtualBox:~/Desktop/code/665-try$ ./output httpsession.pcap 1
filter expression: port 80Packet Number: 0
  From: 145.254.160.237
  To: 65.208.228.223
   Protocol: TCP
 Src port: 3372
 Dst port: 80
Packet Number: 1
  From: 65.208.228.223
  To: 145.254.160.237
   Protocol: TCP
 Src port: 80
 Dst port: 3372
Packet Number: 2
  From: 145.254.160.237
  To: 65.208.228.223
   Protocol: TCP
 Src port: 3372
 Dst port: 80
Packet Number: 3
  From: 145.254.160.237
  To: 65.208.228.223
   Protocol: TCP
 Src port: 3372
 Dst port: 80
   Payload (479 bytes):
000 47 45 54 20 2f 64 6f 77  6e 6c 6f 61 64 2e 68 74    GET /download.ht
016 6d 6c 20 48 54 54 50 2f  31 2e 31 0d 0a 48 6f 73    ml HTTP/1.1..Hos
032 74 3a 20 77 77 77 2e 65  74 68 65 72 65 61 6c 2e    t: www.ethereal.
048 63 6f 6d 0d 0a 55 73 65  72 2d 41 67 65 6e 74 3a    com..User-Agent:
064 20 4d 6f 7a 69 6c 6c 61  2f 35 2e 30 20 28 57 69     Mozilla/5.0 (Wi
080 6e 64 6f 77 73 3b 20 55  3b 20 57 69 6e 64 6f 77    ndows; U; Window
096 73 20 4e 54 20 35 2e 31  3b 20 65 6e 2d 55 53 3b    s NT 5.1; en-US;
```

Analysing the information captured packet by packet:

//TCP SYN packet (TCP three way handshake)

Packet Number: 1
From: 145.254.160.237 *//source IP address*
To: 65.208.228.223 *//destination IP address (IP address of the http server)*
Protocol: TCP
Src port: 3372 *//source port, dynamic port selected for this connection*
Dst port: 80 *//destination port, for http it will be 80*
(all the packets for this connection will have matching MAC addresses, IP addresses and port numbers)

//TCP SYN/ACK packet (TCP three way handshake)
Packet Number: 2
From: 65.208.228.223
To: 145.254.160.237
Protocol: TCP
Src port: 80
Dst port: 3372

//TCP ACK packet (TCP three way handshake)
Packet Number: 3
From: 145.254.160.237
To: 65.208.228.223
Protocol: TCP
Src port: 3372
Dst port: 80

//First HTTP packet – GET/
Packet Number: 4
From: 145.254.160.237
To: 65.208.228.223
Protocol: TCP
Src port: 3372
Dst port: 80

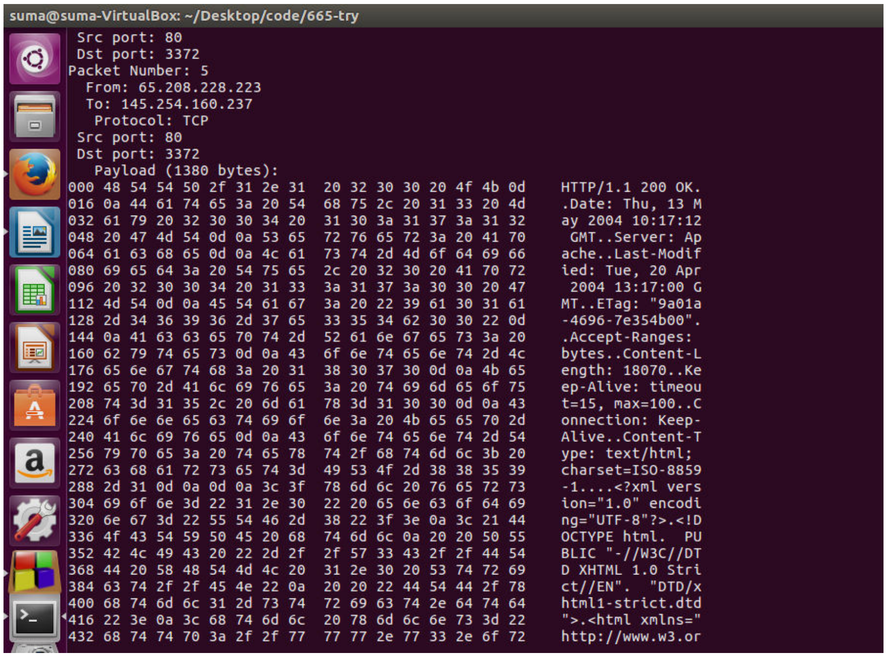
Payload (479 bytes):

GET /download.html HTTP/1.1.. *//GET -Access Control Request Method*
Host: www.ethereal.com.. *//Host- The domain name of the server*
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.6) Gecko/20040113.. *//User-Agent //HTTP identifies the client software originating the request using the user-agent header*

Accept:ext/xml,application/xml,application/xhtml+xml,text/html;q=0.9,ext/plain;q=0.8,image/png,image/jpeg,image/gif; q=0.2,*/*;q=0.1.. *//Accept*
//Media type that is acceptable for the response (Content negotiation)

Accept-Language: en-us,en;q=0.5.. *//Accept-Language*
//list of acceptable human languages for response
Accept-Encoding: gzip,deflate.. *//list of acceptable encodings, HTTP compression*
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7.. *//Character sets that are acceptable*

Keep-Alive: 300..   *//Keep-Alive*
Connection: keep-alive..   *//Connection*
Referer: http://www.ethereal.com/development.html.… *//Referrer*
//This is the address of the previous web page from which a link to the currently requested page has followed.



```
suma@suma-VirtualBox: ~/Desktop/code/665-try
 Src port: 80
 Dst port: 3372
Packet Number: 5
  From: 65.208.228.223
  To: 145.254.160.237
   Protocol: TCP
 Src port: 80
 Dst port: 3372
   Payload (1380 bytes):
000 48 54 54 50 2f 31 2e 31  20 32 30 30 20 4f 4b 0d    HTTP/1.1 200 OK.
016 0a 44 61 74 65 3a 20 54  68 75 2c 20 31 33 20 4d    .Date: Thu, 13 M
032 61 79 20 32 30 30 34 20  31 30 3a 31 37 3a 31 32    ay 2004 10:17:12
048 20 47 4d 54 0d 0a 53 65  72 76 65 72 3a 20 41 70     GMT..Server: Ap
064 61 63 68 65 0d 0a 4c 61  73 74 2d 4d 6f 64 69 66    ache..Last-Modif
080 69 65 64 3a 20 54 75 65  2c 20 32 30 20 41 70 72    ied: Tue, 20 Apr
096 20 32 30 30 34 20 31 33  3a 31 37 3a 30 30 20 47     2004 13:17:00 G
112 4d 54 0d 0a 45 54 61 67  3a 20 22 39 61 30 31 61    MT..ETag: "9a01a
128 2d 34 36 39 36 2d 37 65  33 35 34 62 30 30 22 0d    -4696-7e354b00".
144 0a 41 63 63 65 70 74 2d  52 61 6e 67 65 73 3a 20    .Accept-Ranges:
160 62 79 74 65 73 0d 0a 43  6f 6e 74 65 6e 74 2d 4c    bytes..Content-L
176 65 6e 67 74 68 3a 20 31  38 30 37 30 0d 0a 4b 65    ength: 18070..Ke
192 65 70 2d 41 6c 69 76 65  3a 20 74 69 6d 65 6f 75    ep-Alive: timeou
208 74 3d 31 35 2c 20 6d 61  78 3d 31 30 30 0d 0a 43    t=15, max=100..C
224 6f 6e 6e 65 63 74 69 6f  6e 3a 20 4b 65 65 70 2d    onnection: Keep-
240 41 6c 69 76 65 0d 0a 43  6f 6e 74 65 6e 74 2d 54    Alive..Content-T
256 79 70 65 3a 20 74 65 78  74 2f 68 74 6d 6c 3b 20    ype: text/html;
272 63 68 61 72 73 65 74 3d  49 53 4f 2d 38 38 35 39    charset=ISO-8859
288 2d 31 0d 0a 0d 0a 3c 3f  78 6d 6c 20 76 65 72 73    -1....<?xml vers
304 69 6f 6e 3d 22 31 2e 30  22 20 65 6e 63 6f 64 69    ion="1.0" encodi
320 6e 67 3d 22 55 54 46 2d  38 22 3f 3e 0a 3c 21 44    ng="UTF-8"?>.<!D
336 4f 43 54 59 50 45 20 68  74 6d 6c 0a 20 20 50 55    OCTYPE html.  PU
352 42 4c 49 43 20 22 2d 2f  2f 57 33 43 2f 2f 44 54    BLIC "-//W3C//DT
368 44 20 58 48 54 4d 4c 20  31 2e 30 20 53 74 72 69    D XHTML 1.0 Stri
384 63 74 2f 2f 45 4e 22 0a  20 20 22 44 54 44 2f 78    ct//EN".  "DTD/x
400 68 74 6d 6c 31 2d 73 74  72 69 63 74 2e 64 74 64    html1-strict.dtd
416 22 3e 0a 3c 68 74 6d 6c  20 78 6d 6c 6e 73 3d 22    ">.<html xmlns="
432 68 74 74 70 3a 2f 2f 77  77 77 2e 77 33 2e 6f 72    http://www.w3.or
```

//TCP ACK (Server TCP acknowledgement of receiving the GET request)
Packet Number: 5
From: 65.208.228.223
To: 145.254.160.237
Protocol: TCP
Src port: 80
Dst port: 3372

Packet Number: 6
From: 65.208.228.223
To: 145.254.160.237
Protocol: TCP
Src port: 80
Dst port: 3372

Payload (1380 bytes):

HTTP/1.1 200 OK..   *//the request has succeeded, the information returned with the response is dependent on the method used in the request – Status Line*

Date: Thu, 13 May 2004 10:17:12 GMT..\   *//General Headers*

Server: Apache..   *//Response Headers and Entity Headers*
Last-Modified: Tue, 20 Apr 2004 13:17:00 GMT..

```
ETag: "9a01a-4696-7e354b00"..
Accept-Ranges: bytes..
Content-Length: 18070..
Keep-Alive: timeout=15, max=100..
Connection: Keep-Alive..
Content-Type: text/html;

charset=ISO-8859-1.…        download                         //Style details of a HTML page for ethereal
<?xml version="1.0" encoding="UTF-8"?>.
<!DOCTYPE html.  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN".  "DTD/xhtml1-strict.dtd">.
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">.  <head>.
   <title>Ethereal: Download</title>.
 <style type="text/css" media="all">..@import url("mm/css/ethereal-3-0.css");.
</style>.</head>.  <body>.   <div class="top">.    <table width="100%" cellspacing="0"
cellpadding="0" border="0" summary="">.     <tr>.
<td valign="middle" width="1">.. <a href="/"><img class="logo" title="Ethereal home"
src="mm/image\elogo-64-trans.gif" alt="" width="64" height="64"></img></a>.
  </td>.   <td align="left" valign="middle">.  <h2>Ethereal</h2>.     <h5 style="white-space:
nowrap;">Download</h5>.    </td>.    <td align="right">..
<table style="margin-right: 10px;" cellspacing="0" cellpadding="0" border="0" summary="">.
 <form name="search" method="post" action="http://www.ethereal.com/cgi-bin/htsearch">.
    <tr>..   <td>..    <div class="topformtext">.
```

(This packet capture is basically in reference to downloading ethereal (network troubleshooting and alnalysis software from www.ethereal.com) //href contains the URL (actual link) and the clickable text on the page.

//TCP packet sent with FIN bit sent (the connection is no longer needed and request for it to be closed)

Packet Number: 39
From: 145.254.160.237
To: 65.208.228.223
Protocol: TCP
Src port: 3372
Dst port: 80

//The server sends an ACK for the client's FIN and also the FIN

Packet Number: 40
From: 145.254.160.237
To: 65.208.228.223
Protocol: TCP
Src port: 3372
Dst port: 80

//The client sends and ACK for the server's FIN

Packet Number: 41
From: 65.208.228.223
To: 145.254.160.237
Protocol: TCP
Src port: 80

Dst port: 3372

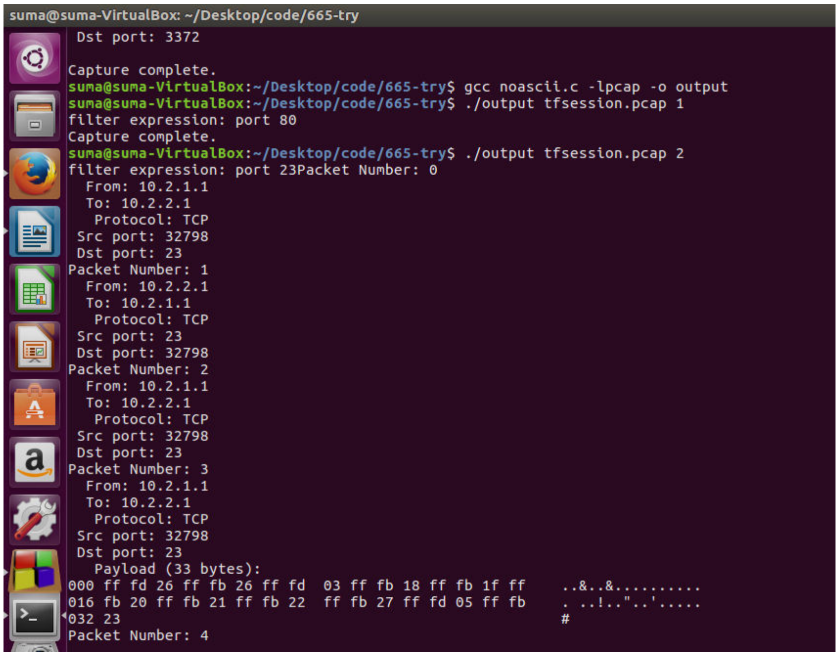
Capture complete. In the similar way we can analyse all the packets obtained for HTTP.

2. We now capture telnet packets from the tfsession.pcap file given.

TELNET is an interactive data transfer protocol. For each character typed, we send 3 packets:

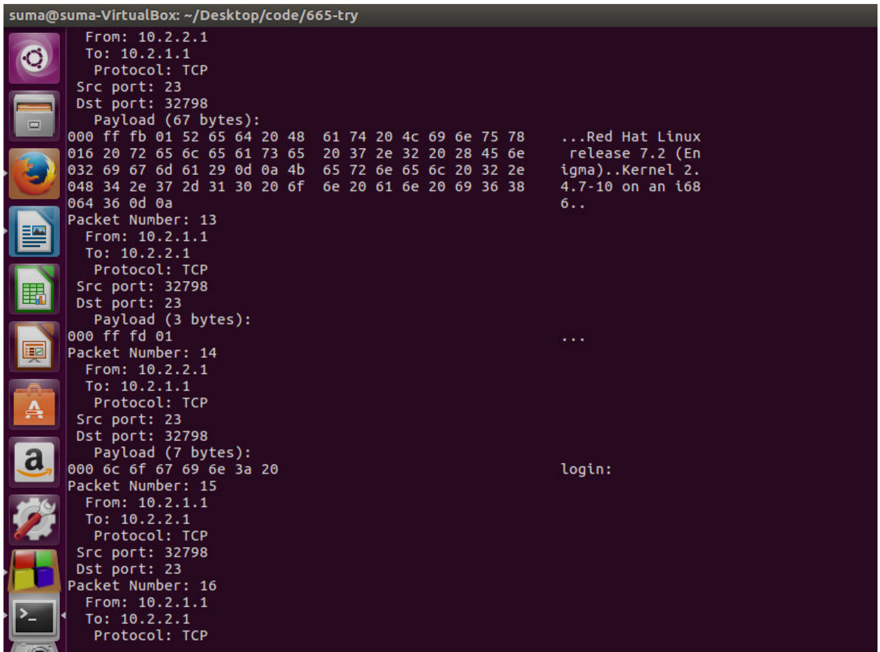
1. client → server : Send typed character
2. server → client : echo of character and acknowledgement of 1st packet
3. client → server : acknowledgement of second packet

Filter expression: port 23



```
suma@suma-VirtualBox: ~/Desktop/code/665-try
 Dst port: 3372

Capture complete.
suma@suma-VirtualBox:~/Desktop/code/665-try$ gcc noascii.c -lpcap -o output
suma@suma-VirtualBox:~/Desktop/code/665-try$ ./output tfsession.pcap 1
filter expression: port 80
Capture complete.
suma@suma-VirtualBox:~/Desktop/code/665-try$ ./output tfsession.pcap 2
filter expression: port 23Packet Number: 0
  From: 10.2.1.1
  To: 10.2.2.1
   Protocol: TCP
 Src port: 32798
 Dst port: 23
Packet Number: 1
  From: 10.2.2.1
  To: 10.2.1.1
   Protocol: TCP
 Src port: 23
 Dst port: 32798
Packet Number: 2
  From: 10.2.1.1
  To: 10.2.2.1
   Protocol: TCP
 Src port: 32798
 Dst port: 23
Packet Number: 3
  From: 10.2.1.1
  To: 10.2.2.1
   Protocol: TCP
 Src port: 32798
 Dst port: 23
   Payload (33 bytes):
000 ff fd 26 ff fb 26 ff fd  03 ff fb 18 ff fb 1f ff    ..&..&.........
016 fb 20 ff fb 21 ff fb 22  ff fb 27 ff fd 05 ff fb    . ..!.."..'.....
032 23                                                  #
Packet Number: 4
```

The initial 3 packets are for the TCP three way handshake as mentioned earlier in the HTTP session. Using the information mentioned above, we decoded the login id and passwords attempted over the TELNET connection.



```
suma@suma-VirtualBox: ~/Desktop/code/665-try
  From: 10.2.2.1
  To: 10.2.1.1
   Protocol: TCP
 Src port: 23
 Dst port: 32798
   Payload (67 bytes):
000 ff fb 01 52 65 64 20 48  61 74 20 4c 69 6e 75 78    ...Red Hat Linux
016 20 72 65 6c 65 61 73 65  20 37 2e 32 20 28 45 6e     release 7.2 (En
032 69 67 6d 61 29 0d 0a 4b  65 72 6e 65 6c 20 32 2e    igma)..Kernel 2.
048 34 2e 37 2d 31 30 20 6f  6e 20 61 6e 20 69 36 38    4.7-10 on an i68
064 36 0d 0a                                            6..
Packet Number: 13
  From: 10.2.1.1
  To: 10.2.2.1
   Protocol: TCP
 Src port: 32798
 Dst port: 23
   Payload (3 bytes):
000 ff fd 01                                            ...
Packet Number: 14
  From: 10.2.2.1
  To: 10.2.1.1
   Protocol: TCP
 Src port: 23
 Dst port: 32798
   Payload (7 bytes):
000 6c 6f 67 69 6e 3a 20                                login:
Packet Number: 15
  From: 10.2.1.1
  To: 10.2.2.1
   Protocol: TCP
 Src port: 32798
 Dst port: 23
Packet Number: 16
  From: 10.2.1.1
  To: 10.2.2.1
   Protocol: TCP
```

1. login : cs6262
password: welkfjwe

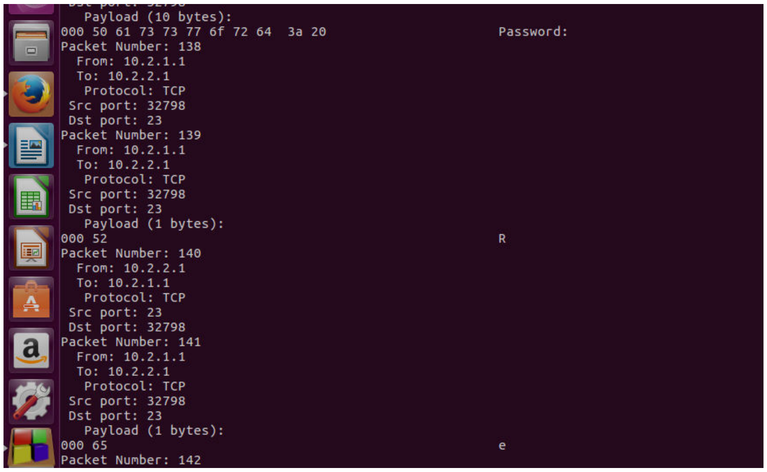
Login incorrect

2. login: cs6262
password: w;lerkwel;f

Login incorrect

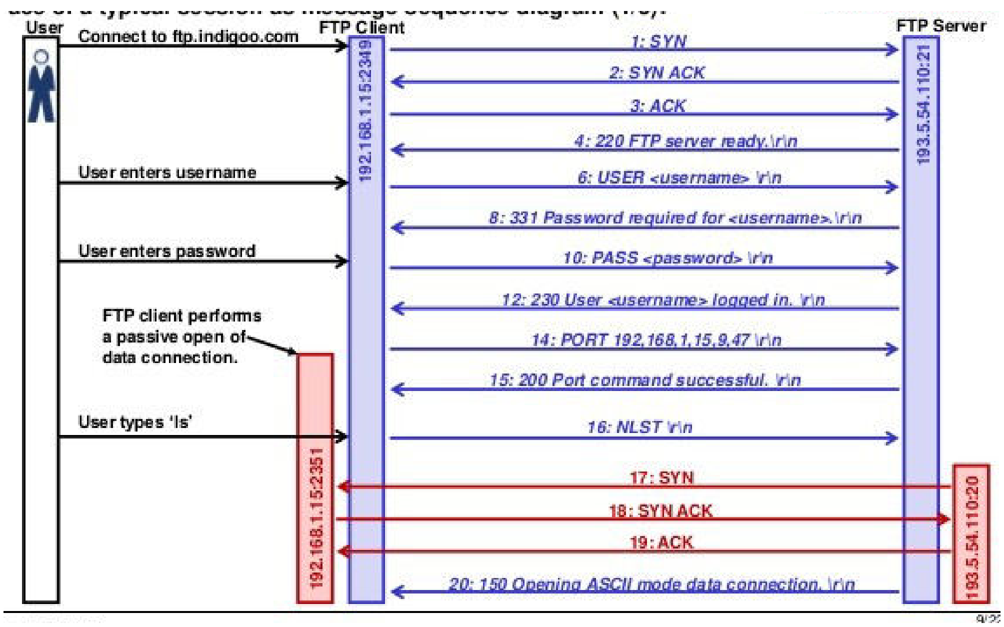
3. login: cs6262
password: Re=mi3vE4

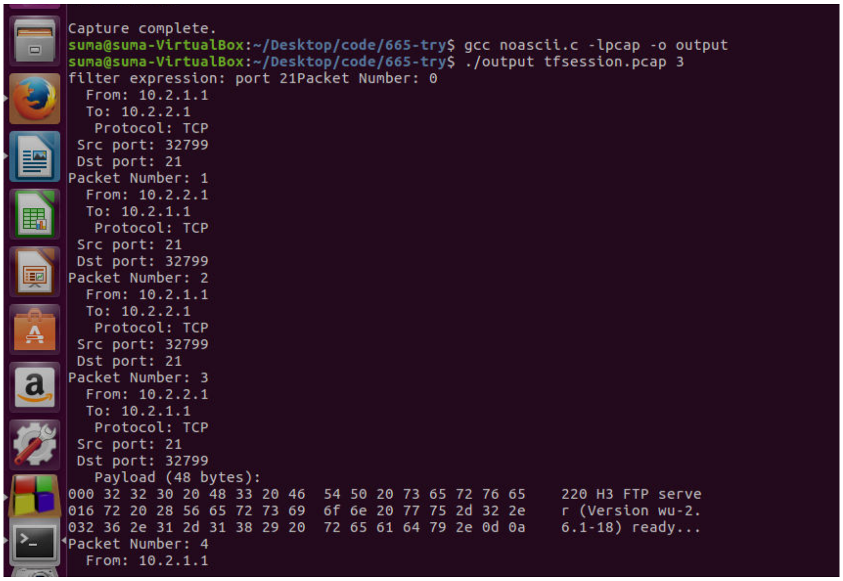
This time we have a successful login, and last login's details are mentioned on the terminal



```
   Payload (10 bytes):
000 50 61 73 73 77 6f 72 64  3a 20                        Password:
Packet Number: 138
  From: 10.2.1.1
  To: 10.2.2.1
   Protocol: TCP
 Src port: 32798
 Dst port: 23
Packet Number: 139
  From: 10.2.1.1
  To: 10.2.2.1
   Protocol: TCP
 Src port: 32798
 Dst port: 23
   Payload (1 bytes):
000 52                                                    R
Packet Number: 140
  From: 10.2.2.1
  To: 10.2.1.1
   Protocol: TCP
 Src port: 23
 Dst port: 32798
Packet Number: 141
  From: 10.2.1.1
  To: 10.2.2.1
   Protocol: TCP
 Src port: 32798
 Dst port: 23
   Payload (1 bytes):
000 65                                                    e
Packet Number: 142
```

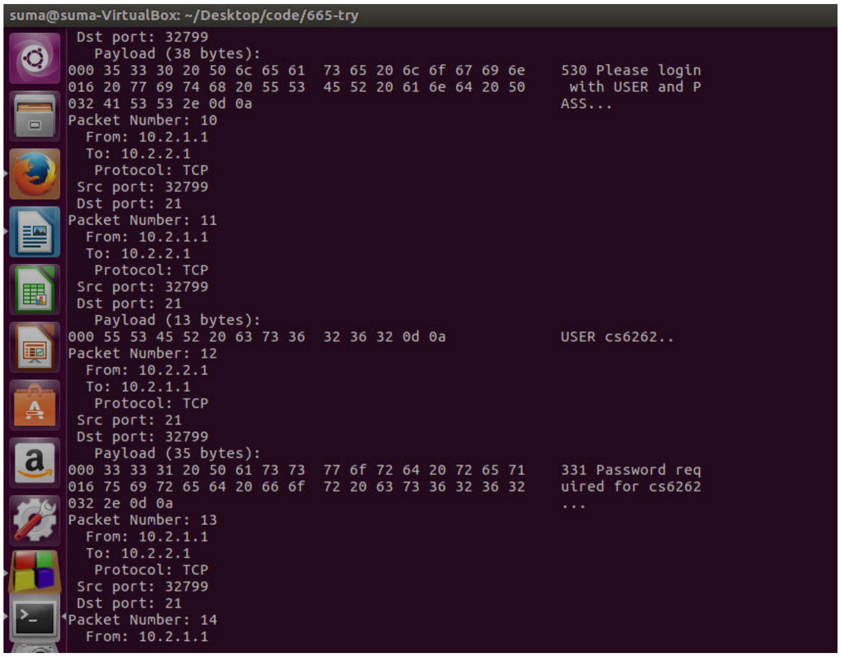
3. Tracing FTP packets: (Blue- Control Connection, Red- Data Connection)


User
FTP Client
FTP Server
Connect to ftp.indigoo.com
192.168.1.15:2349
193.5.54.110:21
1: SYN
2: SYN ACK
3: ACK
4: 220 FTP server ready.\r\n
User enters username
6: USER <username> \r\n
8: 331 Password required for <username>.\r\n
User enters password
10: PASS <password> \r\n
12: 230 User <username> logged in. \r\n
FTP client performs a passive open of data connection.
14: PORT 192,168,1,15,9,47 \r\n
15: 200 Port command successful. \r\n
User types 'ls'
16: NLST \r\n
192.168.1.15:2351
193.5.54.110:20
17: SYN
18: SYN ACK
19: ACK
20: 150 Opening ASCII mode data connection. \r\n



```
Capture complete.
suma@suma-VirtualBox:~/Desktop/code/665-try$ gcc noascii.c -lpcap -o output
suma@suma-VirtualBox:~/Desktop/code/665-try$ ./output tfsession.pcap 3
filter expression: port 21Packet Number: 0
  From: 10.2.1.1
  To: 10.2.2.1
   Protocol: TCP
 Src port: 32799
 Dst port: 21
Packet Number: 1
  From: 10.2.2.1
  To: 10.2.1.1
   Protocol: TCP
 Src port: 21
 Dst port: 32799
Packet Number: 2
  From: 10.2.1.1
  To: 10.2.2.1
   Protocol: TCP
 Src port: 32799
 Dst port: 21
Packet Number: 3
  From: 10.2.2.1
  To: 10.2.1.1
   Protocol: TCP
 Src port: 21
 Dst port: 32799
   Payload (48 bytes):
000 32 32 30 20 48 33 20 46  54 50 20 73 65 72 76 65    220 H3 FTP serve
016 72 20 28 56 65 72 73 69  6f 6e 20 77 75 2d 32 2e    r (Version wu-2.
032 36 2e 31 2d 31 38 29 20  72 65 61 64 79 2e 0d 0a    6.1-18) ready...
Packet Number: 4
  From: 10.2.1.1
```

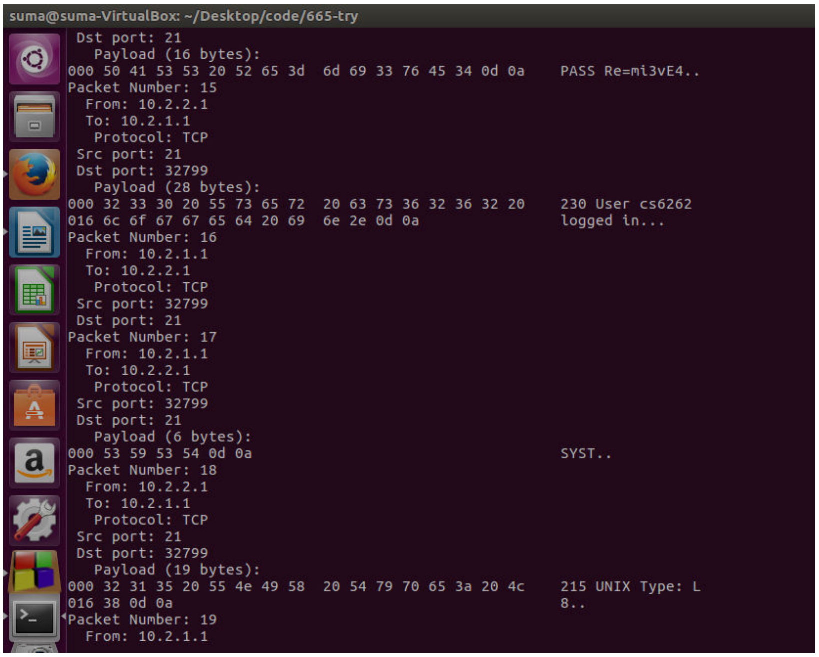
//220 – Service ready for new user



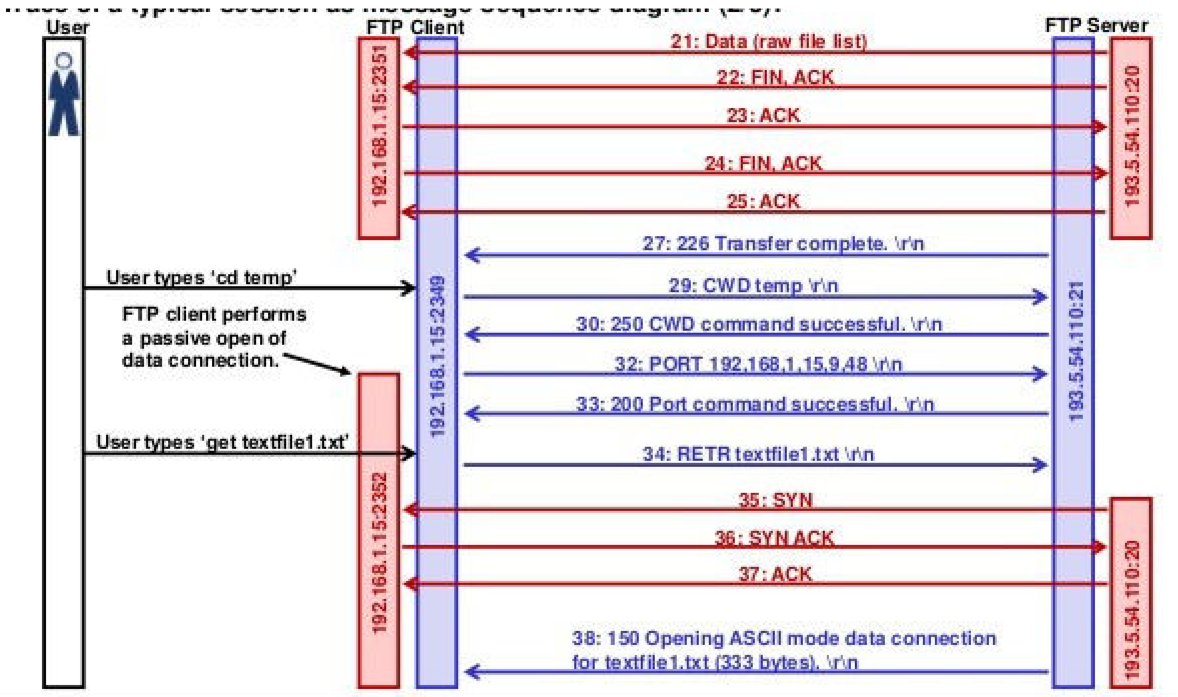
suma@suma-VirtualBox: ~/Desktop/code/665-try

```
 Dst port: 32799
   Payload (38 bytes):
000 35 33 30 20 50 6c 65 61  73 65 20 6c 6f 67 69 6e    530 Please login
016 20 77 69 74 68 20 55 53  45 52 20 61 6e 64 20 50     with USER and P
032 41 53 53 2e 0d 0a                                   ASS...
Packet Number: 10
  From: 10.2.1.1
  To: 10.2.2.1
   Protocol: TCP
 Src port: 32799
 Dst port: 21
Packet Number: 11
  From: 10.2.1.1
  To: 10.2.2.1
   Protocol: TCP
 Src port: 32799
 Dst port: 21
   Payload (13 bytes):
000 55 53 45 52 20 63 73 36  32 36 32 0d 0a             USER cs6262..
Packet Number: 12
  From: 10.2.2.1
  To: 10.2.1.1
   Protocol: TCP
 Src port: 21
 Dst port: 32799
   Payload (35 bytes):
000 33 33 31 20 50 61 73 73  77 6f 72 64 20 72 65 71    331 Password req
016 75 69 72 65 64 20 66 6f  72 20 63 73 36 32 36 32    uired for cs6262
032 2e 0d 0a                                            ...
Packet Number: 13
  From: 10.2.1.1
  To: 10.2.2.1
   Protocol: TCP
 Src port: 32799
 Dst port: 21
Packet Number: 14
  From: 10.2.1.1
```

//530 – Not logged in
//331 – User name okay, Password needed
//230 – User logged in, appropriate
//SYST - requesting information about the server's operating system



```
 Dst port: 21
   Payload (16 bytes):
000 50 41 53 53 20 52 65 3d  6d 69 33 76 45 34 0d 0a    PASS Re=mi3vE4..
Packet Number: 15
  From: 10.2.2.1
  To: 10.2.1.1
   Protocol: TCP
 Src port: 21
 Dst port: 32799
   Payload (28 bytes):
000 32 33 30 20 55 73 65 72  20 63 73 36 32 36 32 20    230 User cs6262
016 6c 6f 67 67 65 64 20 69  6e 2e 0d 0a                logged in...
Packet Number: 16
  From: 10.2.1.1
  To: 10.2.2.1
   Protocol: TCP
 Src port: 32799
 Dst port: 21
Packet Number: 17
  From: 10.2.1.1
  To: 10.2.2.1
   Protocol: TCP
 Src port: 32799
 Dst port: 21
   Payload (6 bytes):
000 53 59 53 54 0d 0a                                   SYST..
Packet Number: 18
  From: 10.2.2.1
  To: 10.2.1.1
   Protocol: TCP
 Src port: 21
 Dst port: 32799
   Payload (19 bytes):
000 32 31 35 20 55 4e 49 58  20 54 79 70 65 3a 20 4c    215 UNIX Type: L
016 38 0d 0a                                            8..
Packet Number: 19
  From: 10.2.1.1
```



Some relevant captured packets:

Packet Number: 21
Payload (8 bytes):
TYPE I..   *// TYPE I – image (binary data), Type E – EBCDIC text, Type L- Local Format*

Packet Number: 24

Payload (8 bytes):
TYPE A.. *// Type A – ASCII text*
Packet Number: 27
Payload (6 bytes):
PASV.. *//The passive FTP command involves a more secure form of data transfer initiated by the client rather than the FTP server program. Any corporate firewall recognizes input from the outside only in response to user requests, hence the PASV command. If not used, the connection from server will not be accepted when it accepts the request and sends to ephemeral port + 1.*

Packet Number: 28
Payload (45 bytes):
227 Entering Passive Mode (10,2,2,1,23,145).. *//227- Entering Passive Mode (h1,h2,h3,h4,p1,p2)*

Packet Number: 29
Payload (6 bytes):
LIST.. *//If remote-filespec refers to a directory, sends information about each file in that directory*

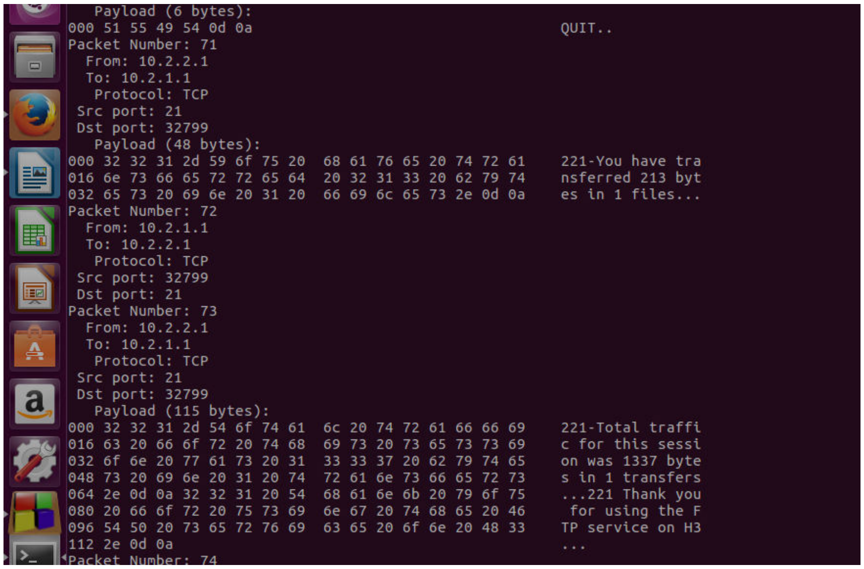
Packet Number: 30
Payload (63 bytes):
150 Opening ASCII mode data connection for directory listing…
//150 - File status okay; about to open data connection.

Packet Number: 42
Payload (10 bytes):
CWD /tmp.. *//change working directory*

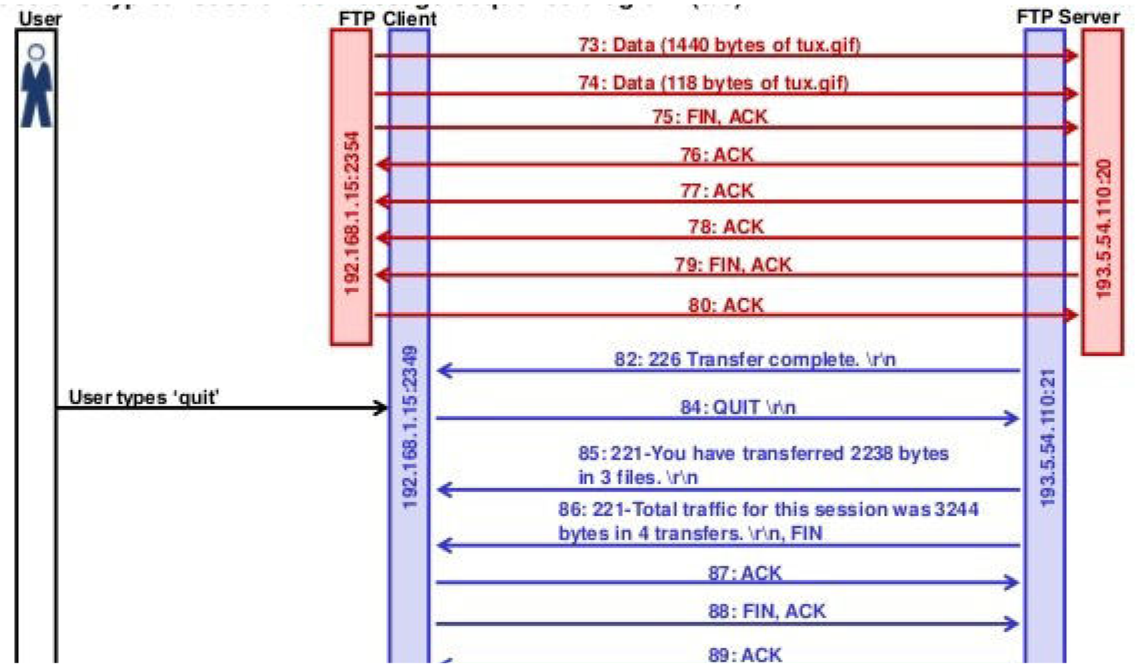
Packet Number: 66
Payload (12 bytes):
RETR hosts.. *// retrieve a remote file*

Packet Number: 67
Payload (64 bytes):
150 Opening BINARY mode data connection for hosts (213 bytes)…

//221- Service Closing Control Connection



```
   Payload (6 bytes):
000 51 55 49 54 0d 0a                                      QUIT..
Packet Number: 71
  From: 10.2.2.1
  To: 10.2.1.1
   Protocol: TCP
 Src port: 21
 Dst port: 32799
   Payload (48 bytes):
000 32 32 31 2d 59 6f 75 20  68 61 76 65 20 74 72 61    221-You have tra
016 6e 73 66 65 72 72 65 64  20 32 31 33 20 62 79 74    nsferred 213 byt
032 65 73 20 69 6e 20 31 20  66 69 6c 65 73 2e 0d 0a    es in 1 files...
Packet Number: 72
  From: 10.2.1.1
  To: 10.2.2.1
   Protocol: TCP
 Src port: 32799
 Dst port: 21
Packet Number: 73
  From: 10.2.2.1
  To: 10.2.1.1
   Protocol: TCP
 Src port: 21
 Dst port: 32799
   Payload (115 bytes):
000 32 32 31 2d 54 6f 74 61  6c 20 74 72 61 66 66 69    221-Total traffi
016 63 20 66 6f 72 20 74 68  69 73 20 73 65 73 73 69    c for this sessi
032 6f 6e 20 77 61 73 20 31  33 33 37 20 62 79 74 65    on was 1337 byte
048 73 20 69 6e 20 31 20 74  72 61 6e 73 66 65 72 73    s in 1 transfers
064 2e 0d 0a 32 32 31 20 54  68 61 6e 6b 20 79 6f 75    ...221 Thank you
080 20 66 6f 72 20 75 73 69  6e 67 20 74 68 65 20 46     for using the F
096 54 50 20 73 65 72 76 69  63 65 20 6f 6e 20 48 33    TP service on H3
112 2e 0d 0a                                            ...
Packet Number: 74
```


User
FTP Client
FTP Server
192.168.1.15:2354
193.5.54.110:20
73: Data (1440 bytes of tux.gif)
74: Data (118 bytes of tux.gif)
75: FIN, ACK
76: ACK
77: ACK
78: ACK
79: FIN, ACK
80: ACK
192.168.1.15:2349
193.5.54.110:21
82: 226 Transfer complete. \r\n
User types 'quit'
84: QUIT \r\n
85: 221-You have transferred 2238 bytes in 3 files. \r\n
86: 221-Total traffic for this session was 3244 bytes in 4 transfers. \r\n, FIN
87: ACK
88: FIN, ACK
89: ACK

Takeaways from packet level data:

I had never never known how easy it is to tap into passwords sent over applications such as TELNET and FTP (As they have plain-text authentication- login and passwords are sent in clear)using lipcap. The code was very easy to implement through minor changes, something that can be done by a novice programmer).

I realized that not only are the login credentials sent in plain-text, but also the payload. Hence, the replacement of TELNET and FTP must be done by SSH and SFTP as a necessity.

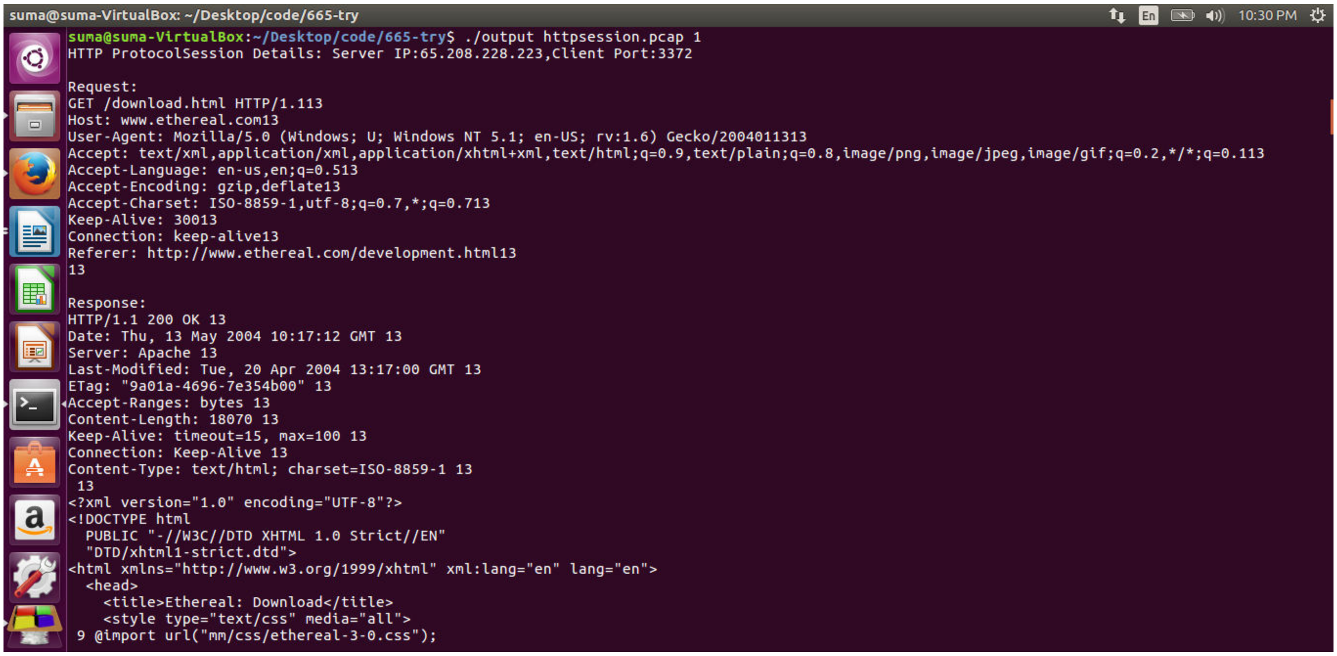
Moreover, by using the promiscuous mode we can capture the traffic not only passing through the router, but the traffic in the whole network.

This plaintext or even encrypted in terms of SSH and SFTP can be replayed for attacks, these are vulnerable even for man-in-the-middle attacks.

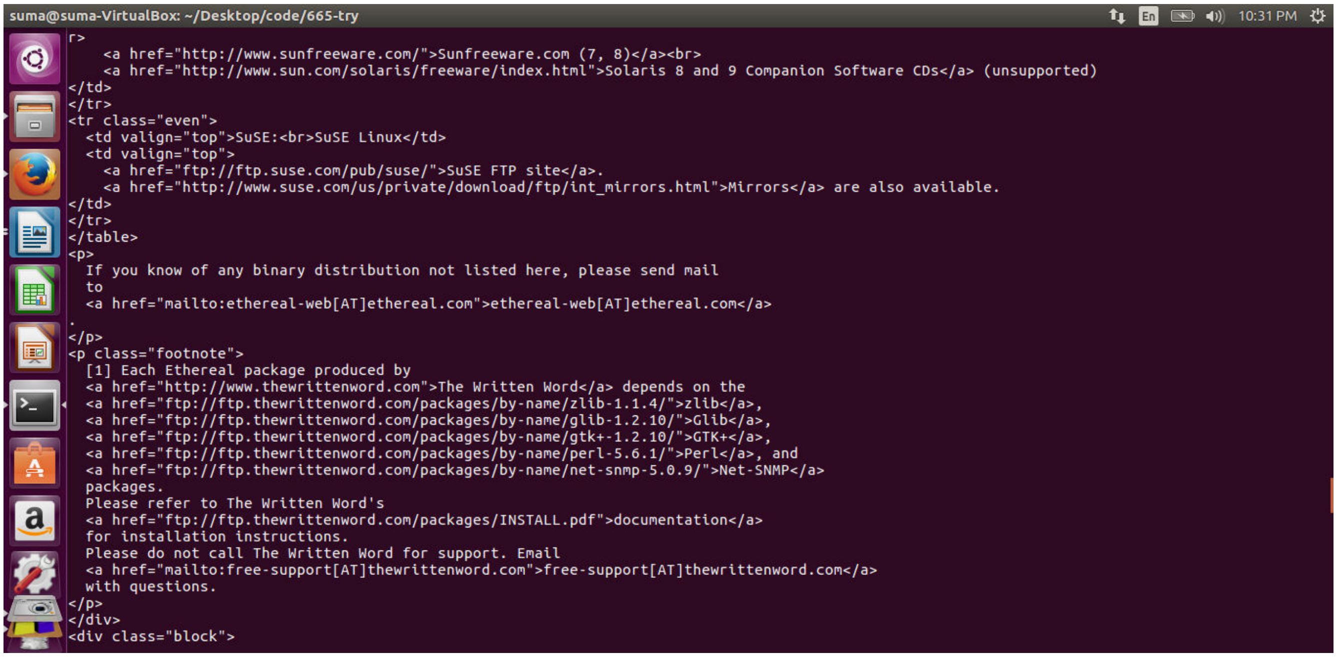
Anti- Sniffing tools must be employed at all levels (not necessarily for home or switched networks). There are many of them already in practise such as Traffscrambler, Sniff Joke, Kitty-Litter, AciD (ARP change intrusion detector).

Next Page – Application Level Screenshots

1. HTTP

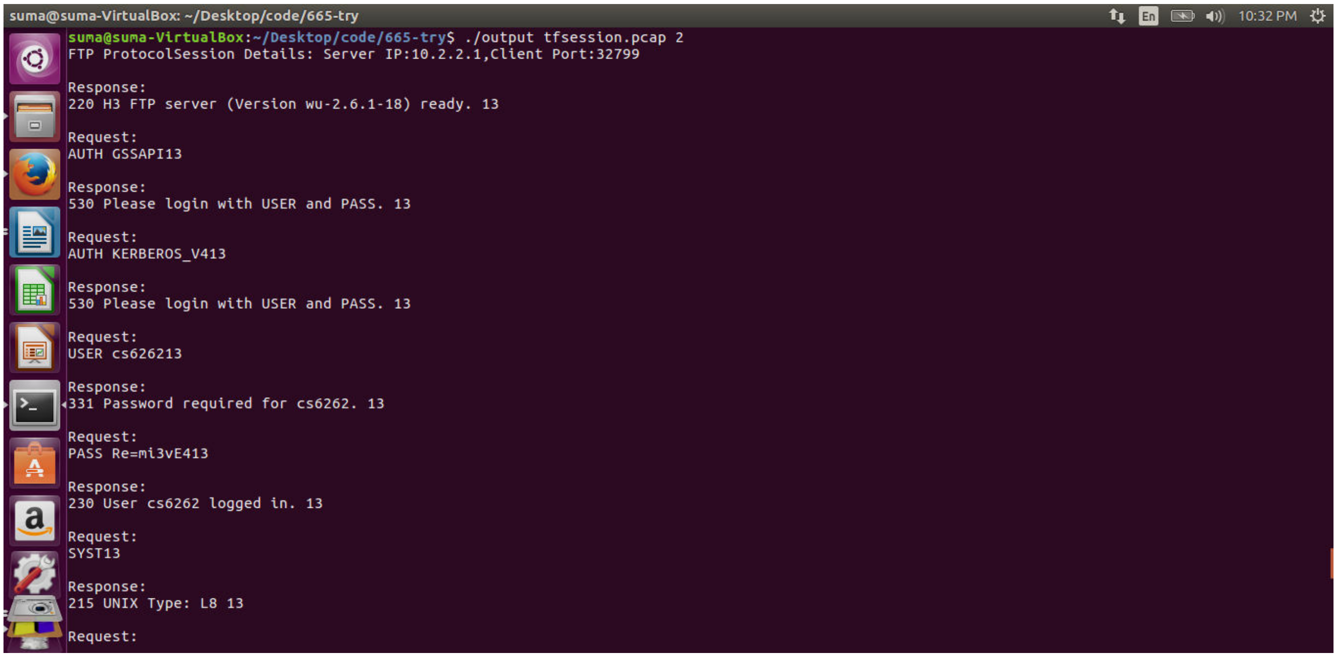


```
suma@suma-VirtualBox:~/Desktop/code/665-try$ ./output httpsession.pcap 1
HTTP ProtocolSession Details: Server IP:65.208.228.223,Client Port:3372

Request:
GET /download.html HTTP/1.113
Host: www.ethereal.com13
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.6) Gecko/2004011313
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,image/jpeg,image/gif;q=0.2,*/*;q=0.113
Accept-Language: en-us,en;q=0.513
Accept-Encoding: gzip,deflate13
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.713
Keep-Alive: 30013
Connection: keep-alive13
Referer: http://www.ethereal.com/development.html13
13

Response:
HTTP/1.1 200 OK 13
Date: Thu, 13 May 2004 10:17:12 GMT 13
Server: Apache 13
Last-Modified: Tue, 20 Apr 2004 13:17:00 GMT 13
ETag: "9a01a-4696-7e354b00" 13
Accept-Ranges: bytes 13
Content-Length: 18070 13
Keep-Alive: timeout=15, max=100 13
Connection: Keep-Alive 13
Content-Type: text/html; charset=ISO-8859-1 13
 13
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <title>Ethereal: Download</title>
    <style type="text/css" media="all">
 9 @import url("mm/css/ethereal-3-0.css");
```

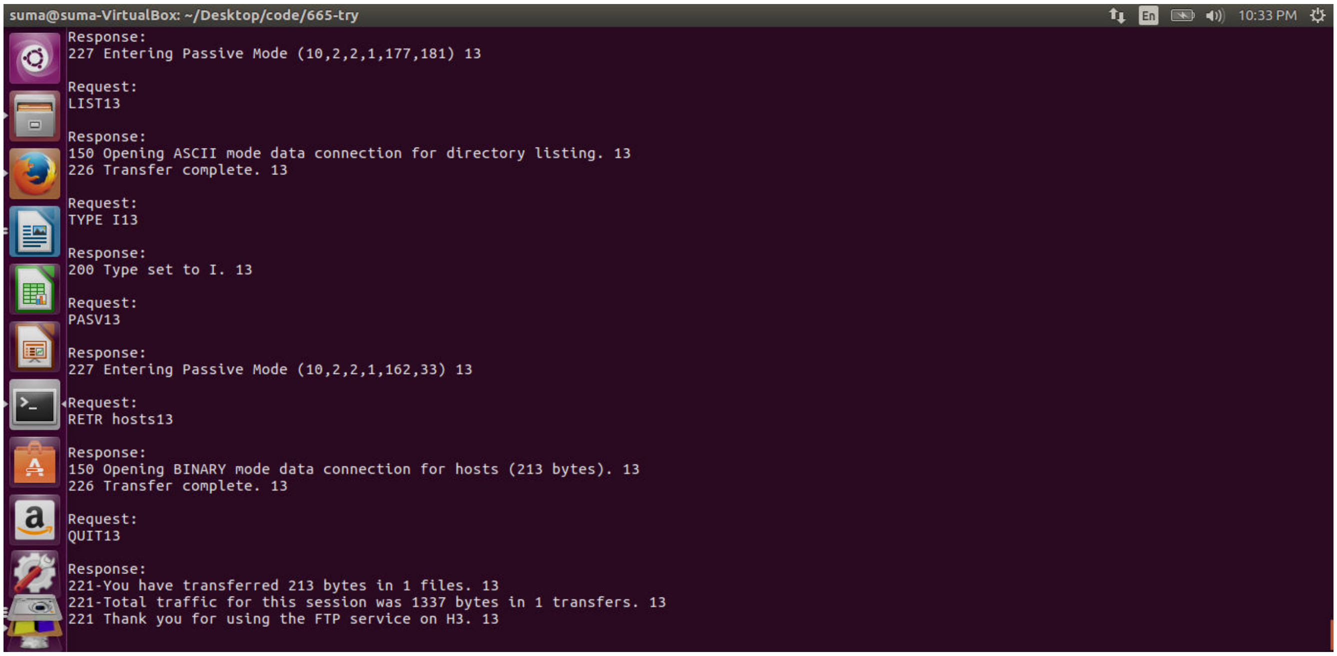


```
r>
    <a href="http://www.sunfreeware.com/">Sunfreeware.com (7, 8)</a><br>
    <a href="http://www.sun.com/solaris/freeware/index.html">Solaris 8 and 9 Companion Software CDs</a> (unsupported)
</td>
</tr>
<tr class="even">
  <td valign="top">SuSE:<br>SuSE Linux</td>
  <td valign="top">
    <a href="ftp://ftp.suse.com/pub/suse/">SuSE FTP site</a>.
    <a href="http://www.suse.com/us/private/download/ftp/int_mirrors.html">Mirrors</a> are also available.
</td>
</tr>
</table>
<p>
  If you know of any binary distribution not listed here, please send mail
  to
  <a href="mailto:ethereal-web[AT]ethereal.com">ethereal-web[AT]ethereal.com</a>
.
</p>
<p class="footnote">
  [1] Each Ethereal package produced by
  <a href="http://www.thewrittenword.com">The Written Word</a> depends on the
  <a href="ftp://ftp.thewrittenword.com/packages/by-name/zlib-1.1.4/">zlib</a>,
  <a href="ftp://ftp.thewrittenword.com/packages/by-name/glib-1.2.10/">Glib</a>,
  <a href="ftp://ftp.thewrittenword.com/packages/by-name/gtk+-1.2.10/">GTK+</a>,
  <a href="ftp://ftp.thewrittenword.com/packages/by-name/perl-5.6.1/">Perl</a>, and
  <a href="ftp://ftp.thewrittenword.com/packages/by-name/net-snmp-5.0.9/">Net-SNMP</a>
  packages.
  Please refer to The Written Word's
  <a href="ftp://ftp.thewrittenword.com/packages/INSTALL.pdf">documentation</a>
  for installation instructions.
  Please do not call The Written Word for support. Email
  <a href="mailto:free-support[AT]thewrittenword.com">free-support[AT]thewrittenword.com</a>
  with questions.
</p>
</div>
<div class="block">
```

2. FTP

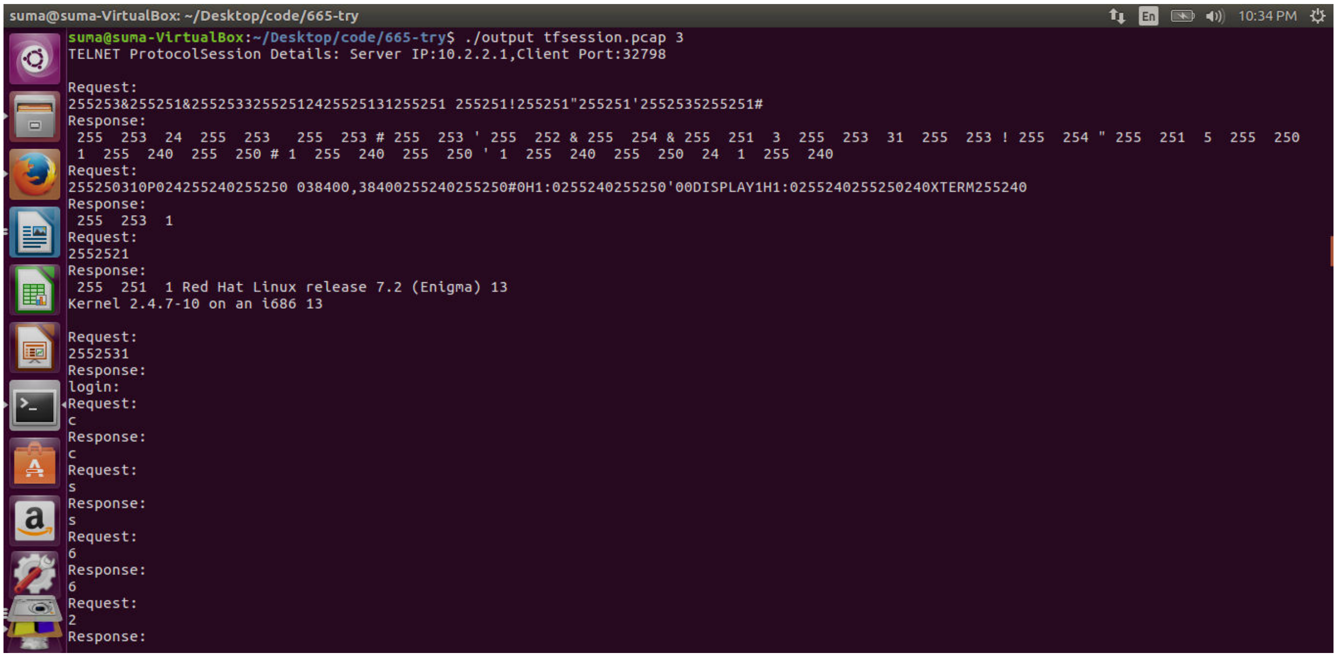


```
suma@suma-VirtualBox:~/Desktop/code/665-try$ ./output tfsession.pcap 2
FTP ProtocolSession Details: Server IP:10.2.2.1,Client Port:32799

Response:
220 H3 FTP server (Version wu-2.6.1-18) ready. 13

Request:
AUTH GSSAPI13

Response:
530 Please login with USER and PASS. 13

Request:
AUTH KERBEROS_V413

Response:
530 Please login with USER and PASS. 13

Request:
USER cs626213

Response:
331 Password required for cs6262. 13

Request:
PASS Re=mi3vE413

Response:
230 User cs6262 logged in. 13

Request:
SYST13

Response:
215 UNIX Type: L8 13

Request:
```

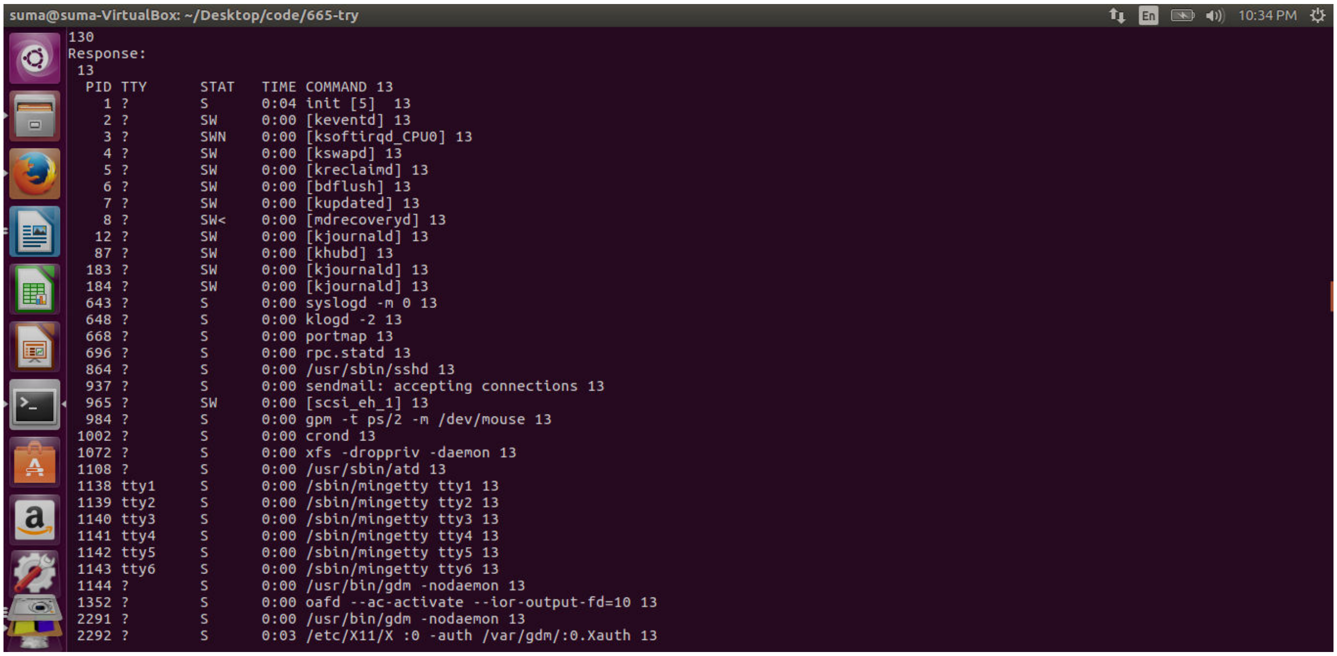


```
Response:
227 Entering Passive Mode (10,2,2,1,177,181) 13

Request:
LIST13

Response:
150 Opening ASCII mode data connection for directory listing. 13
226 Transfer complete. 13

Request:
TYPE I13

Response:
200 Type set to I. 13

Request:
PASV13

Response:
227 Entering Passive Mode (10,2,2,1,162,33) 13

Request:
RETR hosts13

Response:
150 Opening BINARY mode data connection for hosts (213 bytes). 13
226 Transfer complete. 13

Request:
QUIT13

Response:
221-You have transferred 213 bytes in 1 files. 13
221-Total traffic for this session was 1337 bytes in 1 transfers. 13
221 Thank you for using the FTP service on H3. 13
```

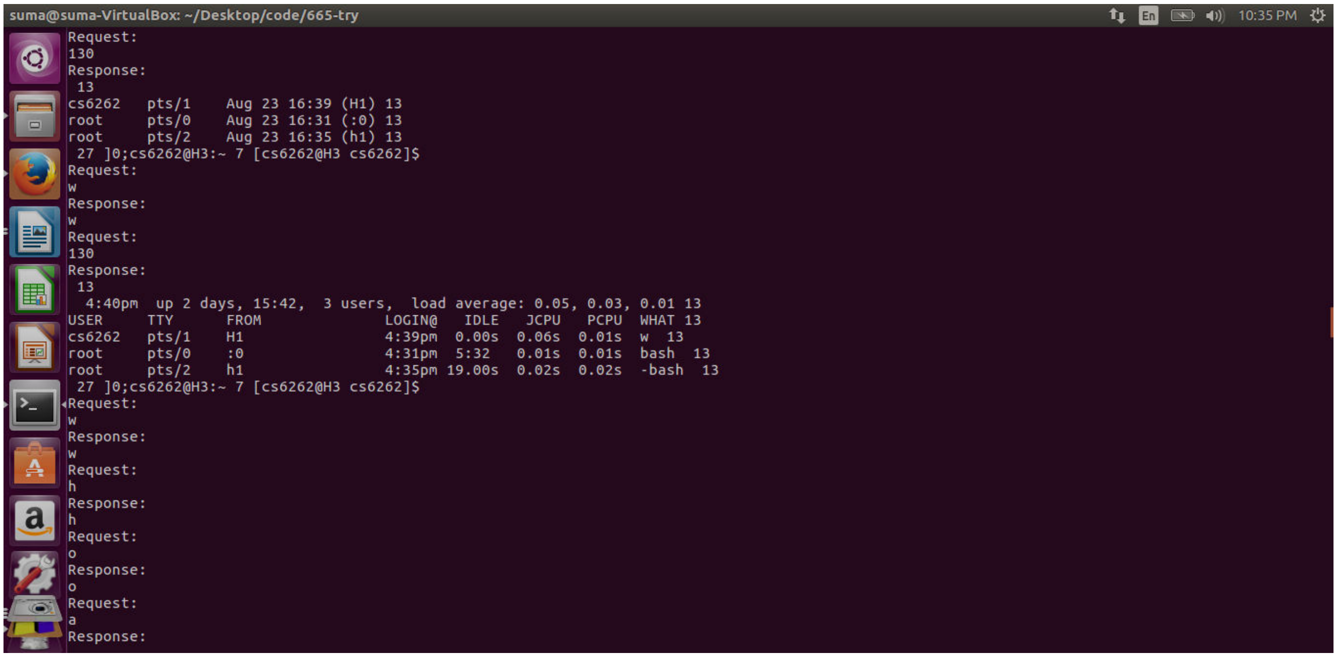
3. TELNET



```
suma@suma-VirtualBox:~/Desktop/code/665-try$ ./output tfsession.pcap 3
TELNET ProtocolSession Details: Server IP:10.2.2.1,Client Port:32798

Request:
255253&255251&25525332552512425525131255251 255251!255251"255251'2552535255251#
Response:
 255  253  24  255  253   255  253 # 255  253 ' 255  252 & 255  254 & 255  251  3  255  253  31  255  253 ! 255  254 " 255  251  5  255  250
 1  255  240  255  250 # 1  255  240  255  250 ' 1  255  240  255  250  24  1  255  240
Request:
255250310P024255240255250 038400,38400255240255250#0H1:0255240255250'00DISPLAY1H1:0255240255250240XTERM255240
Response:
 255  253  1
Request:
2552521
Response:
 255  251  1 Red Hat Linux release 7.2 (Enigma) 13
Kernel 2.4.7-10 on an i686 13

Request:
2552531
Response:
login:
Request:
c
Response:
c
Request:
s
Response:
s
Request:
6
Response:
6
Request:
2
Response:
```



```
130
Response:
 13
  PID TTY      STAT   TIME COMMAND 13
    1 ?        S      0:04 init [5]  13
    2 ?        SW     0:00 [keventd] 13
    3 ?        SWN    0:00 [ksoftirqd_CPU0] 13
    4 ?        SW     0:00 [kswapd] 13
    5 ?        SW     0:00 [kreclaimd] 13
    6 ?        SW     0:00 [bdflush] 13
    7 ?        SW     0:00 [kupdated] 13
    8 ?        SW<    0:00 [mdrecoveryd] 13
   12 ?        SW     0:00 [kjournald] 13
   87 ?        SW     0:00 [khubd] 13
  183 ?        SW     0:00 [kjournald] 13
  184 ?        SW     0:00 [kjournald] 13
  643 ?        S      0:00 syslogd -m 0 13
  648 ?        S      0:00 klogd -2 13
  668 ?        S      0:00 portmap 13
  696 ?        S      0:00 rpc.statd 13
  864 ?        S      0:00 /usr/sbin/sshd 13
  937 ?        S      0:00 sendmail: accepting connections 13
  965 ?        SW     0:00 [scsi_eh_1] 13
  984 ?        S      0:00 gpm -t ps/2 -m /dev/mouse 13
 1002 ?        S      0:00 crond 13
 1072 ?        S      0:00 xfs -droppriv -daemon 13
 1108 ?        S      0:00 /usr/sbin/atd 13
 1138 tty1     S      0:00 /sbin/mingetty tty1 13
 1139 tty2     S      0:00 /sbin/mingetty tty2 13
 1140 tty3     S      0:00 /sbin/mingetty tty3 13
 1141 tty4     S      0:00 /sbin/mingetty tty4 13
 1142 tty5     S      0:00 /sbin/mingetty tty5 13
 1143 tty6     S      0:00 /sbin/mingetty tty6 13
 1144 ?        S      0:00 /usr/bin/gdm -nodaemon 13
 1352 ?        S      0:00 oafd --ac-activate --ior-output-fd=10 13
 2291 ?        S      0:00 /usr/bin/gdm -nodaemon 13
 2292 ?        S      0:03 /etc/X11/X :0 -auth /var/gdm/:0.Xauth 13
```



```
Request:
130
Response:
 13
cs6262   pts/1    Aug 23 16:39 (H1) 13
root     pts/0    Aug 23 16:31 (:0) 13
root     pts/2    Aug 23 16:35 (h1) 13
 27 ]0;cs6262@H3:~ 7 [cs6262@H3 cs6262]$
Request:
w
Response:
w
Request:
130
Response:
 13
  4:40pm  up 2 days, 15:42,  3 users,  load average: 0.05, 0.03, 0.01 13
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU  WHAT 13
cs6262   pts/1    H1                4:39pm  0.00s  0.06s  0.01s  w  13
root     pts/0    :0                4:31pm  5:32   0.01s  0.01s  bash  13
root     pts/2    h1                4:35pm 19.00s  0.02s  0.02s  -bash  13
 27 ]0;cs6262@H3:~ 7 [cs6262@H3 cs6262]$
Request:
w
Response:
w
Request:
h
Response:
h
Request:
o
Response:
o
Request:
a
Response:
```