Computer Networks Midsem Lab

CS F303

Susmit Wani 2018A7PS0116G

Midsem Lab Exam

Vinayak Naik • Mar 13 (Edited 1:07 PM)

Midsem • 18 points **Due Tomorrow, 6:59 PM**

Notes
1. All the students will upload C code along with a PDF file containing the screenshots of the executed program on Google classroom
2. Submit a REAMDE file giving instruction for compilation and execution
3. Submit the captured pcap file
4. Write the wireshark filter, measured values, and the justification in the PDF
5. We will use MOSS to detect cheating cases

Question
Write a TCP server and a client for the following.
1. The server as a command line argument accepts the port number to which it should bind. (2 marks)
2. The client, as command line arguments, accepts the IP address and the port number at which it will find the server. (2 marks)
3. After connecting to the server, the client reads a line from the standard input and sends it to the server. (2 marks)
4. The server prints the received line in the reverser order (2 marks) and reads a line from the standard input and sends it to the client. (2 marks)
5. The client prints the received line in the reverse order and exits. The server is ready to accept a new client. (2 marks)
6. Use Wireshark to capture the packets (2 marks) and write a filter that will find what were the sizes of the TCP segment sent from the client to the server. (2 marks) Compare the number of bytes in the line with those observed in the TCP segment. Justify how the values match. (2 marks)

Output of C codes.

1. server.c

```
lenovo@susmits-lenovo:~/Desktop/CN_Midsem_Lab$ gcc -o server.out server.c
lenovo@susmits-lenovo:~/Desktop/CN_Midsem_Lab$ ./server.out 8000
Attempting to create socket on port number 8000
Socket created successfully.
Binded to the server successfully
Server listening
server has acccepted the client

olleH
Enter your message:Hello

Waiting for new client.

server has acccepted the client

niaga olleH
Enter your message:Okay then

Waiting for new client.
```

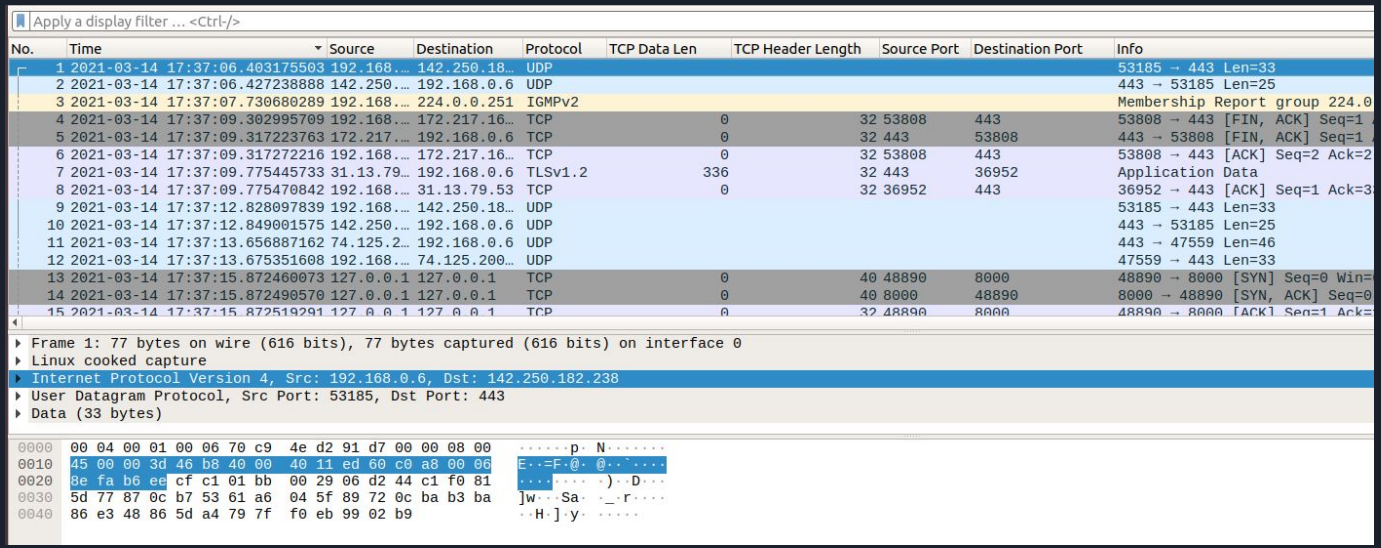
Output of C codes.
2. client.c

```
lenovo@susmits-lenovo:~/Desktop/CN_Midsem_Lab$ gcc -o client.out client.c
lenovo@susmits-lenovo:~/Desktop/CN_Midsem_Lab$ ./client.out 127.0.0.1 8000
Connecting to IP address: 127.0.0.1
Attempting to connect to port: 8000
Socket created successfully.
connected to the server successfully
Enter your message:Hello
Message from server:

olleH
lenovo@susmits-lenovo:~/Desktop/CN_Midsem_Lab$ ./client.out 127.0.0.1 8000
Connecting to IP address: 127.0.0.1
Attempting to connect to port: 8000
Socket created successfully.
connected to the server successfully
Enter your message:Hello again
Message from server:

neht yakO
lenovo@susmits-lenovo:~/Desktop/CN_Midsem_Lab$
```

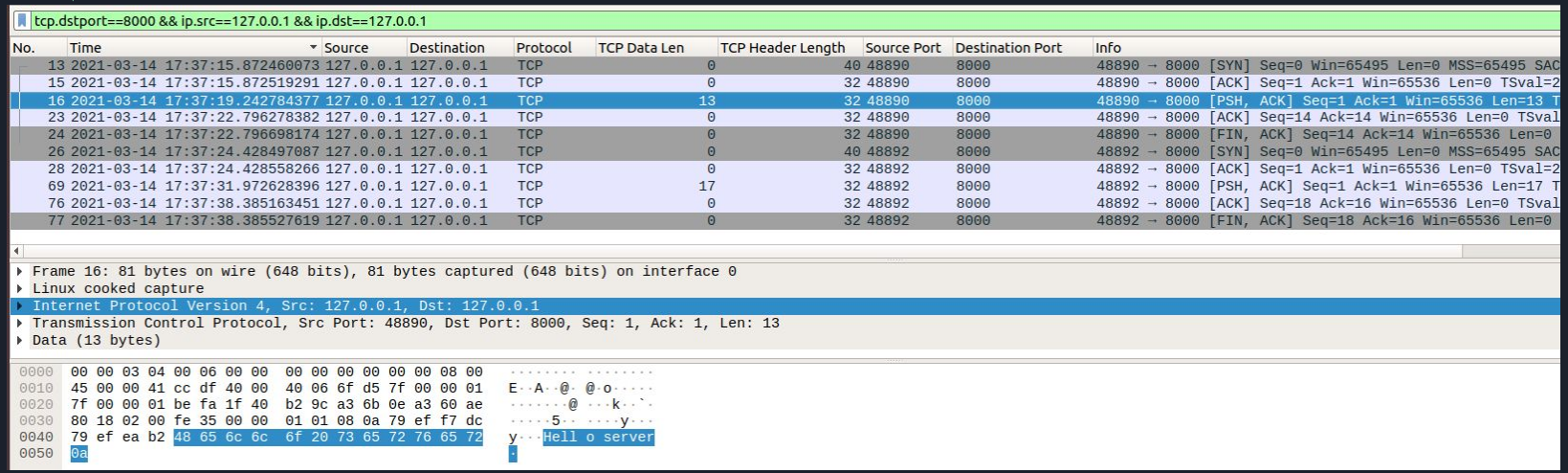
Wireshark packets from the pcapng file and the columns used



The columns TCP Dara Len and TCP Header Len are custom added columns.
TCP Data Len: tcp.len
TCP Header Length: tcp.hdr_len

Filter to be used to display packets we want:

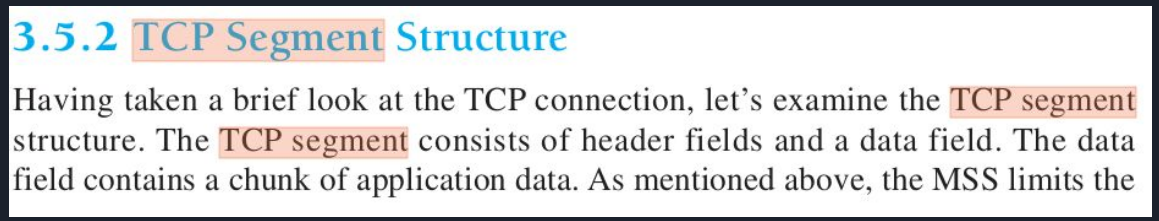
tcp.dstport==8000 && ip.src==127.0.0.1 && ip.dst==127.0.0.1



tcp.dstport allows us to get packets which were sent to port 8000. This along with the ip.dst and ip.src allow us to capture exactly those packets which we want, i.e. those packets which were sent from client to server.

The file in the submission contains only these tcp dumps. The previous slide is only an example of the pcap output.

Comparison between bytes in line and TCP segment



3.5.2 TCP Segment Structure

Having taken a brief look at the TCP connection, let's examine the TCP segment structure. The TCP segment consists of header fields and a data field. The data field contains a chunk of application data. As mentioned above, the MSS limits the

We can know that the TCP segment is made up of header+data. So as we can see in the TCP header length column, the length is 32 bytes for ACK message and 40 for SYN message. This is because in SYN we need to synchronise the sequence numbers, hence the extra size.

Now if we check the data messages, the first message is “Hello server\n” which is 13 bytes long. The second data message is 17 bytes long and it says “this is client 2\n”.

The TCP data length column denotes the length of data sent via the TCP message. It is 0 in case of ACK and SYN and 13 and 17 where actual data is being sent.

*Thus we can clearly see that the **TCP segment size is equal to 32 plus the data length** sent which is nothing but bytes in line sent.*