**\*\*\*\*\*Use file Capture 1 for questions 1-2\*\*\*\***

1. Apply a capture filter to obtain all the packets that are either coming from a hardware address of 00:06:25:da:af:73 from TCP Port 80 or 00:08:74:4f:36:23 from TCP Port 4247. What is the filter you write? [0.5]

Ans: (eth.src == 00:06:25:da:af:73 && tcp.srcport == 80) || (eth.src == 00:08:74:4f:36:23 && tcp.srcport == 4247)

1. Suppose that we assign Number 1 to the first packet obtained after applying filter with each successive packet number increasing by one (2nd Filtered Packet will be called Number 2 and so on). Also suppose that someone replaced the recent data present on the server (gaia.cs.umass.edu) with another bogus data having 250 Bytes on 9 Nov, 2018. What would have been **the sequence number and acknowledgement number** in Packet Number 10 if we would have applied the same HTTP get request as present in packet Number 8? Explain in detail the calculated values.[2.5]

Sequence Number: 1116 (Remain same because both get requests are same)

Acknowledgement Number: 1+ 685(first GET Response) + 314 (200 OK Header length) + 250 (Data was changed to 250 Bytes) = 1250

**\*\*\*\*\*Use file Capture 2 for questions 3-7\*\*\*\*\*\***

1. Apply a capture filter to obtain all the packets that are either going from client IP: 192.168.1.2 to server TCP Port:21 or from server IP:195.89.6.167 and server TCP Port: 20 or 21. Write the filter which you have applied? [0.5]

Ans: (ip.src == 192.168.1.2 && tcp.dstport==21) || (ip.src==195.89.6.167 && (tcp.srcport==21 || tcp.srcport==20))

1. In the filtered packets, client asks for data from server over FTP connection for two times and server sends the data at the port specified by the client. Now suppose that at the start of FTP connection, the first command was issued using Port No: 16979 and IP: 178.128.12.54. Now, if the client wants to get the data third time, what will be the specific command for such a request made by the client to the server? [1.5]

PORT: 178,128,12,54,66,86 (66=0x42 and 86=0x56 ->0x4256=16982)

1. In regard to the request made by client to server in above question what will be Destination Port and Destination IP in the response generated by the server? [1]

Destination Port:16982, Destination IP:178.128.12.54

1. Apply a capture filter to get all the packets which are directed from client with MAC Address:60:67:20:55:7b:ac to the Server UDP Port: 53. What filter did you apply? [0.5]

eth.src ==60:67:20:55:7b:ac && udp.dstport==53

1. If we take the first filtered packet as Packet No. 1 with each successive packet increasing one in number, then what will be the acknowledgement number of the DNS request made in Packet No. 11? [0.5]

No acknowledgement in UDP

**\*\*\*\*\*Use file Capture 3 for questions 8-10\*\*\*\*\*\***

1. Apply a filter to obtain all those packets which are either going from client with IP:192.168.1.102 to server tcp port: 80 or from server with IP:128.119.245.21 to client tcp port: 1161. What is the filter which you applied? [0.5]

(ip.src ==192.168.1.102 && tcp.dstport==80) || (ip.src == 128.119.245.12 && tcp.dstport==1161)

1. If we analyze all the filtered packets, then we come to know that sequence number in all the packets since the beginning remains at one while acknowledgement number keep on increasing. Why is this so? [0.5]

The sequence remains at one because of HTTP Post in which only client is uploading some data onto server and server is basically sending the acknowledgement back to client due to which Seq=1(As server is ending no data-only receiving) and Ack keeps on increasing because client keeps on sending the data

1. If we look at the last packet in the filtered packets then Seq=164091 and Ack=731. How come so that the Seq=1 has abruptly converted into such big value even when the total packet length (including headers) of last packet is only 784 bytes? Why the Acknowledgement Number has decreased so much only with the transmission of one packet? Explain in detail? [2]

The last packet is the acknowledgment packet from client to server after server accepts all the data and send the response that it has received all the data from client. So Seq=164091 because client has sent this much data to the server and Ack=731 because it is the length of the only response sent by server to the client after getting all the data from client using HTTP Post