15CS10038 15CS10055

for HTTP part

steps :

* browzer cache is disabled and network proxy is bypassed to access the given pages

* The pages are analyzed using wireshark.

* appropriate filter is applied . (ip.addr = 10.5.20.222 and ip.addr = 10.105.65.40)

* we observe the various http get messages and their response

* we can get the http version(1.0 or 1.1) by analyzing the packet information . Similarly we can get the persistent or non-persistent for case 0f 1.1 by analyzing the network packet information .

* we can record the time for every get request and correspondng response from packet details of each response which has a field time since
request.

* difference b/w times of last response and 1st get message gives the page download time. We can note both the times from the Time coulmn
in wireshark captured packets .

* To make job easy we can sort on basis of protocols so that all HTTP come together in the sequence.

* we can check the http header to get the OS and browser information .

justification :

* To differentiate b/w 1.1 persistent and non-persistent connections , we can count the no. of syn requests . In case of non-persistent
connections , syn request is more as it has to establish the connection every time which is not so in case of persistent connections.

* Also persistent connection is keep-alive which we can get from HTTP header in the packet information .

* we note that page downloading time is less in case of persistent connection as it involves fewer tcp connections so reduced network congestion thus less downloading time compared to non-persistent connections .

* HTTP version is determined from the packet details as stated earlier . Also 1.1 is better than 1.0 .

* we can intutively say that no of HTTP get requests will be same as same page is accessed .

for FTP part :

steps :

* we start wireshark for capturing by applying the filter (ip.addr = 10.5.20.222 and ip.addr = 10.105.65.40)

* first the command ftp -d is run on the terminal.

* then open 10.5.20.222 asks for username and password .

* after successful login into the server

* we type passive if we want to use passive ftp mode.

* now we can use different ftp commands here. for this experiment we use ls (this shows everything in the current directory of the server)

* we can filter out the ftp messages in different streams by right clicking on a packet -> follow -> tcp stream this gives the tcp stream of this packet.

* we get client and server ip address and the ports for all ftp messages directly from packet information.

justification:

* Active mode:Client informs the port number where it is listening, and the server initiates the TCP connection to that port (TCP server is running at the client side).

* Passive mode:The server selects a random port, and the client initiates a TCP connection to that server port.

* for command channel in both active and passive , the client initiates a TCP connection from some port no. to port 21(always) of the server.

* for data channel connection , a new port no. is used on the client side and port 20 on server side and the server initiates the TCP connection in case of active mode .

* for data channel connection , a new port no. is used on the client side and a new port is assigned in the server side and client initiates the TCP connection in case of passive mode.

* note: IN 1 TCP CONNECTION OF DATA CHANNEL ONLY 1 FILE IS TRANSFERRED AND A NEW DATA CHANNEL CONNECTION NEEDS TO BE ESTABLISHED FOR FURTHER TRANSFER.