**TASK 1: Capture a HTTP Request Sequence**

Start wire shark. Start capturing on wireshark and view a webpage on a web browser. Stop capturing when you have a sufficient number of packets. Take **screenshots** where you **clearly annotate** a listing of the packets captured, identifying:

1. DNS request for the IP address that corresponds to the Uniform Resource Locator (URL)

Type dns

Click on one

Domain name system

Response time

screenshot

1. DNS response returned with the IP address clearly seen

Same but screenshot of the IP address

1. HTTP – TCP starting the 3-way handshake (SYN bit =1) – note sequence numbers

Same IP address as the one on B

TCP

Transmission Control Protocol

Flags

Sync: one

1. HTTP – TCP response from server (port 80) (SYN bit =1, ACK bit = 1)

Screenshot of the port 80 of the same IP address

1. HTTP – TCP acknowledgement to the server (ACK bit =1) – look at seq numbers again

ACK

1. now browser starts to request web pages

HYPERTEXT TRANSFER PROTOCOL highlight

Screenshot with the requested domain name

1. HTTP GET - which the request to retrieve the web page

HTTP GET too

1. HTTP returned with data – this will be the header for the web page

Checksum 0x217

Webpage header

1. HTTP data packets each with Hyper Text Mark Up Language (HTML) code in it – the data to display the Web page

**TASK 2: Capture a HTTP Request Sequence**

Log into Comms - you may need to reboot the machine.

Select the 'Reset Network Settings' program icon on the desktop, to make sure the machine has an IP address starting with 10.0.\*.\*

Start Wireshark and begin packet sniffing by selecting the network card at the top of the list.

Make sure you do not have the wireless card selected.

Click start capturing.

Type pidgin in the Start menu to open the chat client.

You should see other users in the list.

Go to Show Buddies - This is automatically populated as more people come on line. To join a chat, Tools > Room List and then click Get List. Chat with another person in the room.

Log off from pidgin and stop the Wireshark capture to investigate the results.

List the different protocols captured

What is the protocol that delivers the chat packets? Double click on any packet to open it in a new window to see the packet structure. *Example - Ethernet frame > IP datagram > TCP segment > HTTP* Find a packet belonging to the Chat protocol.

c) What ports are involved in the Chat messaging?

d) What else can you find out about the chat protocol? Apply a filter on Wire shark to capture only jabber packets. Select xmpp and click Apply in the filter pane.

e) Why is the source IP address always the same, regardless of which Buddy you are chatting with?

f) Open one of your captured packets

**TASK 3: Capture an FTP Session** *(Stay logged in Comms for this task too)* Start Wire Shark and Set the Wire Shark filter to FTP-data and click the Apply button. Start capturing. Start a command prompt. Type **ftp ubuntu-server**. You will be asked for an ID and password. These are the same as your PC’s ID.

Use **ls -a** to list all the files present. Identify the hidden file. The file has been named with a single space character. While running wireshark in capture mode, copy the hidden file across to your PC using ftp commands.Type **get '' "** Stop capturing.

Can you see the login/password details from the captured data?

Use the *ftp-data* filter on wireshark.

What message did you receive?

Change the filter to look at only *ftp* capture.

c) What was the date and time of your message?

d) What was the IP address of your machine (from the packet capture only)?

Change your filter to look at both *ftp* and *ftp-data: ftp || ftp-data*

Don’t forget to Apply the filter.

Set the capture going again.

e) How has this changed what is captured?

<http://staffweb.cms.gre.ac.uk/~sp02/PacketSniffing/MarkingScheme.html>