WEEK 1.10

PACKET SNIFFING

Wireshark is a free and open-source packet analyser, used for network troubleshooting, analysis, software and communications protocol development and security. It captures network packets and displays detailed packet information.

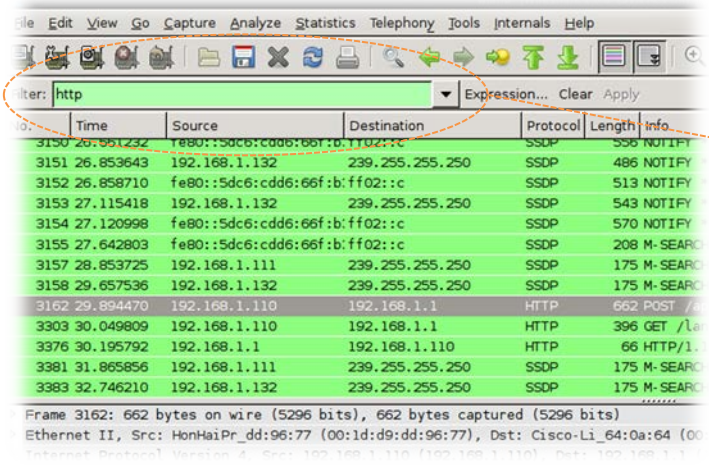
You have 90 minutes to complete all tasks. In the last 30 minutes you will present your work and be marked. If you fail to complete the laboratory within the allocated time, you will receive a mark of zero. There is no opportunity to retake the laboratory.

Form **groups of four**

It may be helpful to divide tasks, so as to finish in time

Start a document where you will be recording your answers for each task

Watch the Introduction to Wireshark video on the Wireshark web site, and then watch Mike Lehrter's - video Introduction to Wireshark - Part1 or Google and watch any other introduction to Wireshark video of your choice.



Make sure you understand how to capture packets and filter the results

How is a webpage retrieved from a server?

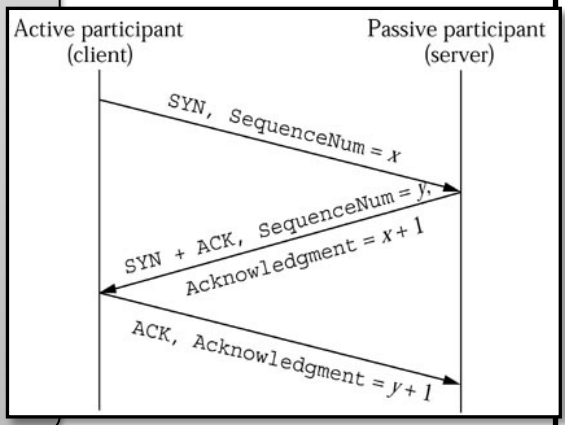
When a web browser fetches a file from a web server, it does so using Hypertext Transfer Protocol (**HTTP**), with which a computer sends a request for some file, and the web server sends back a response, followed by the file itself. There are three stages to retrieving a web page:

❑ Establish the IP address of a web server with the Domain Name System (DNS).

❑ Before a client attempts to connect with a server, the server must first bind to a port to open it up for connections: this is called a passive open. Once the passive open is established, a client may initiate an active open. To establish a connection, the three-way (or 3-step) handshake occurs:

- The client sends a synchronization (SYN flag set) packet to initiate a connection. The SYN packet holds a Sequence Number (a 32-bit field in the TCP header). Let the Sequence Number be x.
- The server receives the packet, records x from the client, and replies with an acknowledgment and synchronization (SYN-ACK) with the sequence number that this host is expecting to receive (x + 1). The server also initiates a return session. This includes a TCP segment with its own initial Sequence Number value of y.
- Finally the client sends an ACK back to the server with an acknowledgment number value of y + 1

DNS is a hierarchical naming system built on a distributed database for resources connected to the Internet or a private network. It translates domain names into numerical identifiers for locating and addressing these devices worldwide.



TASK 1: Capture a HTTP Request Sequence

Start wireshark. 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:

a) DNS request for the IP address that corresponds to the Uniform Resource Locator (URL)
b) DNS response returned with the IP address clearly seen
c) HTTP – TCP starting the 3-way handshake (SYN bit =1) – note sequence numbers
d) HTTP – TCP response from server (port 80) (SYN bit =1, ACK bit = 1)
e) HTTP – TCP acknowledgement to the server (ACK bit =1) – look at seq numbers again now browser starts to request web pages
f) HTTP GET - which the request to retrieve the web page
g) HTTP returned with data – this will be the header for the web page
h) HTTP data packets each with Hyper Text Mark Up Language (HTML) code in it – the data to display the Web page

Pidgin - http://www.pidgin.im/ is an open-source multi-platform instant messaging client which enables log in to accounts on multiple chat networks simultaneously. For example, it is possible to chat to individuals on MSN, while talking to a friend on Google, and sitting in a Yahoo chat room all at the same time. Pidgin runs on Windows, Linux, and other UNIX operating systems, using the Jabber Protocol, also known as the Extensible Messaging and Presence Protocol (XMPP), which is an open-standard communications protocol for message-oriented middleware based on XML (Extensible Mark-up Language) for near-real-time, extensible Instant Messaging (IM), presence information, and contact list maintenance. The protocol also finds application in VoIP and file transfer signalling.

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.

a) List the different protocols captured
b) 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 by double clicking on it and open the jabber part of the packet. What is the chat message?

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.

The password will not show as you type. This is normal.

Look at the top of your monitor for the ID

Use **ls -a** to list all the files present. Identify the hidden file.

In Unix operating systems a dot (.) before a file indicates that it is a hidden file.

The file has been named with a single space character.

This technique is often used to hide virus files on machines, as at first glance the file appears not to be listed.

While running wireshark in capture mode, copy the hidden file across to your PC using ftp commands.Type

get " "

Stop capturing.

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

b) 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?

After the first 90 minutes, attach the marking scheme on the front of your document, print it and hand it in to the tutor, who will then test you and mark you during the last 30 minutes of the laboratory. ONE printout to be handed in per team. In addition, ***ALL*** *team members need to individually upload their documents into* ***week 1.10 (in PDF format) by the end of this laboratory****. Make sure that the marking scheme has the names of all the members of your team.*