COMPUTER NETWORKS (CS F303)
LAB-SHEET – 2
Topic: Learning FTP and DNS using Wireshark

Objectives:

- To learn the working of FTP protocol using Wireshark
- To understand use and working of DNS protocol.

This lab is divided into three parts.
In part one, we will continue doing the experiments with Wireshark to analyze and understand various application layers' protocols. In this lab our focus would be on another very important application layer protocol called **FTP** (File Transfer Protocol).

In the second part of the lab, we will try to understand the working principles of domain name system with the help of Wireshark capture and useful networking command called "**nslookup**".

Part 1: Understanding the functioning of File Transfer Protocol (FTP) using Wireshark

FTP: File Transfer Protocol
FTP is a commonly used application over the Internet for file transfer. The file transfer provided by FTP copies a complete file from one system to another system. To use FTP, we need an account to login on the FTP server.

Figure 1, shows the arrangement of the client and server and the two connections between them.

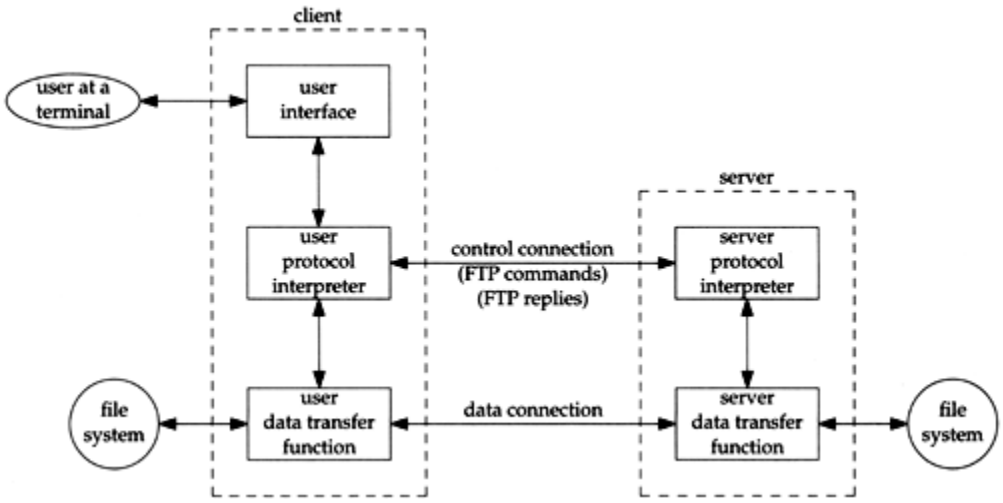

Fig. 1

FTP Commands

The commands and replies sent across the control connection between the client and server are in NVT ASCII (7 bit ASCII). This requires a CR (carriage return), LF (line feed) pair at the end of each line (i.e., each command or each reply). The commands are 3 or 4 bytes of uppercase ASCII characters, some with optional arguments. More than 30 different FTP commands can be sent by the client to the server. Table below shows some of the commonly used commands.

Command	Description
ABOR	abort previous FTP command and any data transfer
LIST *filelist*	list files or directories
PASS *password*	password on server
PORT *n1,n2,n3,n4,n5,n6*	client IP address *(nl.n2.n3.n4)* and port (*n5* x 256 + *n6*)
QUIT	logoff from server
RETR *filename*	retrieve (get) a file
STOP *filename*	store (put) a file

You will find that sometimes there is a one-to-one correspondence between what the user types and the FTP command sent across the control connection, but for some operations a single user command (**what you type in the terminal**) results in multiple FTP commands across the control connection.

The FTP server sends file listings back across the data connection, rather than as multiline replies across the control connection. Unlike control connection which stays up for the duration of the client-server connection, the data connection can come and go, as required.

The usual procedure is as follows:

1. The creation of the data connection is under control of the client, because it's the client that issues the command that requires the data connection (get a file, put a file, or list a directory).
2. The client normally chooses an **ephemeral port number** on the client host for its end of the data connection. The client issues a passive open from this port or starts listening on this port
3. The client sends this port number to the server across the control connection using the **PORT command**.
4. The server receives the port number on the control connection, and issues an active open (try to connect) to that port on the client host. The server's end of the data connection always uses port 20.

Lab Exercise-1: An Interactive FTP Session

1. To start an FTP interactive session type "ftp" from a DOS Command window.

C:\> ftp

The DOS prompt should be replaced with the FTP prompt. The FTP program is now running on the local system. A connection (or session) to a remote system has not been established.

2. The help command or ? (question mark) may be executed without being attached to a remote system and will do a print (usually to the screen) of the FTP commands. The following is an example of an FTP Command to display the FTP Help information.

ftp> help

3. The following FTP Command will perform the FTP OPEN (make the connection) and display the following messages.

ftp> open **ftp.bits-pilani.ac.in**
Username: **csis**, Password: **csis**

4. The following FTP Command will copy a file from the local system to the remote system and display the information.

ftp> put C:\Users\Dell\Documents\readme.txt (This file could be anything from your local computer)

ftp> quit

Lab Exercise-2: Understanding FTP protocol details using Wireshark

Identify TCP Header Fields and Operation Using a Wireshark FTP Session Capture

Step 1: Start a Wireshark capture.

a) Close all unnecessary network traffic, such as the web browser, to limit the amount of traffic during the Wireshark capture.
b) Start the Wireshark capture.

Step 2: Log on to the FTP server

a) From the command prompt, enter **ftp ftp.bits-pilani.ac.in**
b) Enter Username: **csis**
c) Enter Password: **csis**

Step 3: Locate and download the *readme.txt* file from the server.

Step 4: ftp quit

Step 5: Stop the Wireshark capture.

Step 6: View the Wireshark Main Window.

Wireshark would have captured many packets during the FTP session. To limit the amount of data for analysis, type **tcp and ip.addr = = 172.22.1.29** in the Filter: entry area and click Apply. The IP address, **172.22.1.29**, is the address for our FTP server.

Analyze the TCP header fields:

After the TCP filter has been applied, the first three frames in the packet list pane (top section) displays the transport layer protocol TCP creating a reliable session. The sequence of [SYN], [SYN, ACK], and [ACK] illustrates the three-way handshake. (Let us not get into details of SYN, SYN, ACK…., we will learn about them in due course of time!!!)

TCP is routinely used during a session to control datagram delivery, verify datagram arrival, and manage window size. For each data exchange between the FTP client and FTP server, a new TCP session is started. At the conclusion of the data transfer, the TCP session is closed. Finally, when the FTP session is finished, TCP performs an orderly shutdown and termination.

In Wireshark, detailed TCP information is available in the packet details pane (middle section).

Highlight the first TCP datagram from the host computer, and expand the record.

Lab Exercise-3: Understand the working of FTP

Answer the following questions.

1. Did you see an OPTS request going from client to server? What is the purpose of this command?
2. Locate the PORT command in the Wireshark trace and try to correlate it with the command entered by you in the ftp terminal. Analyze the parameters of PORT command.
3. How the server is able to send PORT command successful, even when the three-way handshake required to complete the TCP connection that got initiated due to PORT command has not been completed.
4. The FTP server sends file listings back across the data connection, rather than as multiline replies across the control connection. Try to verify this fact by listing the remote directory using “dir” command and analyzing the Wireshark packet capture.

Part 2: Understanding the functioning of Domain Name System DNS Protocol using Wireshark and *nslookup* command.

DOMAIN NAME SERVICE (DNS)

The Domain Name System (DNS) is a hierarchical decentralized naming system for computers, services, or other resources connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities. Most prominently, it translates more readily memorized domain names to the numerical IP addresses needed for locating and identifying computer services and devices with the underlying network protocols. By providing a worldwide, distributed directory service, the Domain Name System is an essential component of the functionality on the Internet that has been in use since 1985.

The Domain Name System also specifies the technical functionality of the database service that is at its core. It defines the DNS protocol, a detailed specification of the data structures and data communication exchanges used in the DNS, as part of the Internet Protocol Suite. Historically, other directory services preceding DNS were not scalable to large or global directories as they were originally based on text files, prominently the hosts file.A DNS name server is a server that stores the DNS records for a domain; a DNS name server responds with answers to queries against its database.

The most common types of records stored in the DNS database are for Start of Authority (SOA), IP addresses (A and AAAA), SMTP mail exchangers (MX), name servers (NS), pointers for reverse DNS lookups (PTR), and domain name aliases (CNAME).

In this experiment you will sniff DNS packets.

Experiment 4

1. Clear Browsing history.
2. Start Wireshark and type filter name “dns” to see DNS packets.
3. Start browser and type URL:
   https://dst.gov.in
4. Understand that the first task to be done by your system is to get IP address corresponding to the server. See if there is any DNS packet shown in Wireshark to resolve server hostname. Chances are that no DNS packet is shown in Wirehasrk. WHY?? *Think before reading ahead.*
5. Probably you must have thought that DNS resource records are cached locally and clearing browsing history does not clear DNS records. *You were right!* So, now follow the following steps.
6. Open command prompt (or terminal) and type

   C:\ ipconfig/displaydns

   You will see all the cached DNS resource records. Now you need to clear all this. So type

   C:\ ipconfig/flushdns
7. Now, repeat steps 2 to 4 above and observe DNS packets exchanged. Answer the following questions with respect to DNS query message:

   Q1. Which transport layer protocol is used?

   Q2. What is the transaction ID of the message?

Q3. What is the query section? What is the type of query and what does it signify about the type of query? Is there any other type of query you can recall?
Q4. What is the IP address of DNS server?
Q5. What is the destination port number? Remember that it is fixed for all DNS messages.

8. See the DNS response message and answer the following:
   Q1. What is the transport layer protocol?
   Q2. What is the transaction ID of this message? Is it same as that of corresponding DNS query message?
   Q3. Along with the answer, is there any authoritative name server also? What is its type and what does it signifies?
   Q4. Is there any other field (for example additional records, etc.)? What it signifies?
9. Verify that when the webpage in step 3 above was fetched, DNS query and response was generated first then HTTP query and response.

To understand the process of IP address resolution in DNS with the help of Wireshark lets perform the following experiment.

Lab Exercise-4:

Step 1: Start a Wireshark capture.

a) Close all unnecessary network traffic, such as the web browser, to limit the amount traffic during the Wireshark capture.
b) Start the Wireshark capture.

Step 2: Run the following command from the command prompt

nslookup www.meity.gov.in

Step 3: Stop the Wireshark capture.

Step 4: View the Wireshark Main Window.

Answer the following questions by observing traffic captured using Wireshark:

1. Locate the DNS query and response messages. Are they sent over UDP or TCP?
2. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?
3. What is the destination port for the DNS query message?
4. What is the source port of DNS response message?
5. Examine the DNS response message. How many "answers" are provided? What do each of these answers contain?

Repeat the same experiment by running the following command at step 2
nslookup -type=NS www.meity.gov.in and try to answer the same questions on page 6.

Lab Exercise-5:

Run the following commands from the command prompt. We have provided here the corresponding output for each command. The output in your terminal may not be exactly same as the output provided here.

Understand and describe the meaning/purpose of the commands and their corresponding output. (e.g., Different DNS records, i.e. A, NS, MX, CNAME and their purpose.)

Command	Output
C:\Users\Dell>nslookup google.com	Server: Unknown Address: 172.24.2.76 Non-authoritative answer: Name: google.com Addresses: 2404:6800:4009:807::200e 142.250.192.46
C:\Users\Dell>nslookup -type=NS google.com	Server: Unknown Address: 172.24.2.76 Non-authoritative answer: google.com nameserver = ns4.google.com google.com nameserver = ns3.google.com google.com nameserver = ns2.google.com google.com nameserver = ns1.google.com
C:\Users\Dell>nslookup google.com ns2.google.com	Server: ns2.google.com Address: 216.239.34.10 Name: google.com Addresses: 2404:6800:4009:807::200e 142.251.42.46
C:\Users\Dell>nslookup 8.8.8.8	Server: Unknown Address: 172.24.2.76 Name: dns.google Address: 8.8.8.8
C:\Users\Dell>nslookup bits-pilani.ac.in dns.google	Server: dns.google Address: 8.8.8.8 Non-authoritative answer: Name: bits-pilani.ac.in Address: 103.144.92.33 14.139.243.20

*****