**SOLUTION 20L-0921 Aisha Muhammad Nawaz**

**BSCS SECTION 5E1**

**Lab Manual 07**

**Step 5: Analyze the packets**

Carefully analyze the packets in Wireshark windows and answer the following question:

**Use the FTP\_Session.pcapng (Wireshark Capture File) to answer the questions below**

1. FTP uses two port numbers: 20 and 21. Apply **tcp.port==20** and **tcp.port==21**. Analyze the result and write down the purposes of these two ports for FTP.

**Tcp.port==20 is called control connection and is used for the purpose of commands and replies**

**Tcp.port==21 is called data connection and is for data transfer.**

**Analysing the results of port 20 shows that either the requests are sending SYN bit which is used to establish connection, or are sending ACK which is a reply meaning Acknowledgment is sent or are sending TCP Retransmission Request meaning a packet was lost and requires to be sent again.**

**Analysing the results of port 21 shows that data previously requested by client is being sent.**

1. Filter out each packet using either FTP or FTP-DATA Protocol (using **ftp || ftp-data** filter). Mention each packet number and its purpose with reference to request made and response received in the above mentioned FTP Session in command line to get file legal.txt (screenshot show above). Also look for **Response Code** and **Response Arg** in the FTP Header for each packet

**(**There are **19 such packets** and you have to write one/two lines explanation for each packet, what the packet is doing w.r.t FTP Session (Screenshot shown above) **e.g., Packet 104: Client asks server to send the data on IP:192.168.1.2 and Port:16341** [63(0x3F),213(0xD5) and **(0x3FD5=16341**)**] )**

**Packet Number & purpose:**

**Packet 89:**

**Server replies to client with response.**

**Response code: Service ready for new user (220)**

**Response arg: spftp/1.0.0000 Server [195.89.6.167]**

**Packet 94:**

**Client requests or asks server to send the data.**

**Request command: USER**

**Request arg: anonymous**

**Packet 96:**

**Server asks / request client for password.**

**Response code: User name okay, need password (331)**

**Response arg: Password required for USER.**

**Packet 99:**

**Client replies to server with password**

**Request code: -**

**Request command: PASS**

**Packet 100:**

**Server replies to client**

**Response code: User logged in, proceed (230)**

**Response arg:**

**230- ---------------------------------------------------------------------------\r\n**

**230- WARNING: This is a restricted access system. If you do not have explicit\r\n**

**230- permission to access this system, please disconnect immediately!\r\n**

**230 ----------------------------------------------------------------------------\r\n**

**Packet 104:**

**Client sends request to server.**

**Request command: PORT**

**Request arg: 192,168,1,2,63,213**

**Packet 105:**

**Server sends response to client.**

**Response code: Command okay (200)**

**Response arg: PORT command successful.**

**Packet 106:**

**Client sends request to server.**

**Response code: -, Request command: NLST**

**Packet 107:**

**Server sends response to client.**

**Response code: File status okay; about to open data connection (150)**

**Response arg: Opening ASCII mode data connection for /.**

**Packet 125:**

**Server sends response to client saying Transfer Complete**

**Response code: Closing data connection (226)**

**Response arg: Transfer Complete**

**Packet 151:**

**Client sends request to server**

**Request command: PORT**

**Request arg: 192,168,1,2,63,214**

**Packet 152:**

**Server sends response to client that port command is successful.**

**Response code: Command okay (200)**

**Response arg: PORT command successful.**

**Packet 153:**

**Client requests server to return legal.txt**

**Request command: RETR**

**Request arg: legal.txt**

**Packet 155:**

**Server sends response to client**

**Response code: File status okay; about to open data connection (150)**

**Response arg: Opening ASCII mode data connection for legal.txt (1415 bytes).**

**Packet 160:**

**Server sends response to client Transfer Complete**

**Response code: Closing data connection (226)**

**Response arg: Transfer Complete**

**Packet 173:**

**Client requests server to quit.**

**Response code: -, Request command: QUIT**

**Packet 175:**

**Server sends response to client Goodbye**

**Response code: Service closing control connection (221), Response arg: Goodbye.**

**Packet 127:**

**From client to server sends details of data.**

**FTP Data (commonupdater\r\ncommonupdater2\r\ncommonupdater3\r\nf20tools\r\nlegal.txt\r\nlicensed\r\nproducts\r\npub\r\nspamdefs\r\nusage.txt\r\nvirusdefs\r\n)**

**Packet 161:**

**From Server to client sends data**

**FTP Data (\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\r\n\r\nWARNING\r\n \r\nYou are connected to ftp.nai.com\r\n \r\nThis is a private computer system. This computer \r\nsystem,including all related equipment, networks \r\nand net**

**Lab Statement 2**: Multithreaded ECHO server using TCP (10)





