**LAB 17**

**TCP & UDP**

1. Start Wireshark and open the capture-file “**Lab 17 – FTP capture.pcapng”** which includes capturing an FTP session between an FTP client and an FTP server.
2. This Wireshark capture file contains several TCP and UDP sessions. Use the conversation statistics recorded by Wireshark to check how many sessions were captured.

**Number of TCP sessions capped** **=** 61

**Number of captured UDP sessions** **=** 4

What elements jointly identify such a conversation in a unique way?

* Sender IP
* Sender portnumber
* Reciever IP
* Reciever portnumber

1. To view only the segments of a single TCP session, you need to use a filter. You can enter the filter expression yourself, but it is more interesting to have Wireshark do it for you.

Below is explained how you can filter all TCP segments that belong to a session that includes the first frame, for example.

* Right click on the first frame and choose *Follow > TCP Stream.* This opens a new window.
* Close this new window. You will notice that Wireshark automatically generated a filter. This filter ensures that only the segments that belong to the same TCP session as the first segment are shown.

Write down this filter below.

**Filter =** tcp.stream eq 0

So how many segments does the first TCP session consist of?

30

1. Change the configured filter so that you only see the frames that belong to the penultimate (= one but last) TCP session.

Please note that the sessions are counted from 0 onwards!

**Filter =** tcp.stream eq 59

How many segments does the penultimate TCP session according to this set filter consist of?

114

Note that this number is also shown in the table of conversation statistics of the TCP protocol (see previous questions). In order to quickly find the penultimate TCP session in that table, you do need to sort the table differently.

What field do you have to sort for this?

Rel start

1. Which TCP stream does frame number 919 belong to?

Stream 13

1. Answer the following questions about the TCP session to which **frame number 497** belongs:
2. Enter the numbers of the frames which are part of the TCP handshake for this session

488, 489, 495

1. Enter the numbers of the frames which are part of the TCP finalisation of this session

721, 722, 723, 725

1. Which client port was used during this session?

50095

1. Which server port was used during this session?

20

From this “well known port” you can deduce that this TCP session was used by the FTP protocol (see lecture slides).

FTP stands for File Transfer Protocol and used to be widely used to exchange files between a client and a server. Now this is almost always done via the HTTP(S) protocol.

1. What file (name) was transferred by FTP in this session?

The file ‘index.txt’

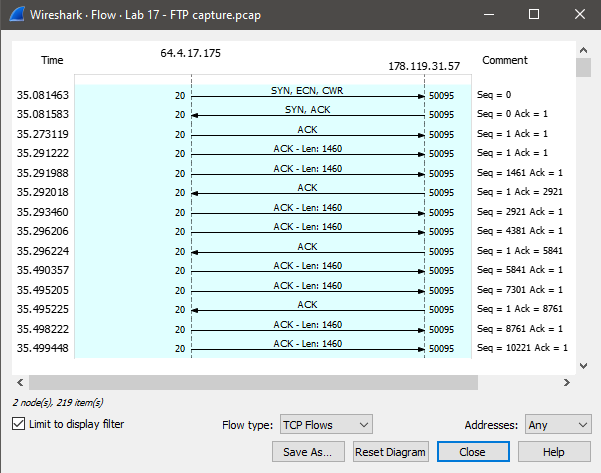
1. What is the IP address of the FTP client in this session?

178.119.31.57

1. What is the IP address of the FTP server in this session?

64.4.17.157

1. You can also have a TCP session graphically displayed in Wireshark. Click on the menu *Statistics* > *Flow Graph*. In the bottom of the window, select “Limit to displayed filter” and the entry “Flow Types” for “TCP Flows”. Paste below a screenshot of this graphic representation.



1. In this graphical presentation you can see the TCP initialisation during the beginning of the session and the TCP finalisation at the end of the session. Between these 2 phases the data transfer takes place (in this case: the transfer of a file using the FTP protocol). Is this a upload or a download of a file (from the client’s point of view)? Explain how you can see this.

The client downloads a file. You can see this because the client only sends ACKs.

After how many segments are acknowledgments sent for the received segments?

2 packets

Then close the “graphic” window.

1. Answer the following questions about the first frame of the TCP session to which frame 497 belongs:
2. According to this frame, how many bytes can be sent by the transmitter before sending a confirmation back to the transmitter?

8192 packets

1. Is the use of selective ACKs (SACK) allowed?

Yes. The info field states ‘SACK\_PERM=1’

1. Note below the sequence number (SEQNR) and acknowledgment number (ACKNR) of the TCP segment with number 497.

**SEQNR = 1**

**ACKNR =** 1

Note below the sequence number (SEQNR) and acknowledgment number (ACKNR) of the frame with number 498.

**SEQNR = 1461**

**ACKNR =** 1

Explain how this new SEQNR is created.

The old SEQNR + amount of sent bytes

Also explain why the ACKNR remains unchanged.

This is the number of the fist byte that is expected. Because the computer sends 2 packets immediately after each other, there are no packets received between there 2.

1. Since the insecure FTP protocol was used during this session, we can also view the content of the file sent in Wireshark. This is in the new window that opens when you follow the TCP stream.

What’s the first word in that traffic jam?

Last

What’s the last word in that traffic jam?

Pack

Save this file under the name “downloaded file” as a separate.txt file on your laptop and then open it with Notepad(++). So normally now you should see the same thing as in Wireshark...

1. In the previous assignments, we concentrated on 1 TCP session containing FTP data from a single transferred file. To display all FTP data that is in the entire capture file, use the filter below:

***ftp-data***

Was more than 1 file transferred in this capture? Explain!

Only 1 file. RETR stands for Retrieve

1. An FTP server uses 2 TCP connections: a data connection (1 per file transferred) and one control connection (for exchanging data to manage the connection).

In the previous question we have already inspected the FTP data connection. So it’s time to check out the control connection. This is used for setting up the FTP connection, asking for a username and password from the FTP server.

Use the filter below to view the FTP control connection:

***ftp***

Take a close look at the Info field of the filtered frames to answer the following questions:

* What port number was used by the FTP server?

21

Note that this is a different port number than that used by the FTP data connection (see command 7d).

* What is the name with which the user logged in to the FTP server?

Anonymous

* What password did this user log in with?

123456

Note that this is an unsafe FTP session, since we can read the password just like that. Of course, there are better (secure) ways to do FTP...

* What’s the name of the FTP server that was contacted?

ftp.microsoft.com

* Which file was downloaded and from which server folder was it downloaded?

The file ‘index.txt’ from the folder ‘softlib’

1. In this capture, not only TCP, but also UDP were used.

Use a filter to only see the frames that belong to the **UDP stream** with number 1.

Why don't you see any handshake segments here?

UDP doesn’t use handshakes

Which UDP port is used by the server?

53

Of which application is this the “well known” port?

DNS

Describe what is happening in this UDP session.

DNS requests the IP-address of the domain ‘ftp.microsoft.com’ and the DNS-server responds with the IP.

1. Enter the numbers of the TCP streams that were properly closed (e.g. using the 4 segments used for this purpose).

(tcp.flags.fin==1)

Streams 56 and 41