Name: Mark.Gardner…… Group: 631……

W4 – TCP

Save the capture files! Mentsd el az adatfájlokat!
Use your own capture file! Használd az általad készített adatfájlt!
Use your browser in incognito mode! Használd a böngésződ inkognitó módban!
Example: Példa:

Write an example for the TCP segment length.
Pckg. nr. 808, Size: 1412 byte

Capture filename: w4_2.pcapng……

1. Capturing a bulk TCP transfer from your computer to a remote server

No questions here 😊. Read the W4_TCP_readme.pdf carefully.

2. A first look at the captured trace

1. What is the IP address and TCP port number used by the client computer (source) that is transferring the alice.txt file to gaia.cs.umass.edu? To answer this question, find and use the HTTP POST message and explore the details of the TCP packet used to carry this HTTP message, using the "details of the selected packet header window" (refer to Figure 2 in the "Getting Started with Wireshark" Lab if you're uncertain about the Wireshark windows).

   Pckg. nr.: 1558……..
   Internet address: 192.168.1.205……..
   TCP port number: 42260……..……..

2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection? (use the package used at question 1)

   Pckg. nr.: 1558……..
   Internet address: 128.119.245.12……..
   TCP port number: 80……..……..

3. TCP Basics

3. What is the relative sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? (Note: this is the sequence number carried in the TCP segment itself; it is NOT the packet # in the "No." column in the Wireshark window. Remember there is no such thing as a "packet number" in TCP or UDP; as you know, there are sequence numbers in TCP and that's what we're after here.

   Pckg. nr.: 1550……..
   Relative sequence number: 0……..

   What is it in this TCP segment that identifies the segment as a SYN segment?

   Field/flag name(s) (e.g.: destination port, reset, …):
   flags……..……..
   Field/flag value(s) (e.g.: 80, 0, …):
   0x002……....

4. What is the relative sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment?

   Pckg. nr.: 1555……..
   Relative sequence number: 0……..
   Acknowledgement value: 1……..

   What is it in the segment that identifies the segment as a SYNACK segment?

   Field/flag name(s) (e.g.: destination port, reset, …):
   flags……..……..
   Field/flag value(s) (e.g.: 80, 0, …):
   0x012……....

5. What is the relative sequence number of the TCP segment containing the header of the HTTP POST command? **Note** that in order to find the POST message header, you'll need to dig into the **packet content field** at the bottom of the Wireshark window, looking for a segment with the **ASCII** text "POST" within its DATA field[1,2].

---

[1] Hint: **this TCP segment is sent by the client soon** (but not always immediately) **after the SYNACK segment** is received from the server.

[2] Note that if you filter to only show "http" messages, you'll see that the TCP segment that Wireshark associates with the HTTP POST message is the *last* TCP segment in the connection (which contains the text at the *end* of alice.txt: "THE END") and *not* the first data-carrying segment in the connection. Students (and teachers!) often find this unexpected and/or confusing.

Pckg. nr.: 1558……..
Relative sequence number: 1……..

How many bytes of data are contained in the payload (data) field of this TCP segment?

Number of bytes: .1428……..

Did all of the data in the transferred file alice.txt fit into this single segment?

Answer: NO……..

6. Consider the TCP segment containing the HTTP "POST" as the first segment in the data transfer part of the TCP connection.
What are the package and relative sequence numbers of the first four segments in the TCP connection (including the segment containing the HTTP POST)?

Pckg. numbers (e.g.: 15, 18, 20, …):
1558, 1559, 1560, 1561…….……..……..
Relative sequence numbers (e.g.: 1, 505, 1232, …):
1, 1429, 2857, 4285…….……..……..

What are the package and acknowledgement numbers of the packages containing the ACKs?

Pckg. numbers (e.g.: 15, 18, 20, …):
1585,1588, 1589…….……..……..
Acknowledgement numbers (e.g.: 1, 505, 1232, …):
1429, 2857, 5713…….……..……..

At what time was the first segment (the one containing the HTTP POST) in the data-transfer part of the TCP connection sent? (at least 4 decimal precision)

Pckg nr.: 1558…… Time (e.g. 12.3456 seconds):.10.80878…..seconds

At what time was the ACK for this first data-containing segment received? (at least 4 decimal precision)

Pckg nr.: 1585…… Time (e.g. 12.3456 seconds):.10.932487….seconds

What is the RTT for this first data-containing segment?

RTT: .0.1237…… seconds (at least 4 decimal precision)

What is the RTT value the second data-carrying TCP segment and its ACK?

RTT: .0.1265…… seconds (at least 4 decimal precision)

What is the `EstimatedRTT` value (see Section 3.5.3, in the text) after the ACK for the second data-carrying segment is received? Assume that in making this calculation after the received of the ACK for the second segment, that the initial value of `EstimatedRTT` is equal to the measured RTT for the first segment, and then is computed using the `EstimatedRTT` equation on page 242, and a value of $\alpha = 0.125$.

Estimated RTT: .0.1240…… seconds (at least 4 decimal precision)

7. What is the length (header plus payload) of each of the first four data-carrying TCP segments?

   1. pckg. nr.: 1558…….. Segment length: 1494……..
   2. pckg. nr.: 1559…….. Segment length: 1494……..
   3. pckg. nr.: 1560…….. Segment length: 1494……..
   4. pckg. nr.: 1561…….. Segment length: 1494……..

8. What is the minimum amount of available buffer space advertised to the client by gaia.cs.umass.edu among these first four data-carrying TCP segments[3]? (Fill the fields based on the package containing the minimum amount of available buffer space).

   Pckg. nr.: 1558……
   Window size: 126……
   Window scaling factor: 512……
   Number of available bytes at the server: 64512……

   Does the lack of receiver buffer space ever throttle the sender for these first four data-carrying segments?

   Answer: NO

9. Are there any retransmitted segments in the trace file? If yes, give an example.

   Answer: NO

   Pckg. nr (if the answer is yes): ……

10. How much data does the receiver typically acknowledge in an ACK? Give an example where the ACK acknowledges a single segment.

    Pckg. nr. of ACK: 1588……..
    Pckg. nr. acknowledged segment 1559……..

[3] The Window size value must be multiplied by the Window Scaling Factor to give the actual number of bytes available at gaia.cs.umass.edu for this connection.

Number of acknowledged bytes: 1428……..

Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 in the text)?

Pckg. nr. of ACK: 1589……..
Pckg. nrs of acknowledged segments (e.g.: 15,16): 1560, 1561…..
Total number of acknowledged bytes: 2856……..

11. -

4. TCP congestion control in action

No questions here 😊 However you can find questions in the readme. We recommend to check them and think about the answers.