

Lab 8 (Week 9)

TCP Connection Analysis

CAN201

Dr. Wenjun Fan

Outline

- General feedback for the last Lab
- TCP connection analysis
- Lab Exercise

General feedback the last Lab

- No module named ‘mininet’
- \$sudo pip3 install or #pip3 install
- ip addresses inconsistent

<https://box.xjtu.edu.cn/f/c4c3d6cc73964c7b9db0/>



wf@vm: ~/CAN201

File Edit View Search Terminal Help

wf@vm:~/CAN201\$



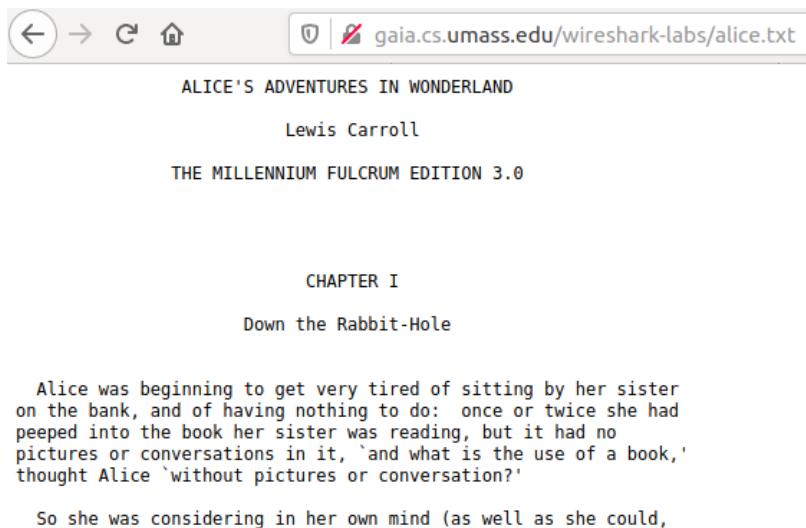
Right Ctrl

TCP connection analysis

1. We'll analyze a trace of the TCP segments sent and received in transferring a 150KB file from your computer to a remote server.
2. To this end, we'll need to use Wireshark to obtain a packet trace of the TCP transfer of a file from your computer to a remote server. You'll do so by accessing a Web page that will allow you to enter the name of a file stored on your computer, and then transfer the file to a Web server using the HTTP POST method while running wireshark during this time to obtain the trace of TCP segments.

The test run for the TCP connection analysis

1. Start up your web browser. Go to <http://gaia.cs.umass.edu/wireshark-labs/alice.txt> and retrieve an ASCII copy of *Alice in Wonderland*. Store this file somewhere on your computer.
 - Note: you just save the page as alice.txt on your local computer (Ubuntu OS)



The test run for the TCP connection analysis

2. Next go to <http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html>. You should see a screen like this

The screenshot shows a web browser window with the title "Upload page for TCP Wireshark Lab". The address bar displays the URL "gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html". The page content includes the following text:

Upload page for TCP Wireshark Lab
Computer Networking: A Top Down Approach, 6th edition
Copyright 2012 J.F. Kurose and K.W. Ross, All Rights Reserved

If you have followed the instructions for the TCP Wireshark Lab, you have *already* downloaded an ASCII copy of Alice and Wonderland from <http://gaia.cs.umass.edu/wireshark-labs/alice.txt> and you also *already* have the Wireshark packet sniffer running and capturing packets on your computer.

Click on the Browse button below to select the directory/file name for the copy of alice.txt that is stored on your computer.

No file selected.

Once you have selected the file, click on the "Upload alice.txt file" button below. This will cause your browser to send a copy of alice.txt over an HTTP connection (using TCP) to the web server at gaia.cs.umass.edu. After clicking on the button, wait until a short message is displayed indicating the the upload is complete. Then stop your Wireshark packet sniffer - you're ready to begin analyzing the TCP transfer of alice.txt from your computer to gaia.cs.umass.edu!!

The test run for the TCP connection analysis

3. Use the *Browse* button in this form to enter the name of the file (alice.txt) on your computer.
 - Note: Don't yet press the “*Upload alice.txt file*” button.

Upload page for TCP Wireshark x +

← → ⌛ ⌂ gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html

Upload page for TCP Wireshark Lab
Computer Networking: A Top Down Approach, 6th edition
Copyright 2012 J.F. Kurose and K.W. Ross, All Rights Reserved

If you have followed the instructions for the TCP Wireshark Lab, you have *already* downloaded a <http://gaia.cs.umass.edu/wireshark-labs/alice.txt> and you also *already* have the Wireshark packet capture file on your computer.

Click on the Browse button below to select the directory/file name for the copy of alice.txt that is

alice.txt

Once you have selected the file, click on the "Upload alice.txt file" button below. This will cause your browser to establish a connection (using TCP) to the web server at gaia.cs.umass.edu. After clicking on the button, wait for the upload to complete. Then stop your Wireshark packet sniffer - you're ready to begin analyzing the file on gaia.cs.umass.edu!!

The test run for the TCP connection analysis

4. Now start up Wireshark and begin packet capture (*Capture->Start*) and then press *OK* on the Wireshark Packet Capture Options screen (we'll not need to select any options here).
5. Returning to your browser, press the “*Upload alice.txt file*” button to upload the file to the `gaia.cs.umass.edu` server. Once the file has been uploaded, a short congratulations message will be displayed in your browser window.
6. Stop Wireshark packet capture. Your Wireshark window should look similar to the window shown below.

The test run for the TCP connection analysis

No.	Time	Source	Destination	Protocol	Length	Info
208	18.497182758	10.0.2.9	128.119.245.12	TCP	74	59298 → 80 [SYN] Seq=0 Win=42340 Len=0 MSS=1460
209	18.557937230	128.119.245.12	10.0.2.9	TCP	60	80 → 59296 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0
210	18.557979294	10.0.2.9	128.119.245.12	TCP	54	59296 → 80 [ACK] Seq=1 Ack=1 Win=42340 Len=0
211	18.558499238	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=1 Ack=1 Win=42340 Len=2920
212	18.558527709	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=2921 Ack=1 Win=42340 Len=2920
213	18.558537730	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=5841 Ack=1 Win=42340 Len=2920
214	18.558678362	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=8761 Ack=1 Win=42340 Len=2920
215	18.558695855	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=11681 Ack=1 Win=42340 Len=2920
216	18.559006789	128.119.245.12	10.0.2.9	TCP	60	80 → 59296 [ACK] Seq=1 Ack=2921 Win=32768 Len=0
217	18.559022071	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=14601 Ack=1 Win=42340 Len=0
218	18.559038039	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=17521 Ack=1 Win=42340 Len=0
219	18.559133314	128.119.245.12	10.0.2.9	TCP	60	80 → 59296 [ACK] Seq=1 Ack=5841 Win=32768 Len=0
220	18.559141794	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=20441 Ack=1 Win=42340 Len=0
221	18.559156489	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=23361 Ack=1 Win=42340 Len=0
222	18.559276628	128.119.245.12	10.0.2.9	TCP	60	80 → 59296 [ACK] Seq=1 Ack=8761 Win=32768 Len=0
223	18.559287559	10.0.2.9	128.119.245.12	TCP	1514	59296 → 80 [PSH, ACK] Seq=26281 Ack=1 Win=42340

```
▶ Frame 208: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_7e:1f:c1 (08:00:27:7e:1f:c1), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
▶ Internet Protocol Version 4, Src: 10.0.2.9, Dst: 128.119.245.12
▶ Transmission Control Protocol, Src Port: 59298, Dst Port: 80, Seq: 0, Len: 0
```

0000	52	54	00	12	35	00	08	00	27	7e	1f	c1	08	00	45	00	RT	5	'~	E
0010	00	3c	99	81	40	00	40	06	1f	ae	0a	00	02	09	80	77	<	00	..	W
0020	f5	0c	e7	a2	00	50	53	27	48	bf	00	00	00	00	a0	02	PS	'	H	..
0030	a5	64	81	bb	00	00	02	04	05	b4	04	02	08	0a	e4	ac	d
0040	1f	50	00	00	00	00	01	03	03	09	P

The test run for the TCP connection analysis

- To find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.

Frame 334: 2634 bytes on wire (21072 bits), 2634 bytes captured (21072 bits) on interface 0
Ethernet II, Src: PcsCompu_7e:1f:c1 (08:00:27:7e:1f:c1), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
Internet Protocol Version 4, Src: 10.0.2.9, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 59296, Dst Port: 80, Seq: 150381, Ack: 1, Len: 2580
[72 Reassembled TCP Segments (152960 bytes): #211(2920), #212(2920), #213(2920), #214(2920), #215(2920), #217(2920), #218(2920)]
Hypertext Transfer Protocol
POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1\r\nHost: gaia.cs.umass.edu\r\nUser-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:85.0) Gecko/20100101 Firefox/85.0\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\nAccept-Language: en-US,en;q=0.5\r\nAccept-Encoding: gzip, deflate\r\nContent-Type: multipart/form-data; boundary=-----43359184729323619052430048947\r\n

Lab Exercise

- Start your Wireshark to listen on the working interface, and then turn on your browser to open this link
<http://gaia.cs.umass.edu/wireshark-labs/alice.txt> and retrieve an ASCII copy of *Alice in Wonderland*.
 - *To see if you can get the similar ASCII characters as follows that were captured by Wireshark when the browser sent an HTTP GET message*

```
GET /wireshark-labs/alice.txt HTTP/1.1
Host: gaia.cs.umass.edu
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:106.0) Gecko/20100101 Firefox/106.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

Lab Exercise

- With the above string of *ASCII characters*, *answer the following questions:*
 - 1) What is the URL of the document requested by the browser?
 - 2) Which version of HTTP is the browser running?
 - 3) Does the browser request a non-persistent or a persistent (TCP) connection?
 - 4) What type of browser initiates this message?