Omar Salih

INFSYS 3842/6836

Assignment 3 (Lab): Understanding Internet Protocol (IPv4) using Wireshark

Points Possible: 100

Due Date: April 16, 2016 by 11:59pm Central Time

IMPORTANT NOTE: THIS LAB MUST BE CARRIED OUT ON YOUR OWN COMPUTER *AND* OWN NETWORK. PLEASE DO NOT CAPTURE PACKETS ON A NETWORK THAT YOU DO NOT OWN. YOU'VE BEEN WARNED!

Lab Overview: Building on our previous lab where we captured some HTTP traffic but focused on Ethernet, we will now look at the Internet Protocol (IP).

Lab Purpose:

1) To continue learning the basics of how to use Wireshark to capture network traffic (from students own computers and own networks)
2) To learn about basic "Capture Filters" available in Wireshark
3) Understand the syntax of IP datagram (packet) headers, its key fields, and some concepts specific to IPv4.

TASK 1: (This task has 6 questions)

In this task you will use Wireshark to capture some HTTP traffic and complete the activities and questions as described below.

I recommend you watch some videos on YouTube on capturing HTTP traffic using Wireshark. The process is fairly simply as demonstrated in class but feel free to learn more.

Also familiarize yourself with the basic interface of Wireshark (the menus, options, **filters** etc.). Again, plenty of videos are available online and the Wireshark wiki is good for more advanced users.

STEPS:

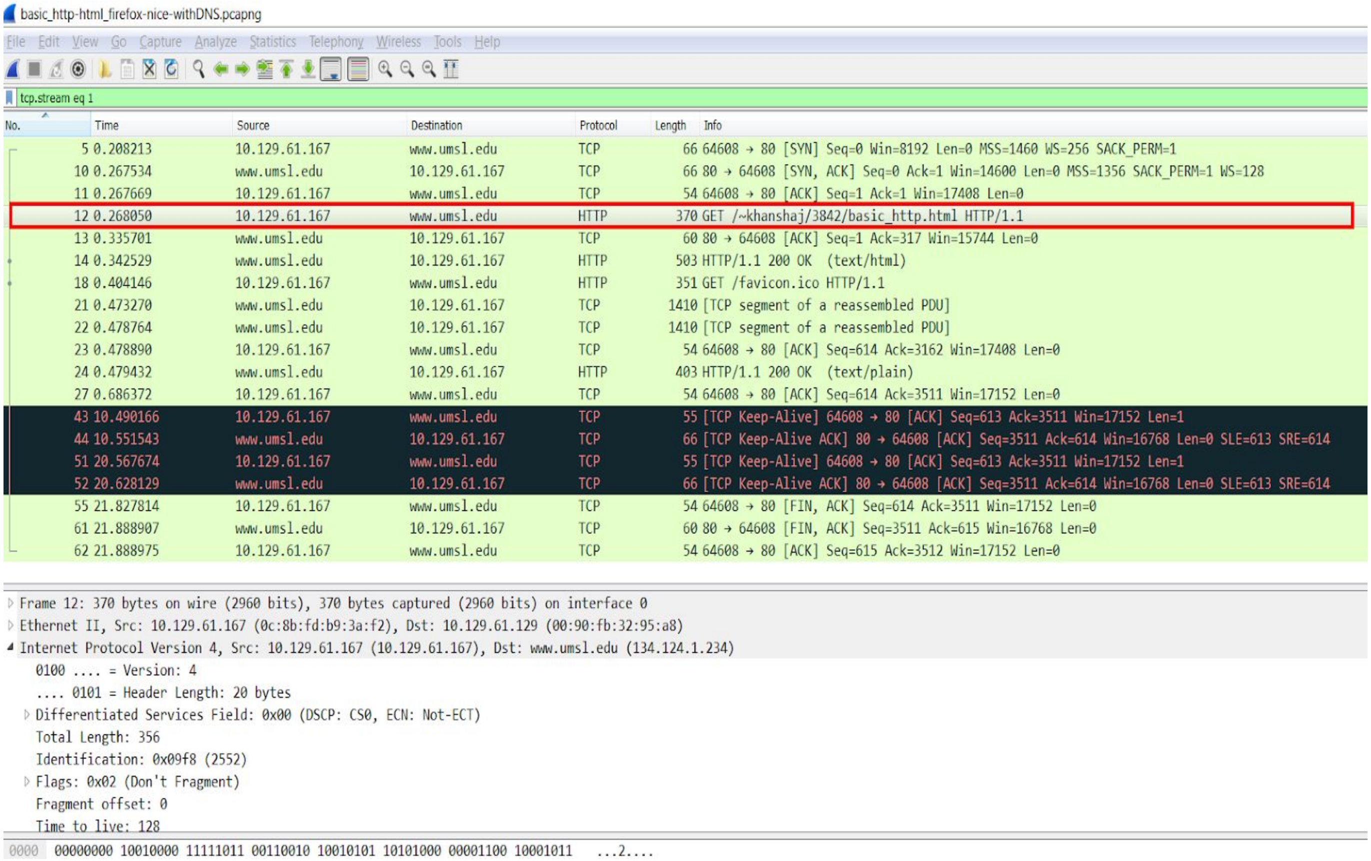
1) Clear history and temporary internet files/cache from your browser. **Note**: If you have to repeat the steps below (e.g., you want to capture again to make sure) then Clear browser cache again before trying steps below.

LAB Question 1: What interface are you using to capture traffic (**Wired Ethernet LAN/** Wireless / Virtual?)? Wired Ethernet LAN

LAB Question 2: Please identify the browser you used in the steps below. Chrome

CONTINUE:

2) Close all browser windows and other applications. Open WireShark. Open a browser window and type in **`http://www.umsl.edu/~khanshaj/3842/basic_http.html`**. DO NOT Press Enter yet.
3) Go back to Wireshark and **Start** a capture.
4) Go back to the browser window and now hit enter to visit the page.
5) Once the page loads, return to Wireshark and **Stop** the capture. **Save the capture on your computer. Call it "FirstName_LastName_Assignment3_Capture"**.
6) Examine the packets captured and scroll to find "green colored rows" that denote "TCP" based traffic.
7) FIND THE PACKET that belongs to the **HTTP GET** request asking for the `basic_http.html` file. [You may be able to use the search packets feature "CTRL+F" but be sure that you are searching for a "SRTING" as a display filter. Once located, right click on that packet and choose "Follow TCP Stream". This should remove all other packets allowing you to focus on just this TCP session. Another window showing the HTTP requests and responses should also open. You can minimize it.


basic_http-html_firefox-nice-withDNS.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 1

No.	Time	Source	Destination	Protocol	Length	Info
5	0.208213	10.129.61.167	www.umsl.edu	TCP	66	64608 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
10	0.267534	www.umsl.edu	10.129.61.167	TCP	66	80 → 64608 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1356 SACK_PERM=1 WS=128
11	0.267669	10.129.61.167	www.umsl.edu	TCP	54	64608 → 80 [ACK] Seq=1 Ack=1 Win=17408 Len=0
12	0.268050	10.129.61.167	www.umsl.edu	HTTP	370	GET /~khanshaj/3842/basic_http.html HTTP/1.1
13	0.335701	www.umsl.edu	10.129.61.167	TCP	60	80 → 64608 [ACK] Seq=1 Ack=317 Win=15744 Len=0
14	0.342529	www.umsl.edu	10.129.61.167	HTTP	503	HTTP/1.1 200 OK (text/html)
18	0.404146	10.129.61.167	www.umsl.edu	HTTP	351	GET /favicon.ico HTTP/1.1
21	0.473270	www.umsl.edu	10.129.61.167	TCP	1410	[TCP segment of a reassembled PDU]
22	0.478764	www.umsl.edu	10.129.61.167	TCP	1410	[TCP segment of a reassembled PDU]
23	0.478890	10.129.61.167	www.umsl.edu	TCP	54	64608 → 80 [ACK] Seq=614 Ack=3162 Win=17408 Len=0
24	0.479432	www.umsl.edu	10.129.61.167	HTTP	403	HTTP/1.1 200 OK (text/plain)
27	0.686372	10.129.61.167	www.umsl.edu	TCP	54	64608 → 80 [ACK] Seq=614 Ack=3511 Win=17152 Len=0
43	10.490166	10.129.61.167	www.umsl.edu	TCP	55	[TCP Keep-Alive] 64608 → 80 [ACK] Seq=613 Ack=3511 Win=17152 Len=1
44	10.551543	www.umsl.edu	10.129.61.167	TCP	66	[TCP Keep-Alive ACK] 80 → 64608 [ACK] Seq=3511 Ack=614 Win=16768 Len=0 SLE=613 SRE=614
51	20.567674	10.129.61.167	www.umsl.edu	TCP	55	[TCP Keep-Alive] 64608 → 80 [ACK] Seq=613 Ack=3511 Win=17152 Len=1
52	20.628129	www.umsl.edu	10.129.61.167	TCP	66	[TCP Keep-Alive ACK] 80 → 64608 [ACK] Seq=3511 Ack=614 Win=16768 Len=0 SLE=613 SRE=614
55	21.827814	10.129.61.167	www.umsl.edu	TCP	54	64608 → 80 [FIN, ACK] Seq=614 Ack=3511 Win=17152 Len=0
61	21.888907	www.umsl.edu	10.129.61.167	TCP	60	80 → 64608 [FIN, ACK] Seq=3511 Ack=615 Win=16768 Len=0
62	21.888975	10.129.61.167	www.umsl.edu	TCP	54	64608 → 80 [ACK] Seq=615 Ack=3512 Win=17152 Len=0

```
▷ Frame 12: 370 bytes on wire (2960 bits), 370 bytes captured (2960 bits) on interface 0
▷ Ethernet II, Src: 10.129.61.167 (0c:8b:fd:b9:3a:f2), Dst: 10.129.61.129 (00:90:fb:32:95:a8)
◢ Internet Protocol Version 4, Src: 10.129.61.167 (10.129.61.167), Dst: www.umsl.edu (134.124.1.234)
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes
  ▷ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 356
    Identification: 0x09f8 (2552)
  ▷ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 128
0000  00000000 10010000 11111011 00110010 10010101 10101000 00001100 10001011   ...2....
```

8) **With the GET request packet selected (it should remain highlighted as shown above), please complete the rest of this lab**

LAB Question 3: Using the packet associated with the GET request as mentioned above, please complete the following fields in the IPv4 Datagram (a.k.a. packet) header below. **The idea here is to learn about the different fields and what they accomplish**. NOTE: Although the field values are always in binary (as seen in the bottom most section of your capture) please feel free to provide the values as they appear in the MIDDLE part (i.e. either decimal or hex as the case maybe). Type in your values in the "Grey" shaded areas immediately below each field.

[Hint: please visit http://en.wikipedia.org/wiki/IPv4 to learn more about IP and look for the section "Packet Structure" to better understand what each of the fields below mean]

IP Version 4 Header

<table>
<tr><td>ffset</td><td>Octet</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>ctet</td><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>3</td></tr>
<tr><td rowspan="2">0</td><td rowspan="2">0</td><td colspan="4">Version
(4 bits)</td><td colspan="4">IP Header Length
(4 bits)</td><td colspan="6">Differentiated Services Code Point (6 bits)</td><td colspan="2">ECN
(2 bits)</td><td colspan="16">Total Length
(16 bits)</td></tr>
<tr><td colspan="4">0100</td><td colspan="4">0101</td><td colspan="6">0x00</td><td colspan="2">Not-ECT</td><td colspan="16">466</td></tr>
<tr><td rowspan="3">4</td><td rowspan="3">32</td><td colspan="16" rowspan="2">Identification
(16 bits)</td><td colspan="3">Flags</td><td colspan="13" rowspan="2">Fragment Offset
(13 bits)</td></tr>
<tr><td>R</td><td>DF</td><td>MF</td></tr>
<tr><td colspan="16">0x13f2 (5106)</td><td>0</td><td>1</td><td>0</td><td colspan="13">0</td></tr>
<tr><td rowspan="2">8</td><td rowspan="2">64</td><td colspan="8">Time to Live
(8 bits)</td><td colspan="8">Protocol
(8 bits)</td><td colspan="16">Header Checksum
(16 bits)</td></tr>
<tr><td colspan="8">128</td><td colspan="8">TCP (6)</td><td colspan="16">0x8923</td></tr>
<tr><td rowspan="2">12</td><td rowspan="2">96</td><td colspan="32">Source IP Address (32 bits)</td></tr>
<tr><td colspan="32">192.168.19.2</td></tr>
<tr><td rowspan="2">16</td><td rowspan="2">128</td><td colspan="32">Destination IP Address (32 bits)</td></tr>
<tr><td colspan="32">134.124.1.234</td></tr>
<tr><td>20</td><td>160</td><td colspan="32">Options (if IP Header Length > 5)</td></tr>
<tr><td></td><td></td><td colspan="32">DATA
(or PAYLOAD OF THE IP v4 Packet/Datagram)
[The type of "payload" i.e. upper layer protocol packet the IP datagram is carrying is specified in the Protocol field above. See http://en.wikipedia.org/wiki/List_of_IP_protocol_numbers for a full list)</td></tr>
</table>

LAB Question 4: For each of the IPv4 datagram header fields you completed above, please briefly describe its purpose/meaning. When possible, please answer this question both in general as well as in particular to the datagram you are currently examining. For example, *the current value in the Protocol field is 6 which denotes that this datagram is carrying a TCP segment. This makes sense as we are looking at HTTP data being sent in TCP segments which are being carried by IPv4 packets enclosed in Ethernet Frames!.*)

[Hint: A good reading of the above link and understanding the packet structure will help with this task. The answers are given on the link above but try to actually understand what each field does.]

- Version tells us what protocols to follow. If its version 4 it is different than version 6.
- IP Header Length helps us to find out what the Fragment Offset is.
- Differentiated Services Code Point is the Type of Service (ToS)
- ECN it does end to end notification of network congestion without dropping packets if the both sides have and are willing to use ECN
- Total Length defines the entire packet size
- Identification it is practically used for identifying the group of fragments of a single IP datagram.
- Flags lets us know if this packet is the last of datagram. Also, the last bit by being
  - bit 0: Reserved; must be zero.[note 1]
  - bit 1: Don't Fragment (DF)
  - bit 2: More Fragments (MF)
- Fragment Offset lets us know by how much we can increment each packet by
- Time to Live it helps the datagram from being in an infinite loop
- Protocol tells us what type of protocol to use what type of connection we are on if it's a TCP, HTTP, …
- Header Checksum it is used for error checking the header. The router when it receives the packet header it compares it and if it matches no problem, if not it discards it.
- Source IP Address tells us where the datagram originated from
- Destination IP Address tells us where the datagram is going to

LAB Question 5:

a) What is the MINIMUM size (in bytes/octets) of an IPv4 packet's header? 20 bytes
b) What is the MAXIMUM size (in bytes/octets) of an IPv4 packet's header? 65,535 bytes

LAB Question 6: IPv4 allows for packet fragmentation (whereas IPv6 DOES NOT allow for packet fragmentation). On the IPv4 page on Wikipedia read about Fragmentation and Reassembly (https://en.wikipedia.org/wiki/IPv4#Fragmentation_and_reassembly). Please briefly explain the idea behind IP fragmentation (why it is sometimes necessary, which device typically performs this function in IPv4 networks, and what IP fragmentation is?).

For IPv4 networks to communicate with networks that have different maximum transmission units MTU's, hardware, and transmission speeds, the networks that have to communicate with networks with lower or higher MTU's can fragment the datagrams at the internet layer. When a router receives a packet it will use the outgoing interface it is using and the interfaces MTU of the destination address. If the MTU is not as big as the packet size and the Do Not Fragment (last bit) is set to 0, then the router will decide to fragment the packet. Each packet that's fragmented will then be put into its own fragmented packet. That packet and all following fragments will then have its own: total length field, the more fragment flag will be set to 0 except for the last one set to 1, fragment offset field will be set, and a fresh header checksum field.

Depending on what type of MTU Link you are on, if it's a MTU of 2,500 bytes, then files or packets are can have a max size of 2,500 bytes. A packet size is the max MTU size + header size (usually 20 – 60 bytes). In a case of a packet size of 3,235 bytes, the router will break the packets down into 2,500 bytes and 735 bytes packets. When the router makes the packets it changes total length field, more fragments flag is set for all except the last packet which will be set to 0, fragment offset field will be set to the total MTU size of 3,235 bytes + header size of 20 bytes = 3,255/8 =406.875. The fragment offset would be in multiples of 407. So packet 1 would be 2,500 bytes big with data size of 2480 bytes More Fragments would be 1 and the Frag offset would be 0. Packet 2 would be 735 bytes big with data size of the packet would be 735bytes – header size of 20 bytes = 715bytes, the More Fragments would be a 0, but if it's the last packet it will be a 1, and 407 would be the Fragment offset increment for each subsequent packet.

LAB DELIVERABLES (to be uploaded to MyGateway):

1) **Upload completed Word document (saved as "FirstName_LastName_Assignment3")**
2) **Upload the Wireshark capture file**

GETTING HELP:

1) Call (314-489-9733) / email (shajikhan@umsl.edu) me anytime. Feel free to walk-in to my office if you see me there or setup appointment. Tutors are also available (see MyGateway/Faculty Information for hours, or simply walk in to cybersecurity lab, ESH 204 during lab hours).
2) Of course try to help each other out. If some students are already familiar with the tasks listed above, I encourage you to help others. "Teaching" and helping others is by far the best way to learn!