Assignment 1 - Wireshark
COMP 307 - Principles of Web Development
Prof. Joseph Vybihal
Fall 2018

LE, Nhat Hung
McGill ID: 260793376
Date: September 14, 2018
Due date: September 17, 2018

1. Identify the exact locations you were when during your two Wi-Fi packet recording sessions.

Public network: Sherbrooke Tim Hortons.
Home network: At home in my bed.

2. Experiment 1 (at home)

a. Select a packet and include screenshots.

No.	Delta	Time	Source	Destination	Protocol	Length	Info
351	0.015245279	41.763300877	104.208.165.109	192.168.0.103	TCP	74	443 → 44600 [SYN, ACK]
352	0.000046844	41.763347721	192.168.0.103	104.208.165.109	TCP	66	44600 → 443 [ACK] Seq=
353	0.000478701	41.763826422	192.168.0.103	104.208.165.109	TLSv1.2	583	Client Hello
354	0.043456035	41.807282457	104.208.165.109	192.168.0.103	TCP	1514	443 → 44600 [ACK] Seq=
355	0.000041650	41.807324107	192.168.0.103	104.208.165.109	TCP	66	44600 → 443 [ACK] Seq=

Figure 1: Selected packet highlighted in blue.

```
▸ Frame 353: 583 bytes on wire (4664 bits), 583 bytes captured (4664 bits) on interface 0
▸ Ethernet II, Src: HonHaiPr_f4:cf:8b (48:e2:44:f4:cf:8b), Dst: ZyxelCom_d2:5d:ce (b8:ec:a3:d2:5d:ce)
▸ Internet Protocol Version 4, Src: 192.168.0.103, Dst: 104.208.165.109
▸ Transmission Control Protocol, Src Port: 44600, Dst Port: 443, Seq: 1, Ack: 1, Len: 517
▸ Secure Sockets Layer
```

Figure 2: Packet info.

```
0000  b8 ec a3 d2 5d ce 48 e2  44 f4 cf 8b 08 00 45 00   ····]·H· D·····E·
0010  02 39 ec 73 40 00 40 06  7c fe c0 a8 00 67 68 d0   ·9·s@·@· |····gh·
0020  a5 6d ae 38 01 bb 46 c1  62 97 86 94 06 dc 80 18   ·m·8··F· b·······
0030  00 e5 2a a1 00 00 01 01  08 0a c2 9c 98 8a 03 ce   ··*····· ········
0040  99 3e 16 03 01 02 00 01  00 01 fc 03 03 aa 71 2e   ·>······ ······q.
0050  42 04 59 35 32 d4 b9 89  9d e3 39 24 14 1f ed bd   B·Y52··· ··9$····
0060  26 ad 2f f8 8f 56 75 bd  87 ab 84 eb f8 20 72 11   &·/··Vu· ····· r·
0070  00 00 3f d5 e1 23 35 61  76 38 4c 18 b3 31 6e 26   ··?··#5a v8L··1n&
0080  a1 e4 15 fd 05 85 d7 77  91 a8 d6 c1 f3 59 00 22   ·······w ·····Y·"
0090  ea ea 13 01 13 02 13 03  c0 2b c0 2f c0 2c c0 30   ········ ·+·/·,·0
00a0  cc a9 cc a8 c0 13 c0 14  00 9c 00 9d 00 2f 00 35   ········ ·····/·5
00b0  00 0a 01 00 01 91 2a 2a  00 00 ff 01 00 01 00 00   ······** ········
00c0  00 00 19 00 17 00 00 14  73 74 61 74 69 63 2e 61   ········ static.a
00d0  73 6d 2e 73 6b 79 70 65  2e 63 6f 6d 00 17 00 00   sm.skype .com····
00e0  00 23 00 00 00 0d 00 14  00 12 04 03 08 04 04 01   ·#····· ········
00f0  05 03 08 05 05 01 08 06  06 01 02 01 00 05 00 05   ········ ········
0100  01 00 00 00 00 00 12 00  00 00 10 00 0e 00 0c 02   ········ ········
0110  68 32 08 68 74 74 70 2f  31 2e 31 00 0b 00 02 01   h2·http/ 1.1·····
0120  00 00 33 00 2b 00 29 8a  8a 00 01 00 00 1d 00 20   ··3·+·)· ·· ····
0130  96 77 0b 11 fd 91 2f 69  6e 4c 82 35 87 4e db d9   ·w····/i nL·5·N··
0140  cc 10 31 c7 00 5c 0d f4  98 00 4d 1e 37 f5 2c 24   ··1··\·· ··M·7·,$
0150  00 2d 00 02 01 01 00 2b  00 0b 0a ca ca 7f 17 03   ·-·····+ ········
0160  03 03 02 03 01 00 0a 00  0a 00 08 8a 8a 00 1d 00   ········ ········
0170  17 00 18 ea ea 00 01 00  00 15 00 cb 00 00 00 00   ········ ········
0180  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0190  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
01a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
01b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
01c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
01d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
01e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
01f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0200  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0210  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0220  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0230  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0240  00 00 00 00 00 00 00                               ·······
```

Figure 3: Packet's raw data.

b. Identify all the fields contained within the packet. Compare with lecture slides.

Will now compare with the following figures, taken from lecture slides:

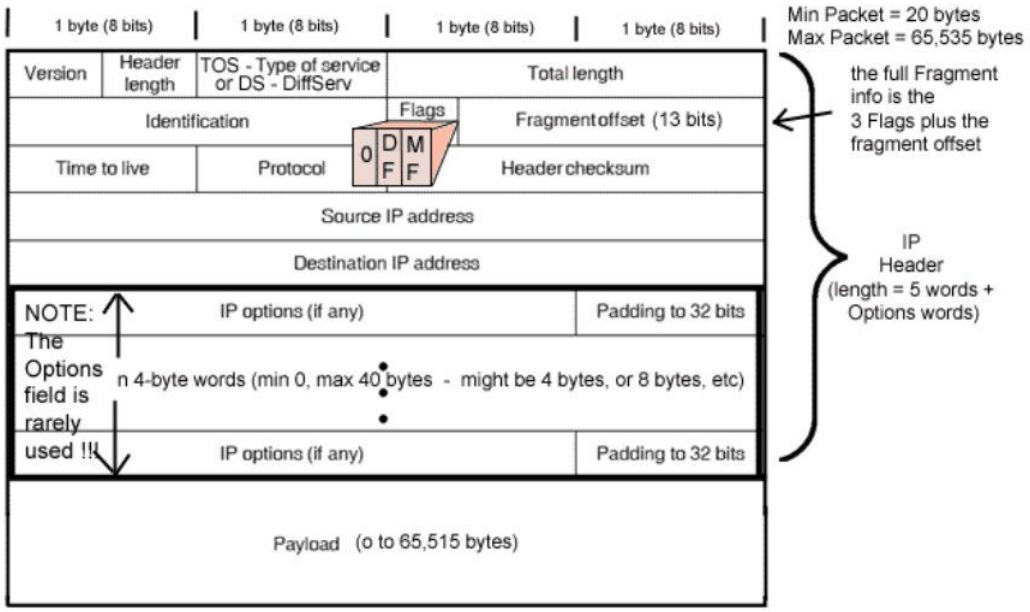


Figure 4: Packet data structure

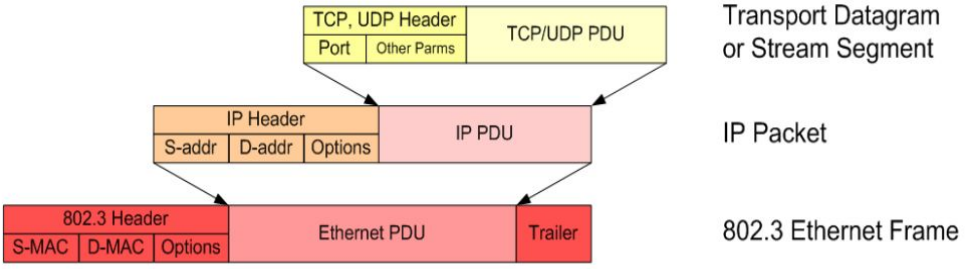


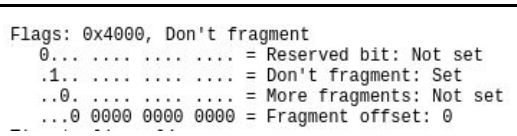
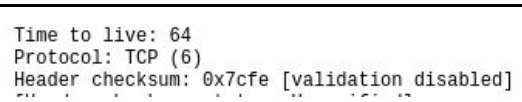
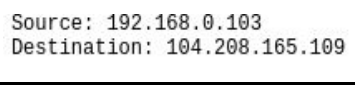
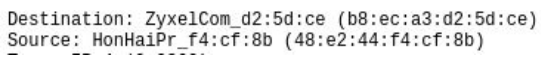
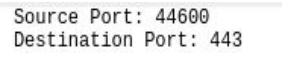
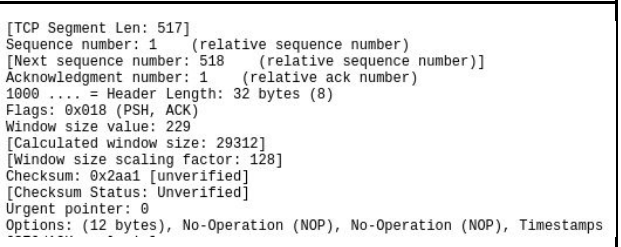
Figure 5: Nested structures of packets

Now identifying fields through the Wireshark packet info pane:

Fields present in figure 4		

IP Header	Version, header length & DiffServ	`0100 .... = Version: 4`
		`.... 0101 = Header Length: 20 bytes (5)`
		`Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)`

	Total length	`Total Length: 569`
	Identification	`Identification: 0xec73 (60531)`

	Flags & fragment offset	
	Time to live, protocol & header checksum	
	Source & destination IP addresses	
Payload		
Fields present in figure 5		
802.3 header	Source & destination MAC addresses	
TCP, UDP header	Source & destination ports	
	"Other parameters", which from the Wireshark packet info pane include: ● TCP segment length ● Sequence number ● Acknowledgement number ● Header length ● Flags ● Window size ● Checksum ● Urgent pointer ● Various options	

The packet also contains other fields not represented in the two figures from the lecture slides.

These fields are more specific to the protocol, which in this case is TLSv1.2 (Transport Layer Security, a newer and more secure version of SSL). The basic unit of data in SSL (or TLS) is a **record.** The record in this case is the (TCP) payload represented in the table above.

An SSL record consists of

- A 5 bytes **record header,** containing metadata, similar to TCP headers or IP headers. This packet's SSL record header contains the following fields
    - Content type
    - Protocol version
    - Message length

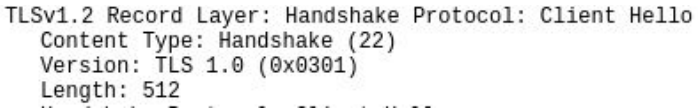


Figure 6: The packet's record header

- The actual data/message. From the content type specified above, we know the data here is a handshake.

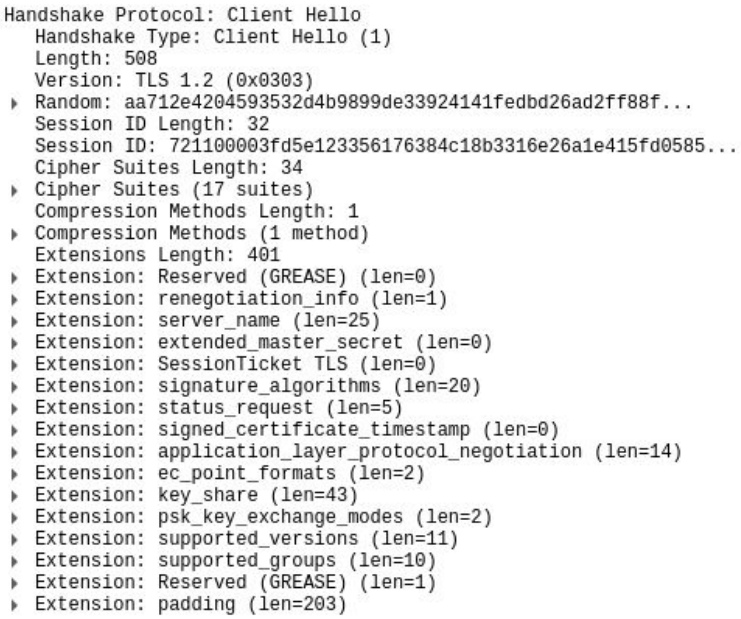


Figure 7: The handshake

c. What information does this packet contain?

Where it came from:
The packet's source MAC address, 48:e2:44:f4:cf:8b, matches my pc's MAC address. The packet then came from my own computer.

Figure 8: MAC address obtained from running ifconfig in command line

Where it's going:
The packet was heading to Microsoft. We can see this through three points.

Exhibit A: Pasting the destination IP address, 104.208.165.109, into Google shows a first page full of IP tracing sites pointing to Microsoft.

Exhibit B: Looking inside the packet data, we find the name the of the destination server: static.asm.skype.com. Skype is a Microsoft product.

```
Server Name Indication extension
   Server Name list length: 23
   Server Name Type: host_name (0)
   Server Name length: 20
   Server Name: static.asm.skype.com
```

Figure 8: Inside the packet's payload

Exhibit C: Wireshark's packet list indicates relationships between captured packets. It shows our selected packet is part of a conversation. The subsequent packets in the conversation show clear traces of Microsoft. Case in point the packet which directly follows our packet is an acknowledgement packet. In it is undeniable proof:

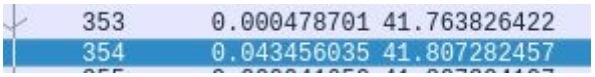


Figure 9: Our packet, no. 353, is acknowledged by the highlighted packet, as shown by the check mark

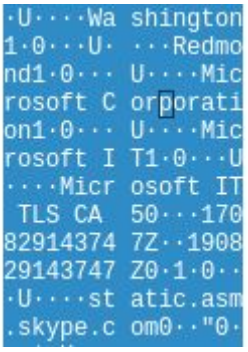


Figure 10: Packet no. 354's raw data

```
01 0 UUS10U
Washington10URedmond10U
Microsoft Corporation10U Microsoft IT10UMicrosoft IT TLS CA 50
170829143747Z
1908291437477Z010Ustatic.asm.skype.com0"0
```

Figure 11: Enhanced (actually just copy pasted into a text editor)

In conclusion the packet's destination is Microsoft, more specifically for a Skype service.

What security it's using:
The packet uses SSL encryption.

What message it contains:
The packet's record header indicates it's a Client Hello message sent to the server as the start of a handshake protocol. It sends the following information (as shown in figure 7):

- Its protocol version is TLS 1.2
- The session ID the client wishes to use for this connection
- A cipher suite containing the combinations of cryptographic algorithms supported by the client
- Compression methods - a list of compression algorithms supported by the client

In summary, the Client Hello message shares the above information and awaits a responding Server Hello, in order to agree on a protocol version, select cryptographic algorithms, optionally authenticate each other, and use public key encryption techniques in order to generate shared secrets.

This packet is then trying to start a secure connection with Microsoft through a handshake protocol.

3. Experiment 2 (at a public hotspot)

a. Select and sort by a sender IP address and try to deduce what they were trying to do.

Now picking sender IP 172.18.20.47.

Source	Destination	Protocol	Length	Info
172.18.20.47	31.13.80.5	TCP	142	39070 → 443 [ACK]
172.18.20.47	31.13.80.5	TCP	158	[TCP Dup ACK 5524
172.18.20.47	31.13.80.5	TCP	142	[TCP Dup ACK 5524
172.18.20.47	31.13.80.5	TCP	158	[TCP Dup ACK 5524

Figure 12: Sorted by source IP - the list goes further

Pasting the destination IP into Google reveals it to be Facebook's. Further down the list is a TLS Client Hello packet, part of the handshake protocol. This packet's payload contains the following line:

Server Name: scontent.fykz2-1.fna.fbcdn.net

Fbcdn.net is a Facebook domain used to serve static content, like media, from a content delivery network (CDN).

Among the following packets, some destination IP's are Google's. Their protocol is TLS, the payload being "application data". A Google service that uses TLS by default is Gmail. We can then guess that the user is also using Gmail.

We can then deduce the user is interacting with Facebook, specifically logging into their account judging by the Client Hello packet, used to exchange security related information and using Gmail at the same time.

b. Include a screenshot of the sorted top summary panel for the above sender IP.

No.	Delta	Time	Source	Destination	Protocol	Length	Info
5524	0.000364	3.789468	172.18.20.47	31.13.80.5	TCP	142	39070 → 443 [ACK]
5525	0.000074	3.789542	172.18.20.47	31.13.80.5	TCP	158	[TCP Dup ACK 5524#
5528	0.000082	3.789711	172.18.20.47	31.13.80.5	TCP	142	[TCP Dup ACK 5524#
5529	0.000043	3.789754	172.18.20.47	31.13.80.5	TCP	158	[TCP Dup ACK 5524#
11665	0.000043	7.552752	172.18.20.47	31.13.80.5	TCP	142	39070 → 443 [ACK]
11666	0.000045	7.552797	172.18.20.47	31.13.80.5	TCP	142	39070 → 443 [ACK]
11667	0.000044	7.552841	172.18.20.47	66.102.1.188	TCP	146	48816 → 443 [ACK]
11690	0.000081	7.559689	172.18.20.47	31.13.71.34	TCP	154	36038 → 443 [ACK]
11691	0.000048	7.559737	172.18.20.47	66.102.1.188	TCP	154	[TCP Dup ACK 11667
11692	0.000043	7.559780	172.18.20.47	31.13.80.5	TCP	154	[TCP Dup ACK 11666
11693	0.000044	7.559824	172.18.20.47	31.13.71.34	TCP	158	[TCP Dup ACK 11690
11709	0.000068	7.564638	172.18.20.47	31.13.80.5	TLSv1.2	181	Application Data
11867	0.000042	7.611210	172.18.20.47	31.13.80.5	TLSv1.2	200	Application Data
12151	0.000043	7.733797	172.18.20.47	31.13.71.34	TLSv1.2	179	Application Data
12458	0.000049	7.944157	172.18.20.47	31.13.80.5	TCP	146	39070 → 443 [ACK]
12803	0.000049	8.109066	172.18.20.47	207.219.36.81	TCP	154	46804 → 443 [SYN]
12840	0.000049	8.119323	172.18.20.47	207.219.36.81	TCP	146	46804 → 443 [ACK]
12858	0.000259	8.125258	172.18.20.47	207.219.36.81	TLSv1.2	341	Client Hello

Figure 13: Sorted top summary panel for 172.18.20.47

c. Include a screenshot of one detailed packet from the sender IP that most helped you identify what that person was doing.

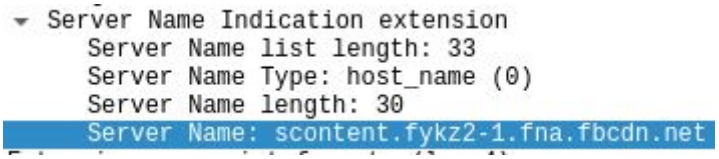


Figure 14: Packet 12858, Client Hello message sent to Facebook's fbcdn.net

Figure 15: Packet 11709, sent to a Google IP - payload is TLS application data

4. Which location was more secure?

We will define security here as higher levels packet protocols. In the public network recording, more than half of the packets have 802.11 protocol which is very low level. On the other hand, the home network recording all had higher level protocols: TCP, UDP, GQUIC, SSL or TLS.

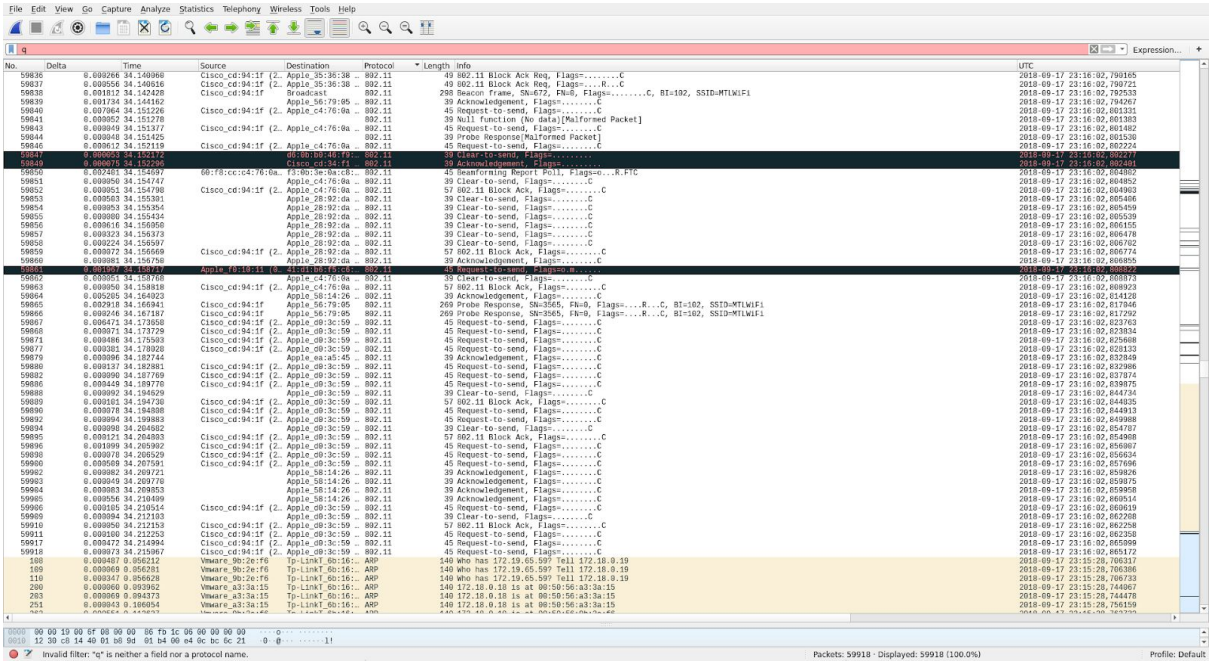


Figure 16: Public network recording - all white packets are in 802.11.
The white packets go back to the beginning of the list.

We can then conclude that the home network was secure.