

CS342– Lab 2

Ashish Kumar Barnawal (180123006) – MnC

Trace files: <http://cutt.ly/pf2CWSA>

The packet capturing for the Skype application happened on a Linux Mint (64 bit) system. (but analysis is carried out on Windows)

Question 1

List out all the protocols used by the application at different layers (only those which you can figure out from traces). Study and briefly describe their packet formats. Mention and explain the observed values for at least 5 fields of the packets of each layer. Example: Source or destination IP address, port number, Ethernet address, protocol number, etc.

Ans.

The following protocols could be figures out from the traces:

- UDP, TCP
- TLS, DNS, HTTP, STUN
- ICMP

The following are some fields of a packet (packet 51 in skype_signin.pcapng):

- Checksum: used to verify the integrity of the packet data (here values is 0x5b64)
- Windows size: The buffer space available for incoming data (size of sender's receive window, value as shown in screenshot)
- Source Port, Destination Port: Used to identify processes at the source and receiver's system (source port is 59540, dst port is 443. The destination port tells that HTTP protocol is being used)
- Source & Destination IP: to identify the source and destination computers on the network (Source address is 192.168.43.2 which is my PC's address and the destination is 40.90.22.185 which belongs to Microsoft Azure, makes sense as Microsoft owns Skype)
- TTL (Time to live): maintains a counter that gradually decrements to zero, at which point the datagram is discarded. This prevents the packet from looping endlessly (Here the value is 64 which is the default for Linux)
- Protocol: Specifies the protocol used (here TCP, this is obvious)
- Total Length: Specifies the total length of the packet in bytes (here 52 bytes)
- Version: the version of IP currently used (value is 4 here as we're using IP version 4)
- Source: Tp-LinkT_61:86:d8 (50:3e:aa:61:86:d8) (My PC's Tp-link usb wifi adapter)
- Destination: XiaomiCo_72:ad:8b (4c:49:e3:72:ad:8b) (My Phone's wifi MAC address, as I am using mobile hotspot)

```

> Frame 47: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface wlx503eaa6186d8, id 0
▼ Ethernet II, Src: Tp-LinkT_61:86:d8 (50:3e:aa:61:86:d8), Dst: XiaomiCo_72:ad:8b (4c:49:e3:72:ad:8b)
  > Destination: XiaomiCo_72:ad:8b (4c:49:e3:72:ad:8b)
  > Source: Tp-LinkT_61:86:d8 (50:3e:aa:61:86:d8)
  Type: IPv4 (0x0800)
▼ Internet Protocol Version 4, Src: 192.168.43.2, Dst: 40.90.22.185
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 52
  Identification: 0x57d6 (22486)
  > Flags: 0x4000, Don't fragment
  Fragment offset: 0
  Time to live: 64
  Protocol: TCP (6)
  Header checksum: 0xb830 [validation disabled]
  [Header checksum status: Unverified]
  Source: 192.168.43.2
  Destination: 40.90.22.185

```

```

> Source: Tp-LinkT_61:86:d8 (50:3e:aa:61:86:d8)
  Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.43.2, Dst: 40.90.22.185
▼ Transmission Control Protocol, Src Port: 59540, Dst Port: 443, Seq: 518, Ack: 3774, Len: 0
  Source Port: 59540
  Destination Port: 443
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 518 (relative sequence number)
  Sequence number (raw): 1985781851
  [Next sequence number: 518 (relative sequence number)]
  Acknowledgment number: 3774 (relative ack number)
  Acknowledgment number (raw): 2795179959
  1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x010 (ACK)
  Window size value: 501
  [Calculated window size: 64128]
  [Window size scaling factor: 128]
  Checksum: 0x5b64 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [SEQ/ACK analysis]
  > [Timestamps]

```

Question 2. Mention the important functionalities of the application as many as you can discover. (Two example functionalities for each application is given in Table 1). Explain which protocols are being used by which functionalities of the application. Give reason why those protocols are used for the functionalities.

Ans.

Some of the important functionalities of the application are:

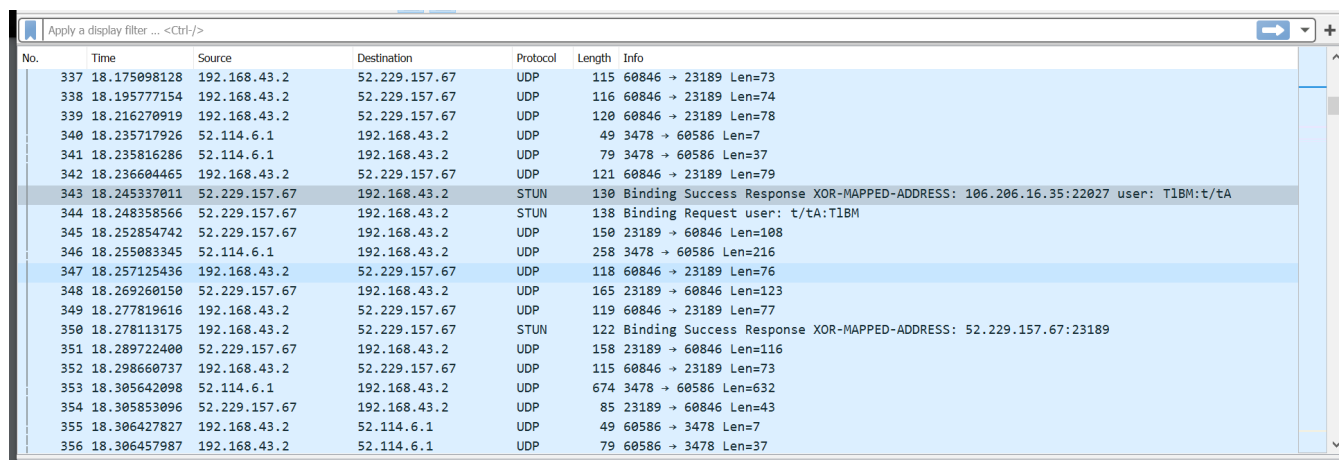
Functionality	Protocol Used	Explanation
Sign In	HTTPS: HTTP over TLS	A reliable and secure connection is needed for this step as sensitive information (passwords, emails) will be transferred. So HTTPS is used. DNS is used to get the IP of the skype server.
Get contacts	DNS	
Get account information		
Call Begin	STUN	STUN is used for traversal of UDP packets through NAT. As most of the communication during the call happens via UDP, this is less reliability about the packets reaching the destination. The STUN protocol gets over the Network Address Translators or NAT (the routers work as NAT) by asking a STUN server for the address it can be reached at. This address is used as destination by the application on the other side of the call.
In Call	UDP STUN	The call data (voice) is transferred over UDP. Here TCP is not used as it would cause lag. STUN is used repeatedly to keep the NAT connection alive with the server. In the trace file it can be seen that every couple 50-200 UDP packets, STUN packets are transmitted
Sign out	HTTPS: HTTP over TLS	Similar to Sign In. Sensitive information might be passing and we need reliability

Question 3. For any two functionalities of the application (mentioned in question 2), show the sequence of messages (attach screenshot) exchanged to achieve those functionalities. Explain those message sequences. Check whether there are any handshaking sequences in the messages, and briefly explain the reason

Ans.

In Call

Here as shown in the snapshots below, we see there are STUN packets after every 50-200 UDP packets.



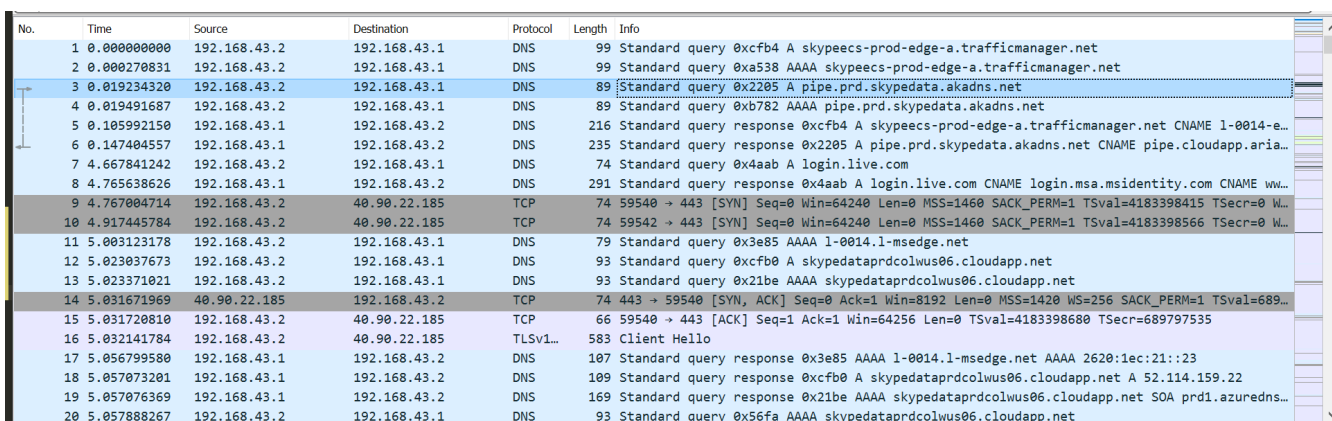
No.	Time	Source	Destination	Protocol	Length	Info
337	18.175098128	192.168.43.2	52.229.157.67	UDP	115	60846 → 23189 Len=73
338	18.195777154	192.168.43.2	52.229.157.67	UDP	116	60846 → 23189 Len=74
339	18.216270919	192.168.43.2	52.229.157.67	UDP	120	60846 → 23189 Len=78
340	18.235717926	52.114.6.1	192.168.43.2	UDP	49	3478 → 60586 Len=7
341	18.235816286	52.114.6.1	192.168.43.2	UDP	79	3478 → 60586 Len=37
342	18.236604465	192.168.43.2	52.229.157.67	UDP	121	60846 → 23189 Len=79
343	18.245337011	52.229.157.67	192.168.43.2	STUN	130	Binding Success Response XOR-MAPPED-ADDRESS: 106.206.16.35:22027 user: T1BM:t/ta
344	18.248358566	52.229.157.67	192.168.43.2	STUN	138	Binding Request user: t/ta:T1BM
345	18.252854742	52.229.157.67	192.168.43.2	UDP	150	23189 → 60846 Len=108
346	18.255083345	52.114.6.1	192.168.43.2	UDP	258	3478 → 60586 Len=216
347	18.257125436	192.168.43.2	52.229.157.67	UDP	118	60846 → 23189 Len=76
348	18.269260150	52.229.157.67	192.168.43.2	UDP	165	23189 → 60846 Len=123
349	18.277819616	192.168.43.2	52.229.157.67	UDP	119	60846 → 23189 Len=77
350	18.278113175	192.168.43.2	52.229.157.67	STUN	122	Binding Success Response XOR-MAPPED-ADDRESS: 52.229.157.67:23189
351	18.289722400	52.229.157.67	192.168.43.2	UDP	158	23189 → 60846 Len=116
352	18.298660737	192.168.43.2	52.229.157.67	UDP	115	60846 → 23189 Len=73
353	18.305642098	52.114.6.1	192.168.43.2	UDP	674	3478 → 60586 Len=632
354	18.305853096	52.229.157.67	192.168.43.2	UDP	85	23189 → 60846 Len=43
355	18.306427827	192.168.43.2	52.114.6.1	UDP	49	60586 → 3478 Len=7
356	18.306457987	192.168.43.2	52.114.6.1	UDP	79	60586 → 3478 Len=37

As explained in Q2, periodic STUN packets need to be sent periodically to keep the connection alive on the NAT (router).

The UDP packets sent in between contains the audio/video data in an encrypted format.

Sign in

The Sign In is done by first making a DNS queries for various Skype servers. The screenshot below shows this



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.43.2	192.168.43.1	DNS	99	Standard query 0xcfb4 A skypeecs-prod-edge-a.trafficmanager.net
2	0.000270831	192.168.43.2	192.168.43.1	DNS	99	Standard query 0xa538 AAAA skypeecs-prod-edge-a.trafficmanager.net
3	0.019234320	192.168.43.2	192.168.43.1	DNS	89	Standard query 0x2205 A pipe.prn.skypedata.akadns.net
4	0.019491687	192.168.43.2	192.168.43.1	DNS	89	Standard query 0xb782 AAAA pipe.prn.skypedata.akadns.net
5	0.105992150	192.168.43.1	192.168.43.2	DNS	216	Standard query response 0xcfb4 A skypeecs-prod-edge-a.trafficmanager.net CNAME 1-0014-e...
6	0.147404557	192.168.43.1	192.168.43.2	DNS	235	Standard query response 0x2205 A pipe.prn.skypedata.akadns.net CNAME pipe.cloudapp.aria...
7	4.667841242	192.168.43.2	192.168.43.1	DNS	74	Standard query 0x4aab A login.live.com
8	4.765638626	192.168.43.1	192.168.43.2	DNS	291	Standard query response 0x4aab A login.live.com CNAME login.msa.msidentity.com CNAME ww...
9	4.767004714	192.168.43.2	40.90.22.185	TCP	74	59540 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=4183398415 TSecr=0 W...
10	4.917445784	192.168.43.2	40.90.22.185	TCP	74	59542 → 443 [ACK] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=4183398566 TSecr=0 W...
11	5.003123178	192.168.43.2	192.168.43.1	DNS	79	Standard query 0x3e85 AAAA 1-0014.1-msedge.net
12	5.023037673	192.168.43.2	192.168.43.1	DNS	93	Standard query 0xcfb0 A skypedataprdcolwus06.cloudapp.net
13	5.023371021	192.168.43.2	192.168.43.1	DNS	93	Standard query 0x21be AAAA skypedataprdcolwus06.cloudapp.net
14	5.031671969	40.90.22.185	192.168.43.2	TCP	74	443 → 59540 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1420 WS=256 SACK_PERM=1 TSval=689...
15	5.031720810	192.168.43.2	40.90.22.185	TCP	66	59540 → 443 [ACK] Seq=0 Ack=1 Win=64256 Len=0 TSval=4183398680 TSecr=689797535
16	5.032141784	192.168.43.2	40.90.22.185	TLSv1...	583	Client Hello
17	5.056799580	192.168.43.1	192.168.43.2	DNS	107	Standard query response 0x3e85 AAAA 1-0014.1-msedge.net AAAA 2620:1ec:21::23
18	5.057073201	192.168.43.1	192.168.43.2	DNS	109	Standard query response 0xcfb0 A skypedataprdcolwus06.cloudapp.net A 52.114.159.22
19	5.057076369	192.168.43.1	192.168.43.2	DNS	169	Standard query response 0x21be AAAA skypedataprdcolwus06.cloudapp.net SOA prd1.azuredns...
20	5.057888267	192.168.43.2	192.168.43.1	DNS	93	Standard query 0x56fa AAAA skypedataprdcolwus06.cloudapp.net

Then TLS handshake is done.

skype_signin.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
25	5.116874891	Tp-LinkT_61:86:d8	XiaomiCo_72:ad:8b	ARP	42	192.168.43.2 is at 50:3e:aa:61:86:d8
26	5.174381581	192.168.43.2	192.168.43.2	TCP	66	443 → 37060 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1420 WS=256 SACK_PERM=1
27	5.174447673	192.168.43.2	13.107.42.23	TCP	54	37060 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0
28	5.175228470	192.168.43.2	13.107.42.23	TLSv1...	311	Client Hello
29	5.191418893	40.90.22.185	192.168.43.2	TCP	74	443 → 59542 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1420 WS=256 SACK_PERM=1 TSval=243...
30	5.191468854	192.168.43.2	40.90.22.185	TCP	66	59542 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=4183398840 TSecr=243395866
31	5.191787921	192.168.43.2	40.90.22.185	TLSv1...	583	Client Hello
32	5.257155424	13.107.42.23	192.168.43.2	TCP	54	443 → 37060 [ACK] Seq=1 Ack=258 Win=525056 Len=0
33	5.257674406	13.107.42.23	192.168.43.2	TCP	1474	443 → 37060 [ACK] Seq=1 Ack=258 Win=525056 Len=1420 [TCP segment of a reassembled PDU]
34	5.257695640	192.168.43.2	13.107.42.23	TCP	54	37060 → 443 [ACK] Seq=258 Ack=1421 Win=64128 Len=0
35	5.257731343	13.107.42.23	192.168.43.2	TCP	1474	443 → 37060 [ACK] Seq=1421 Ack=258 Win=525056 Len=1420 [TCP segment of a reassembled PD...
36	5.257741396	192.168.43.2	13.107.42.23	TCP	54	37060 → 443 [ACK] Seq=258 Ack=2841 Win=63360 Len=0
37	5.258002216	13.107.42.23	192.168.43.2	TCP	1474	443 → 37060 [ACK] Seq=2841 Ack=258 Win=525056 Len=1420 [TCP segment of a reassembled PD...
38	5.258017656	192.168.43.2	13.107.42.23	TCP	54	37060 → 443 [ACK] Seq=258 Ack=4261 Win=64128 Len=0
39	5.260649006	13.107.42.23	192.168.43.2	TLSv1...	755	Server Hello, Certificate, Server Key Exchange, Server Hello Done
40	5.260664152	192.168.43.2	13.107.42.23	TCP	54	37060 → 443 [ACK] Seq=258 Ack=4962 Win=64128 Len=0

> Extension: status_request (len=5)
 > Extension: signature_algorithms (len=20)
 Type: signature_algorithms (13)
 Length: 20
 Signature Hash Algorithms Length: 18
 > Signature Hash Algorithms (9 algorithms)
 > Signature Algorithm: ecdsa_secp256r1_sha256 (0x0403)
 > Signature Algorithm: rsa_pss_rsae_sha256 (0x0804)
 > Signature Algorithm: rsa_pkcs1_sha256 (0x0401)
 > Signature Algorithm: ecdsa_secp384r1_sha384 (0x0503)
 > Signature Algorithm: rsa_pss_rsae_sha384 (0x0805)
 > Signature Algorithm: rsa_pkcs1_sha384 (0x0501)

0000 4c 49 e3 72 ad 8b 50 3e aa 61 86 d8 08 00 45 00 LI r...P> a...E-

Type here to search

5:53 PM 28-Sep-20

The client Hello packet as in screenshot shows this. This packet contains the hash algorithms, cipher suites etc. supported by the client. Then several packet exchanges happen b/w the client and the server to complete the transaction and establish an encrypted connection. Then encrypted data is transferred over TCP as shown by multiple TCP packets.

Yes, there are handshaking sequences in the messages. The client hello is described previously. Some other messages are shown in the screenshot below:

Packet list Narrow & Wide Case sensitive String server hello Find Cancel

No.	Time	Source	Destination	Protocol	Length	Info
30	5.191468854	192.168.43.2	40.90.22.185	TCP	66	59542 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=4183398840 TSecr=243395866
31	5.191787921	192.168.43.2	40.90.22.185	TLSv1...	583	Client Hello
32	5.257155424	13.107.42.23	192.168.43.2	TCP	54	443 → 37060 [ACK] Seq=1 Ack=258 Win=525056 Len=0
33	5.257674406	13.107.42.23	192.168.43.2	TCP	1474	443 → 37060 [ACK] Seq=1 Ack=258 Win=525056 Len=1420 [TCP segment of a reassembled PDU]
34	5.257695640	192.168.43.2	13.107.42.23	TCP	54	37060 → 443 [ACK] Seq=258 Ack=1421 Win=64128 Len=0
35	5.257731343	13.107.42.23	192.168.43.2	TCP	1474	443 → 37060 [ACK] Seq=1421 Ack=258 Win=525056 Len=1420 [TCP segment of a reassembled PD...
36	5.257741396	192.168.43.2	13.107.42.23	TCP	54	37060 → 443 [ACK] Seq=258 Ack=2841 Win=63360 Len=0
37	5.258002216	13.107.42.23	192.168.43.2	TCP	1474	443 → 37060 [ACK] Seq=2841 Ack=258 Win=525056 Len=1420 [TCP segment of a reassembled PD...
38	5.258017656	192.168.43.2	13.107.42.23	TCP	54	37060 → 443 [ACK] Seq=258 Ack=4261 Win=64128 Len=0
39	5.260649006	13.107.42.23	192.168.43.2	TLSv1...	755	Server Hello, Certificate, Server Key Exchange, Server Hello Done
40	5.260664152	192.168.43.2	13.107.42.23	TCP	54	37060 → 443 [ACK] Seq=258 Ack=4962 Win=64128 Len=0
41	5.264582825	192.168.43.2	13.107.42.23	TLSv1...	212	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
42	5.297273735	40.90.22.185	192.168.43.2	TCP	1474	443 → 59540 [ACK] Seq=1 Ack=518 Win=263168 Len=1408 TSval=689797807 TSecr=4183398680 [T...
43	5.297317511	192.168.43.2	40.90.22.185	TCP	66	59540 → 443 [ACK] Seq=518 Ack=1409 Win=64128 Len=0 TSval=4183398945 TSecr=689797807
44	5.297632566	40.90.22.185	192.168.43.2	TCP	1474	443 → 59540 [ACK] Seq=1409 Ack=518 Win=263168 Len=1408 TSval=689797807 TSecr=4183398680...

> Frame 39: 755 bytes on wire (6040 bits), 755 bytes captured (6040 bits) on interface wlx503aaa6186d8, id 0

Question 4. Calculate the following statistics from your traces while performing experiments at three different times (morning, afternoon, night) of the day: a) Throughput, b) RTT, c) Packet size, d) Number of packets lost, e) Number of UDP & TCP packets, f) Number of responses received with respect to one request sent. Report the observed values in your answer, preferably using tables.

Ans.

Time of day	Throughput	RTT	Packet Size	Number of packets lost	No.of UDP Packets	No. of TCP packets	Responses per request
4 pm	145kb/s	113ms	316B	100	1827	1949	0.657
12 pm	162kb/s	83ms	323B	9	1794	1745	0.740
6pm	138kb/s	88ms	307B	48	2455	1492	0.818

Question 5. Check whether the content is being sent/fetched by the application to/from the same or different destination(s)/source(s) during the three different times of the day used in question 4. If multiple destinations /sources exist, list out their IP addresses, and explain the reason behind this

Ans.

Only some of the addresses have been listed

29 Sept 6pm : 40.90.22.185, 13.107.3.128, 52.114.14.47, 20.185.212.106, 13.76.142.69

29 Sept 12pm: 13.107.42.23, 52.114.76.37, 40.90.22.187, 13.107.3.128, 13.76.142.98

28 Sept 4pm: 13.107.42.23, 52.114.128.70, 40.90.22.192, 13.107.3.128, 52.229.135.178

We can find both common and different IP addresses in the above list. The reason for this is that at different times of the day, different servers maybe allocated to us depending on a variety of factors like traffic, latency, region etc.