Aryan Gupta
ECGR-3123-001
Dr. Tao Han
May 1, 2018

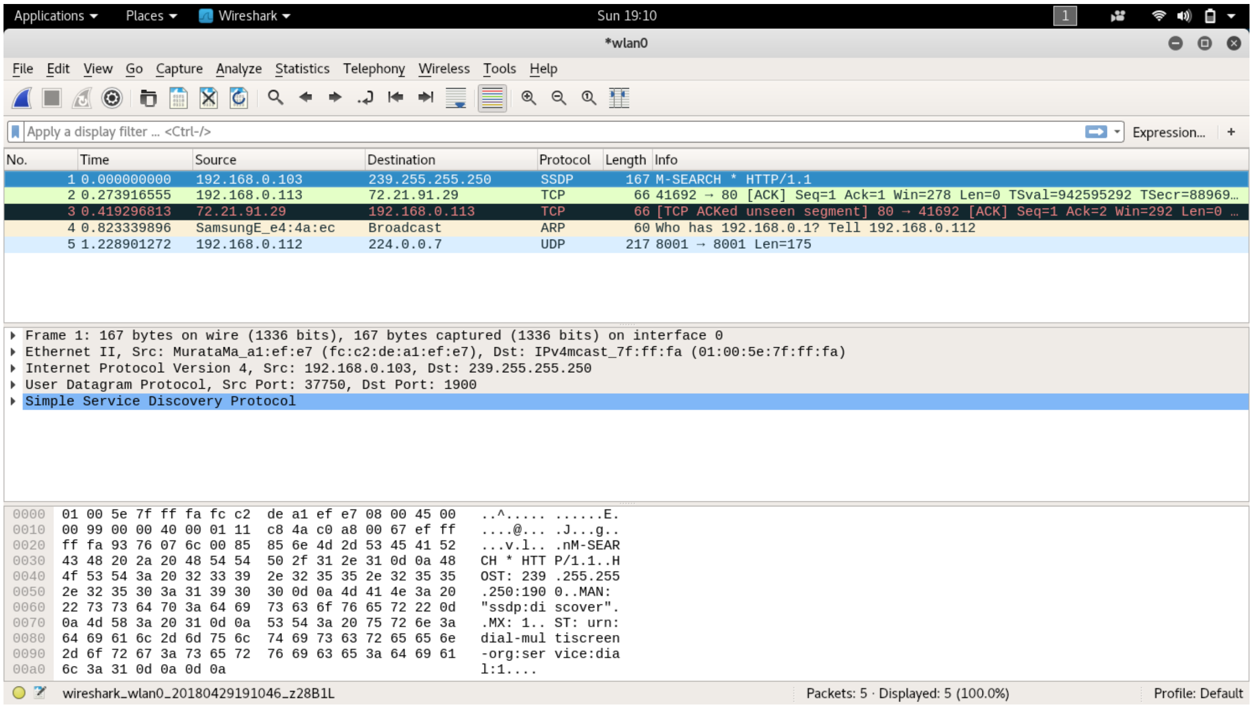
Wireshark Packet Sniffing Project

Computer Details

This report was done on a Lenovo Ideapad 320-15IAP with Kali Linux 4.14.0 at my home network. I have Spectrum as my ISP.

Wireshark

Wireshark is an application that captures packets that are sent on a wireless network. For example, if my computer wants to ping another computer, it will send an ICMP packet to the client and wait for a response. Wireshark can pick up this packet and display it in a table. The UI is very simple to use; first you must choose a device to start capturing packets from. I chose to use the wireless LAN port because there are some issues with trying to use Ethernet LAN. One of these issues is that if you have a switch and not a hub, you will only receive packets that belong to you or are addressed to you. In a Wi-Fi network, you can capture all packets that are being sent in the air, ones that belong to you and ones that don't.


Applications
Places
Wireshark
Sun 19:10
1
*wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
+

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.103	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
2	0.273916555	192.168.0.113	72.21.91.29	TCP	66	41692 → 80 [ACK] Seq=1 Ack=1 Win=278 Len=0 TSval=942595292 TSecr=88969…
3	0.419296813	72.21.91.29	192.168.0.113	TCP	66	[TCP ACKed unseen segment] 80 → 41692 [ACK] Seq=1 Ack=2 Win=292 Len=0 …
4	0.823339896	SamsungE_e4:4a:ec	Broadcast	ARP	60	Who has 192.168.0.1? Tell 192.168.0.112
5	1.228901272	192.168.0.112	224.0.0.7	UDP	217	8001 → 8001 Len=175

```
▸ Frame 1: 167 bytes on wire (1336 bits), 167 bytes captured (1336 bits) on interface 0
▸ Ethernet II, Src: MurataMa_a1:ef:e7 (fc:c2:de:a1:ef:e7), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
▸ Internet Protocol Version 4, Src: 192.168.0.103, Dst: 239.255.255.250
▸ User Datagram Protocol, Src Port: 37750, Dst Port: 1900
▸ Simple Service Discovery Protocol
```

```
0000  01 00 5e 7f ff fa fc c2  de a1 ef e7 08 00 45 00   ..^..... ......E.
0010  00 99 00 00 40 00 01 11  c8 4a c0 a8 00 67 ef ff   ....@... .J...g..
0020  ff fa 93 76 07 6c 00 85  85 6e 4d 2d 53 45 41 52   ...v.l.. .nM-SEAR
0030  43 48 20 2a 20 48 54 54  50 2f 31 2e 31 0d 0a 48   CH * HTT P/1.1..H
0040  4f 53 54 3a 20 32 33 39  2e 32 35 35 2e 32 35 35   OST: 239 .255.255
0050  2e 32 35 30 3a 31 39 30  30 0d 0a 4d 41 4e 3a 20   .250:190 0..MAN:
0060  22 73 73 64 70 3a 64 69  73 63 6f 76 65 72 22 0d   "ssdp:di scover".
0070  0a 4d 58 3a 20 31 0d 0a  53 54 3a 20 75 72 6e 3a   .MX: 1.. ST: urn:
0080  64 69 61 6c 2d 6d 75 6c  74 69 73 63 72 65 65 6e   dial-mul tiscreen
0090  2d 6f 72 67 3a 73 65 72  76 69 63 65 3a 64 69 61   -org:ser vice:dia
00a0  6c 3a 31 0d 0a 0d 0a                               l:1....
```

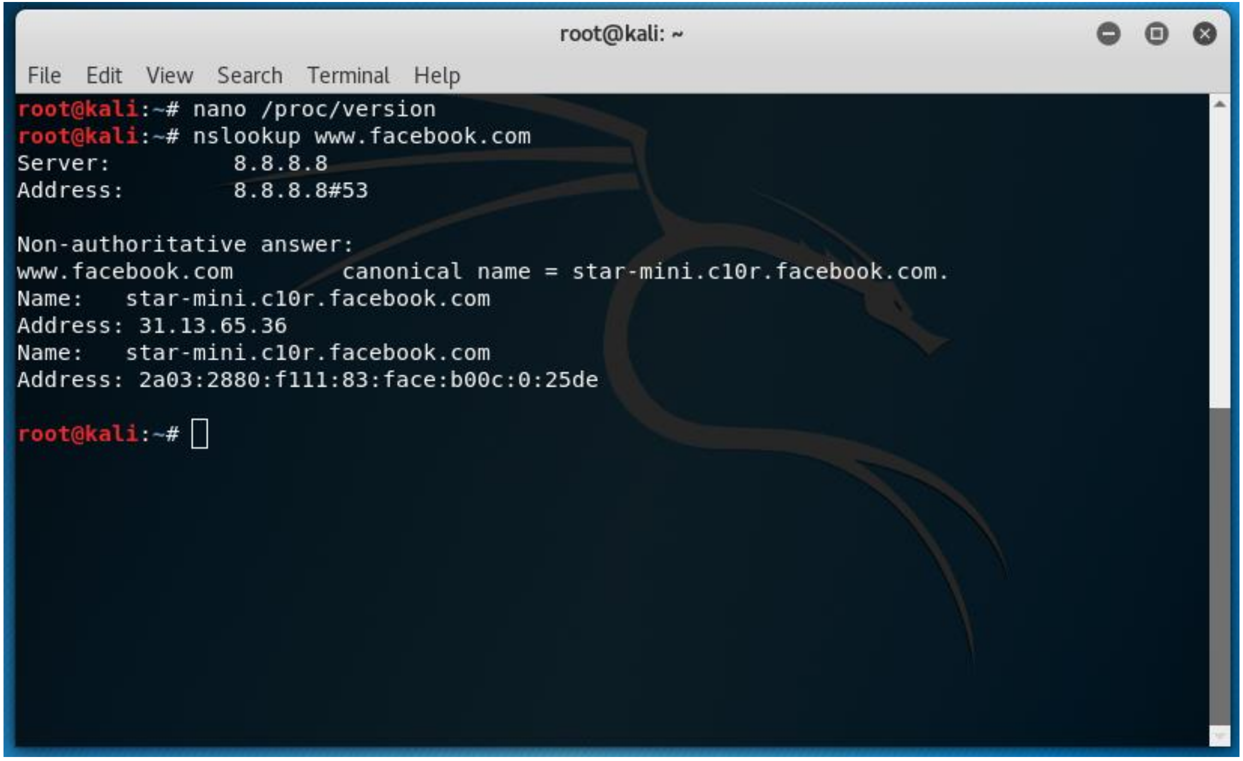
wireshark_wlan0_20180429191046_z28B1L   Packets: 5 · Displayed: 5 (100.0%)   Profile: Default

As you can see in this image, as soon as I start the capturing process I get many packets being sent over the air. One of these packets, packet number 4, belongs to my Dad's Samsung phone.

Facebook

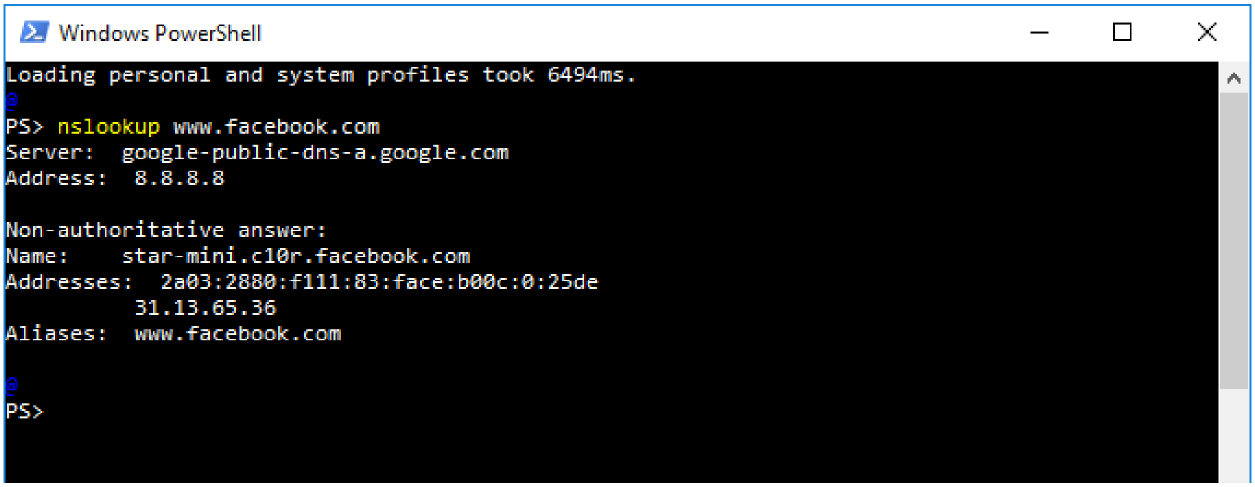
One of the first things I did was figure out my IP address. Because I have a DHCP server running, my IP will be dynamically decided by the server.

Aryan Gupta
ECGR-3123-001
Dr. Tao Han
May 1, 2018



```
root@kali:~# nano /proc/version
root@kali:~# nslookup www.facebook.com
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
www.facebook.com        canonical name = star-mini.c10r.facebook.com.
Name:   star-mini.c10r.facebook.com
Address: 31.13.65.36
Name:   star-mini.c10r.facebook.com
Address: 2a03:2880:f111:83:face:b00c:0:25de

root@kali:~# 
```

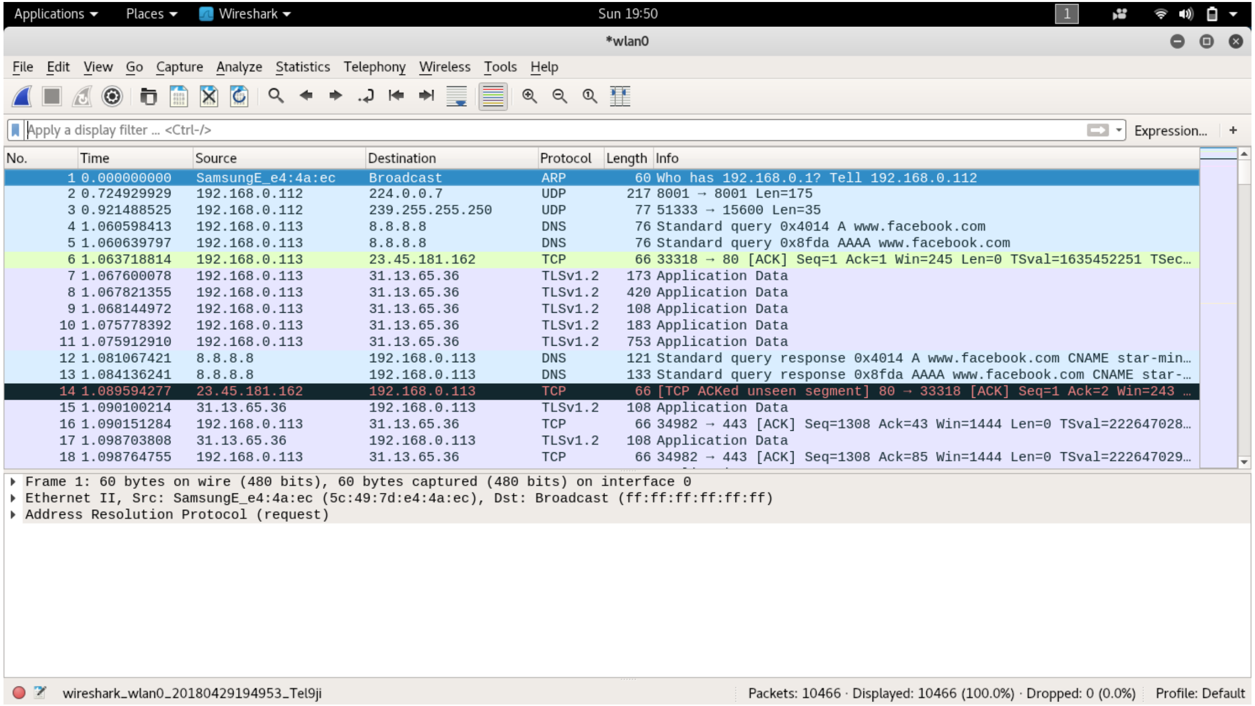
From this image you can see that I have the IP address: 192.168.0.113. Another thing I did was figure out the IP address of Facebook so we can filter out these 2 IP addresses.



```
Loading personal and system profiles took 6494ms.
@
PS> nslookup www.facebook.com
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
Name:    star-mini.c10r.facebook.com
Addresses:  2a03:2880:f111:83:face:b00c:0:25de
          31.13.65.36
Aliases:  www.facebook.com

@
PS>
```

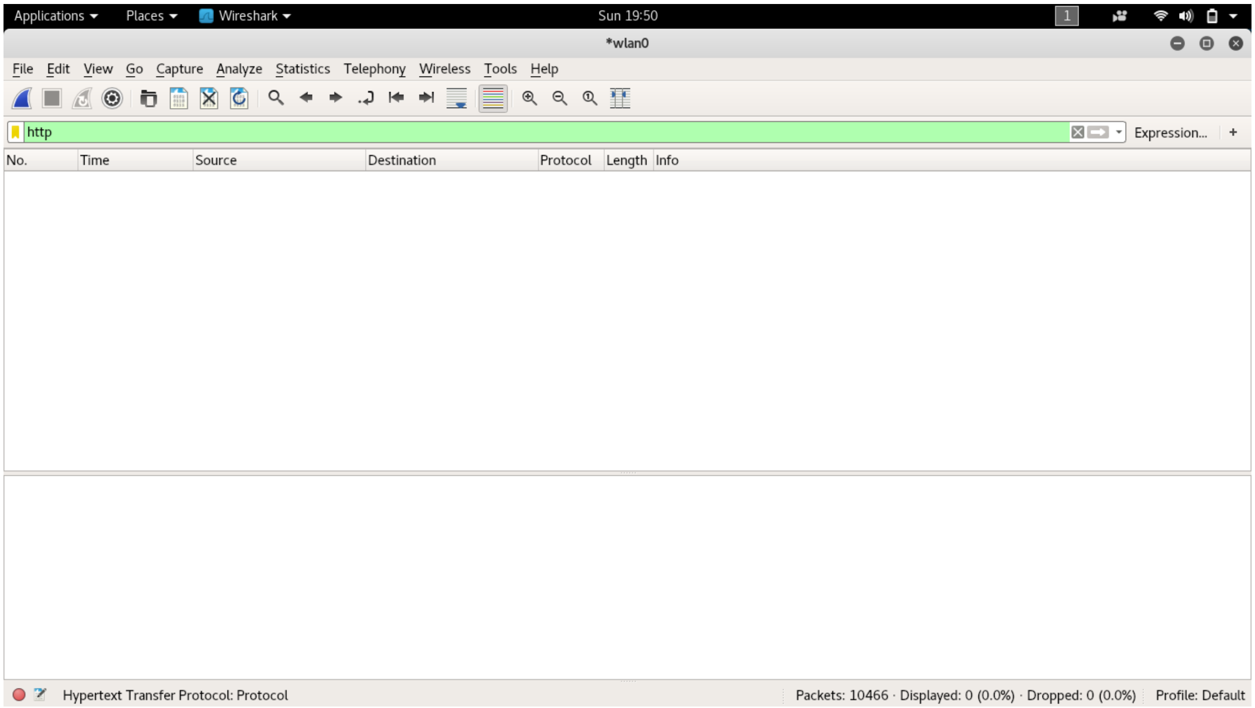
From this image, you can see that the DNS resolves Facebook to IP: 31.13.65.36. (I was unable to find the screenshot; however, I took one my Windows partition)

Aryan Gupta
ECGR-3123-001
Dr. Tao Han
May 1, 2018


Applications
Places
Wireshark
Sun 19:50
*wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…

```
No.   Time          Source              Destination       Protocol Length Info
    1 0.000000000   SamsungE_e4:4a:ec   Broadcast         ARP         60 Who has 192.168.0.1? Tell 192.168.0.112
    2 0.724929929   192.168.0.112       224.0.0.7         UDP        217 8001 → 8001 Len=175
    3 0.921488525   192.168.0.112       239.255.255.250   UDP         77 51333 → 15600 Len=35
    4 1.060598413   192.168.0.113       8.8.8.8           DNS         76 Standard query 0x4014 A www.facebook.com
    5 1.060639797   192.168.0.113       8.8.8.8           DNS         76 Standard query 0x8fda AAAA www.facebook.com
    6 1.063718814   192.168.0.113       23.45.181.162     TCP         66 33318 → 80 [ACK] Seq=1 Ack=1 Win=245 Len=0 TSval=1635452251 TSec…
    7 1.067600078   192.168.0.113       31.13.65.36       TLSv1.2    173 Application Data
    8 1.067821355   192.168.0.113       31.13.65.36       TLSv1.2    420 Application Data
    9 1.068144972   192.168.0.113       31.13.65.36       TLSv1.2    108 Application Data
   10 1.075778392   192.168.0.113       31.13.65.36       TLSv1.2    183 Application Data
   11 1.075912910   192.168.0.113       31.13.65.36       TLSv1.2    753 Application Data
   12 1.081067421   8.8.8.8             192.168.0.113     DNS        121 Standard query response 0x4014 A www.facebook.com CNAME star-min…
   13 1.084136241   8.8.8.8             192.168.0.113     DNS        133 Standard query response 0x8fda AAAA www.facebook.com CNAME star-…
   14 1.089594277   23.45.181.162       192.168.0.113     TCP         66 [TCP ACKed unseen segment] 80 → 33318 [ACK] Seq=1 Ack=2 Win=243 …
   15 1.090100214   31.13.65.36         192.168.0.113     TLSv1.2    108 Application Data
   16 1.090151284   192.168.0.113       31.13.65.36       TCP         66 34982 → 443 [ACK] Seq=1308 Ack=43 Win=1444 Len=0 TSval=222647028…
   17 1.098703808   31.13.65.36         192.168.0.113     TLSv1.2    108 Application Data
   18 1.098764755   192.168.0.113       31.13.65.36       TCP         66 34982 → 443 [ACK] Seq=1308 Ack=85 Win=1444 Len=0 TSval=222647029…

▸ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▸ Ethernet II, Src: SamsungE_e4:4a:ec (5c:49:7d:e4:4a:ec), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▸ Address Resolution Protocol (request)

wireshark_wlan0_20180429194953_Tel9ji    Packets: 10466 · Displayed: 10466 (100.0%) · Dropped: 0 (0.0%)    Profile: Default
```

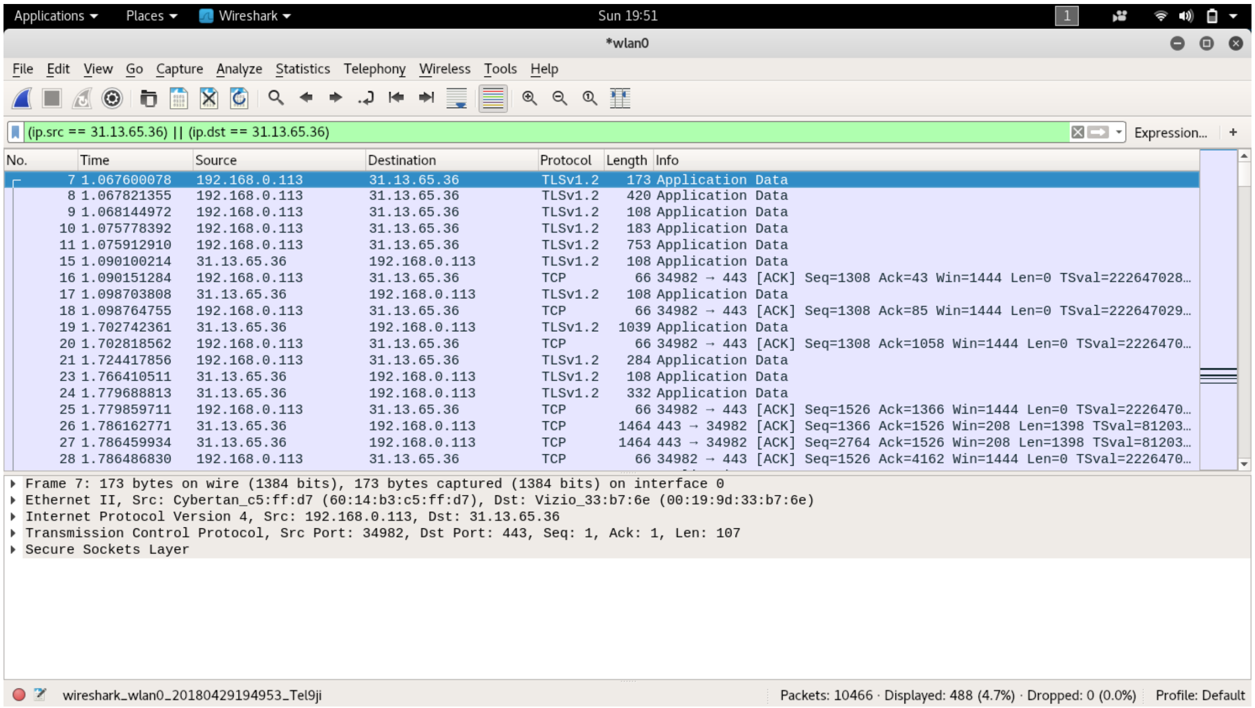
After starting restarting the capturing process and logging in to my Facebook account, we see a lot of traffic between my computer and Facebook's server.


Applications
Places
Wireshark
Sun 19:50
*wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…

```
http

No.   Time          Source              Destination       Protocol Length Info

Hypertext Transfer Protocol: Protocol    Packets: 10466 · Displayed: 0 (0.0%) · Dropped: 0 (0.0%)    Profile: Default
```

Filter	Meaning
http	Only TCP packets that use the http protocol are displayed

By filtering all packets so that only HTTP packets are shown we get a peculiar result: no packets. After some googling and rereading the prompt, I realized that because all the traffic between my computer and Facebook's server is encrypted, no HTTP packets were exchanged. I googled many ways to give Wireshark my private RSA key so that it could unencrypt the packets and see the data, however my research shows that after Firefox version 48 this was disabled.


Applications
Places
Wireshark
Sun 19:51
*wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
+

```
(ip.src == 31.13.65.36) || (ip.dst == 31.13.65.36)
```

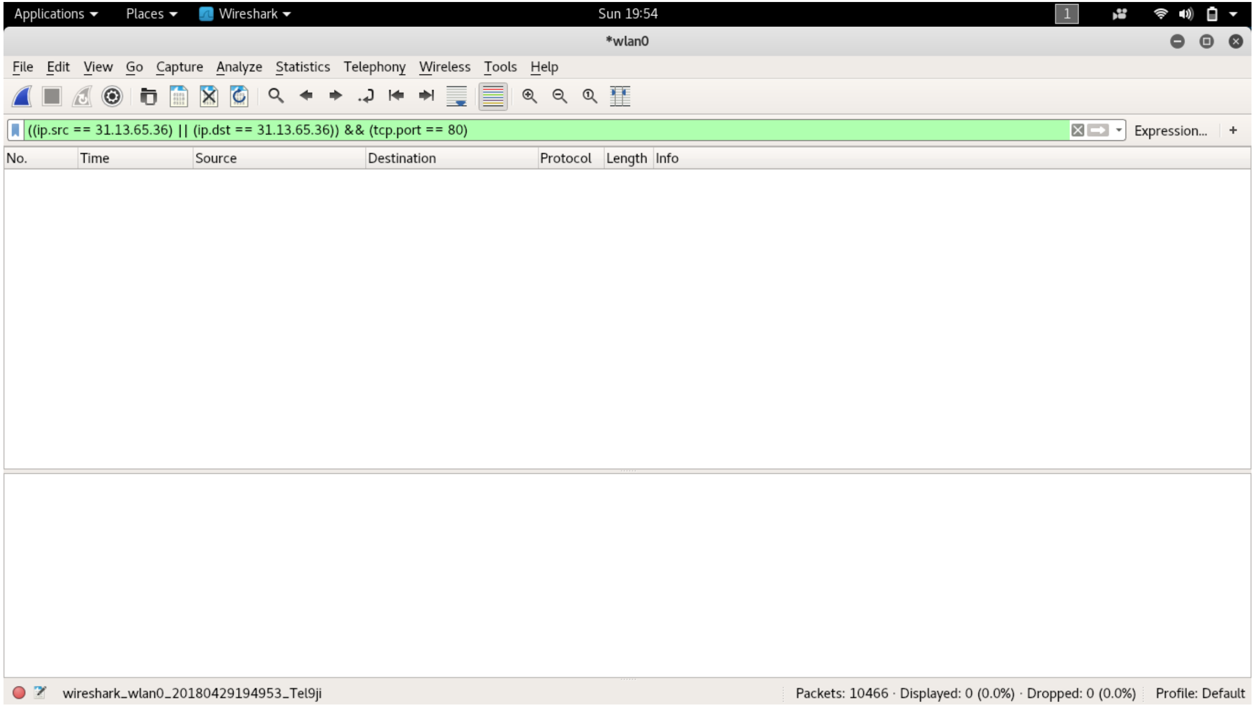
```
No.  Time         Source           Destination      Protocol  Length Info
   7 1.067600078  192.168.0.113    31.13.65.36      TLSv1.2    173 Application Data
   8 1.067821355  192.168.0.113    31.13.65.36      TLSv1.2    420 Application Data
   9 1.068144972  192.168.0.113    31.13.65.36      TLSv1.2    108 Application Data
  10 1.075778392  192.168.0.113    31.13.65.36      TLSv1.2    183 Application Data
  11 1.075912910  192.168.0.113    31.13.65.36      TLSv1.2    753 Application Data
  15 1.090100214  31.13.65.36      192.168.0.113    TLSv1.2    108 Application Data
  16 1.090151284  192.168.0.113    31.13.65.36      TCP         66 34982 → 443 [ACK] Seq=1308 Ack=43 Win=1444 Len=0 TSval=22264702８…
  17 1.098703808  31.13.65.36      192.168.0.113    TLSv1.2    108 Application Data
  18 1.098764755  192.168.0.113    31.13.65.36      TCP         66 34982 → 443 [ACK] Seq=1308 Ack=85 Win=1444 Len=0 TSval=222647029…
  19 1.702742361  31.13.65.36      192.168.0.113    TLSv1.2   1039 Application Data
  20 1.702818562  192.168.0.113    31.13.65.36      TCP         66 34982 → 443 [ACK] Seq=1308 Ack=1058 Win=1444 Len=0 TSval=2226470…
  21 1.724417856  192.168.0.113    31.13.65.36      TLSv1.2    284 Application Data
  23 1.766410511  31.13.65.36      192.168.0.113    TLSv1.2    108 Application Data
  24 1.779688813  31.13.65.36      192.168.0.113    TLSv1.2    332 Application Data
  25 1.779859711  192.168.0.113    31.13.65.36      TCP         66 34982 → 443 [ACK] Seq=1526 Ack=1366 Win=1444 Len=0 TSval=2226470…
  26 1.786162771  31.13.65.36      192.168.0.113    TCP       1464 443 → 34982 [ACK] Seq=1366 Ack=1526 Win=208 Len=1398 TSval=81203…
  27 1.786459934  31.13.65.36      192.168.0.113    TCP       1464 443 → 34982 [ACK] Seq=2764 Ack=1526 Win=208 Len=1398 TSval=81203…
  28 1.786486830  192.168.0.113    31.13.65.36      TCP         66 34982 → 443 [ACK] Seq=1526 Ack=4162 Win=1444 Len=0 TSval=2226470…
```

```
▸ Frame 7: 173 bytes on wire (1384 bits), 173 bytes captured (1384 bits) on interface 0
▸ Ethernet II, Src: Cybertan_c5:ff:d7 (60:14:b3:c5:ff:d7), Dst: Vizio_33:b7:6e (00:19:9d:33:b7:6e)
▸ Internet Protocol Version 4, Src: 192.168.0.113, Dst: 31.13.65.36
▸ Transmission Control Protocol, Src Port: 34982, Dst Port: 443, Seq: 1, Ack: 1, Len: 107
▸ Secure Sockets Layer
```

```
wireshark_wlan0_20180429194953_Tel9ji          Packets: 10466 · Displayed: 488 (4.7%) · Dropped: 0 (0.0%)   Profile: Default
```

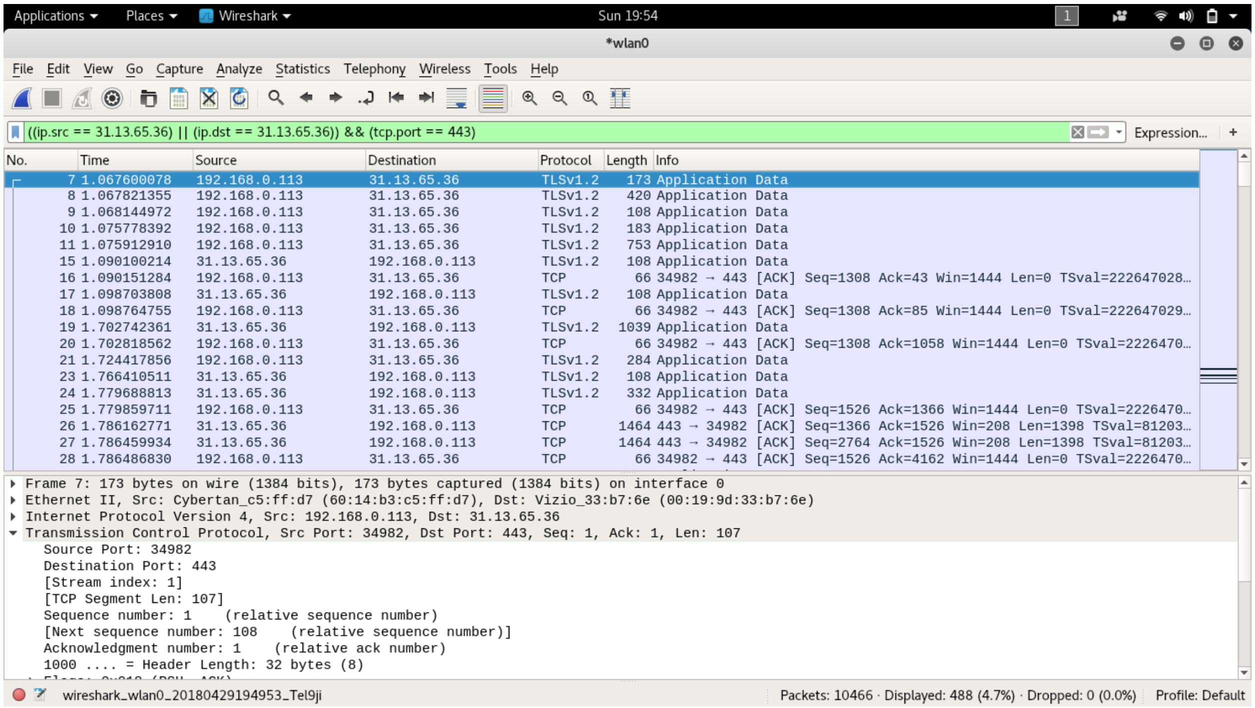
Filter	**Meaning**
(ip.src == 31.13.65.36) || (ip.dst == 31.13.65.36)	Only packets that are sent from Facebook or sent to Facebook are displayed

I then chose to filter by IP address rather than protocol and we see all the packets between my computer and Facebook. As you can see, there are many TLS packets being sent, this is the new encryption standard after SSL.


Applications
Places
Wireshark
Sun 19:54
*wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
+

```
((ip.src == 31.13.65.36) || (ip.dst == 31.13.65.36)) && (tcp.port == 80)

No.     Time          Source              Destination         Protocol  Length Info

wireshark_wlan0_20180429194953_Tel9ji      Packets: 10466 · Displayed: 0 (0.0%) · Dropped: 0 (0.0%)    Profile: Default
```

Filter	**Meaning**
((ip.src == 31.13.65.36) || (ip.dst == 31.13.65.36)) && (tcp.port == 80)	Only packets that are sent from Facebook or sent to Facebook and are addressed to port 80 are displayed

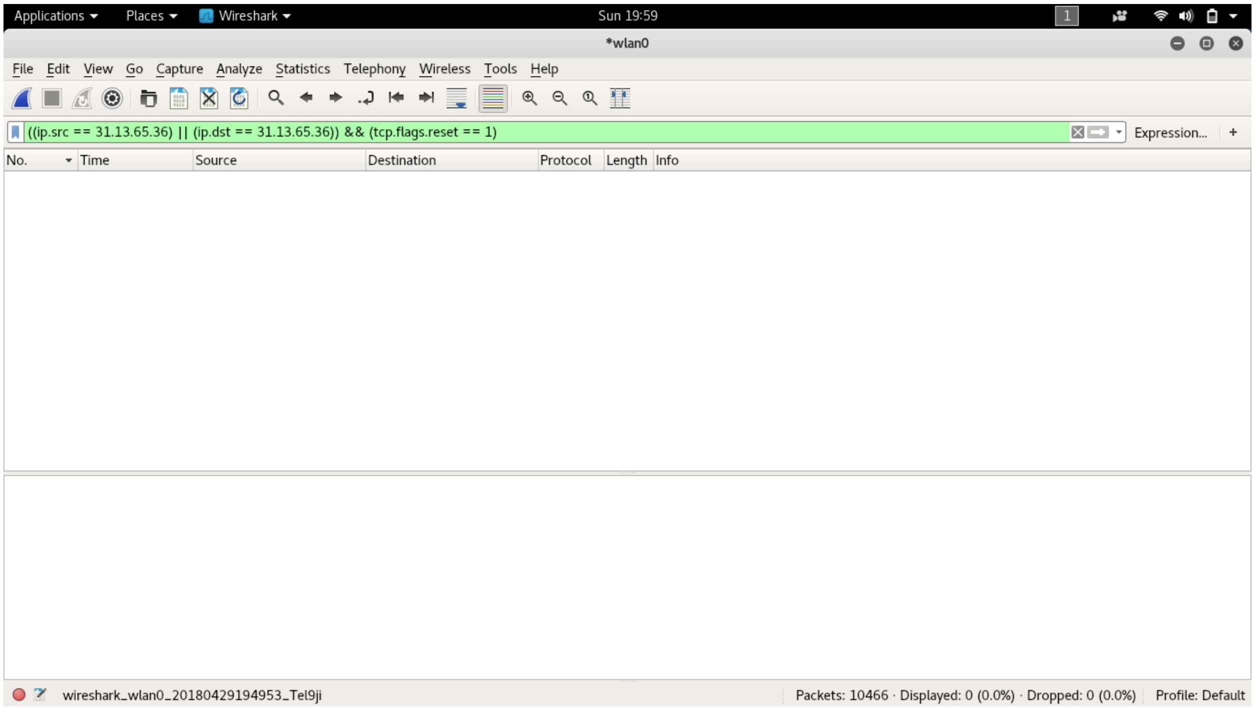

Applications
Places
Wireshark
Sun 19:54
*wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
+

```
((ip.src == 31.13.65.36) || (ip.dst == 31.13.65.36)) && (tcp.port == 443)

No.     Time          Source              Destination         Protocol  Length Info
      7 1.067600078   192.168.0.113       31.13.65.36         TLSv1.2    173 Application Data
      8 1.067821355   192.168.0.113       31.13.65.36         TLSv1.2    420 Application Data
      9 1.068144972   192.168.0.113       31.13.65.36         TLSv1.2    108 Application Data
     10 1.075778392   192.168.0.113       31.13.65.36         TLSv1.2    183 Application Data
     11 1.075912910   192.168.0.113       31.13.65.36         TLSv1.2    753 Application Data
     15 1.090100214   31.13.65.36         192.168.0.113       TLSv1.2    108 Application Data
     16 1.090151284   192.168.0.113       31.13.65.36         TCP         66 34982 → 443 [ACK] Seq=1308 Ack=43 Win=1444 Len=0 TSval=222647028…
     17 1.098703808   31.13.65.36         192.168.0.113       TLSv1.2    108 Application Data
     18 1.098764755   192.168.0.113       31.13.65.36         TCP         66 34982 → 443 [ACK] Seq=1308 Ack=85 Win=1444 Len=0 TSval=222647029…
     19 1.702742361   31.13.65.36         192.168.0.113       TLSv1.2   1039 Application Data
     20 1.702818562   192.168.0.113       31.13.65.36         TCP         66 34982 → 443 [ACK] Seq=1308 Ack=1058 Win=1444 Len=0 TSval=2226470…
     21 1.724417856   192.168.0.113       31.13.65.36         TLSv1.2    284 Application Data
     23 1.766410511   31.13.65.36         192.168.0.113       TLSv1.2    108 Application Data
     24 1.779688813   31.13.65.36         192.168.0.113       TLSv1.2    332 Application Data
     25 1.779859711   192.168.0.113       31.13.65.36         TCP         66 34982 → 443 [ACK] Seq=1526 Ack=1366 Win=1444 Len=0 TSval=2226470…
     26 1.786162771   31.13.65.36         192.168.0.113       TCP       1464 443 → 34982 [ACK] Seq=1366 Ack=1526 Win=208 Len=1398 TSval=81203…
     27 1.786459934   31.13.65.36         192.168.0.113       TCP       1464 443 → 34982 [ACK] Seq=2764 Ack=1526 Win=208 Len=1398 TSval=81203…
     28 1.786486830   192.168.0.113       31.13.65.36         TCP         66 34982 → 443 [ACK] Seq=1526 Ack=4162 Win=1444 Len=0 TSval=2226470…

▸ Frame 7: 173 bytes on wire (1384 bits), 173 bytes captured (1384 bits) on interface 0
▸ Ethernet II, Src: Cybertan_c5:ff:d7 (60:14:b3:c5:ff:d7), Dst: Vizio_33:b7:6e (00:19:9d:33:b7:6e)
▸ Internet Protocol Version 4, Src: 192.168.0.113, Dst: 31.13.65.36
▾ Transmission Control Protocol, Src Port: 34982, Dst Port: 443, Seq: 1, Ack: 1, Len: 107
    Source Port: 34982
    Destination Port: 443
    [Stream index: 1]
    [TCP Segment Len: 107]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 108    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)

wireshark_wlan0_20180429194953_Tel9ji      Packets: 10466 · Displayed: 488 (4.7%) · Dropped: 0 (0.0%)    Profile: Default
```

Aryan Gupta
ECGR-3123-001
Dr. Tao Han
May 1, 2018

Filter	Meaning
((ip.src == 31.13.65.36) || (ip.dst == 31.13.65.36)) && (tcp.port == 443)	Only packets that are sent from Facebook or sent to Facebook and are addressed to port 443 are displayed

As you can see, HTTPS does not use port 80 but port 443. Normal HTTP uses port 80.

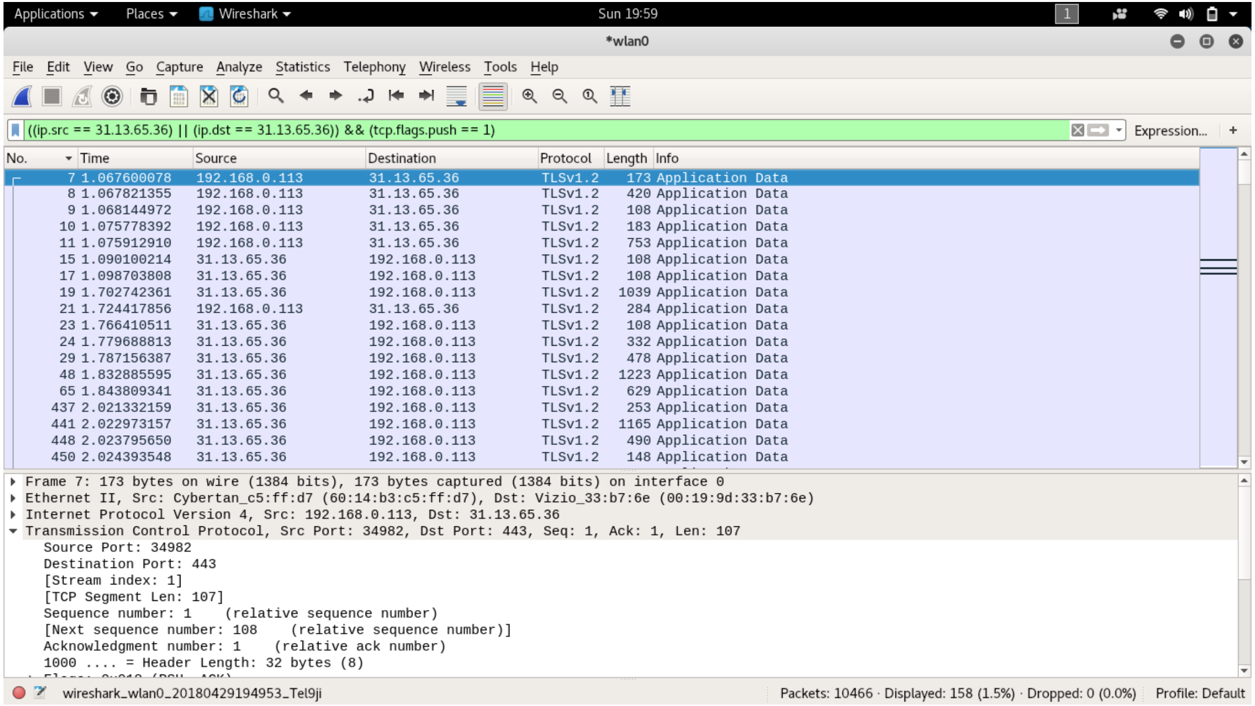

Applications
Places
Wireshark
Sun 19:59
*wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
No. Time Source Destination Protocol Length Info

((ip.src == 31.13.65.36) || (ip.dst == 31.13.65.36)) && (tcp.flags.reset == 1)

wireshark_wlan0_20180429194953_Tel9ji Packets: 10466 · Displayed: 0 (0.0%) · Dropped: 0 (0.0%) Profile: Default

Filter	Meaning
((ip.src == 31.13.65.36) || (ip.dst == 31.13.65.36)) && (tcp.flags.reset == 1)	Only packets that are sent from Facebook or sent to Facebook and have the RST flag are displayed

Each TCP packet has various flags set. The reset flag is set when we receive a packet that we weren't expecting. As you can see, there were no packets with RST flag set.



Applications
Places
Wireshark
Sun 19:59
*wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...

((ip.src == 31.13.65.36) || (ip.dst == 31.13.65.36)) && (tcp.flags.push == 1)

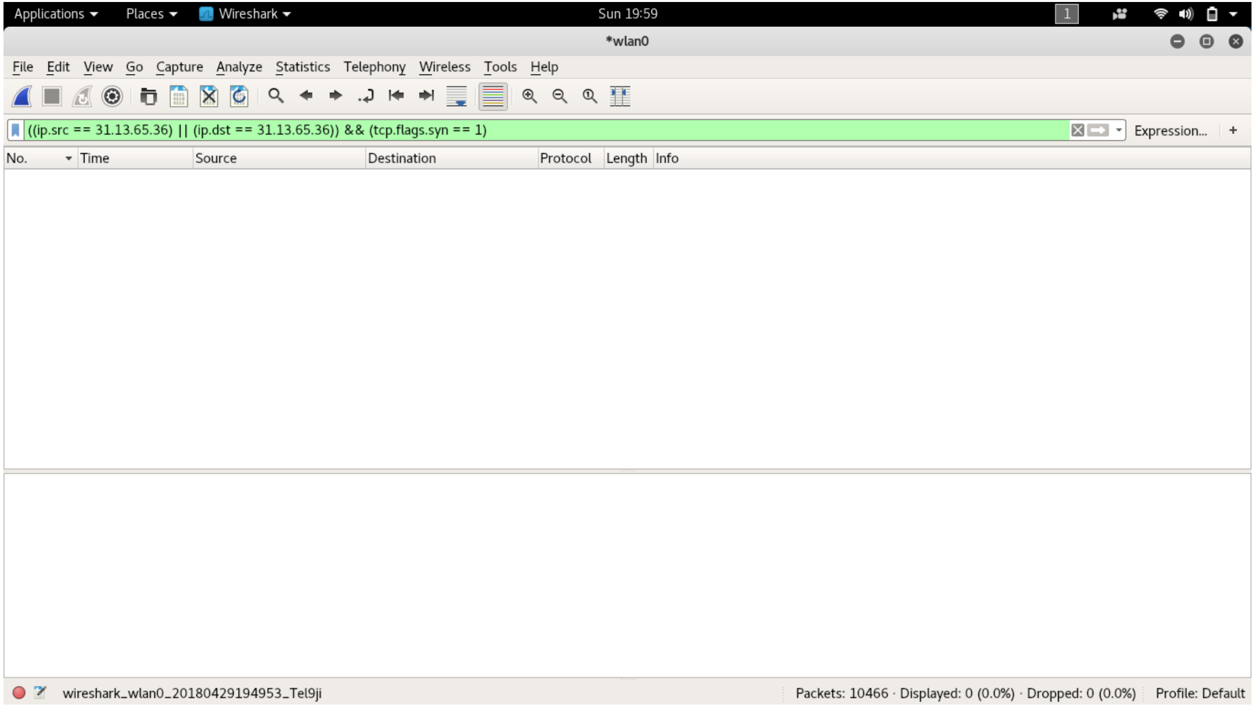
```
No.     Time          Source          Destination     Protocol  Length Info
      7 1.067600078   192.168.0.113   31.13.65.36     TLSv1.2    173 Application Data
      8 1.067821355   192.168.0.113   31.13.65.36     TLSv1.2    420 Application Data
      9 1.068144972   192.168.0.113   31.13.65.36     TLSv1.2    108 Application Data
     10 1.075778392   192.168.0.113   31.13.65.36     TLSv1.2    183 Application Data
     11 1.075912910   192.168.0.113   31.13.65.36     TLSv1.2    753 Application Data
     15 1.090100214   31.13.65.36     192.168.0.113   TLSv1.2    108 Application Data
     17 1.098703808   31.13.65.36     192.168.0.113   TLSv1.2    108 Application Data
     19 1.702742361   31.13.65.36     192.168.0.113   TLSv1.2   1039 Application Data
     21 1.724417856   192.168.0.113   31.13.65.36     TLSv1.2    284 Application Data
     23 1.766410511   31.13.65.36     192.168.0.113   TLSv1.2    108 Application Data
     24 1.779688813   31.13.65.36     192.168.0.113   TLSv1.2    332 Application Data
     29 1.787156387   31.13.65.36     192.168.0.113   TLSv1.2    478 Application Data
     48 1.832885595   31.13.65.36     192.168.0.113   TLSv1.2   1223 Application Data
     65 1.843809341   31.13.65.36     192.168.0.113   TLSv1.2    629 Application Data
    437 2.021332159   31.13.65.36     192.168.0.113   TLSv1.2    253 Application Data
    441 2.022973157   31.13.65.36     192.168.0.113   TLSv1.2   1165 Application Data
    448 2.023795650   31.13.65.36     192.168.0.113   TLSv1.2    490 Application Data
    450 2.024393548   31.13.65.36     192.168.0.113   TLSv1.2    148 Application Data
```

```
▸ Frame 7: 173 bytes on wire (1384 bits), 173 bytes captured (1384 bits) on interface 0
▸ Ethernet II, Src: Cybertan_c5:ff:d7 (60:14:b3:c5:ff:d7), Dst: Vizio_33:b7:6e (00:19:9d:33:b7:6e)
▸ Internet Protocol Version 4, Src: 192.168.0.113, Dst: 31.13.65.36
▾ Transmission Control Protocol, Src Port: 34982, Dst Port: 443, Seq: 1, Ack: 1, Len: 107
    Source Port: 34982
    Destination Port: 443
    [Stream index: 1]
    [TCP Segment Len: 107]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 108    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
```

wireshark_wlan0_20180429194953_Tel9ji   Packets: 10466 · Displayed: 158 (1.5%) · Dropped: 0 (0.0%)   Profile: Default

Filter	Meaning
((ip.src == 31.13.65.36) || (ip.dst == 31.13.65.36)) && (tcp.flags.push == 1)	Only packets that are sent from Facebook or sent to Facebook and have the PSH flag are displayed

Another common flag is the push flag. This flag is sent to the receiver to notify it to process the packets as they receive, rather than buffer them. There were many PSH packets.



((ip.src == 31.13.65.36) || (ip.dst == 31.13.65.36)) && (tcp.flags.syn == 1)

wireshark_wlan0_20180429194953_Tel9ji

Packets: 10466 · Displayed: 0 (0.0%) · Dropped: 0 (0.0%)

Profile: Default

Filter	Meaning
((ip.src == 31.13.65.36) || (ip.dst == 31.13.65.36)) && (tcp.flags.syn == 1)	Only packets that are sent from Facebook or sent to Facebook and have the SYN flag are displayed

The last flag that we needed to analyze was the Synchronization flag. This flag is used to establish a 3-way handshake between 2 hosts. Surprisingly, I was unable to find any packets with the SYN flag set.

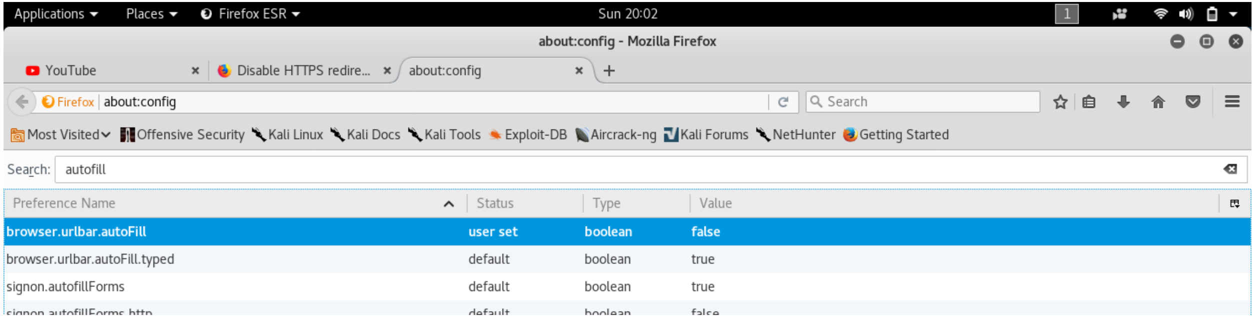
FLAG	**SYN**	**PSH**	**RST**
FACEBOOK	0 (0%)	158 (32.4%)	0 (0%)

Above is a table of the 3 flags and the percent of packets that each flag contained. There was a total of 488 packets.

YouTube

The YouTube video I used is called “*Can Light be Black? Mind-Blowing Dark Light Experiments!”* https://youtu.be/p-OCfiglZRQ . With the Facebook connection, I did not attempt to turn off HTTPS because I didn’t want my Facebook password sent through the internet as plaintext, however with YouTube, there is no ‘security’ that I need to worry about. I Googled the internet to find a way to turn off HTTPS and use the HTTP version of the website.

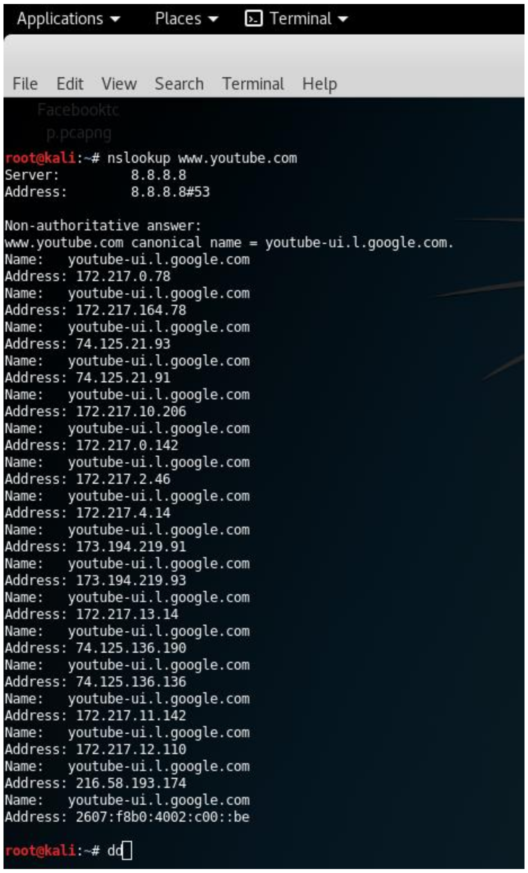
Aryan Gupta
ECGR-3123-001
Dr. Tao Han
May 1, 2018



Search: autofill

Preference Name	Status	Type	Value
browser.urlbar.autoFill	user set	boolean	false
browser.urlbar.autoFill.typed	default	boolean	true
signon.autofillForms	default	boolean	true
signon.autofillForms.http	default	boolean	false

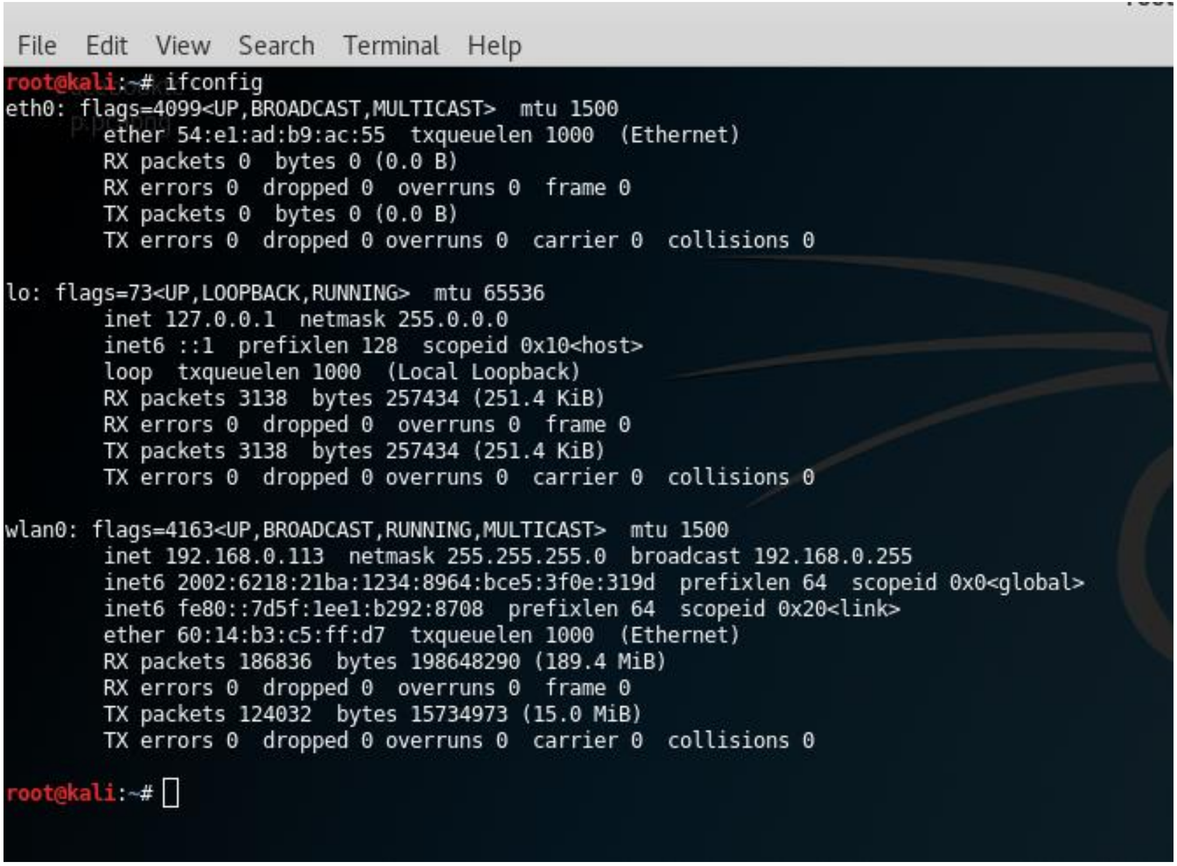
One way I found was to force Firefox to redirect me to the HTTPS website. However, this did not do anything and I was still being redirected to the HTTPS website. I decided it was best to continue with the project using HTTPS and maybe later come back to using HTTP only.



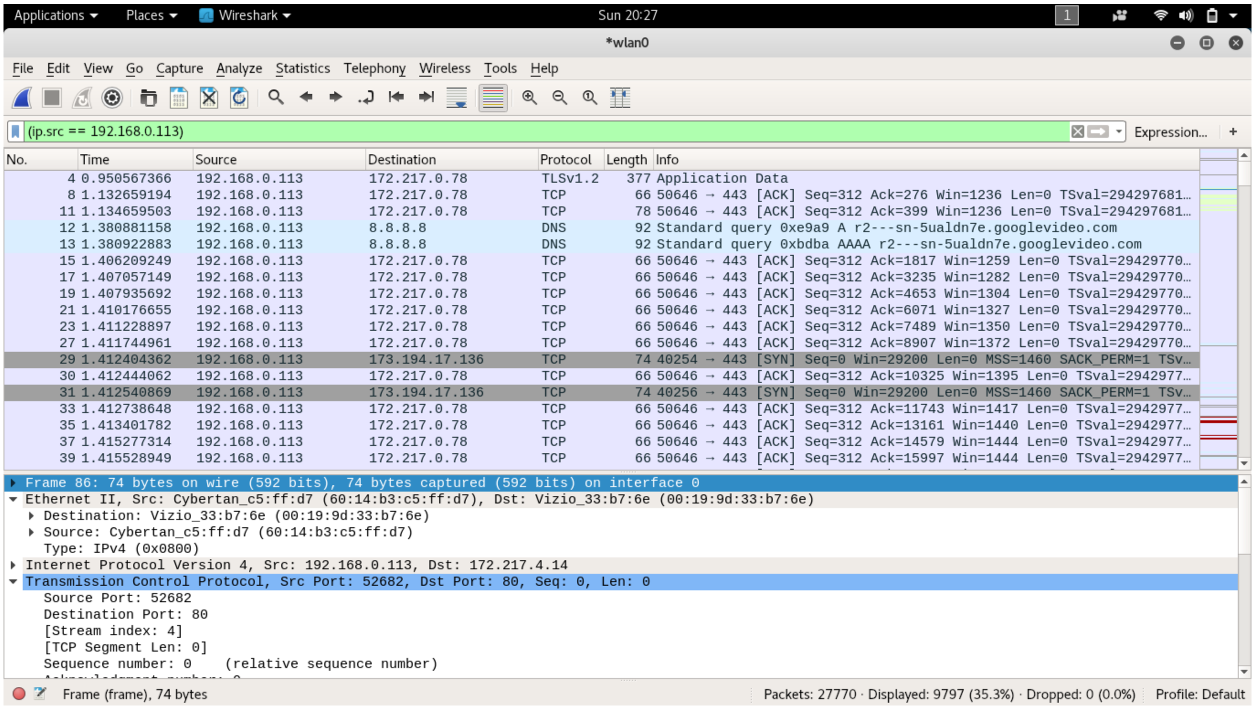
```
root@kali:~# nslookup www.youtube.com
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
www.youtube.com canonical name = youtube-ui.l.google.com.
Name:   youtube-ui.l.google.com
Address: 172.217.0.78
Name:   youtube-ui.l.google.com
Address: 172.217.164.78
Name:   youtube-ui.l.google.com
Address: 74.125.21.93
Name:   youtube-ui.l.google.com
Address: 74.125.21.91
Name:   youtube-ui.l.google.com
Address: 172.217.10.206
Name:   youtube-ui.l.google.com
Address: 172.217.0.142
Name:   youtube-ui.l.google.com
Address: 172.217.2.46
Name:   youtube-ui.l.google.com
Address: 172.217.4.14
Name:   youtube-ui.l.google.com
Address: 173.194.219.91
Name:   youtube-ui.l.google.com
Address: 173.194.219.93
Name:   youtube-ui.l.google.com
Address: 172.217.13.14
Name:   youtube-ui.l.google.com
Address: 74.125.136.190
Name:   youtube-ui.l.google.com
Address: 74.125.136.136
Name:   youtube-ui.l.google.com
Address: 172.217.11.142
Name:   youtube-ui.l.google.com
Address: 172.217.12.110
Name:   youtube-ui.l.google.com
Address: 216.58.193.174
Name:   youtube-ui.l.google.com
Address: 2607:f8b0:4002:c00::be

root@kali:~# dd
```

One of the first things I did was figure out the IP address of YouTube. As you can see YouTube resolves to many IP addresses, I'm assuming to load distributing. I was a bit worried, would I have to create a long filter, one for each IP address? I decided to just use my own IP address and create a filter with that.

Aryan Gupta
ECGR-3123-001
Dr. Tao Han
May 1, 2018



```
root@kali:~# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 54:e1:ad:b9:ac:55  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 3138  bytes 257434 (251.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3138  bytes 257434 (251.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.113  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 2002:6218:21ba:1234:8964:bce5:3f0e:319d  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::7d5f:1ee1:b292:8708  prefixlen 64  scopeid 0x20<link>
        ether 60:14:b3:c5:ff:d7  txqueuelen 1000  (Ethernet)
        RX packets 186836  bytes 198648290 (189.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 124032  bytes 15734973 (15.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@kali:~# 
```

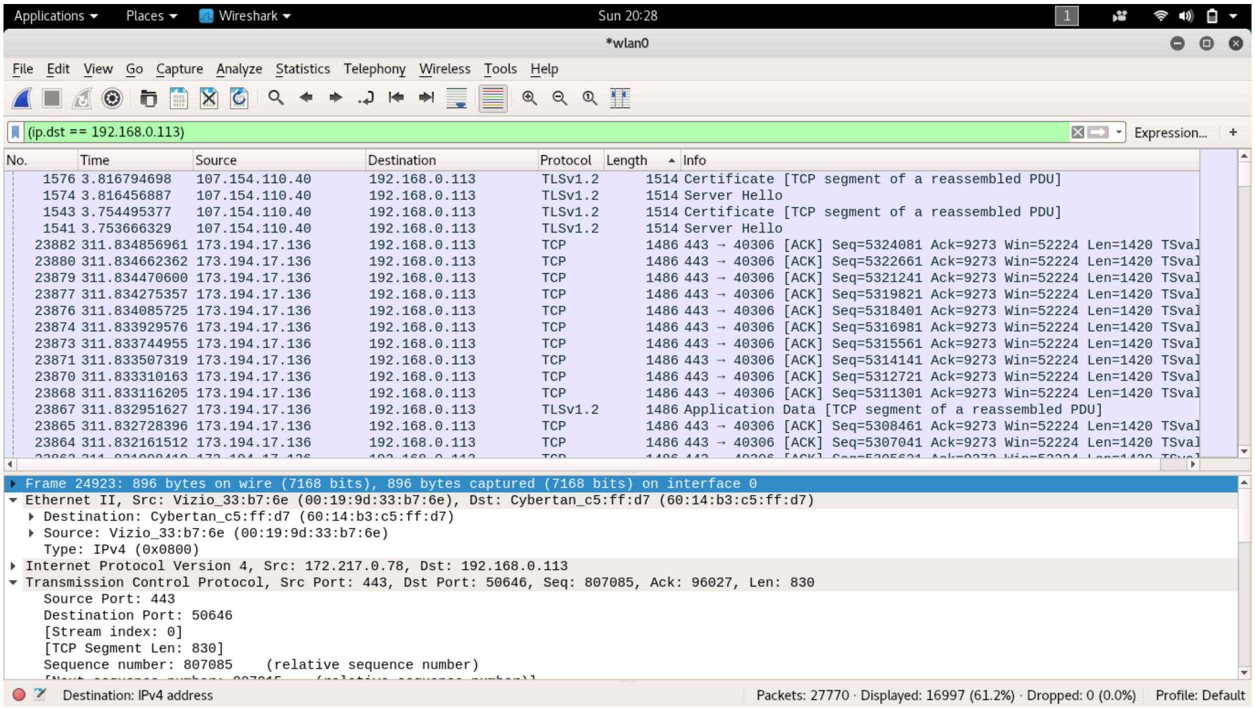

Applications
Places
Wireshark
Sun 20:27
*wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...

```
(ip.src == 192.168.0.113)

No.  Time         Source         Destination     Protocol Length Info
   4 0.950567366  192.168.0.113  172.217.0.78    TLSv1.2    377 Application Data
   8 1.132659194  192.168.0.113  172.217.0.78    TCP         66 50646 → 443 [ACK] Seq=312 Ack=276 Win=1236 Len=0 TSval=294297681…
  11 1.134659503  192.168.0.113  172.217.0.78    TCP         78 50646 → 443 [ACK] Seq=312 Ack=399 Win=1236 Len=0 TSval=294297681…
  12 1.380881158  192.168.0.113  8.8.8.8         DNS         92 Standard query 0xe9a9 A r2---sn-5ualdn7e.googlevideo.com
  13 1.380922883  192.168.0.113  8.8.8.8         DNS         92 Standard query 0xbdba AAAA r2---sn-5ualdn7e.googlevideo.com
  15 1.406209249  192.168.0.113  172.217.0.78    TCP         66 50646 → 443 [ACK] Seq=312 Ack=1817 Win=1259 Len=0 TSval=29429770…
  17 1.407057149  192.168.0.113  172.217.0.78    TCP         66 50646 → 443 [ACK] Seq=312 Ack=3235 Win=1282 Len=0 TSval=29429770…
  19 1.407935692  192.168.0.113  172.217.0.78    TCP         66 50646 → 443 [ACK] Seq=312 Ack=4653 Win=1304 Len=0 TSval=29429770…
  21 1.410176655  192.168.0.113  172.217.0.78    TCP         66 50646 → 443 [ACK] Seq=312 Ack=6071 Win=1327 Len=0 TSval=29429770…
  23 1.411228897  192.168.0.113  172.217.0.78    TCP         66 50646 → 443 [ACK] Seq=312 Ack=7489 Win=1350 Len=0 TSval=29429770…
  27 1.411744961  192.168.0.113  172.217.0.78    TCP         66 50646 → 443 [ACK] Seq=312 Ack=8907 Win=1372 Len=0 TSval=29429770…
  29 1.412404362  192.168.0.113  173.194.17.136  TCP         74 40254 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
  30 1.412444062  192.168.0.113  172.217.0.78    TCP         66 50646 → 443 [ACK] Seq=312 Ack=10325 Win=1395 Len=0 TSval=2942977…
  31 1.412540869  192.168.0.113  173.194.17.136  TCP         74 40256 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv…
  33 1.412738648  192.168.0.113  172.217.0.78    TCP         66 50646 → 443 [ACK] Seq=312 Ack=11743 Win=1417 Len=0 TSval=2942977…
  35 1.413401782  192.168.0.113  172.217.0.78    TCP         66 50646 → 443 [ACK] Seq=312 Ack=13161 Win=1440 Len=0 TSval=2942977…
  37 1.415277314  192.168.0.113  172.217.0.78    TCP         66 50646 → 443 [ACK] Seq=312 Ack=14579 Win=1444 Len=0 TSval=2942977…
  39 1.415528949  192.168.0.113  172.217.0.78    TCP         66 50646 → 443 [ACK] Seq=312 Ack=15997 Win=1444 Len=0 TSval=2942977…

▸ Frame 86: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▾ Ethernet II, Src: Cybertan_c5:ff:d7 (60:14:b3:c5:ff:d7), Dst: Vizio_33:b7:6e (00:19:9d:33:b7:6e)
  ▸ Destination: Vizio_33:b7:6e (00:19:9d:33:b7:6e)
  ▸ Source: Cybertan_c5:ff:d7 (60:14:b3:c5:ff:d7)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 192.168.0.113, Dst: 172.217.4.14
▾ Transmission Control Protocol, Src Port: 52682, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 52682
    Destination Port: 80
    [Stream index: 4]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)

Frame (frame), 74 bytes    Packets: 27770 · Displayed: 9797 (35.3%) · Dropped: 0 (0.0%)    Profile: Default
```

Aryan Gupta
ECGR-3123-001
Dr. Tao Han
May 1, 2018

Filter	**Meaning**
(ip.src == 192.168.0.113)	Only packets that are sent from my computer are displayed

In this filter I got both YouTube packets and DNS packets, and may other IP addresses that didn't show up in the YouTube DNS lookup.


Applications
Places
Wireshark
Sun 20:28
*wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…

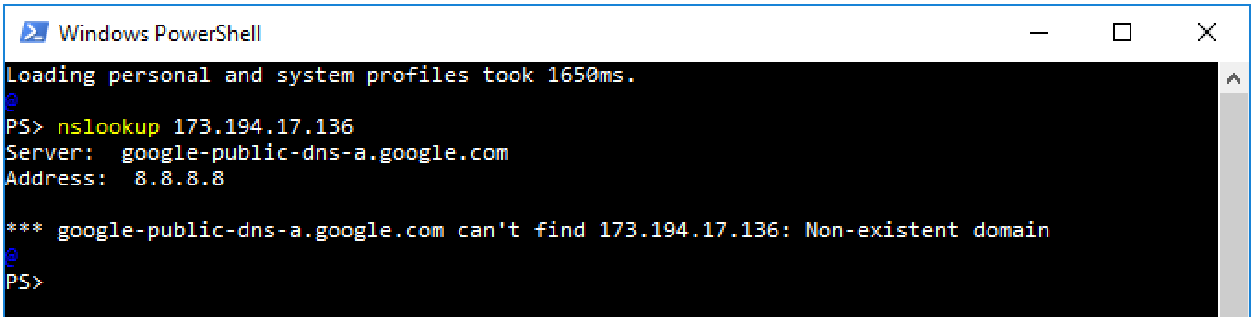
(ip.dst == 192.168.0.113)

```
No.   Time          Source          Destination    Protocol  Length  Info
 1576 3.816794698   107.154.110.40  192.168.0.113  TLSv1.2   1514 Certificate [TCP segment of a reassembled PDU]
 1574 3.816456887   107.154.110.40  192.168.0.113  TLSv1.2   1514 Server Hello
 1543 3.754495377   107.154.110.40  192.168.0.113  TLSv1.2   1514 Certificate [TCP segment of a reassembled PDU]
 1541 3.753666329   107.154.110.40  192.168.0.113  TLSv1.2   1514 Server Hello
23882 311.834856961 173.194.17.136  192.168.0.113  TCP       1486 443 → 40306 [ACK] Seq=5324081 Ack=9273 Win=52224 Len=1420 TSval
23880 311.834662362 173.194.17.136  192.168.0.113  TCP       1486 443 → 40306 [ACK] Seq=5322661 Ack=9273 Win=52224 Len=1420 TSval
23879 311.834470600 173.194.17.136  192.168.0.113  TCP       1486 443 → 40306 [ACK] Seq=5321241 Ack=9273 Win=52224 Len=1420 TSval
23877 311.834275357 173.194.17.136  192.168.0.113  TCP       1486 443 → 40306 [ACK] Seq=5319821 Ack=9273 Win=52224 Len=1420 TSval
23876 311.834085725 173.194.17.136  192.168.0.113  TCP       1486 443 → 40306 [ACK] Seq=5318401 Ack=9273 Win=52224 Len=1420 TSval
23874 311.833929576 173.194.17.136  192.168.0.113  TCP       1486 443 → 40306 [ACK] Seq=5316981 Ack=9273 Win=52224 Len=1420 TSval
23873 311.833744955 173.194.17.136  192.168.0.113  TCP       1486 443 → 40306 [ACK] Seq=5315561 Ack=9273 Win=52224 Len=1420 TSval
23871 311.833507319 173.194.17.136  192.168.0.113  TCP       1486 443 → 40306 [ACK] Seq=5314141 Ack=9273 Win=52224 Len=1420 TSval
23870 311.833310163 173.194.17.136  192.168.0.113  TCP       1486 443 → 40306 [ACK] Seq=5312721 Ack=9273 Win=52224 Len=1420 TSval
23868 311.833116205 173.194.17.136  192.168.0.113  TCP       1486 443 → 40306 [ACK] Seq=5311301 Ack=9273 Win=52224 Len=1420 TSval
23867 311.832951627 173.194.17.136  192.168.0.113  TLSv1.2   1486 Application Data [TCP segment of a reassembled PDU]
23865 311.832728396 173.194.17.136  192.168.0.113  TCP       1486 443 → 40306 [ACK] Seq=5308461 Ack=9273 Win=52224 Len=1420 TSval
23864 311.832161512 173.194.17.136  192.168.0.113  TCP       1486 443 → 40306 [ACK] Seq=5307041 Ack=9273 Win=52224 Len=1420 TSval
```

```
▸ Frame 24923: 896 bytes on wire (7168 bits), 896 bytes captured (7168 bits) on interface 0
▾ Ethernet II, Src: Vizio_33:b7:6e (00:19:9d:33:b7:6e), Dst: Cybertan_c5:ff:d7 (60:14:b3:c5:ff:d7)
  ▸ Destination: Cybertan_c5:ff:d7 (60:14:b3:c5:ff:d7)
  ▸ Source: Vizio_33:b7:6e (00:19:9d:33:b7:6e)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 172.217.0.78, Dst: 192.168.0.113
▾ Transmission Control Protocol, Src Port: 443, Dst Port: 50646, Seq: 807085, Ack: 96027, Len: 830
    Source Port: 443
    Destination Port: 50646
    [Stream index: 0]
    [TCP Segment Len: 830]
    Sequence number: 807085    (relative sequence number)
```

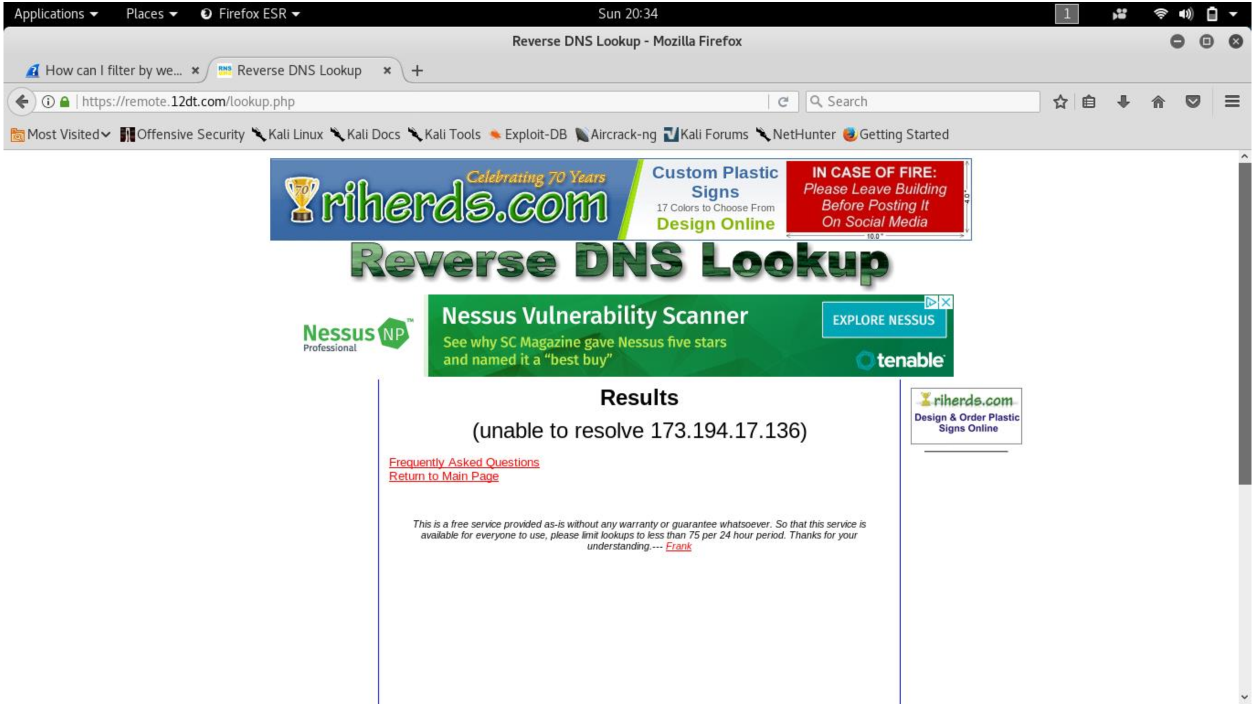
Destination: IPv4 address   Packets: 27770 · Displayed: 16997 (61.2%) · Dropped: 0 (0.0%)   Profile: Default

I got a bit curious because I had a lot of packets with IP addresses that didn't belong to YouTube. I then decided that I should figure out who's IP address this is.



```
Loading personal and system profiles took 1650ms.
@
PS> nslookup 173.194.17.136
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

*** google-public-dns-a.google.com can't find 173.194.17.136: Non-existent domain
@
PS>
```

I did a reverse DNS lookup, but I couldn't find a host or a server name.

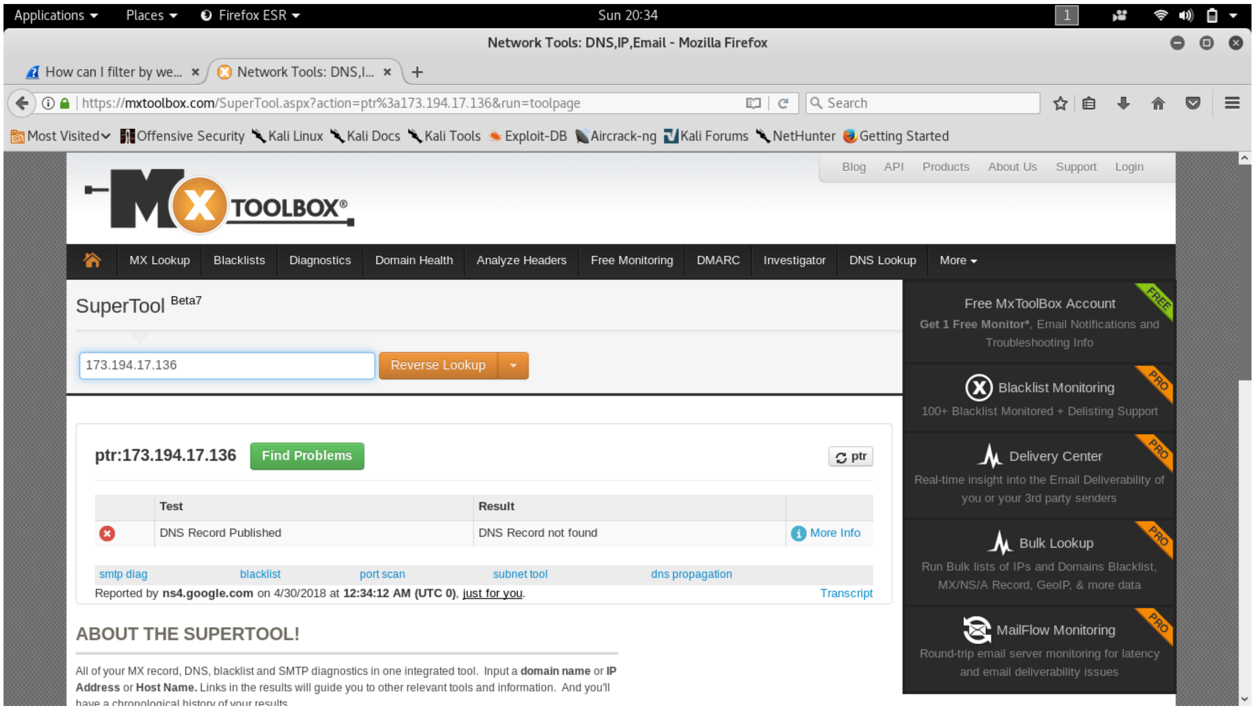


Results

(unable to resolve 173.194.17.136)

Frequently Asked Questions
Return to Main Page

This is a free service provided as-is without any warranty or guarantee whatsoever. So that this service is available for everyone to use, please limit lookups to less than 75 per 24 hour period. Thanks for your understanding.--- Frank



ptr:173.194.17.136

	Test	Result	
✗	DNS Record Published	DNS Record not found	More Info

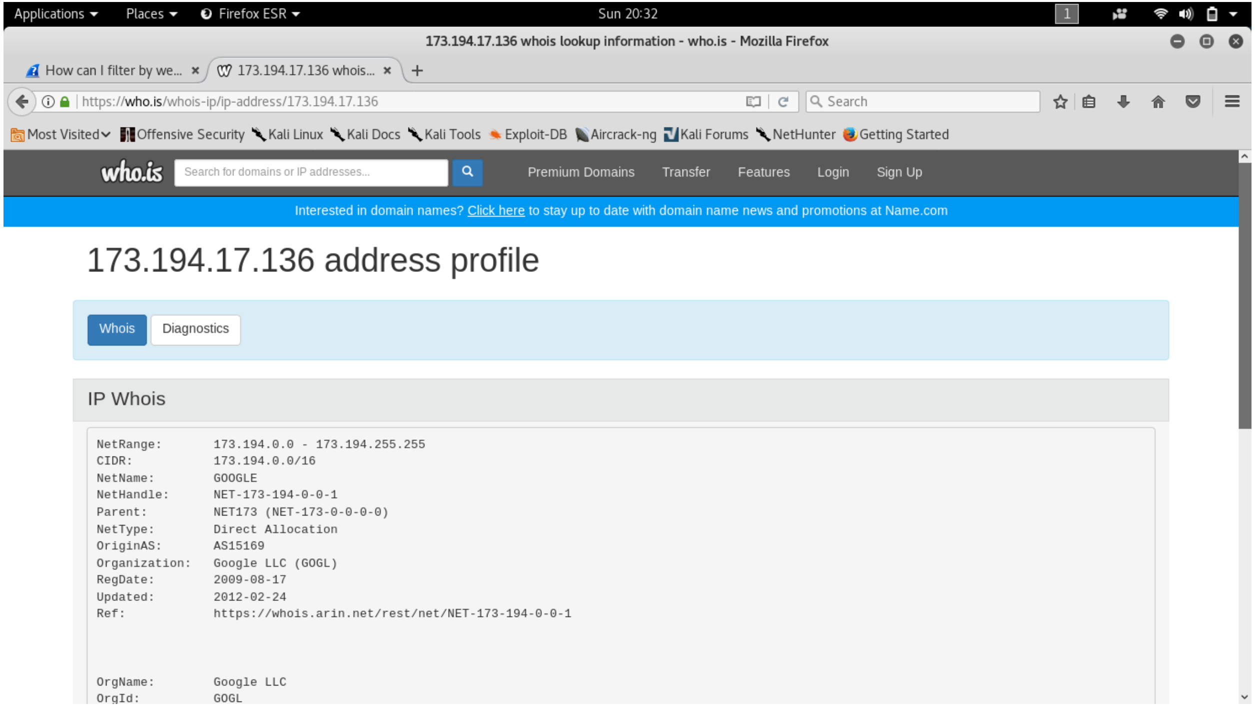
Reported by **ns4.google.com** on 4/30/2018 at **12:34:12 AM (UTC 0)**, just for you.

ABOUT THE SUPERTOOL!

All of your MX record, DNS, blacklist and SMTP diagnostics in one integrated tool. Input a **domain name** or **IP Address** or **Host Name.** Links in the results will guide you to other relevant tools and information. And you'll have a chronological history of your results.

I decided to go to reverse DNS websites to see if I can get a domain name. However, these websites just confirmed that the command line told me. The IP address can't be found. I then went to who.is to see if that can tell me what's up with that IP address.

Aryan Gupta
ECGR-3123-001
Dr. Tao Han
May 1, 2018

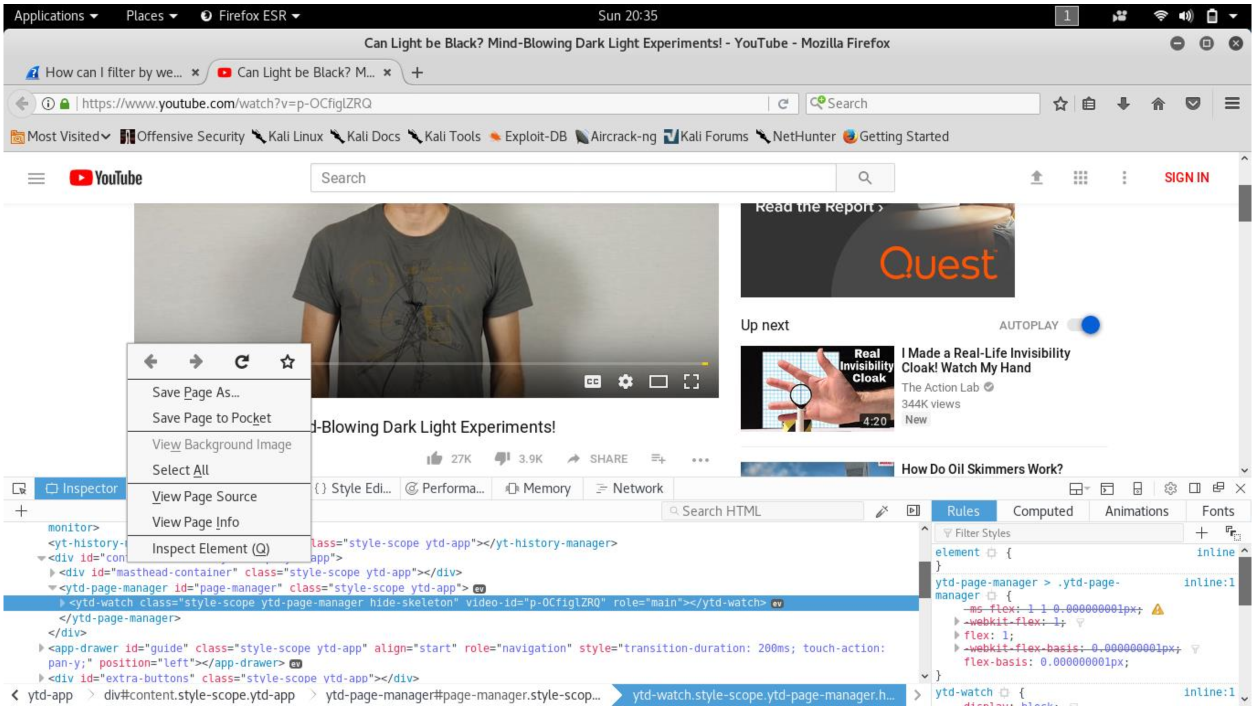

Applications
Places
Firefox ESR
Sun 20:32
173.194.17.136 whois lookup information - who.is - Mozilla Firefox
How can I filter by we...
173.194.17.136 whois...
https://who.is/whois-ip/ip-address/173.194.17.136
Search
Most Visited
Offensive Security
Kali Linux
Kali Docs
Kali Tools
Exploit-DB
Aircrack-ng
Kali Forums
NetHunter
Getting Started
who.is
Search for domains or IP addresses...
Premium Domains
Transfer
Features
Login
Sign Up
Interested in domain names? Click here to stay up to date with domain name news and promotions at Name.com
Whois
Diagnostics

173.194.17.136 address profile

IP Whois

```
NetRange:       173.194.0.0 - 173.194.255.255
CIDR:           173.194.0.0/16
NetName:        GOOGLE
NetHandle:      NET-173-194-0-0-1
Parent:         NET173 (NET-173-0-0-0-0)
NetType:        Direct Allocation
OriginAS:       AS15169
Organization:   Google LLC (GOGL)
RegDate:        2009-08-17
Updated:        2012-02-24
Ref:            https://whois.arin.net/rest/net/NET-173-194-0-0-1



OrgName:        Google LLC
OrgId:          GOGL
```

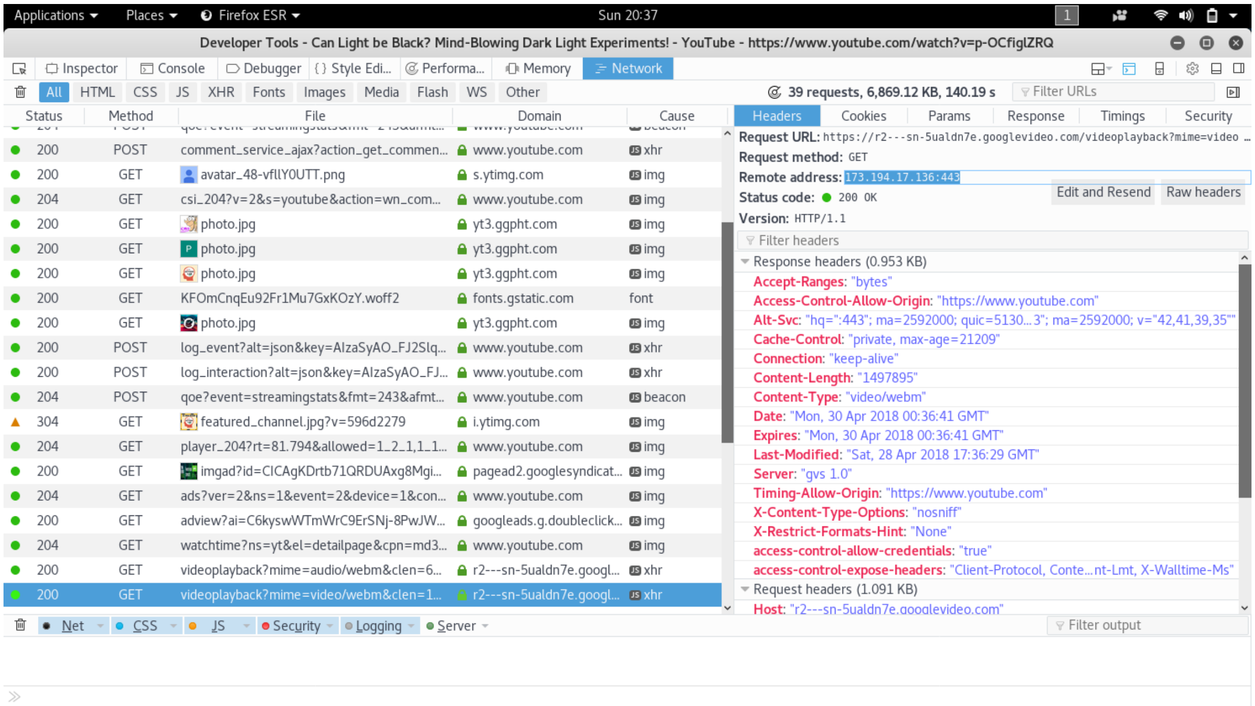
Who.is showed that that that block of IP address is owned by Google. I did this similarly to the other IP addresses and it showed me similar results. I was still intrigued, I wanted to figure out how I can get the domain of the server. So, I opened up Developer Options in Firefox.



```
monitor>
<yt-history-                        lass="style-scope ytd-app"></yt-history-manager>
<div id="con                        app">
  <div id="masthead-container" class="style-scope ytd-app"></div>
  <ytd-page-manager id="page-manager" class="style-scope ytd-app"> ev
    <ytd-watch class="style-scope ytd-page-manager hide-skeleton" video-id="p-OCfiglZRQ" role="main"></ytd-watch> ev
  </ytd-page-manager>
</div>
<app-drawer id="guide" class="style-scope ytd-app" align="start" role="navigation" style="transition-duration: 200ms; touch-action:
pan-y;" position="left"></app-drawer> ev
<div id="extra-buttons" class="style-scope ytd-app"></div>
```

```
element { inline
}
ytd-page-manager > .ytd-page-   inline:1
manager {
  ~~-ms-flex: 1 1 0.000000001px;~~
  ~~-webkit-flex: 1;~~
  flex: 1;
  ~~-webkit-flex-basis: 0.000000001px;~~
  flex-basis: 0.000000001px;
}
ytd-watch {   inline:1
```

Going to the Network tab of the window, and then restarting the video we can see all the network traffic from Firefox, some of this traffic was of webm type. Exactly what we needed for figuring out the domain for that IP address.



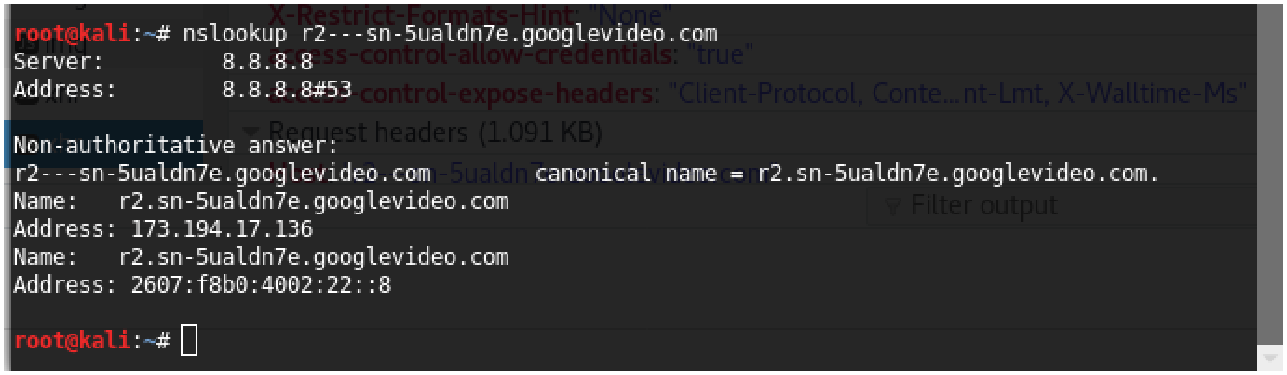
39 requests, 6,869.12 KB, 140.19 s

Status	Method	File	Domain	Cause
● 200	POST	comment_service_ajax?action_get_commen...	www.youtube.com	xhr
● 200	GET	avatar_48-vfllY0UTT.png	s.ytimg.com	img
● 204	GET	csi_204?v=2&s=youtube&action=wn_com...	www.youtube.com	img
● 200	GET	photo.jpg	yt3.ggpht.com	img
● 200	GET	photo.jpg	yt3.ggpht.com	img
● 200	GET	photo.jpg	yt3.ggpht.com	img
● 200	GET	KFOmCnqEu92Fr1Mu7GxKOzY.woff2	fonts.gstatic.com	font
● 200	GET	photo.jpg	yt3.ggpht.com	img
● 200	POST	log_event?alt=json&key=AlzaSyAO_FJ2Slq...	www.youtube.com	xhr
● 200	POST	log_interaction?alt=json&key=AlzaSyAO_FJ...	www.youtube.com	xhr
● 204	POST	qoe?event=streamingstats&fmt=243&afmt...	www.youtube.com	beacon
▲ 304	GET	featured_channel.jpg?v=596d2279	i.ytimg.com	img
● 204	GET	player_204?rt=81.794&allowed=1_2_1,1_1...	www.youtube.com	img
● 200	GET	imgad?id=CICAgKDrtb71QRDUAxg8Mgi...	pagead2.googlesyndicat...	img
● 204	GET	ads?ver=2&ns=1&event=2&device=1&con...	www.youtube.com	img
● 200	GET	adview?ai=C6kyswWTmWrC9ErSNj-8PwJW...	googleads.g.doubleclick...	img
● 204	GET	watchtime?ns=yt&el=detailpage&cpn=md3...	www.youtube.com	img
● 200	GET	videoplayback?mime=audio/webm&clen=6...	r2---sn-5ualdn7e.googl...	xhr
● 200	GET	videoplayback?mime=video/webm&clen=1...	r2---sn-5ualdn7e.googl...	xhr

Request URL: https://r2---sn-5ualdn7e.googlevideo.com/videoplayback?mime=video ...
Request method: GET
Remote address: 173.194.17.136:443
Status code: ● 200 OK
Version: HTTP/1.1

Response headers (0.953 KB)
Accept-Ranges: "bytes"
Access-Control-Allow-Origin: "https://www.youtube.com"
Alt-Svc: "hq=":443"; ma=2592000; quic=5130...3"; ma=2592000; v="42,41,39,35""
Cache-Control: "private, max-age=21209"
Connection: "keep-alive"
Content-Length: "1497895"
Content-Type: "video/webm"
Date: "Mon, 30 Apr 2018 00:36:41 GMT"
Expires: "Mon, 30 Apr 2018 00:36:41 GMT"
Last-Modified: "Sat, 28 Apr 2018 17:36:29 GMT"
Server: "gvs 1.0"
Timing-Allow-Origin: "https://www.youtube.com"
X-Content-Type-Options: "nosniff"
X-Restrict-Formats-Hint: "None"
access-control-allow-credentials: "true"
access-control-expose-headers: "Client-Protocol, Conte...nt-Lmt, X-Walltime-Ms"
Request headers (1.091 KB)
Host: "r2---sn-5ualdn7e.googlevideo.com"

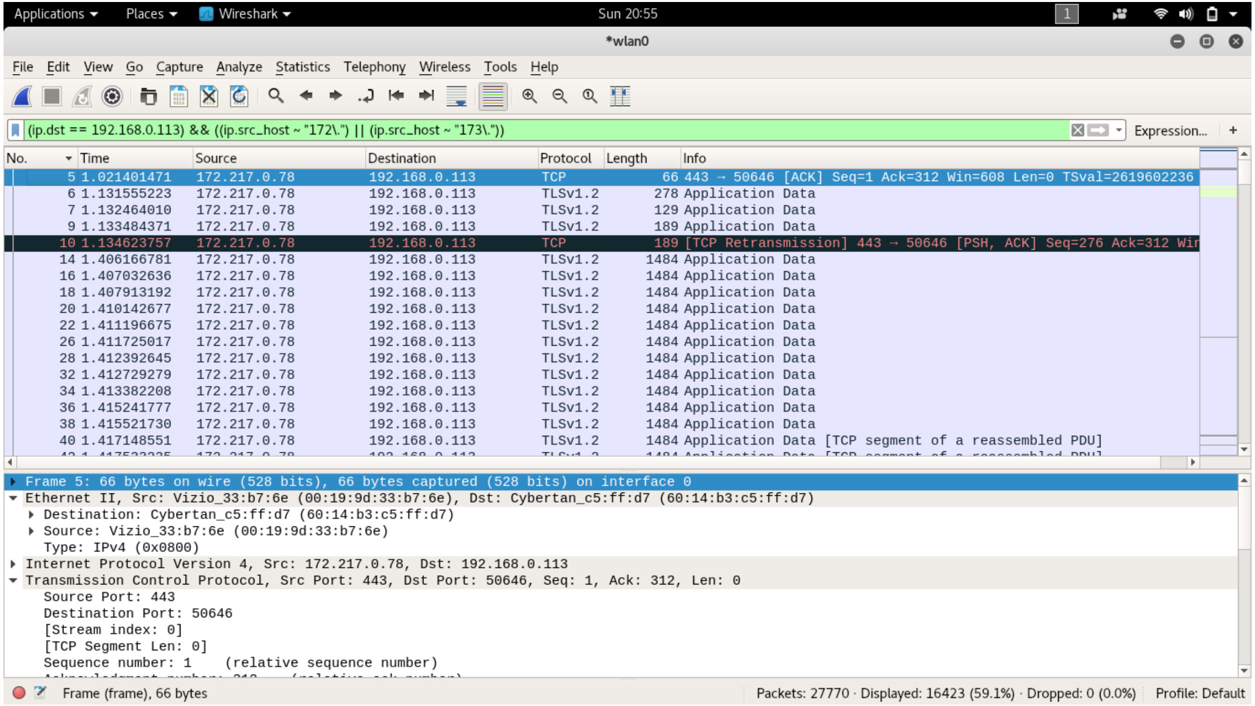
Expanding that network request, we see on the right the IP address that we wanted and the URL of the network recourse.



```
root@kali:~# nslookup r2---sn-5ualdn7e.googlevideo.com
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
r2---sn-5ualdn7e.googlevideo.com        canonical name = r2.sn-5ualdn7e.googlevideo.com.
Name:    r2.sn-5ualdn7e.googlevideo.com
Address: 173.194.17.136
Name:    r2.sn-5ualdn7e.googlevideo.com
Address: 2607:f8b0:4002:22::8

root@kali:~# 
```

Looking up that domain name we get what we wanted, the IP address.

Aryan Gupta
ECGR-3123-001
Dr. Tao Han
May 1, 2018



(ip.dst == 192.168.0.113) && ((ip.src_host ~ "172\.") || (ip.src_host ~ "173\."))

```
No.   Time         Source         Destination    Protocol  Length Info
    5 1.021401471  172.217.0.78   192.168.0.113  TCP           66 443 → 50646 [ACK] Seq=1 Ack=312 Win=608 Len=0 TSval=2619602236
    6 1.131555223  172.217.0.78   192.168.0.113  TLSv1.2      278 Application Data
    7 1.132464010  172.217.0.78   192.168.0.113  TLSv1.2      129 Application Data
    9 1.133484371  172.217.0.78   192.168.0.113  TLSv1.2      189 Application Data
   10 1.134623757  172.217.0.78   192.168.0.113  TCP          189 [TCP Retransmission] 443 → 50646 [PSH, ACK] Seq=276 Ack=312 Wir
   14 1.406166781  172.217.0.78   192.168.0.113  TLSv1.2     1484 Application Data
   16 1.407032636  172.217.0.78   192.168.0.113  TLSv1.2     1484 Application Data
   18 1.407913192  172.217.0.78   192.168.0.113  TLSv1.2     1484 Application Data
   20 1.410142677  172.217.0.78   192.168.0.113  TLSv1.2     1484 Application Data
   22 1.411196675  172.217.0.78   192.168.0.113  TLSv1.2     1484 Application Data
   26 1.411725017  172.217.0.78   192.168.0.113  TLSv1.2     1484 Application Data
   28 1.412392645  172.217.0.78   192.168.0.113  TLSv1.2     1484 Application Data
   32 1.412729279  172.217.0.78   192.168.0.113  TLSv1.2     1484 Application Data
   34 1.413382208  172.217.0.78   192.168.0.113  TLSv1.2     1484 Application Data
   36 1.415241777  172.217.0.78   192.168.0.113  TLSv1.2     1484 Application Data
   38 1.415521730  172.217.0.78   192.168.0.113  TLSv1.2     1484 Application Data
   40 1.417148551  172.217.0.78   192.168.0.113  TLSv1.2     1484 Application Data [TCP segment of a reassembled PDU]
```

```
▸ Frame 5: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▾ Ethernet II, Src: Vizio_33:b7:6e (00:19:9d:33:b7:6e), Dst: Cybertan_c5:ff:d7 (60:14:b3:c5:ff:d7)
  ▸ Destination: Cybertan_c5:ff:d7 (60:14:b3:c5:ff:d7)
  ▸ Source: Vizio_33:b7:6e (00:19:9d:33:b7:6e)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 172.217.0.78, Dst: 192.168.0.113
▾ Transmission Control Protocol, Src Port: 443, Dst Port: 50646, Seq: 1, Ack: 312, Len: 0
    Source Port: 443
    Destination Port: 50646
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
```

Frame (frame), 66 bytes | Packets: 27770 · Displayed: 16423 (59.1%) · Dropped: 0 (0.0%) | Profile: Default

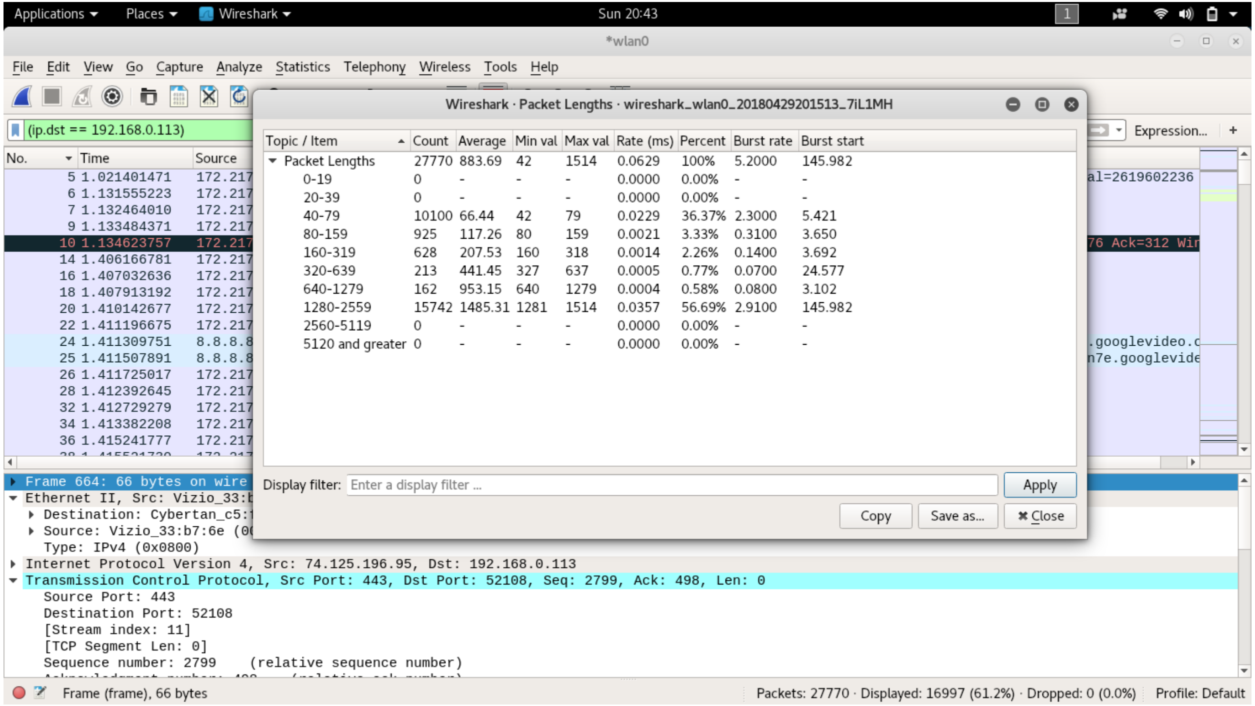
Filter	Meaning
(ip.src == 192.168.0.113) && ((ip.src_host ~ "172\.") || (ip.src_host ~ "173\."))	Only packets that are sent from my computer and that are received from IP addresses that match 172.*.*.* or 173.*.*.* are displayed

I created a simple filter that filtered by the first byte of the IP address. This would filter out the DNS packets but keep all the YouTube packets.

FLAG	SYN	PSH	RST
YOUTUBE	21 (0.13%)	539 (3.3%)	12 (0.073%)

Here is the table for the TCP flags. There was a total of 16,423 packets.

Aryan Gupta
ECGR-3123-001
Dr. Tao Han
May 1, 2018


Applications
Places
Wireshark
Sun 20:43
*wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
(ip.dst == 192.168.0.113)
Expression...
Display filter:
Enter a display filter ...
Apply
Copy
Save as...
Close

Wireshark · Packet Lengths · wireshark_wlan0_20180429201513_7iL1MH

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Packet Lengths	27770	883.69	42	1514	0.0629	100%	5.2000	145.982
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	10100	66.44	42	79	0.0229	36.37%	2.3000	5.421
80-159	925	117.26	80	159	0.0021	3.33%	0.3100	3.650
160-319	628	207.53	160	318	0.0014	2.26%	0.1400	3.692
320-639	213	441.45	327	637	0.0005	0.77%	0.0700	24.577
640-1279	162	953.15	640	1279	0.0004	0.58%	0.0800	3.102
1280-2559	15742	1485.31	1281	1514	0.0357	56.69%	2.9100	145.982
2560-5119	0	-	-	-	0.0000	0.00%	-	-
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

```
No.     Time         Source
    5 1.021401471  172.217
    6 1.131555223  172.217
    7 1.132464010  172.217
    9 1.133484371  172.217
   10 1.134623757  172.217
   14 1.406166781  172.217
   16 1.407032636  172.217
   18 1.407913192  172.217
   20 1.410142677  172.217
   22 1.411196675  172.217
   24 1.411309751  8.8.8.8
   25 1.411507891  8.8.8.8
   26 1.411725017  172.217
   28 1.412392645  172.217
   32 1.412729279  172.217
   34 1.413382208  172.217
   36 1.415241777  172.217
```

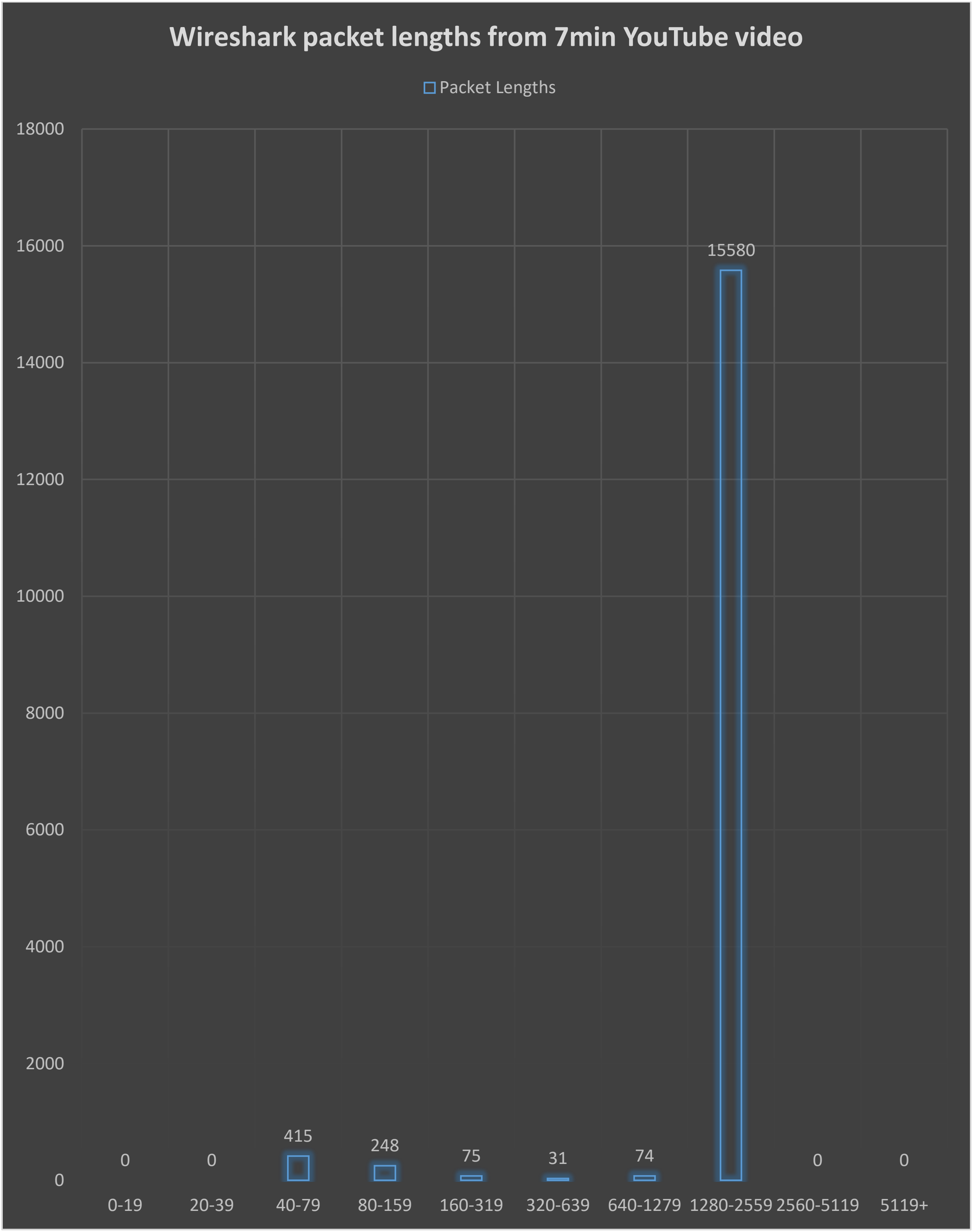
```
al=2619602236
76 Ack=312 Wir
.googlevideo.c
n7e.googlevide
```

```
▸ Frame 664: 66 bytes on wire
▾ Ethernet II, Src: Vizio_33:b
  ▸ Destination: Cybertan_c5:
  ▸ Source: Vizio_33:b7:6e (0
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 74.125.196.95, Dst: 192.168.0.113
▾ Transmission Control Protocol, Src Port: 443, Dst Port: 52108, Seq: 2799, Ack: 498, Len: 0
    Source Port: 443
    Destination Port: 52108
    [Stream index: 11]
    [TCP Segment Len: 0]
    Sequence number: 2799    (relative sequence number)
```

```
Frame (frame), 66 bytes          Packets: 27770 · Displayed: 16997 (61.2%) · Dropped: 0 (0.0%)    Profile: Default
```

Wireshark has a cool feature that allows you to see the packet lengths as a table. This made it easy to create the histogram, shown below.

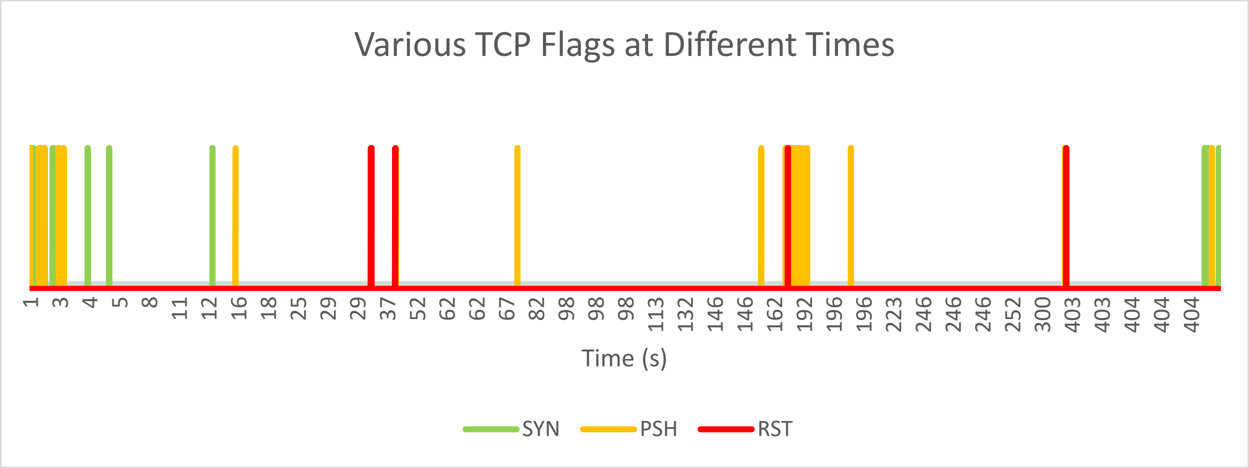
Aryan Gupta
ECGR-3123-001
Dr. Tao Han
May 1, 2018



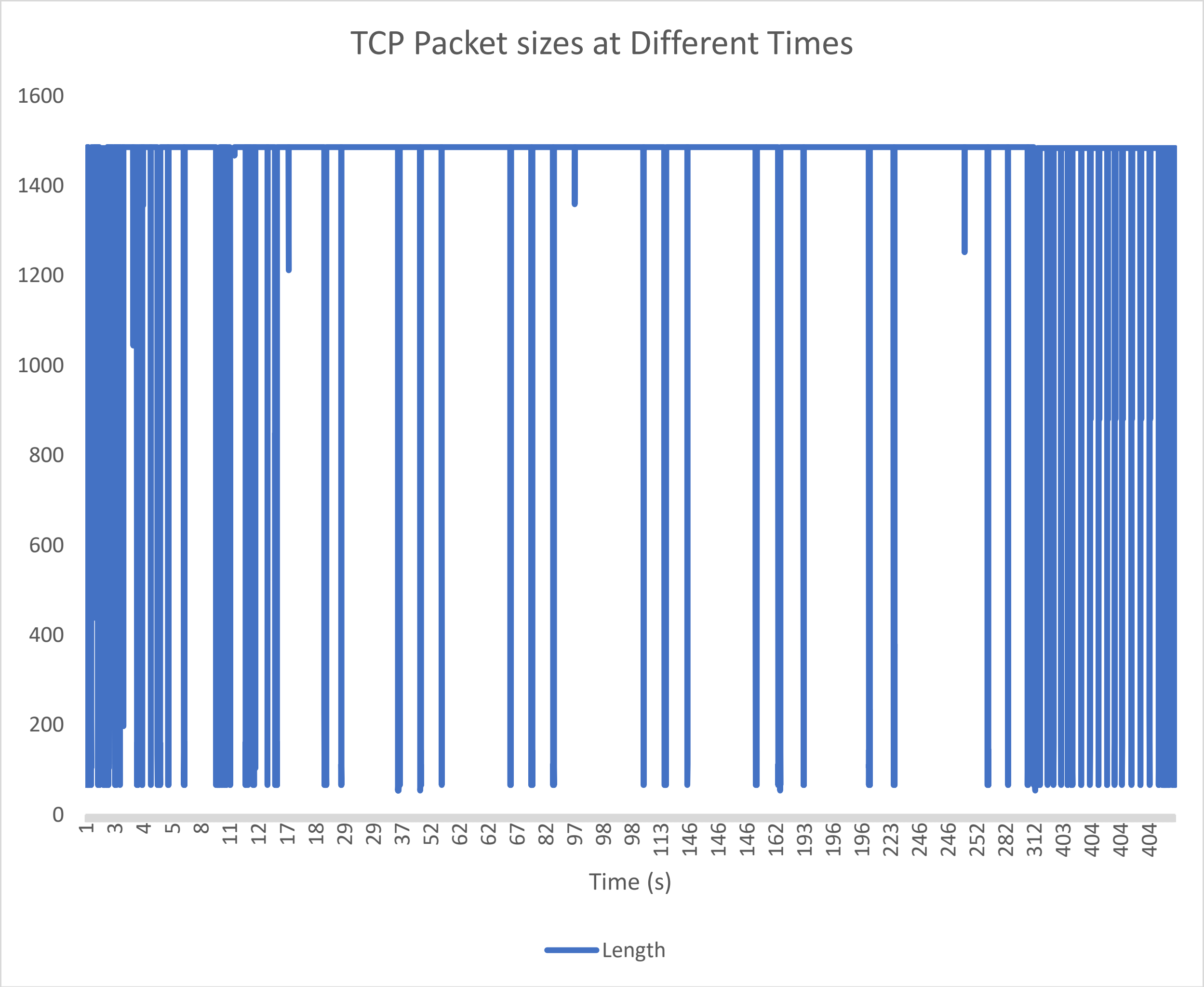
This is the histogram for the packet sizes during the YouTube part of this project.

Aryan Gupta
ECGR-3123-001
Dr. Tao Han
May 1, 2018

I exported all the data to a comma separated value file (*.csv), and uses excel to create a timeline for the data, shown below:



Below is a graph of the size of the packets versus the time on the video.


TCP Packet sizes at Different Times
Time (s)
Length