Zicheng Huang 921016568
Dhilan Patel 921025821

Part 1a.

Code: Project1a.py

Ping google.com 20 times

google.pcap:

Total: 145
DNS: 24
HTTPS: 53
ICMP (ping): 43
Other App: 2
QUIC: 23

ip:
22:12:49.191120 -> 170.114.14.50
22:12:51.289407 -> 169.237.250.250
22:12:51.300918 -> 168.150.49.156
22:12:51.314235 -> 151.101.43.6
22:12:52.861195 -> 198.189.166.16
22:12:52.876095 -> 17.253.4.85
22:12:52.950142 -> 17.253.144.10
22:13:01.365184 -> 142.251.32.46
22:13:05.083760 -> 168.150.63.254
22:13:09.322609 -> 17.248.192.3
22:13:19.418432 -> 173.194.65.188

Ping uses ICMP protocol, which is on transport layer. Here it has 43, which is close to what we expect, being 40 (20 ping and 20 pong)

example.com

example.pcap:

Total: 149
QUIC: 149

ip:
22:23:57.396151 -> 168.150.49.156
22:23:57.407175 -> 23.220.75.232
22:23:57.459888 -> 142.250.191.36
22:23:58.282601 -> 142.251.46.206

22:24:02.233403 -> 35.83.224.87
22:24:02.987337 -> 18.155.192.117
22:24:05.100672 -> 34.107.204.85

example.com uses mostly QUIC along with some HTTPS. A quick internet search told us that there is no HTML content in example.com, and it is internationally used as a special documentation browser so the application layer works differently. We only recorded UDP and TCP packets.

httpforever.com

httpforever.pcap:

Total: 1030
DNS: 82
HTTP: 8
HTTPS: 61
ICMP (ping): 9
Other App: 52
QUIC: 818

ip:
22:24:46.544959 -> 142.250.191.36
22:24:46.564454 -> 168.150.49.156
22:24:48.024008 -> 18.155.192.117
22:24:50.101007 -> 173.194.65.108
22:24:50.109043 -> 34.107.204.85
22:24:50.359922 -> 169.237.1.250
22:24:50.364843 -> 142.251.32.46
22:24:52.357617 -> 142.250.189.225
22:24:52.358103 -> 172.217.12.98
22:24:52.484430 -> 142.250.189.234
22:24:53.833719 -> 142.250.191.78
22:24:54.281959 -> 142.250.189.206
22:24:54.386777 -> 23.220.75.245
22:24:54.387156 -> 142.251.214.129
22:24:54.387334 -> 172.217.12.106
22:24:54.387600 -> 142.250.191.74
22:24:54.388243 -> 35.83.224.87
22:24:54.389342 -> 104.17.24.14
22:24:54.397939 -> 142.250.191.67
22:24:54.575967 -> 17.248.245.36
22:24:55.202346 -> 146.190.62.39
22:24:55.275849 -> 169.237.250.250
22:24:55.320697 -> 23.221.77.93
22:24:57.407396 -> 142.251.46.174

22:25:01.566972 -> 35.228.14.46
22:25:02.653745 -> 3.230.114.92
22:25:05.603829 -> 52.23.24.49
22:25:06.202057 -> 3.220.158.94
22:25:06.674311 -> 3.212.169.172
22:25:07.129520 -> 34.36.213.229
22:25:07.194953 -> 54.175.128.23
22:25:07.509106 -> 23.220.75.232
22:25:08.345956 -> 173.194.65.188

httpforever.com is shown to be more complicated than example.com, using QUIC, HTTPS, DNS, and HTTP

tmz.com

tmz.pcap:

Total: 15693
DNS: 413
HTTPS: 12908
ICMP (ping): 19
Other App: 4
QUIC: 2349

ip:
22:25:30.052987 -> 142.250.191.36
22:25:30.077724 -> 168.150.49.156
22:25:30.277848 -> 169.237.1.250
22:25:30.292060 -> 142.250.189.206
22:25:30.454421 -> 142.251.46.195
22:25:30.455087 -> 142.250.189.195
22:25:31.649672 -> 142.251.32.46
22:25:32.817068 -> 142.250.189.225
22:25:33.048214 -> 18.155.192.117
22:25:35.142145 -> 34.107.204.85
22:25:36.480304 -> 104.17.24.14
22:25:36.837077 -> 142.250.189.234
22:25:36.844085 -> 35.83.224.87
22:25:36.856747 -> 18.238.192.108
22:25:37.012641 -> 3.169.183.17
22:25:37.264037 -> 142.250.189.226
22:25:37.268003 -> 13.249.70.91
22:25:37.268069 -> 142.251.32.40
22:25:37.283304 -> 151.101.41.91
22:25:37.416104 -> 23.220.75.232

22:25:37.430165 -> 151.101.40.157
22:25:37.482411 -> 96.16.55.134
22:25:37.532060 -> 34.110.146.185
22:25:37.602779 -> 52.11.247.82
22:25:37.607732 -> 65.8.176.2
22:25:37.640133 -> 35.186.224.24
22:25:37.671618 -> 216.239.38.181
22:25:37.711581 -> 184.30.148.86
22:25:37.725481 -> 142.250.191.46
22:25:37.748127 -> 142.251.46.226
22:25:37.759493 -> 18.155.192.91
22:25:37.766780 -> 151.101.42.202
22:25:37.892876 -> 142.251.46.168
22:25:37.893542 -> 18.155.192.106
22:25:37.893659 -> 172.217.12.110
22:25:37.901964 -> 162.159.140.229
22:25:37.902192 -> 44.239.68.199
22:25:37.933766 -> 18.205.65.202
22:25:37.945521 -> 52.32.212.61
22:25:37.948617 -> 18.155.192.125
22:25:38.003393 -> 96.16.55.166
22:25:38.004860 -> 142.250.191.35
22:25:38.007257 -> 142.250.189.162
22:25:38.028425 -> 169.237.250.250
22:25:38.061567 -> 151.101.42.132
22:25:38.079146 -> 172.217.78.155
22:25:38.080775 -> 142.250.189.164
22:25:38.125051 -> 142.250.189.214
22:25:38.126493 -> 104.79.0.27
22:25:38.185587 -> 54.69.126.129
22:25:38.185893 -> 142.250.189.161
22:25:38.274146 -> 44.232.191.163
22:25:38.293399 -> 142.250.189.194
22:25:38.349256 -> 23.46.216.85
22:25:38.391119 -> 23.67.33.74
22:25:38.567139 -> 184.30.148.83
22:25:38.616859 -> 23.67.33.105
22:25:38.617324 -> 184.30.148.78
22:25:38.621351 -> 151.101.42.208
22:25:38.712488 -> 142.250.189.227
22:25:38.789936 -> 151.101.40.159
22:25:38.835789 -> 142.251.214.129
22:25:38.840090 -> 71.18.135.245
22:25:38.879409 -> 151.101.43.42

22:25:38.998676 -> 63.140.37.238
22:25:39.318446 -> 142.251.46.170
22:25:39.326414 -> 96.16.55.96
22:25:39.461099 -> 104.18.40.226
22:25:39.472650 -> 166.117.21.130
22:25:39.536817 -> 96.16.55.72
22:25:39.752551 -> 142.250.191.70
22:25:39.839942 -> 34.8.113.107
22:25:40.025381 -> 96.16.55.95
22:25:40.218055 -> 146.190.62.39
22:25:40.343180 -> 23.221.77.93
22:25:40.471622 -> 23.212.59.27
22:25:40.471968 -> 74.125.250.129
22:25:40.558159 -> 74.119.118.73
22:25:40.584258 -> 104.18.36.146
22:25:40.596158 -> 142.251.214.142
22:25:40.626837 -> 108.138.246.105
22:25:41.305325 -> 18.173.121.89
22:25:42.009233 -> 107.23.182.55
22:25:42.034432 -> 142.250.191.66
22:25:42.037611 -> 151.101.43.52
22:25:42.146833 -> 35.82.246.97
22:25:42.450579 -> 142.251.46.225
22:25:42.628280 -> 74.119.118.98
22:25:42.817272 -> 104.18.25.18
22:25:42.817416 -> 23.37.17.107
22:25:42.817499 -> 34.98.64.218
22:25:42.817569 -> 23.39.189.14
22:25:42.817639 -> 35.71.139.29
22:25:42.877576 -> 151.101.41.108
22:25:43.375891 -> 34.120.195.249
22:25:43.377091 -> 34.218.207.190
22:25:43.377186 -> 104.254.151.36
22:25:43.401712 -> 3.33.220.150

tmz.com uses mostly the same protocols as httpforever.com; however ,its packet number is way higher, showing that there are more objects needed to construct its HTML

FTP server:

ftp.pcap:

Total: 238
DNS: 22
FTP: 5

HTTPS: 45
QUIC: 166

ip:
22:30:55.108078 -> 169.237.250.250
22:30:55.139921 -> 169.237.1.250
22:30:55.165714 -> 168.150.34.184
22:30:55.167491 -> 184.28.81.204
22:30:55.661555 -> 142.250.189.238
22:30:56.345095 -> 142.250.189.170
22:30:56.363990 -> 20.189.173.27
22:30:56.440618 -> 142.251.32.42
22:30:57.985579 -> 209.51.188.20
22:30:58.839690 -> 35.153.246.86
22:31:01.314990 -> 172.67.193.137
22:31:02.066423 -> 13.64.159.249
22:31:03.615983 -> 142.250.191.36
22:31:04.795339 -> 142.250.191.67

There are 5 FTP packets, the actual attempt of connecting to the FTP server. The low number of packets indicate that we did not connect for a long time.

SSH to CSIF

SSH.pcap:

Total: 113
HTTPS: 4
Other App: 75
QUIC: 34

ip:
22:43:19.309884 -> 142.251.214.142
22:43:19.330311 -> 168.150.49.156
22:43:20.393392 -> 169.237.240.10
22:43:20.678363 -> 104.18.39.21
22:43:21.268493 -> 142.250.189.170
22:43:22.557290 -> 148.139.180.240

Using SSH to access a CSIF machine is fairly simple, using only QUIC and HTTPS. Looking at Wireshark, it also has a lot of ESP protocols, which are on the network layer. This is probably caused by us using the library VPN to access the CSIF.

In order to see the browser used, we need HTTP protocols. example.com has none, so we cannot see. In httpforever.com's HTTP, its header includes: User Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0

Safari/537.36. This shows it was accessed from a Mac's Chrome Browser. tmz.com has HTTPS but not HTTP, so I can't find the user agent in its header

Part 1b.

Code: Project1b.py

We see in PCAP1_1.pcap at frame = 112 the info says 'GET /?secret=secret1 HTTP/1.1', so we know that this is the secret. Using Python to translate the packet bytes to human-readable texts, it reads:
GET /?secret=secret1 HTTP/1.1
Host: example.com
User-Agent: test-client/2
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
MY-SECRET: Zubair Rocks!!

Looking at PCAP1_2.pcap, most are MDNS protocols with info "Standard query" or "standard query response". The sources are mostly from 10.0.0.86. MDNS is a protocol that allows devices to find each other on a local network without a central DNS server, meaning the activity here is mostly on a local network. 10.0.0.86 is likely the IP address of a computer that is trying to access local machines. In some of the frame's info segments, it mentions "cache-flush Alexs-iPhone" or "_spotify-connect._tcp". This could indicate that an iPhone is trying to connect to something to use Spotify, by airplaying to a Roku TV. The reason iPhone flushes its cache could be that it chose another song, so the phone flushes its current data of the song out and downloads he new one. In addition, it has a packet with a header asking 'Who has 10.0.0.1? Tel 10.0.0.230'. This indicates device 10.0.230 with source ChongqingFug is asking which device has IP 10.0.0.1. Later, a device responds with '10.0.0.1 is at' followed by an address, indicating 10.0.0.1 has noticed the previous request and responded with an answer.

For PCAP1_3.pcap, there are still a lot of MDNS protocols. This indicates that the user is still trying to access something on the local network. However, there is also a lot of ICMPv6 Protocols, with info section displaying "Echo (ping) request" and "Time Exceeded (Hop limit exceeded in transit) [Malformed Packet]. This shows that the user is probably trying to test the network by pinging something over IPv6. However, it does not get a response Pong. Instead, the packet has reached the Hop limit and is probably lost. The user then increases the number of hop limit 1 at a time, testing sending ping in between each change. Finally, the user receives a pong at hop limit = 13.

Part 2.

Code: udp_clientZicheng_921016568_Dhilan_921025821.py,
udp_serverZicheng_921016568_Dhilan_921025821.py

The throughput is 242903.14713256166 kB/s

Part 3.

proxy_serverZicheng_921016568_Dhilan_921025821.py
clientZicheng_921016568_Dhilan_921025821.py
serverZicheng_921016568_Dhilan_921025821.py

Submission Page

Best of luck

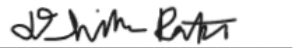
Submission Page
Include this signed page with your submission

I certify that all submitted work is my own work. I have completed all of the assignments on my own without assistance from others except as indicated by appropriate citation. I have read and understand the university policy on plagiarism and academic dishonesty. I further understand that official sanctions will be imposed if there is any evidence of academic dishonesty in this work. I certify that the above statements are true.

Team Member 1:

Zicheng Huang		10/31/25
Full Name (Printed)	Signature	Date

Team Member 2:

Dhilan Patel		10/31/25
Full Name (Printed)	Signature	Date