SIL765: NETWORKS SYSTEMS AND SECURITY

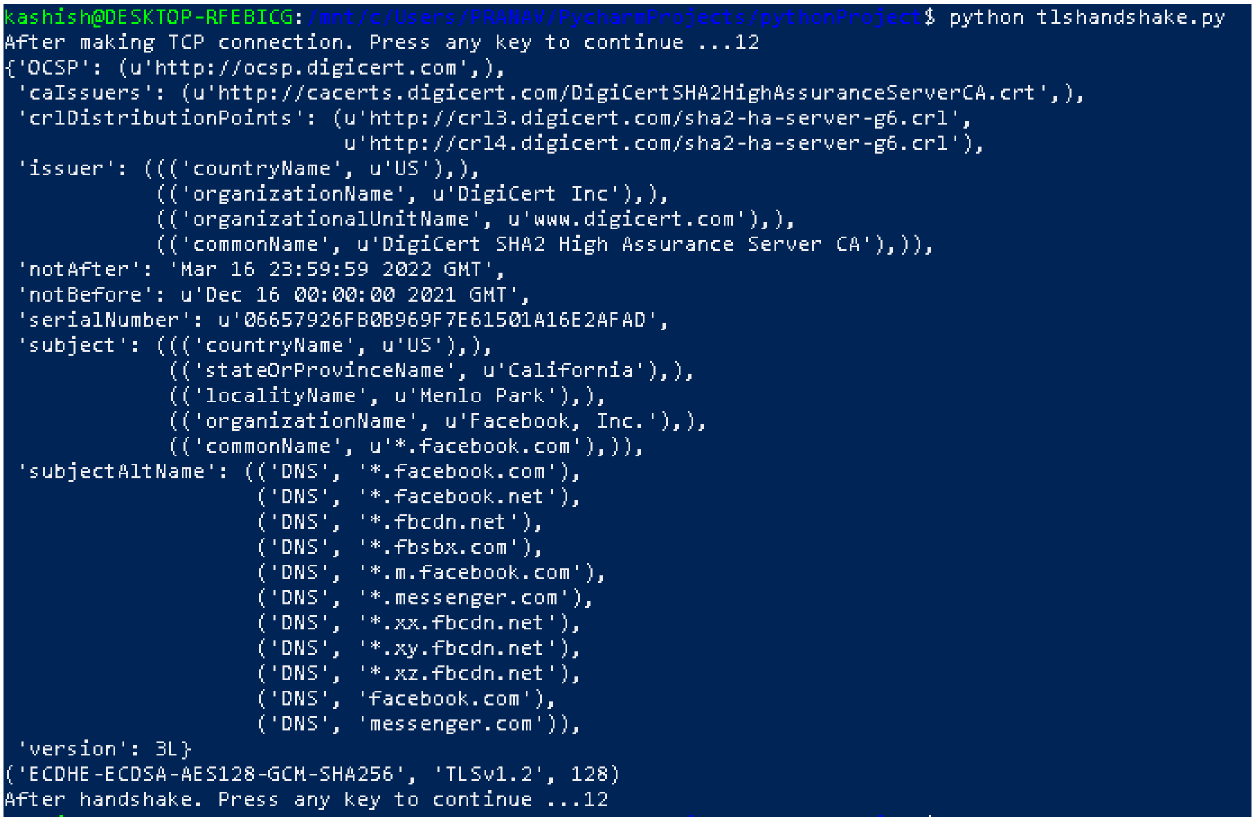
ASSIGNMENT-3

Kashish jain,2021jcs2240

PROBLEM-1

TASK-1

1.Cipher used between the client and server is as shown:



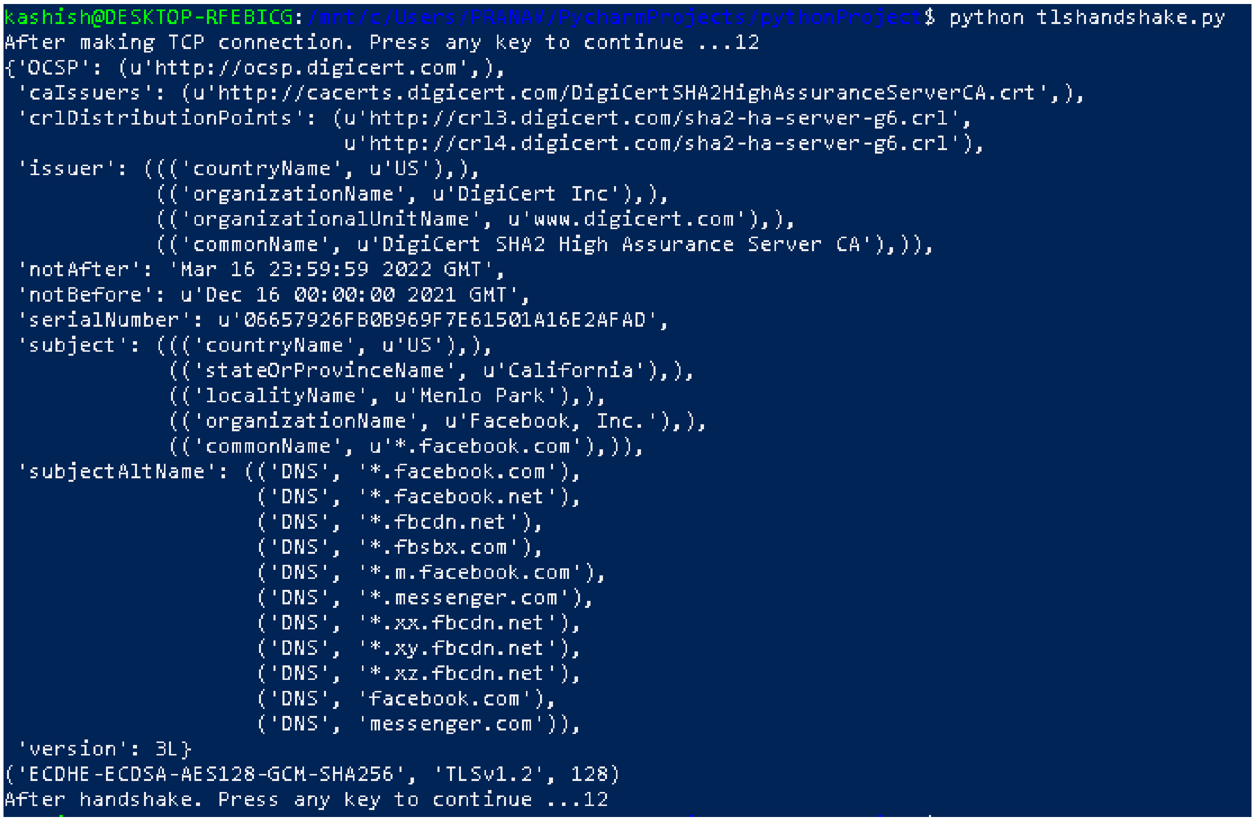
```
kashish@DESKTOP-RFEBICG:/mnt/c/Users/PRANAV/PycharmProjects/pythonProject$ python tlshandshake.py
After making TCP connection. Press any key to continue ...12
{'OCSP': (u'http://ocsp.digicert.com',),
 'caIssuers': (u'http://cacerts.digicert.com/DigiCertSHA2HighAssuranceServerCA.crt',),
 'crlDistributionPoints': (u'http://crl3.digicert.com/sha2-ha-server-g6.crl',
                           u'http://crl4.digicert.com/sha2-ha-server-g6.crl'),
 'issuer': ((('countryName', u'US'),),
            (('organizationName', u'DigiCert Inc'),),
            (('organizationalUnitName', u'www.digicert.com'),),
            (('commonName', u'DigiCert SHA2 High Assurance Server CA'),)),
 'notAfter': 'Mar 16 23:59:59 2022 GMT',
 'notBefore': u'Dec 16 00:00:00 2021 GMT',
 'serialNumber': u'06657926FB0B969F7E61501A16E2AFAD',
 'subject': ((('countryName', u'US'),),
             (('stateOrProvinceName', u'California'),),
             (('localityName', u'Menlo Park'),),
             (('organizationName', u'Facebook, Inc.'),),
             (('commonName', u'*.facebook.com'),)),
 'subjectAltName': (('DNS', '*.facebook.com'),
                    ('DNS', '*.facebook.net'),
                    ('DNS', '*.fbcdn.net'),
                    ('DNS', '*.fbsbx.com'),
                    ('DNS', '*.m.facebook.com'),
                    ('DNS', '*.messenger.com'),
                    ('DNS', '*.xx.fbcdn.net'),
                    ('DNS', '*.xy.fbcdn.net'),
                    ('DNS', '*.xz.fbcdn.net'),
                    ('DNS', 'facebook.com'),
                    ('DNS', 'messenger.com')),
 'version': 3L}
('ECDHE-ECDSA-AES128-GCM-SHA256', 'TLSv1.2', 128)
After handshake. Press any key to continue ...12
```

I used ssock.cipher() function to know about the cipher used.

So from : ‘ECDHE-ECDSA-AES128-GCM-SHA256’, ‘TLSv1.2’, 128, AES 128-GCM is used as encryption cipher.

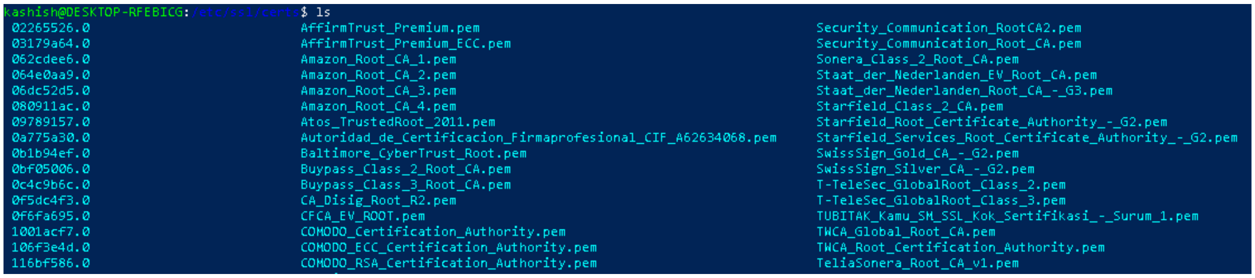
Upon connecting with google.com, I obtained this as:
‘TLS_AES_256_GCM_SHA384’, ‘TLSv1.3’,256

2. Server certificate is as shown:{‘OCSP’:…….3L}



```
kashish@DESKTOP-RFEBICG:/mnt/c/Users/PRANAV/PycharmProjects/pythonProject$ python tlshandshake.py
After making TCP connection. Press any key to continue ...12
{'OCSP': (u'http://ocsp.digicert.com',),
 'caIssuers': (u'http://cacerts.digicert.com/DigiCertSHA2HighAssuranceServerCA.crt',),
 'crlDistributionPoints': (u'http://crl3.digicert.com/sha2-ha-server-g6.crl',
                           u'http://crl4.digicert.com/sha2-ha-server-g6.crl'),
 'issuer': ((('countryName', u'US'),),
            (('organizationName', u'DigiCert Inc'),),
            (('organizationalUnitName', u'www.digicert.com'),),
            (('commonName', u'DigiCert SHA2 High Assurance Server CA'),)),
 'notAfter': 'Mar 16 23:59:59 2022 GMT',
 'notBefore': u'Dec 16 00:00:00 2021 GMT',
 'serialNumber': u'06657926FB0B969F7E61501A16E2AFAD',
 'subject': ((('countryName', u'US'),),
             (('stateOrProvinceName', u'California'),),
             (('localityName', u'Menlo Park'),),
             (('organizationName', u'Facebook, Inc.'),),
             (('commonName', u'*.facebook.com'),)),
 'subjectAltName': (('DNS', '*.facebook.com'),
                    ('DNS', '*.facebook.net'),
                    ('DNS', '*.fbcdn.net'),
                    ('DNS', '*.fbsbx.com'),
                    ('DNS', '*.m.facebook.com'),
                    ('DNS', '*.messenger.com'),
                    ('DNS', '*.xx.fbcdn.net'),
                    ('DNS', '*.xy.fbcdn.net'),
                    ('DNS', '*.xz.fbcdn.net'),
                    ('DNS', 'facebook.com'),
                    ('DNS', 'messenger.com')),
 'version': 3L}
('ECDHE-ECDSA-AES128-GCM-SHA256', 'TLSv1.2', 128)
After handshake. Press any key to continue ...12
```

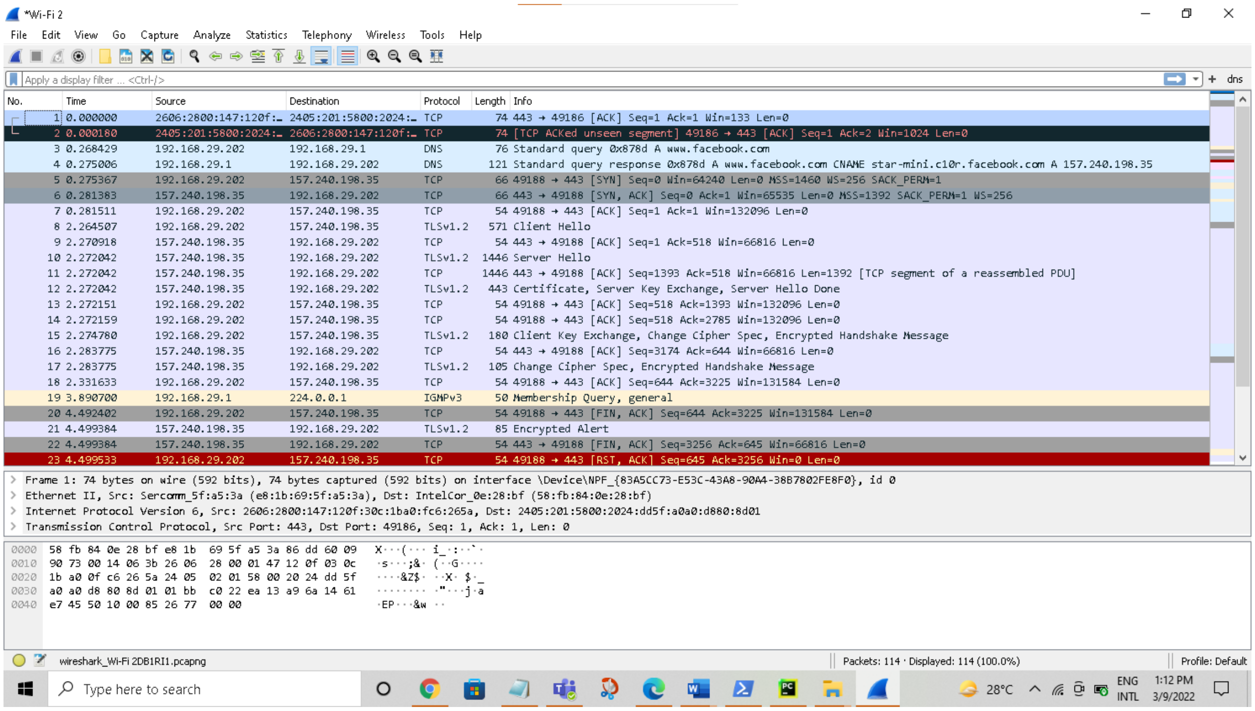
3. Purpose of /etc/ssl/certs is: etc/ssl/certs is the default location to install certficates of ssl, just like etc/ssl/private which is location to install all private keys. So its an OpenSSL compatible certificate directory. This location stores file of format .pem,.crt.



```
kashish@DESKTOP-RFEBICG:/etc/ssl/certs$ ls
 02265526.0                           AffirmTrust_Premium.pem                                           Security_Communication_RootCA2.pem
 03179a64.0                           AffirmTrust_Premium_ECC.pem                                       Security_Communication_Root_CA.pem
 062cdee6.0                           Amazon_Root_CA_1.pem                                              Sonera_Class_2_Root_CA.pem
 064e0aa9.0                           Amazon_Root_CA_2.pem                                              Staat_der_Nederlanden_EV_Root_CA.pem
 06dc52d5.0                           Amazon_Root_CA_3.pem                                              Staat_der_Nederlanden_Root_CA_-_G3.pem
 080911ac.0                           Amazon_Root_CA_4.pem                                              Starfield_Class_2_CA.pem
 09789157.0                           Atos_TrustedRoot_2011.pem                                         Starfield_Root_Certificate_Authority_-_G2.pem
 0a775a30.0                           Autoridad_de_Certificacion_Firmaprofesional_CIF_A62634068.pem     Starfield_Services_Root_Certificate_Authority_-_G2.pem
 0b1b94ef.0                           Baltimore_CyberTrust_Root.pem                                     SwissSign_Gold_CA_-_G2.pem
 0bf05006.0                           Buypass_Class_2_Root_CA.pem                                       SwissSign_Silver_CA_-_G2.pem
 0c4c9b6c.0                           Buypass_Class_3_Root_CA.pem                                       T-TeleSec_GlobalRoot_Class_2.pem
 0f5dc4f3.0                           CA_Disig_Root_R2.pem                                              T-TeleSec_GlobalRoot_Class_3.pem
 0f6fa695.0                           CFCA_EV_ROOT.pem                                                  TUBITAK_Kamu_SM_SSL_Kok_Sertifikasi_-_Surum_1.pem
 1001acf7.0                           COMODO_Certification_Authority.pem                                TWCA_Global_Root_CA.pem
 106f3e4d.0                           COMODO_ECC_Certification_Authority.pem                            TWCA_Root_Certification_Authority.pem
 116bf586.0                           COMODO_RSA_Certification_Authority.pem                            TeliaSonera_Root_CA_v1.pem
```

.pem format (concatenated certificate containers) represents entire certificate chain (private key, public key, root certificates).

4. Below is the screenshot of result obtained after using wireshark.


*Wi-Fi 2
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
dns
Profile: Default
Type here to search
28°C
ENG INTL
1:12 PM 3/9/2022

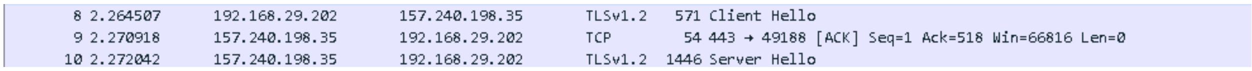
```
No.   Time      Source                 Destination            Protocol Length Info
    1 0.000000  2606:2800:147:120f:…  2405:201:5800:2024:…  TCP        74 443 → 49186 [ACK] Seq=1 Ack=1 Win=133 Len=0
    2 0.000180  2405:201:5800:2024:…  2606:2800:147:120f:…  TCP        74 [TCP ACKed unseen segment] 49186 → 443 [ACK] Seq=1 Ack=2 Win=1024 Len=0
    3 0.268429  192.168.29.202         192.168.29.1           DNS        76 Standard query 0x878d A www.facebook.com
    4 0.275006  192.168.29.1           192.168.29.202         DNS       121 Standard query response 0x878d A www.facebook.com CNAME star-mini.c10r.facebook.com A 157.240.198.35
    5 0.275367  192.168.29.202         157.240.198.35         TCP        66 49188 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
    6 0.281383  157.240.198.35         192.168.29.202         TCP        66 443 → 49188 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1392 SACK_PERM=1 WS=256
    7 0.281511  192.168.29.202         157.240.198.35         TCP        54 49188 → 443 [ACK] Seq=1 Ack=1 Win=132096 Len=0
    8 2.264507  192.168.29.202         157.240.198.35         TLSv1.2   571 Client Hello
    9 2.270918  157.240.198.35         192.168.29.202         TCP        54 443 → 49188 [ACK] Seq=1 Ack=518 Win=66816 Len=0
   10 2.272042  157.240.198.35         192.168.29.202         TLSv1.2  1446 Server Hello
   11 2.272042  157.240.198.35         192.168.29.202         TCP      1446 443 → 49188 [ACK] Seq=1393 Ack=518 Win=66816 Len=1392 [TCP segment of a reassembled PDU]
   12 2.272042  157.240.198.35         192.168.29.202         TLSv1.2   443 Certificate, Server Key Exchange, Server Hello Done
   13 2.272151  192.168.29.202         157.240.198.35         TCP        54 49188 → 443 [ACK] Seq=518 Ack=1393 Win=132096 Len=0
   14 2.272159  192.168.29.202         157.240.198.35         TCP        54 49188 → 443 [ACK] Seq=518 Ack=2785 Win=132096 Len=0
   15 2.274780  192.168.29.202         157.240.198.35         TLSv1.2   180 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
   16 2.283775  157.240.198.35         192.168.29.202         TCP        54 443 → 49188 [ACK] Seq=3174 Ack=644 Win=66816 Len=0
   17 2.283775  157.240.198.35         192.168.29.202         TLSv1.2   105 Change Cipher Spec, Encrypted Handshake Message
   18 2.331633  192.168.29.202         157.240.198.35         TCP        54 49188 → 443 [ACK] Seq=644 Ack=3225 Win=131584 Len=0
   19 3.890700  192.168.29.1           224.0.0.1              IGMPv3     50 Membership Query, general
   20 4.492402  192.168.29.202         157.240.198.35         TCP        54 49188 → 443 [FIN, ACK] Seq=644 Ack=3225 Win=131584 Len=0
   21 4.499384  157.240.198.35         192.168.29.202         TLSv1.2    85 Encrypted Alert
   22 4.499384  157.240.198.35         192.168.29.202         TCP        54 443 → 49188 [FIN, ACK] Seq=3256 Ack=645 Win=66816 Len=0
   23 4.499533  192.168.29.202         157.240.198.35         TCP        54 49188 → 443 [RST, ACK] Seq=645 Ack=3256 Win=0 Len=0
```

```
> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{83A5CC73-E53C-43A8-90A4-38B7802FE8F0}, id 0
> Ethernet II, Src: Sercomm_5f:a5:3a (e8:1b:69:5f:a5:3a), Dst: IntelCor_0e:28:bf (58:fb:84:0e:28:bf)
> Internet Protocol Version 6, Src: 2606:2800:147:120f:30c:1ba0:fc6:265a, Dst: 2405:201:5800:2024:dd5f:a0a0:d880:8d01
> Transmission Control Protocol, Src Port: 443, Dst Port: 49186, Seq: 1, Ack: 1, Len: 0
```

```
0000  58 fb 84 0e 28 bf e8 1b  69 5f a5 3a 86 dd 60 09   X···(··· i_·:··`·
0010  90 73 00 14 06 3b 26 06  28 00 01 47 12 0f 03 0c   ·s···;&· (··G····
0020  1b a0 0f c6 26 5a 24 05  02 01 58 00 20 24 dd 5f   ····&Z$· ··X· $·_
0030  a0 a0 d8 80 8d 01 01 bb  c0 22 ea 13 a9 6a 14 61   ········ ·"···j·a
0040  e7 45 50 10 00 85 26 77  00 00                     ·EP···&w ··
```

```
wireshark_Wi-Fi 2DB1RI1.pcapng          Packets: 114 · Displayed: 114 (100.0%)
```

We can observe from above results TCP handshake is done first before TLS handshake. And TCP handshake is 3 way handshake which includes transfer of SYN,ACK. It initiated from line 5 in above ss.



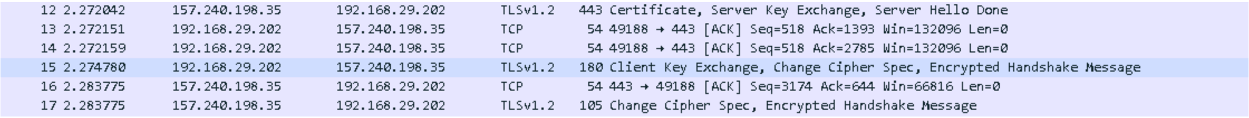
```
    8 2.264507  192.168.29.202   157.240.198.35   TLSv1.2   571 Client Hello
    9 2.270918  157.240.198.35   192.168.29.202   TCP        54 443 → 49188 [ACK] Seq=1 Ack=518 Win=66816 Len=0
   10 2.272042  157.240.198.35   192.168.29.202   TLSv1.2  1446 Server Hello
```

We can observe from the obtained results that:

1.TLS handshake initiated from line 8 in above screenshot.

In TLS handshake, client hello is the first message that client sends to server.

2.Then server responds with many messages, such as server hello message. Also server sends its x.509 certificates to client to authenticate itself.

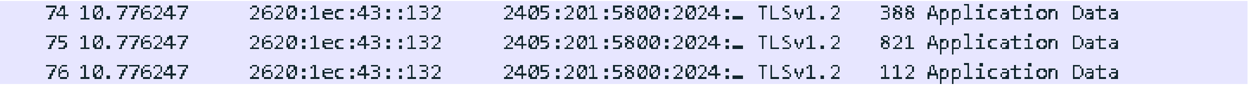


```
   12 2.272042  157.240.198.35   192.168.29.202   TLSv1.2   443 Certificate, Server Key Exchange, Server Hello Done
   13 2.272151  192.168.29.202   157.240.198.35   TCP        54 49188 → 443 [ACK] Seq=518 Ack=1393 Win=132096 Len=0
   14 2.272159  192.168.29.202   157.240.198.35   TCP        54 49188 → 443 [ACK] Seq=518 Ack=2785 Win=132096 Len=0
   15 2.274780  192.168.29.202   157.240.198.35   TLSv1.2   180 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
   16 2.283775  157.240.198.35   192.168.29.202   TCP        54 443 → 49188 [ACK] Seq=3174 Ack=644 Win=66816 Len=0
   17 2.283775  157.240.198.35   192.168.29.202   TLSv1.2   105 Change Cipher Spec, Encrypted Handshake Message
```

3.Then clients responds with many messages. If server certificate is valid ie not expired then client sends a client key exchange and a change cipher spec message and Encrypted handshake message. Encrypted handshake message is the message which indicates client wants to end this TLS negotiation, also known as finished message.

4.After this server also responds with change cipher spec message and Encrypted handshake message or finished message, which lets client fully authenticate the server.

5.After the completion of TLS handshake, authenticated peers can start sending their data to each other.

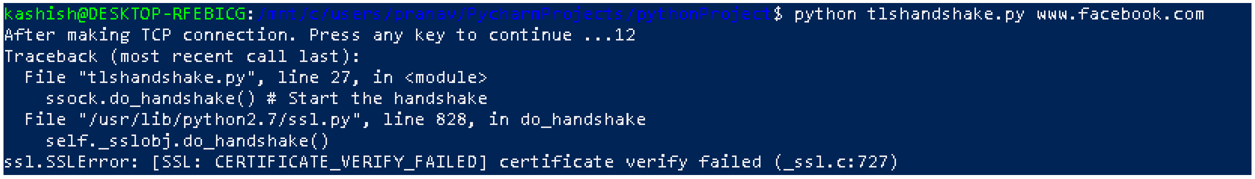


```
   74 10.776247     2620:1ec:43::132     2405:201:5800:2024:… TLSv1.2   388 Application Data
   75 10.776247     2620:1ec:43::132     2405:201:5800:2024:… TLSv1.2   821 Application Data
   76 10.776247     2620:1ec:43::132     2405:201:5800:2024:… TLSv1.2   112 Application Data
```

TLS handshake always happens after TCP handshake. So after the opening of TCP connection, TLS handshake takes place. TCP handshake is a 3 way handshake of SYN,SYN/ACK,ACK which ensures the successful opening of connection whereas in TLS handshake client and server agrees on TLS version, chooses their cipher suite, exchanges their certificates .

TASK-2:

1. After assigning cadir=./certs , I obtained this result:



```
kashish@DESKTOP-RFEBICG:/mnt/c/users/pranav/PycharmProjects/pythonProject$ python tlshandshake.py www.facebook.com
After making TCP connection. Press any key to continue ...12
Traceback (most recent call last):
  File "tlshandshake.py", line 27, in <module>
    ssock.do_handshake() # Start the handshake
  File "/usr/lib/python2.7/ssl.py", line 828, in do_handshake
    self._sslobj.do_handshake()
ssl.SSLError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed (_ssl.c:727)
```

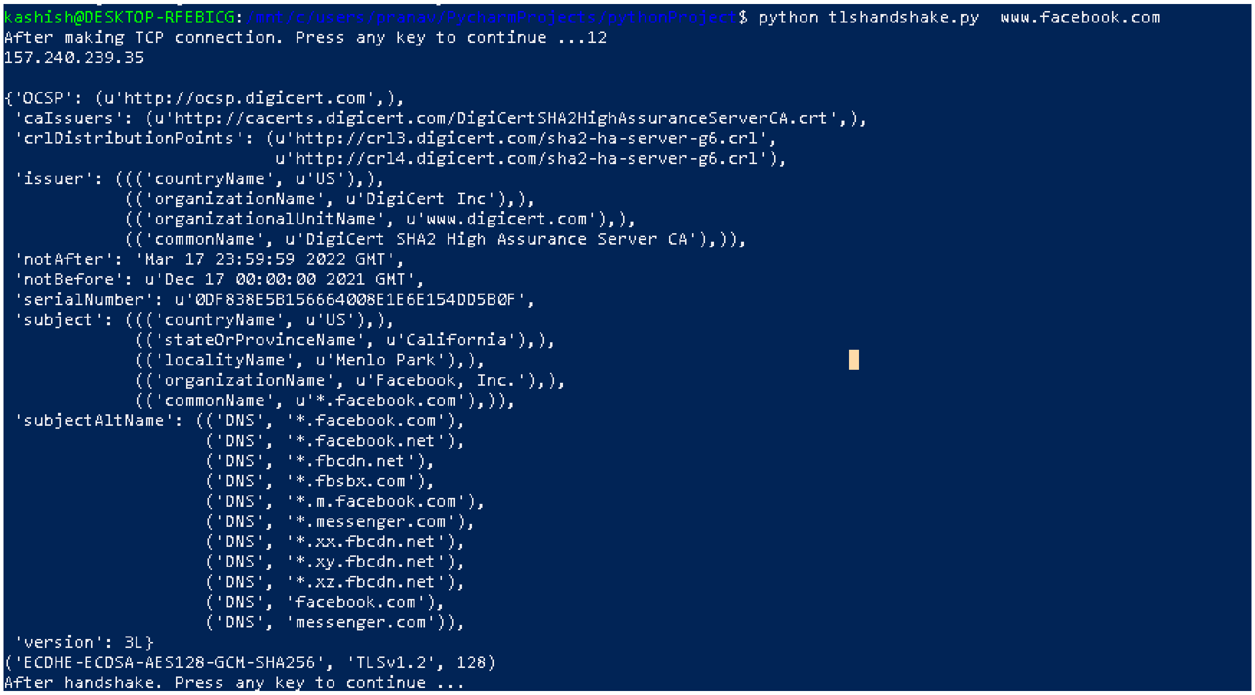
Because all certificates are present in etc/ssl/certs, so after changing cadir to /certs which is an empty folder, server certificates could not be verified (as CA certificate and public key is required to verify), Hence it showed me above result.

2. After observing obtained results for the server facebook.com, I found that I need to put- DigiCert SHA2 High Assurance Server CA.pem in /certs folder. After that my client program is running. Basically public key of CA certificate is required to verify the CA digital signature which is present in the server certificate.

TASK-3

1. To know the IP address of server using dig command, I added following code to my python client program on windows:

```
cmd='dig facebook.com +short'
proc=subprocess.Popen(shlex.split(cmd),stdout=subprocess.PIPE)
out,err=proc.communicate()
print(out)
```



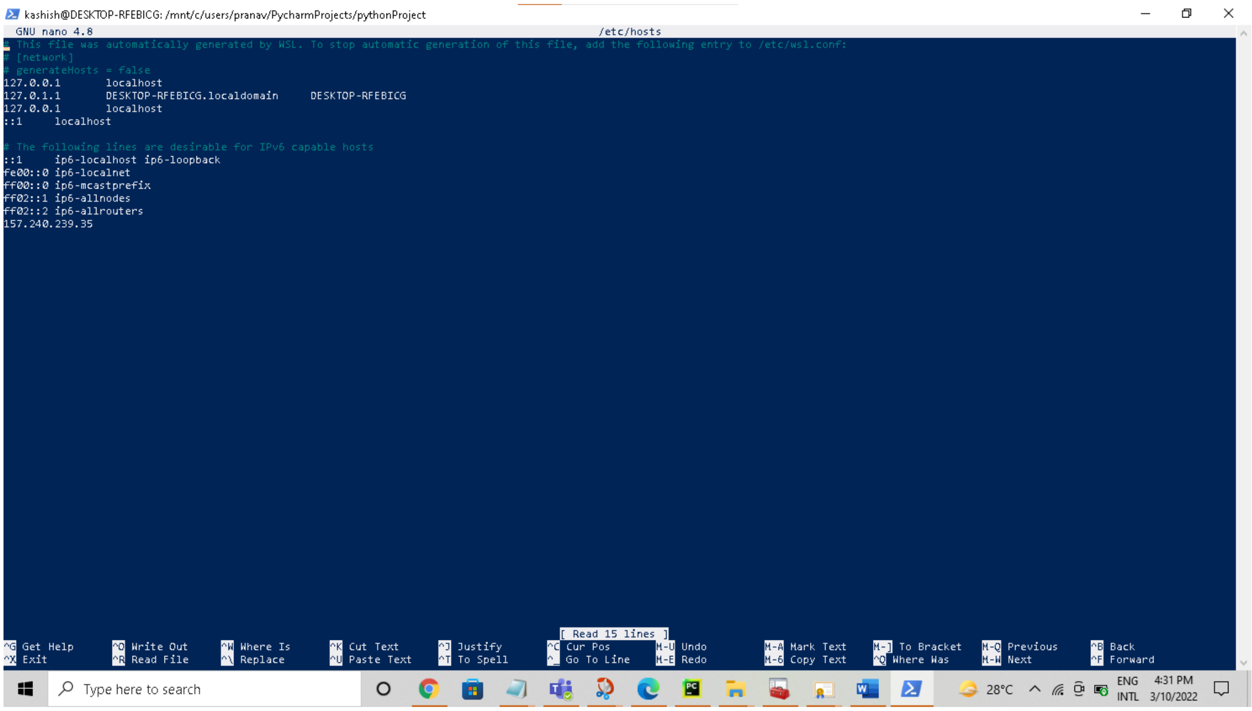
```
kashish@DESKTOP-RFEBICG:/mnt/c/users/pranav/PycharmProjects/pythonProject$ python tlshandshake.py  www.facebook.com
After making TCP connection. Press any key to continue ...12
157.240.239.35

{'OCSP': (u'http://ocsp.digicert.com',),
 'caIssuers': (u'http://cacerts.digicert.com/DigiCertSHA2HighAssuranceServerCA.crt',),
 'crlDistributionPoints': (u'http://crl3.digicert.com/sha2-ha-server-g6.crl',
                           u'http://crl4.digicert.com/sha2-ha-server-g6.crl'),
 'issuer': ((('countryName', u'US'),),
            (('organizationName', u'DigiCert Inc'),),
            (('organizationalUnitName', u'www.digicert.com'),),
            (('commonName', u'DigiCert SHA2 High Assurance Server CA'),)),
 'notAfter': 'Mar 17 23:59:59 2022 GMT',
 'notBefore': u'Dec 17 00:00:00 2021 GMT',
 'serialNumber': u'0DF838E5B156664008E1E6E154DD5B0F',
 'subject': ((('countryName', u'US'),),
             (('stateOrProvinceName', u'California'),),
             (('localityName', u'Menlo Park'),),
             (('organizationName', u'Facebook, Inc.'),),
             (('commonName', u'*.facebook.com'),)),
 'subjectAltName': (('DNS', '*.facebook.com'),
                    ('DNS', '*.facebook.net'),
                    ('DNS', '*.fbcdn.net'),
                    ('DNS', '*.fbsbx.com'),
                    ('DNS', '*.m.facebook.com'),
                    ('DNS', '*.messenger.com'),
                    ('DNS', '*.xx.fbcdn.net'),
                    ('DNS', '*.xy.fbcdn.net'),
                    ('DNS', '*.xz.fbcdn.net'),
                    ('DNS', 'facebook.com'),
                    ('DNS', 'messenger.com')),
 'version': 3L}
('ECDHE-ECDSA-AES128-GCM-SHA256', 'TLSv1.2', 128)
After handshake. Press any key to continue ...
```

Second line :157.240.239.35, in about output is IP address of facebook.com.

2.The etc/hosts file, contains a list of IP host names and their corresponding IP addresses. This file is used to resolve host names to an address. Using following codes in my python program, I wrote IP address of server in /etc/hosts file. Because it is a protected file and access was showing permission denied, so I had to use below code.

```
with open('/etc/hosts', 'rt') as f:
    s = f.read() + out
    with open('/tmp/etc_hosts.tmp', 'wt') as outf:
        outf.write(s)
```

Also using sudo nano etc/hosts command in ubuntu, I opened etc/hosts file. We can see IP address of facebook.com: 157.240.239.35 , written in the last line of file.



kashish@DESKTOP-RFEBICG: /mnt/c/users/pranav/PycharmProjects/pythonProject
Type here to search
28°C
ENG
INTL
4:31 PM
3/10/2022

```
  GNU nano 4.8                                          /etc/hosts
# This file was automatically generated by WSL. To stop automatic generation of this file, add the following entry to /etc/wsl.conf:
# [network]
# generateHosts = false
127.0.0.1       localhost
127.0.1.1       DESKTOP-RFEBICG.localdomain     DESKTOP-RFEBICG
127.0.0.1       localhost
::1     localhost

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
157.240.239.35

                                                     [ Read 15 lines ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text  M-] To Bracket M-Q Previous   ^B Back
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text  ^Q Where Was   M-W Next       ^F Forward
```

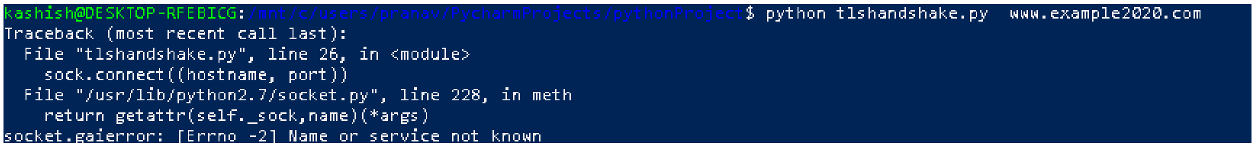
3. context.check_hostname() =False, ensures no verification of hostname ,which leads to successful verification of wrong certificates.

context.check_hostname()=True, ensures verification of hostname ,which always leads to unsuccessful verification of wrong certificates.

When check_hostname is True, sslsocket.do_handshake() performs match_hostname().Also to check authenticity of certificate , check_hostname must be set True. If check_hostname is set False, then by default host will not be matched against the hostname allowed by the server certificate.

Security consequences: If check_hostname is set False, then Attackers will be able to cheat users through their fake websites as clients/users connection would not have verified the hostname of their websites. Attackers will be able to use authorized certificates of other websites as certificates of their fake website and will use them for getting authenticated from clients.

Upon connecting to www.example2020.com, I obtained following results:



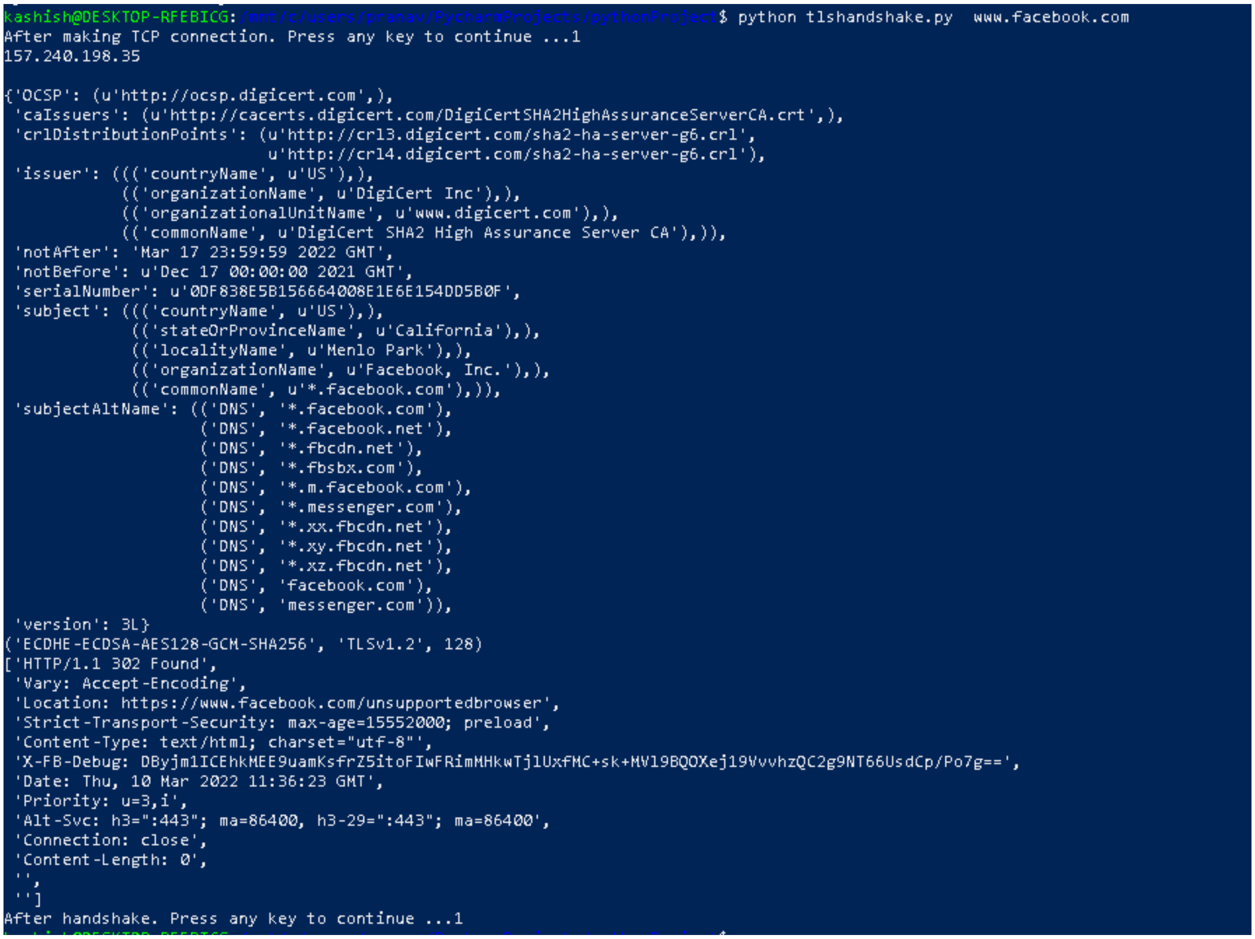
```
kashish@DESKTOP-RFEBICG:/mnt/c/users/pranav/PycharmProjects/pythonProject$ python tlshandshake.py  www.example2020.com
Traceback (most recent call last):
  File "tlshandshake.py", line 26, in <module>
    sock.connect((hostname, port))
  File "/usr/lib/python2.7/socket.py", line 228, in meth
    return getattr(self._sock,name)(*args)
socket.gaierror: [Errno -2] Name or service not known
```

Also when I type URL :www.example2020.com on google, I got 404 Page not found error. But for other servers like google, facebook On making context.check_hostname=False, or True, I found no change in obtained results.

TASK-4

1. After adding following code to client program, I obtained foll. results:

```
request = b"GET / HTTP/1.0\r\nHost: " + \
hostname.encode(’utf-8’) + b"\r\n\r\n"
ssock.sendall(request)
# Read HTTP Response from Server
response = ssock.recv(2048)
while response:
    pprint.pprint(response.split(b"\r\n"))
    response = ssock.recv(2048)
```

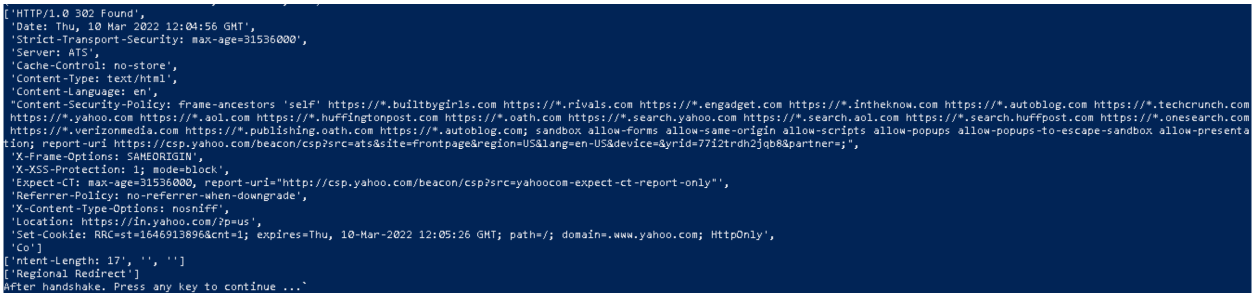


```
kashish@DESKTOP-RFEBICG:/mnt/c/users/pranav/PycharmProjects/pythonProject$ python tlshandshake.py  www.facebook.com
After making TCP connection. Press any key to continue ...1
157.240.198.35

{'OCSP': (u'http://ocsp.digicert.com',),
 'caIssuers': (u'http://cacerts.digicert.com/DigiCertSHA2HighAssuranceServerCA.crt',),
 'crlDistributionPoints': (u'http://crl3.digicert.com/sha2-ha-server-g6.crl',
                           u'http://crl4.digicert.com/sha2-ha-server-g6.crl'),
 'issuer': ((('countryName', u'US'),),
            (('organizationName', u'DigiCert Inc'),),
            (('organizationalUnitName', u'www.digicert.com'),),
            (('commonName', u'DigiCert SHA2 High Assurance Server CA'),)),
 'notAfter': 'Mar 17 23:59:59 2022 GMT',
 'notBefore': u'Dec 17 00:00:00 2021 GMT',
 'serialNumber': u'0DF838E5B156664008E1E6E154DD5B0F',
 'subject': ((('countryName', u'US'),),
             (('stateOrProvinceName', u'California'),),
             (('localityName', u'Menlo Park'),),
             (('organizationName', u'Facebook, Inc.'),),
             (('commonName', u'*.facebook.com'),)),
 'subjectAltName': (('DNS', '*.facebook.com'),
                    ('DNS', '*.facebook.net'),
                    ('DNS', '*.fbcdn.net'),
                    ('DNS', '*.fbsbx.com'),
                    ('DNS', '*.m.facebook.com'),
                    ('DNS', '*.messenger.com'),
                    ('DNS', '*.xx.fbcdn.net'),
                    ('DNS', '*.xy.fbcdn.net'),
                    ('DNS', '*.xz.fbcdn.net'),
                    ('DNS', 'facebook.com'),
                    ('DNS', 'messenger.com')),
 'version': 3L}
('ECDHE-ECDSA-AES128-GCM-SHA256', 'TLSv1.2', 128)
['HTTP/1.1 302 Found',
 'Vary: Accept-Encoding',
 'Location: https://www.facebook.com/unsupportedbrowser',
 'Strict-Transport-Security: max-age=15552000; preload',
 'Content-Type: text/html; charset="utf-8"',
 'X-FB-Debug: DByjm1ICEhkMEE9uamKsfrZ5itoFIwFRimMHkwTj1UxfMC+sk+MV19BQOXej19VvvhzQC2g9NT66UsdCp/Po7g==',
 'Date: Thu, 10 Mar 2022 11:36:23 GMT',
 'Priority: u=3,i',
 'Alt-Svc: h3=":443"; ma=86400, h3-29=":443"; ma=86400',
 'Connection: close',
 'Content-Length: 0',
 '',
 '']
After handshake. Press any key to continue ...1
```

In Above obtained results: [‘HTTP/1.1 302 found,…………’, ‘’] is the response received from server. This 302 found shows that the resource requested has been temporarily moved to the URL given by the location header. Location header is: ‘https://www.facebook.com/unsupportedbrowser’.
So this location header gives the new location of resource.

302 shows a redirect to a temporary location but its not an error.

Also on giving hostname as www.yahoo.com , I obtained same result:



```
['HTTP/1.0 302 Found',
 'Date: Thu, 10 Mar 2022 12:04:56 GMT',
 'Strict-Transport-Security: max-age=31536000',
 'Server: ATS',
 'Cache-Control: no-store',
 'Content-Type: text/html',
 'Content-Language: en',
 "Content-Security-Policy: frame-ancestors 'self' https://*.builtbygirls.com https://*.rivals.com https://*.engadget.com https://*.intheknow.com https://*.autoblog.com https://*.techcrunch.com
 https://*.yahoo.com https://*.aol.com https://*.huffingtonpost.com https://*.oath.com https://*.search.yahoo.com https://*.search.aol.com https://*.search.huffpost.com https://*.onesearch.com
 https://*.verizonmedia.com https://*.publishing.oath.com https://*.autoblog.com; sandbox allow-forms allow-same-origin allow-scripts allow-popups allow-popups-to-escape-sandbox allow-presenta
tion; report-uri https://csp.yahoo.com/beacon/csp?src=ats&site=frontpage&region=US&lang=en-US&device=&yrid=77i2trdh2jqb8&partner=;",
 'X-Frame-Options: SAMEORIGIN',
 'X-XSS-Protection: 1; mode=block',
 'Expect-CT: max-age=31536000, report-uri="http://csp.yahoo.com/beacon/csp?src=yahoocom-expect-ct-report-only"',
 'Referrer-Policy: no-referrer-when-downgrade',
 'X-Content-Type-Options: nosniff',
 'Location: https://in.yahoo.com/?p=us',
 'Set-Cookie: RRC=st=1646913896&cnt=1; expires=Thu, 10-Mar-2022 12:05:26 GMT; path=/; domain=.www.yahoo.com; HttpOnly',
 'Co']
['ntent-Length: 17', '', '']
['Regional Redirect']
After handshake. Press any key to continue ...`
```

SECURITY:

Public keys of server are authenticated using their x.509 certificates, thus the prototype is secured against Man in middle attack.

Also for creating sockets, ssl sockets are used which uses session ID which is a randomly generated unique identifier for a session, also client_hello, server_hello messages uses random number nonce ,hence prototype is secured against Replay attacks.

Downgrade attacks can break the security of system by allowing the attacker to negotiate the use of lower version of TLS. I have set the ssl_version as PROTOCOL_TLSv1_2 ie: TLS 1.2 , Hence downgrade attacks are not possible on this prototype.