Name: Chang Pao Herr
Student ID: 19544
Course Term: Summer 2020 – CS535 - NETWORK SECURITY FUNDAMENTAL
Instructor: Dr. Chang Henry

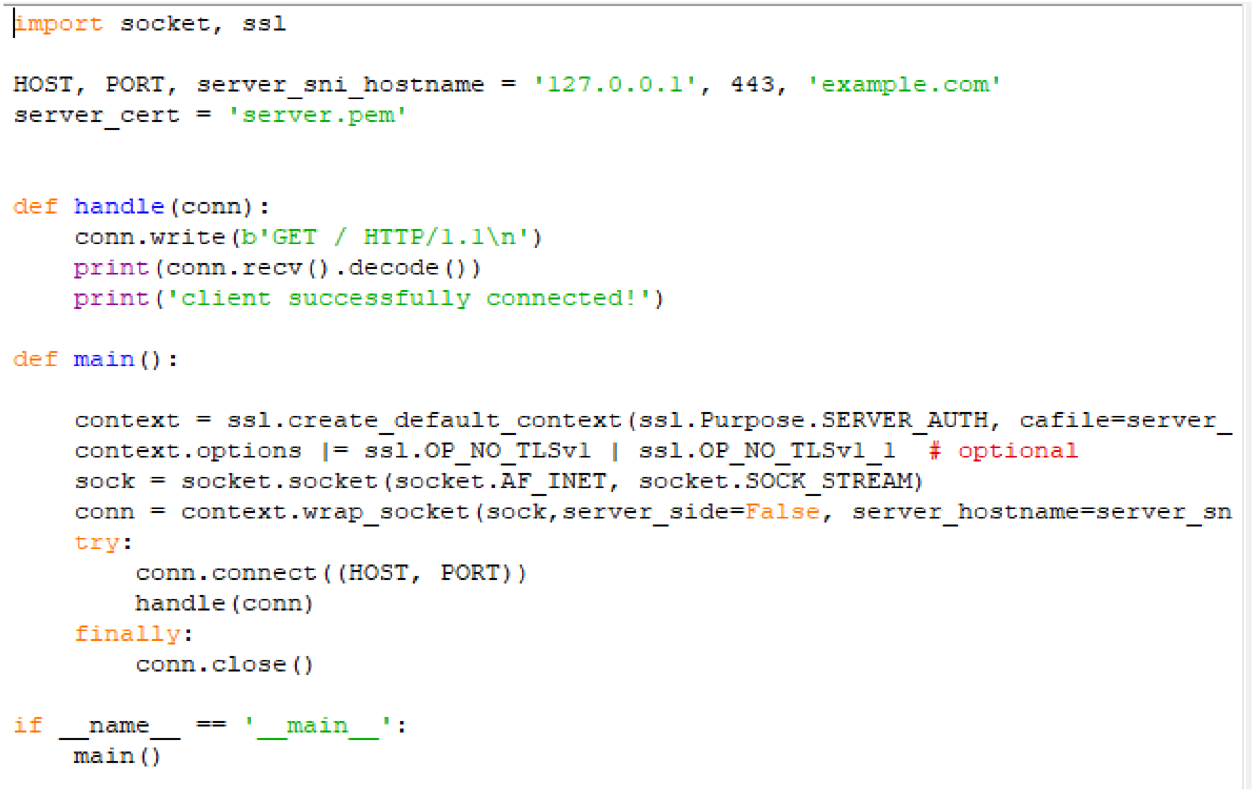
Week# 11 Homework#: 9
Due Date: 7/28/2020 11:30:00 PM
Homework Subject: **Project HTTPS (I)**

Part#1 Question #3: HTTP is not secured

Project Part 1: HTTP is not secured

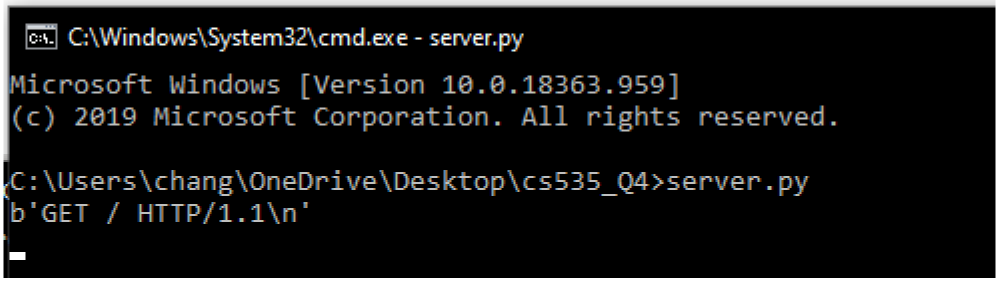
- References
  - Part 1: HTTP is not secured
  - Answers - 2020 Summer

Code: Server.py



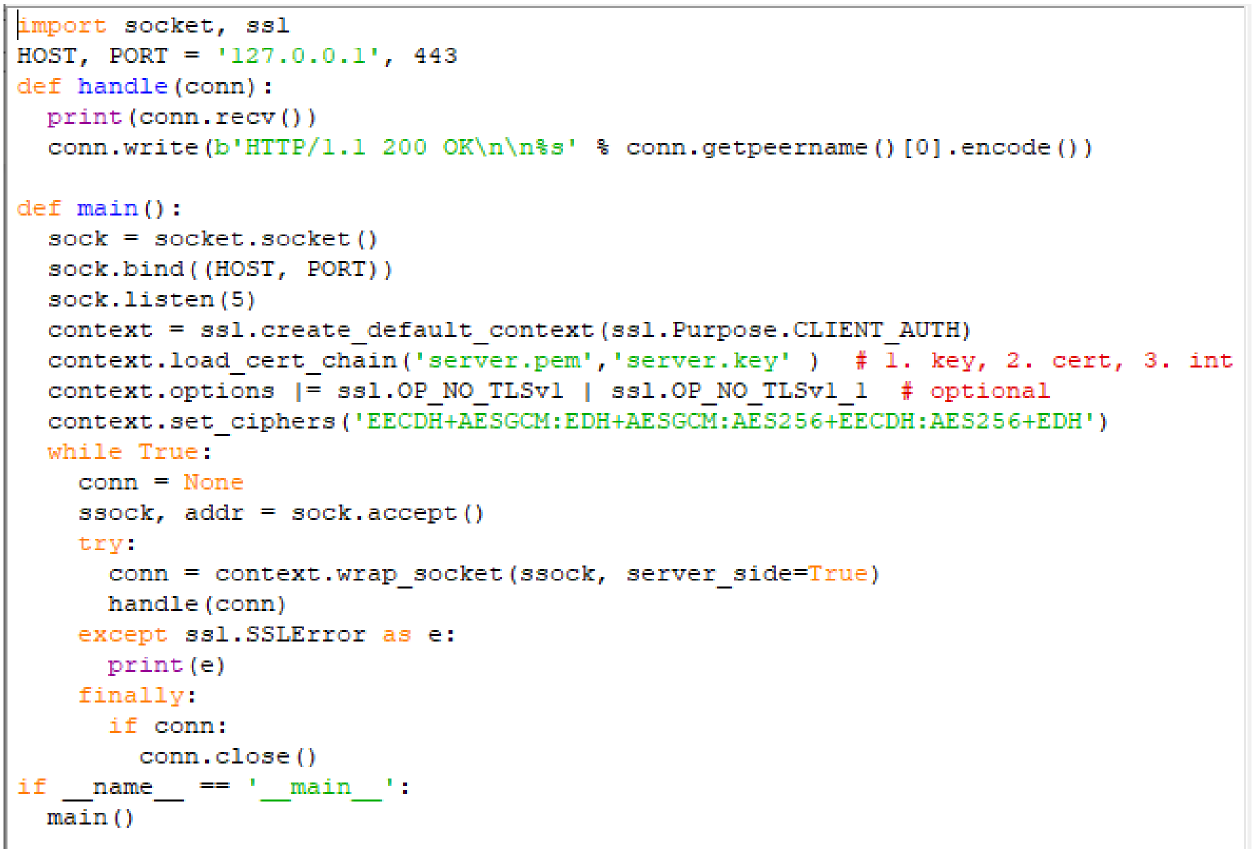
```
import socket, ssl

HOST, PORT, server_sni_hostname = '127.0.0.1', 443, 'example.com'
server_cert = 'server.pem'


def handle(conn):
    conn.write(b'GET / HTTP/1.1\n')
    print(conn.recv().decode())
    print('client successfully connected!')

def main():

    context = ssl.create_default_context(ssl.Purpose.SERVER_AUTH, cafile=server_
    context.options |= ssl.OP_NO_TLSv1 | ssl.OP_NO_TLSv1_1  # optional
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    conn = context.wrap_socket(sock,server_side=False, server_hostname=server_sn
    try:
        conn.connect((HOST, PORT))
        handle(conn)
    finally:
        conn.close()

if __name__ == '__main__':
    main()
```

Server executes display:



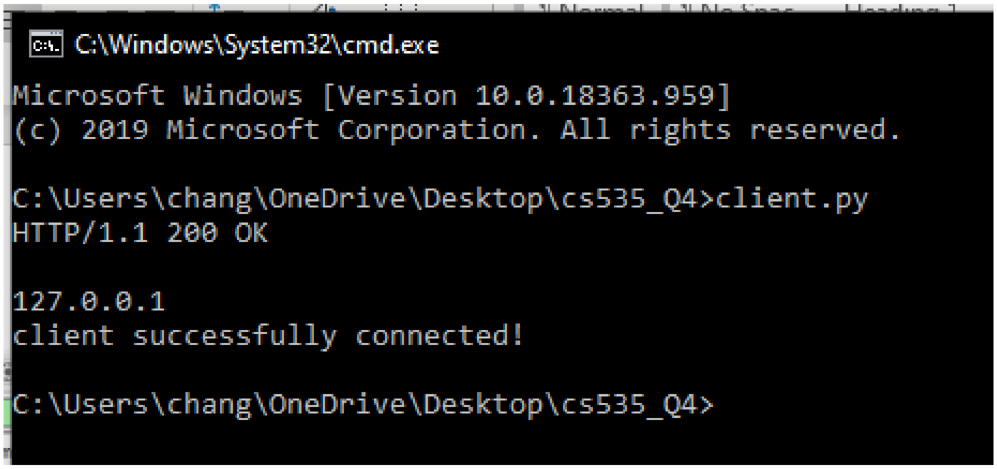
```
Microsoft Windows [Version 10.0.18363.959]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\chang\OneDrive\Desktop\cs535_Q4>server.py
b'GET / HTTP/1.1\n'
```

Code: Client.py



```
import socket, ssl
HOST, PORT = '127.0.0.1', 443
def handle(conn):
  print(conn.recv())
  conn.write(b'HTTP/1.1 200 OK\n\n%s' % conn.getpeername()[0].encode())

def main():
  sock = socket.socket()
  sock.bind((HOST, PORT))
  sock.listen(5)
  context = ssl.create_default_context(ssl.Purpose.CLIENT_AUTH)
  context.load_cert_chain('server.pem','server.key' )  # 1. key, 2. cert, 3. int
  context.options |= ssl.OP_NO_TLSv1 | ssl.OP_NO_TLSv1_1  # optional
  context.set_ciphers('EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH')
  while True:
    conn = None
    ssock, addr = sock.accept()
    try:
      conn = context.wrap_socket(ssock, server_side=True)
      handle(conn)
    except ssl.SSLError as e:
      print(e)
    finally:
      if conn:
        conn.close()
if __name__ == '__main__':
  main()
```

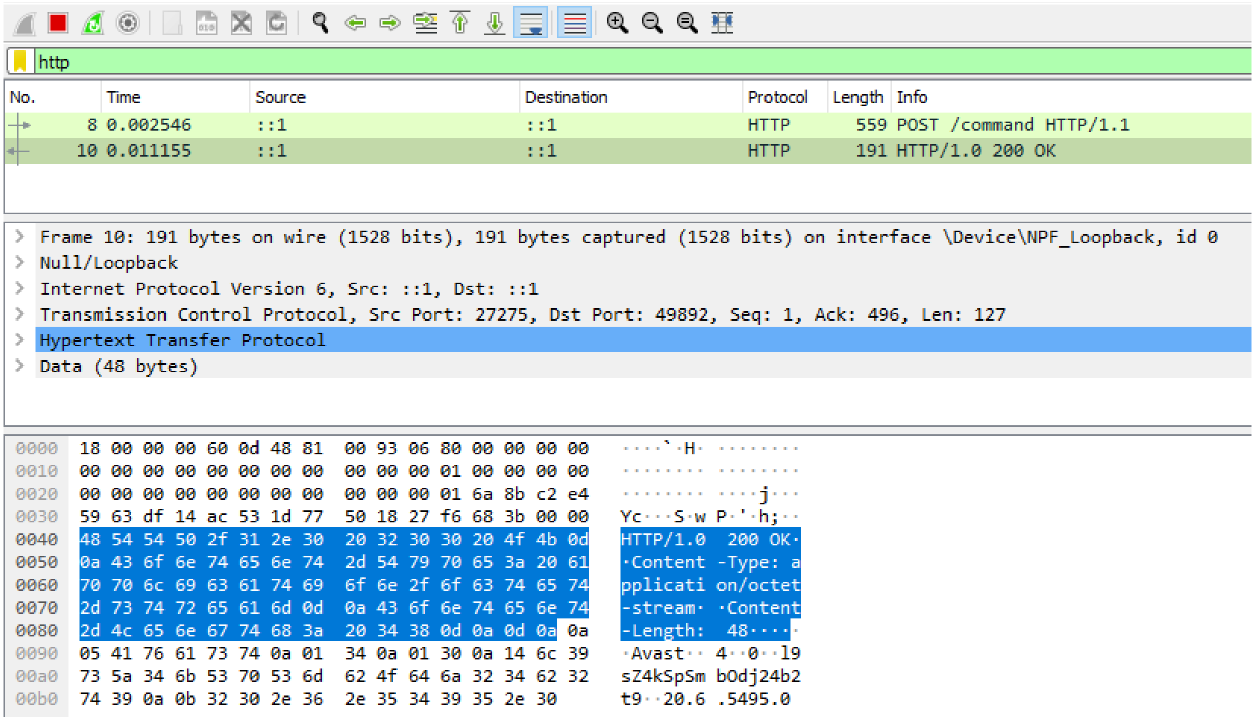
Client.py executes display.



C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.18363.959]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\chang\OneDrive\Desktop\cs535_Q4>client.py
HTTP/1.1 200 OK

127.0.0.1
client successfully connected!

C:\Users\chang\OneDrive\Desktop\cs535_Q4>
```

Wireshark Capture display HTTP protocol is response to request



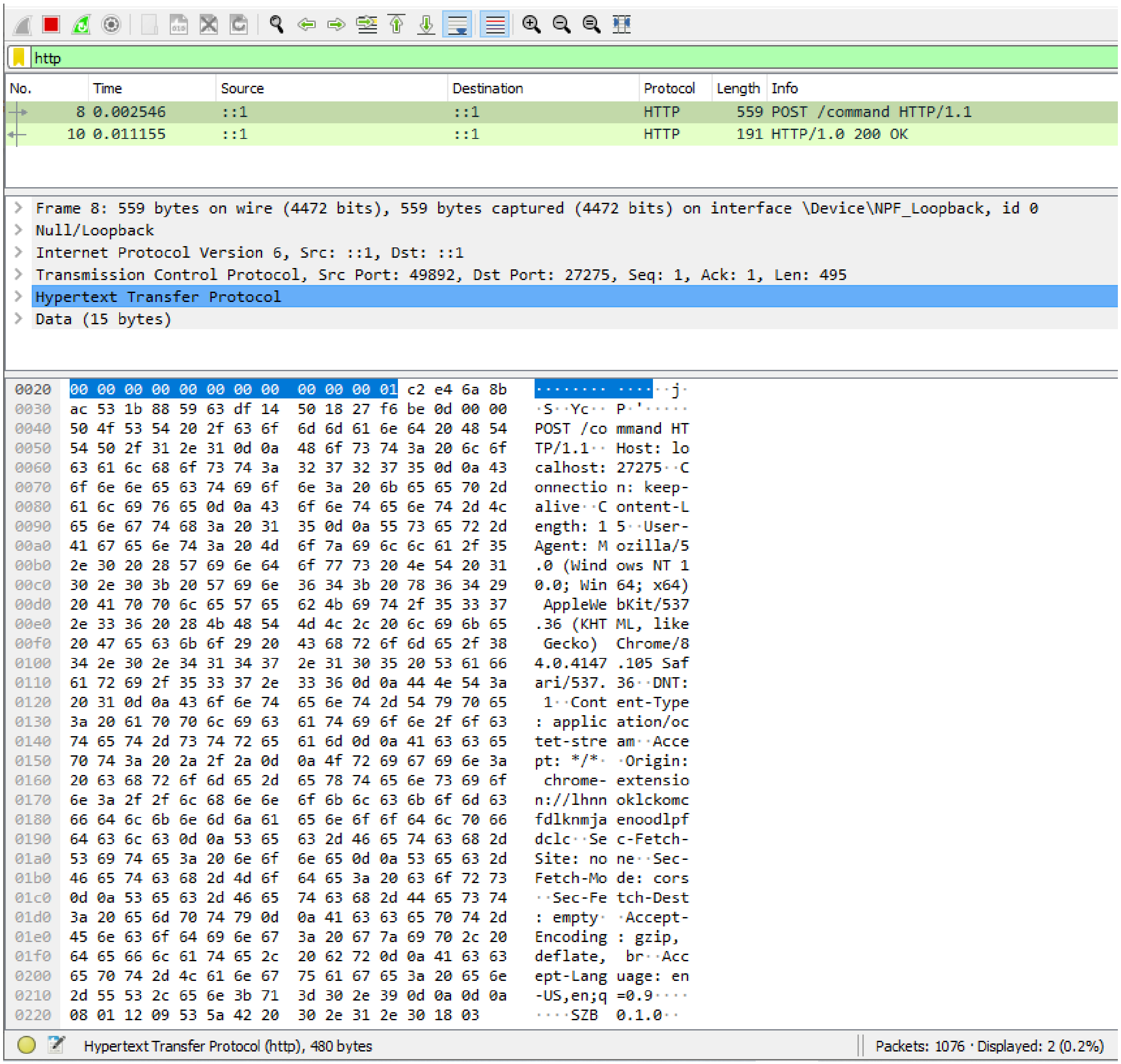
```
http
```

```
No.    Time        Source     Destination   Protocol  Length  Info
    8  0.002546    ::1        ::1           HTTP         559  POST /command HTTP/1.1
   10  0.011155    ::1        ::1           HTTP         191  HTTP/1.0 200 OK
```

```
> Frame 10: 191 bytes on wire (1528 bits), 191 bytes captured (1528 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 6, Src: ::1, Dst: ::1
> Transmission Control Protocol, Src Port: 27275, Dst Port: 49892, Seq: 1, Ack: 496, Len: 127
> Hypertext Transfer Protocol
> Data (48 bytes)
```

```
0000  18 00 00 00 60 0d 48 81  00 93 06 80 00 00 00 00   ····`·H· ········
0010  00 00 00 00 00 00 00 00  00 00 00 01 00 00 00 00   ········ ········
0020  00 00 00 00 00 00 00 00  00 00 00 01 6a 8b c2 e4   ········ ····j···
0030  59 63 df 14 ac 53 1d 77  50 18 27 f6 68 3b 00 00   Yc···S·w P·'·h;··
0040  48 54 54 50 2f 31 2e 30  20 32 30 30 20 4f 4b 0d   HTTP/1.0  200 OK·
0050  0a 43 6f 6e 74 65 6e 74  2d 54 79 70 65 3a 20 61   ·Content -Type: a
0060  70 70 6c 69 63 61 74 69  6f 6e 2f 6f 63 74 65 74   pplicati on/octet
0070  2d 73 74 72 65 61 6d 0d  0a 43 6f 6e 74 65 6e 74   -stream· ·Content
0080  2d 4c 65 6e 67 74 68 3a  20 34 38 0d 0a 0d 0a 0a   -Length:  48·····
0090  05 41 76 61 73 74 0a 01  34 0a 01 30 0a 14 6c 39   ·Avast·· 4··0··l9
00a0  73 5a 34 6b 53 70 53 6d  62 4f 64 6a 32 34 62 32   sZ4kSpSm bOdj24b2
00b0  74 39 0a 0b 32 30 2e 36  2e 35 34 39 35 2e 30      t9··20.6 .5495.0
```

Wireshark Capture display server.py request to server.py



```
No.    Time        Source                    Destination          Protocol  Length Info
     8 0.002546    ::1                       ::1                  HTTP         559 POST /command HTTP/1.1
    10 0.011155    ::1                       ::1                  HTTP         191 HTTP/1.0 200 OK
```

```
> Frame 8: 559 bytes on wire (4472 bits), 559 bytes captured (4472 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 6, Src: ::1, Dst: ::1
> Transmission Control Protocol, Src Port: 49892, Dst Port: 27275, Seq: 1, Ack: 1, Len: 495
> Hypertext Transfer Protocol
> Data (15 bytes)
```

```
0020  00 00 00 00 00 00 00 00  00 00 00 01 c2 e4 6a 8b   ········ ······j·
0030  ac 53 1b 88 59 63 df 14  50 18 27 f6 be 0d 00 00   ·S··Yc·· P·'·····
0040  50 4f 53 54 20 2f 63 6f  6d 6d 61 6e 64 20 48 54   POST /co mmand HT
0050  54 50 2f 31 2e 31 0d 0a  48 6f 73 74 3a 20 6c 6f   TP/1.1·· Host: lo
0060  63 61 6c 68 6f 73 74 3a  32 37 32 37 35 0d 0a 43   calhost: 27275··C
0070  6f 6e 6e 65 63 74 69 6f  6e 3a 20 6b 65 65 70 2d   onnectio n: keep-
0080  61 6c 69 76 65 0d 0a 43  6f 6e 74 65 6e 74 2d 4c   alive··C ontent-L
0090  65 6e 67 74 68 3a 20 31  35 0d 0a 55 73 65 72 2d   ength: 1 5··User-
00a0  41 67 65 6e 74 3a 20 4d  6f 7a 69 6c 6c 61 2f 35   Agent: M ozilla/5
00b0  2e 30 20 28 57 69 6e 64  6f 77 73 20 4e 54 20 31   .0 (Wind ows NT 1
00c0  30 2e 30 3b 20 57 69 6e  36 34 3b 20 78 36 34 29   0.0; Win 64; x64)
00d0  20 41 70 70 6c 65 57 65  62 4b 69 74 2f 35 33 37    AppleWe bKit/537
00e0  2e 33 36 20 28 4b 48 54  4d 4c 2c 20 6c 69 6b 65   .36 (KHT ML, like
00f0  20 47 65 63 6b 6f 29 20  43 68 72 6f 6d 65 2f 38    Gecko)  Chrome/8
0100  34 2e 30 2e 34 31 34 37  2e 31 30 35 20 53 61 66   4.0.4147 .105 Saf
0110  61 72 69 2f 35 33 37 2e  33 36 0d 0a 44 4e 54 3a   ari/537. 36··DNT:
0120  20 31 0d 0a 43 6f 6e 74  65 6e 74 2d 54 79 70 65    1··Cont ent-Type
0130  3a 20 61 70 70 6c 69 63  61 74 69 6f 6e 2f 6f 63   : applic ation/oc
0140  74 65 74 2d 73 74 72 65  61 6d 0d 0a 41 63 63 65   tet-stre am··Acce
0150  70 74 3a 20 2a 2f 2a 0d  0a 4f 72 69 67 69 6e 3a   pt: */*· ·Origin:
0160  20 63 68 72 6f 6d 65 2d  65 78 74 65 6e 73 69 6f    chrome- extensio
0170  6e 3a 2f 2f 6c 68 6e 6e  6f 6b 6c 63 6b 6f 6d 63   n://lhnn oklckomc
0180  66 64 6c 6b 6e 6d 6a 61  65 6e 6f 6f 64 6c 70 66   fdlknmja enoodlpf
0190  64 63 6c 63 0d 0a 53 65  63 2d 46 65 74 63 68 2d   dclc··Se c-Fetch-
01a0  53 69 74 65 3a 20 6e 6f  6e 65 0d 0a 53 65 63 2d   Site: no ne··Sec-
01b0  46 65 74 63 68 2d 4d 6f  64 65 3a 20 63 6f 72 73   Fetch-Mo de: cors
01c0  0d 0a 53 65 63 2d 46 65  74 63 68 2d 44 65 73 74   ··Sec-Fe tch-Dest
01d0  3a 20 65 6d 70 74 79 0d  0a 41 63 63 65 70 74 2d   : empty· ·Accept-
01e0  45 6e 63 6f 64 69 6e 67  3a 20 67 7a 69 70 2c 20   Encoding : gzip,
01f0  64 65 66 6c 61 74 65 2c  20 62 72 0d 0a 41 63 63   deflate,  br··Acc
0200  65 70 74 2d 4c 61 6e 67  75 61 67 65 3a 20 65 6e   ept-Lang uage: en
0210  2d 55 53 2c 65 6e 3b 71  3d 30 2e 39 0d 0a 0d 0a   -US,en;q =0.9····
0220  08 01 12 09 53 5a 42 20  30 2e 31 2e 30 18 03      ····SZB  0.1.0··
```

Hypertext Transfer Protocol (http), 480 bytes | Packets: 1076 · Displayed: 2 (0.2%)

Notes:

When doing this homework, my laptop cannot capture HTTP. I was unable to capture even re-download the Win64bit version 3.0.12. Try to use my desktop computer and able capture HTTP, but it is not GET, it was POST. So the message display does not shows proper note when verify it in the Hypertext Transfer Protocol layer.