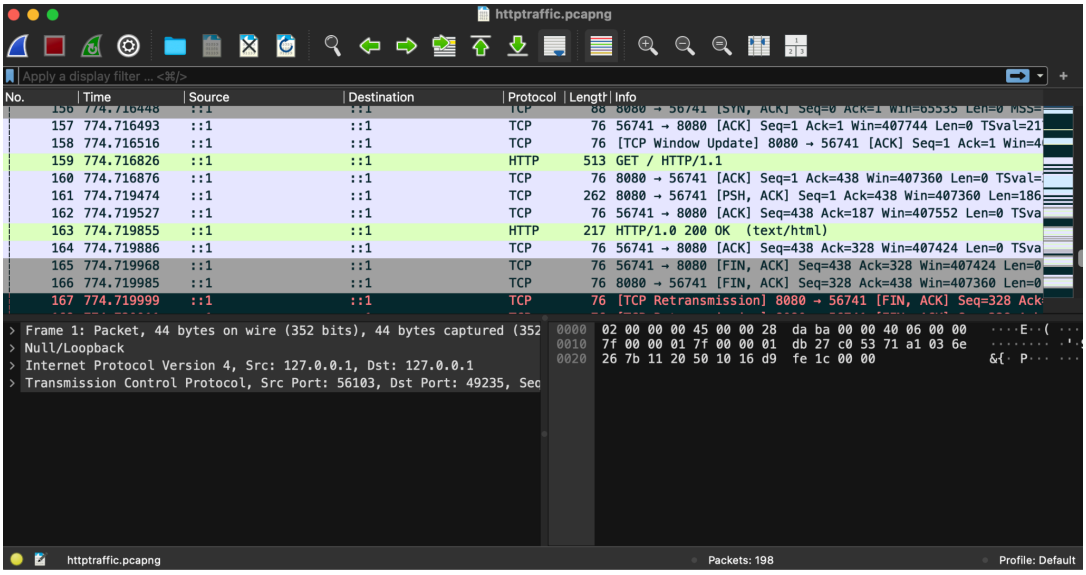
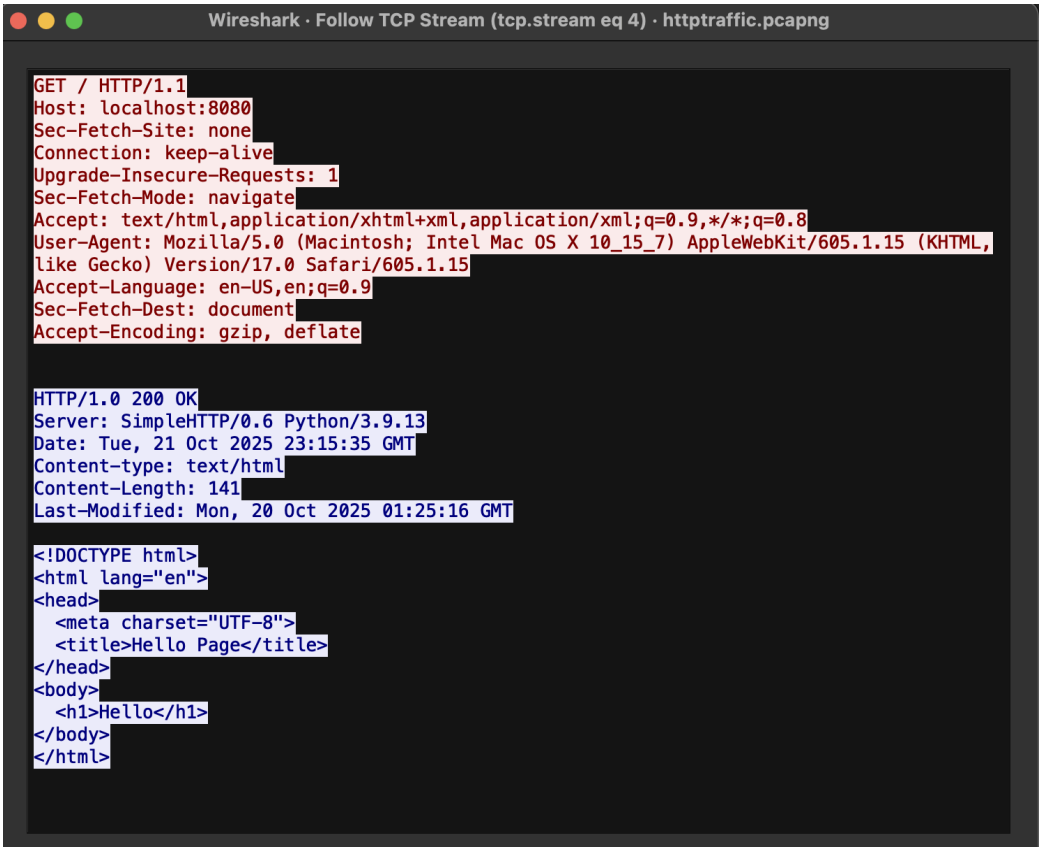
Public Key Infrastructure

1. Host a local web server.
    a. Created a webserver directory
        i. Using mkdir /webserver on terminal
        ii. Added file index.html to directory
    b. Installed Wireshark
        i. brew install --cask wireshark
    c. Started local web server
        i. python3 -m http.server 8080
        ii. Web server started at http://localhost:8080/

2. Identify why HTTP is not secure.
    a. Packet trace of HTTP traffic
        i. Used webshark
        ii. Applied filter http
        iii. Refreshed page a couple of times

    HTTP is not secure because it transmits data without encryption or authentication. This means that the data can be easily intercepted by third parties who are connected to the same network and capturing packets. For example, in the following screenshot, the following TCP stream shows the entirety of my request and the following HTML response from the website. The HTTP packet trace also shows exactly the GET request and HTML response. Typically, unencrypted data can show data such as specific URLs and sites requested, HTTP headers possibly showing authentication cookies, and the full content of web pages. Because of this, attackers can read and modify the communication between client and server, allowing them to inject malicious code, steal login information, etc. Because of these major vulnerabilities, HTTP is replaced by HTTPS in modern applications.





3. Create a self-signed certificate and upgrade your web server to HTTPS
    a. Self-signed certificate
        i. Created in /webserver directory
        ii. openssl req -x509 -newkey rsa:2048 -keyout server.key -out server.crt -days 365 -nodes

iii. Filled in certificate information

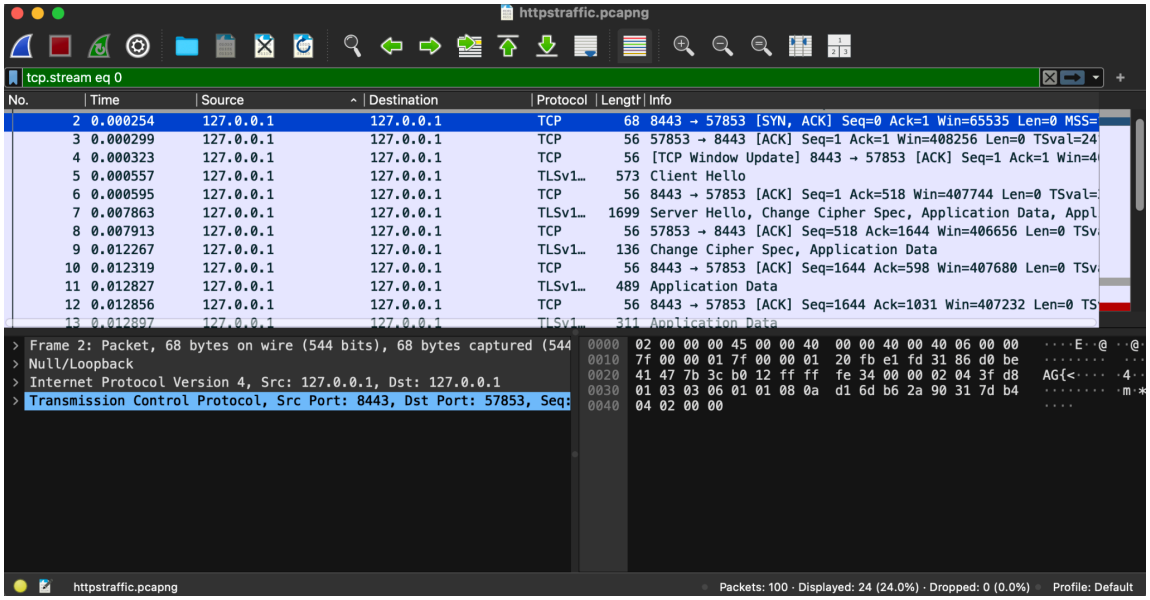
b. Upgraded web server to HTTPS
   i. Created HTTPS script

```
import http.server
import ssl
PORT = 8443
server_address = ('127.0.0.1', PORT)
httpd = http.server.HTTPServer(server_address,
http.server.SimpleHTTPRequestHandler)
httpd.socket = ssl.wrap_socket(
        httpd.socket,
        certfile="server.crt",
        keyfile="server.key",
        server_side=True
)
httpd.serve_forever()
```

   ii. Restarted server
   iii. Certificate added to list of locally trusted roots
c. Packet trace of HTTPS traffic
   i. Used webshark
   ii. Applied filter tls && tcp.port == 8443
   iii. Refreshed a couple of times

I cannot obtain an SSL certificate for my local web server from a certificate authority because they only issue SSL/TLS certificates for public, verifiable domain names, such as github.com and uchicago.edu. When a certificate is requested, the CA performs domain validation by checking DNS records. However, for my localhost server, which is not registered in DNS, it is not verifiable by any external CA; the CA cannot prove that you "own" or control localhost, so it will refuse to issue a trusted certificate for it.

In comparison to HTTP's plaintext, the screenshot below shows that HTTPS traffic only shows encrypted packets, meaning that TLS encrypts and authenticates data. Additionally, the HTTPS packet trace shows Client Hello, Server Hello, and Application Data rather than showing the exact type of request, unlike the GET request shown in HTTP. This shows that HTTPS ensures that communications between client and server remain private and secure. Because of this, attackers are unable to access private information even if connected on the same network.


httpstraffic.pcapng
tcp.stream eq 0
No. | Time | Source | Destination | Protocol | Length | Info
2 0.000254 127.0.0.1 127.0.0.1 TCP 68 8443 → 57853 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=
3 0.000299 127.0.0.1 127.0.0.1 TCP 56 57853 → 8443 [ACK] Seq=1 Ack=1 Win=408256 Len=0 TSval=24
4 0.000323 127.0.0.1 127.0.0.1 TCP 56 [TCP Window Update] 8443 → 57853 [ACK] Seq=1 Ack=1 Win=4
5 0.000557 127.0.0.1 127.0.0.1 TLSv1… 573 Client Hello
6 0.000595 127.0.0.1 127.0.0.1 TCP 56 8443 → 57853 [ACK] Seq=1 Ack=518 Win=407744 Len=0 TSval=
7 0.007863 127.0.0.1 127.0.0.1 TLSv1… 1699 Server Hello, Change Cipher Spec, Application Data, Appl
8 0.007913 127.0.0.1 127.0.0.1 TCP 56 57853 → 8443 [ACK] Seq=518 Ack=1644 Win=406656 Len=0 TSv
9 0.012267 127.0.0.1 127.0.0.1 TLSv1… 136 Change Cipher Spec, Application Data
10 0.012319 127.0.0.1 127.0.0.1 TCP 56 8443 → 57853 [ACK] Seq=1644 Ack=598 Win=407680 Len=0 TSv
11 0.012827 127.0.0.1 127.0.0.1 TLSv1… 489 Application Data
12 0.012856 127.0.0.1 127.0.0.1 TCP 56 8443 → 57853 [ACK] Seq=1644 Ack=1031 Win=407232 Len=0 TS
13 0.012897 127.0.0.1 127.0.0.1 TLSv1… 311 Application Data
Frame 2: Packet, 68 bytes on wire (544 bits), 68 bytes captured (544
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 8443, Dst Port: 57853, Seq:
0000 02 00 00 00 45 00 00 40 00 00 40 00 40 06 00 00
0010 7f 00 00 01 7f 00 00 01 20 fb e1 fd 31 86 d0 be
0020 41 47 7b 3c b0 12 ff ff fe 34 00 00 02 04 3f d8
0030 01 03 03 06 01 01 08 0a d1 6d b6 2a 90 31 7d b4
0040 04 02 00 00
httpstraffic.pcapng
Packets: 100 · Displayed: 24 (24.0%) · Dropped: 0 (0.0%)
Profile: Default


Wireshark · Follow TCP Stream (tcp.stream eq 0) · httpstraffic.pcapng
.....B,T.....Hos.q(o.....|&.c] F.....S.....!G...esz.z..pE....E..*.....,.
+...0./...
. .....5./.....
....jj.....
...
JJ..... .http/1.1.....
.....3.+.)JJ..... ...}.ghX.s.^.....yDa..Q.7..W.r..-.....+..
....z...v...[...56.b.....,.I#....&.I.x#... F.....S.....!G...esz.z..pE....E.....+.....3
.$... .....$z....V4.5.....y.m/c..PAu.....G...."1.Z3.I.;.-..Y....].yJ./.....
..G..e."..+...#V;K.A.....+.'. L.....R..Vm...jL. .!l. .}....Uo[.....y....M.T.T}3.
x.....Hx..d..~.E.h.?...zP-.+..A.8O...)DI.f.
A.$....`....U..~...A...2...P...8.....:0...'.....k.k..Sv.9.!....KlKm....S.4....3B...]
MX:...A.y.....5RAXo3&.m..7j^.....v.@.}k.W.t;].....f'..C..o2....@.YF .<[.....y..-,.."
U.....~..Z%.I..F.....hi..C...l3-C..NF:. .....O.s.v...>{.. .....iL.i.b..D@jZ`
.....6Ol....2..V.,.....b.lW.....'x.....8...*B..e..."4=....Q..RV'.....p.To.....Fbe
....GEm...NP.._^...%..<..P.Y*...
...X.,wI.T...
.i.....Y.8..>R..i...V.\.....h@..;.....}.Q.....^.$......../.3P..&....%.....{..3.....U
....~.'.w].P...QU..".\D....8.r...du@.#...3h,.ujd.bW.....W..9.....2..9~.j...t.....!|P.
.}.^...O.QV...o)....8.....w..}<.....w[...0M...D..!..%..>..p..@.Y...(L.....N.....!../..
P...{24/:.~JN..m..
A.@.3:*%...^..
....UZI :.<.hj!N.....o..lU...\h..93..O.....[..FU{.....o...<V.M.~..+...._...),....g
}.Sf.f|.#.m..S..<.7.....N..d..).[].....Lg.Ba.....k...|..... ....U.^
8[.2..yA.~s.o.4r..rh..$....kT..7eb.....y....A.....$.....T..5..t.(.BE_ .6....A$...u...3.
l..%:...{'.S..P.w..
..T.....A.r...7.U.....*.PIX."....l....f.(?.d).....8...F5c.....B60B ...E.
.8....vk..=...n.<.;.....L."]...Q...i....-EZZ.'Jr.....L.SH...^.....<.'p..s{P#[o.,JQj.v.R.
k...;[vY#.6...9Z..9.3.....O.../....V.F...mQ..qHVu.M..}.....@..c.p.4A.2.".J...3'..9%?
`.....HQ.;..h..../.....o.....E.r.#...*.y}H.8.rW..Q.. ..Qm.....B.g.....WB.4jMd(...v.-m.
.!....<.5sC