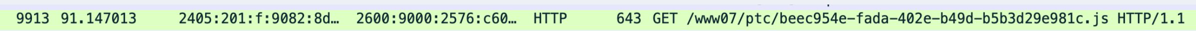
Experiment with Packet Tracers/Analyzers

Q1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?

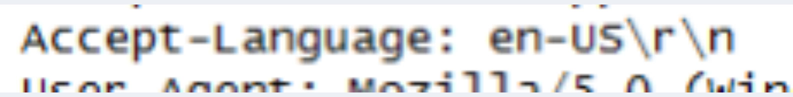
→ *HTTP 1.1*



```
9913 91.147013    2405:201:f:9082:8d…  2600:9000:2576:c60…  HTTP     643  GET /www07/ptc/beec954e-fada-402e-b49d-b5b3d29e981c.js HTTP/1.1
```

Q2. What languages (if any) does your browser indicate that it can accept to the server?

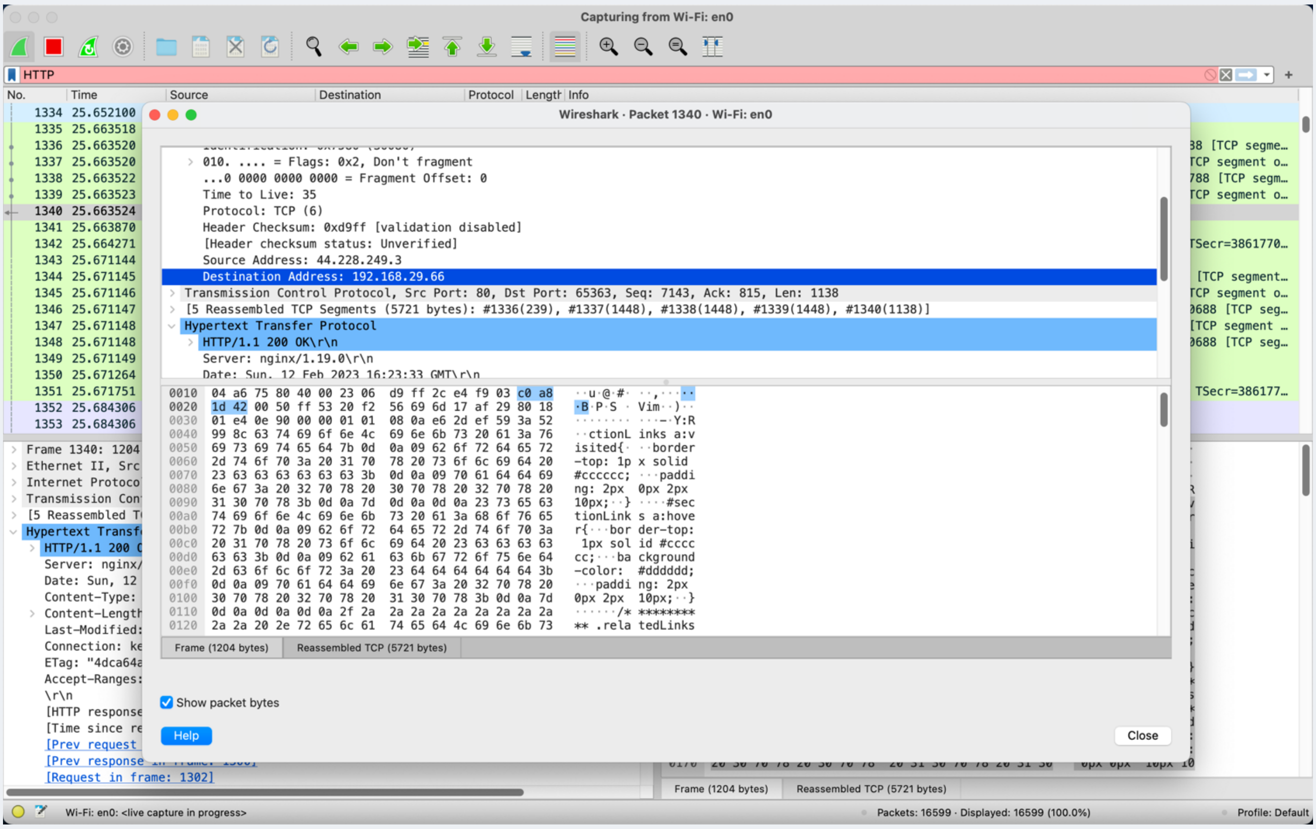
→ en-US (US English)



```
Accept-Language: en-US\r\n
User-Agent: Mozilla/5.0 (Win
```

Q3. What is the IP address of your computer?

→ Source Address: 44.228.249.3
→ Destination Address: 192.168.29.66



Capturing from Wi-Fi: en0

HTTP

```
No.   Time       Source   Destination   Protocol  Length  Info
 1334 25.652100
 1335 25.663518
 1336 25.663520                                            …38 [TCP segme…
 1337 25.663520                                            TCP segment o…
 1338 25.663522                                            …788 [TCP segm…
 1339 25.663523                                            TCP segment o…
 1340 25.663524
 1341 25.663870
 1342 25.664271                                            TSecr=3861770…
 1343 25.671144
 1344 25.671145                                            [TCP segment…
 1345 25.671146                                            TCP segment o…
 1346 25.671147                                            …0688 [TCP seg…
 1347 25.671148                                            [TCP segment …
 1348 25.671148                                            …0688 [TCP seg…
 1349 25.671149
 1350 25.671264
 1351 25.671751                                            TSecr=386177…
 1352 25.684306
 1353 25.684306
```

```
> Frame 1340: 1204
> Ethernet II, Src
> Internet Protoco
> Transmission Con
> [5 Reassembled T
v Hypertext Transf
  > HTTP/1.1 200 O
    Server: nginx/
    Date: Sun, 12
    Content-Type:
  > Content-Length
    Last-Modified:
    Connection: ke
    ETag: "4dca64a
    Accept-Ranges:
    \r\n
    [HTTP response
    [Time since re
    [Prev request
    [Prev response in frame: 1300]
    [Request in frame: 1302]
```

Wireshark · Packet 1340 · Wi-Fi: en0

```
    > 010. .... = Flags: 0x2, Don't fragment
      ...0 0000 0000 0000 = Fragment Offset: 0
      Time to Live: 35
      Protocol: TCP (6)
      Header Checksum: 0xd9ff [validation disabled]
      [Header checksum status: Unverified]
      Source Address: 44.228.249.3
      Destination Address: 192.168.29.66
> Transmission Control Protocol, Src Port: 80, Dst Port: 65363, Seq: 7143, Ack: 815, Len: 1138
> [5 Reassembled TCP Segments (5721 bytes): #1336(239), #1337(1448), #1338(1448), #1339(1448), #1340(1138)]
v Hypertext Transfer Protocol
    > HTTP/1.1 200 OK\r\n
      Server: nginx/1.19.0\r\n
      Date: Sun, 12 Feb 2023 16:23:33 GMT\r\n
```

```
0010  04 a6 75 80 40 00 23 06  d9 ff 2c e4 f9 03 c0 a8   ··u·@·#· ··,·····
0020  1d 42 00 50 ff 53 20 f2  56 69 6d 17 af 29 80 18   ·B·P·S · Vim··)··
0030  01 e4 0e 90 00 00 01 01  08 0a e6 2d ef 59 3a 52   ········ ···-·Y:R
0040  99 8c 63 74 69 6f 6e 4c  69 6e 6b 73 20 61 3a 76   ··ctionL inks a:v
0050  69 73 69 74 65 64 7b 0d  0a 09 62 6f 72 64 65 72   isited{· ··border
0060  2d 74 6f 70 3a 20 31 70  78 20 73 6f 6c 69 64 20   -top: 1p x solid
0070  23 63 63 63 63 63 63 3b  0d 0a 09 70 61 64 64 69   #cccccc; ···paddi
0080  6e 67 3a 20 32 70 78 20  30 70 78 20 32 70 78 20   ng: 2px  0px 2px
0090  31 30 70 78 3b 0d 0a 7d  0d 0a 0d 0a 23 73 65 63   10px;··} ····#sec
00a0  74 69 6f 6e 4c 69 6e 6b  73 20 61 3a 68 6f 76 65   tionLink s a:hove
00b0  72 7b 0d 0a 09 62 6f 72  64 65 72 2d 74 6f 70 3a   r{···bor der-top:
00c0  20 31 70 78 20 73 6f 6c  69 64 20 23 63 63 63 63    1px sol id #cccc
00d0  63 63 3b 0d 0a 09 62 61  63 6b 67 72 6f 75 6e 64   cc;···ba ckground
00e0  2d 63 6f 6c 6f 72 3a 20  23 64 64 64 64 64 64 3b   -color:  #dddddd;
00f0  0d 0a 09 70 61 64 64 69  6e 67 3a 20 32 70 78 20   ···paddi ng: 2px
0100  30 70 78 20 32 70 78 20  31 30 70 78 3b 0d 0a 7d   0px 2px  10px;··}
0110  0d 0a 0d 0a 0d 0a 2f 2a  2a 2a 2a 2a 2a 2a 2a 2a   ······/* ********
0120  2a 2a 20 2e 72 65 6c 61  74 65 64 4c 69 6e 6b 73   ** .rela tedLinks
```

Frame (1204 bytes) | Reassembled TCP (5721 bytes)

☑ Show packet bytes

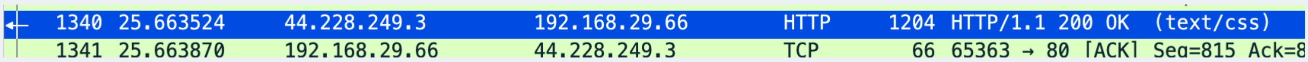
Help | Close

Frame (1204 bytes) | Reassembled TCP (5721 bytes)

Wi-Fi: en0: <live capture in progress>
Packets: 16599 · Displayed: 16599 (100.0%)
Profile: Default

Q4. What is the status code returned from the server to your browser?

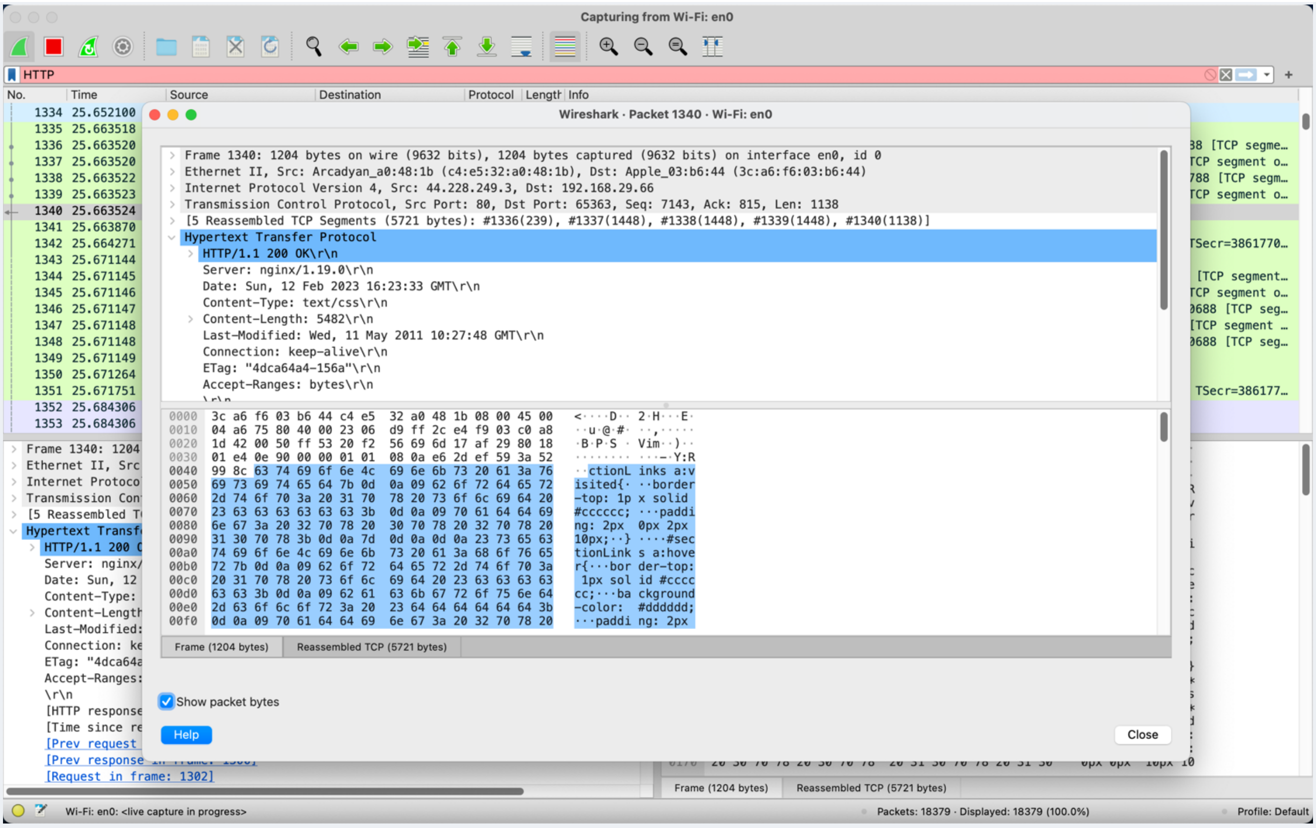
→200 OK



```
1340  25.663524    44.228.249.3     192.168.29.66     HTTP     1204  HTTP/1.1 200 OK  (text/css)
1341  25.663870    192.168.29.66    44.228.249.3      TCP        66  65363 → 80 [ACK] Seq=815 Ack=8
```

Q5. When was the HTML file that you are retrieving last modified at the server?

→ We can filter messages by http.last_modified and we see that the HTTP response I received for the html file doesn't show this field. We do have a `http.last_modified` field in the favicon response however, as shown in the screenshot below.



Capturing from Wi-Fi: en0

Wireshark · Packet 1340 · Wi-Fi: en0

```
> Frame 1340: 1204 bytes on wire (9632 bits), 1204 bytes captured (9632 bits) on interface en0, id 0
> Ethernet II, Src: Arcadyan_a0:48:1b (c4:e5:32:a0:48:1b), Dst: Apple_03:b6:44 (3c:a6:f6:03:b6:44)
> Internet Protocol Version 4, Src: 44.228.249.3, Dst: 192.168.29.66
> Transmission Control Protocol, Src Port: 80, Dst Port: 65363, Seq: 7143, Ack: 815, Len: 1138
> [5 Reassembled TCP Segments (5721 bytes): #1336(239), #1337(1448), #1338(1448), #1339(1448), #1340(1138)]
v Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Server: nginx/1.19.0\r\n
    Date: Sun, 12 Feb 2023 16:23:33 GMT\r\n
    Content-Type: text/css\r\n
  > Content-Length: 5482\r\n
    Last-Modified: Wed, 11 May 2011 10:27:48 GMT\r\n
    Connection: keep-alive\r\n
    ETag: "4dca64a4-156a"\r\n
    Accept-Ranges: bytes\r\n
```

```
0000  3c a6 f6 03 b6 44 c4 e5  32 a0 48 1b 08 00 45 00   <····D·· 2·H···E·
0010  04 a6 75 80 40 00 23 06  d9 ff 2c e4 f9 03 c0 a8   ··u·@·#· ··,·····
0020  1d 42 00 50 ff 53 20 f2  56 69 6d 17 af 29 80 18   ·B·P·S · Vim··)··
0030  01 e4 0e 90 00 00 01 01  08 0a e6 2d ef 59 3a 52   ········ ···-·Y:R
0040  99 8c 63 74 69 6f 6e 4c  69 6e 6b 73 20 61 3a 76   ··ctionL inks a:v
0050  69 73 69 74 65 64 7b 0d  0a 09 62 6f 72 64 65 72   isited{· ··border
0060  2d 74 6f 70 3a 20 31 70  78 20 73 6f 6c 69 64 20   -top: 1p x solid
0070  23 63 63 63 63 63 63 3b  0d 0a 09 70 61 64 64 69   #cccccc; ···paddi
0080  6e 67 3a 20 32 70 78 20  30 70 78 20 32 70 78 20   ng: 2px  0px 2px
0090  31 30 70 78 3b 0d 0a 7d  0d 0a 0d 0a 23 73 65 63   10px;··} ····#sec
00a0  74 69 6f 6e 4c 69 6e 6b  73 20 61 3a 68 6f 76 65   tionLink s a:hove
00b0  72 7b 0d 0a 09 62 6f 72  64 65 72 2d 74 6f 70 3a   r{···bor der-top:
00c0  20 31 70 78 20 73 6f 6c  69 64 20 23 63 63 63 63    1px sol id #cccc
00d0  63 63 3b 0d 0a 09 62 61  63 6b 67 72 6f 75 6e 64   cc;···ba ckground
00e0  2d 63 6f 6c 6f 72 3a 20  23 64 64 64 64 64 64 3b   -color:  #dddddd;
00f0  0d 0a 09 70 61 64 64 69  6e 67 3a 20 32 70 78 20   ···paddi ng: 2px
```

Frame (1204 bytes)
Reassembled TCP (5721 bytes)
☑ Show packet bytes
Help
Close
Wi-Fi: en0: <live capture in progress>
Packets: 18379 · Displayed: 18379 (100.0%)
Profile: Default

Q6. How many bytes of content are being returned to your browser?

→ Content-Length: 5482 bytes

Q7. By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.

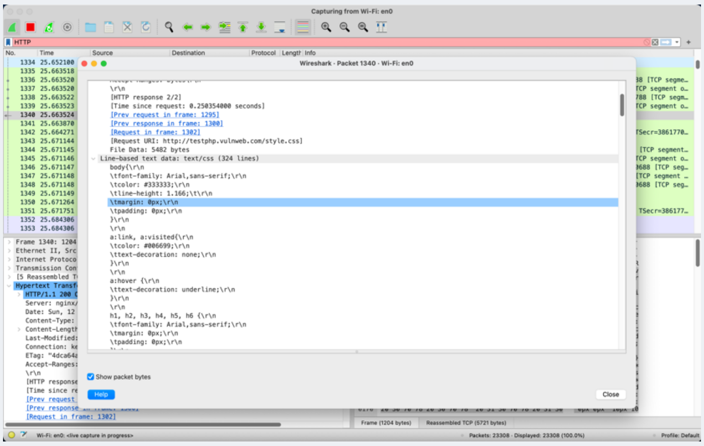
→ No. The raw data appears to match up exactly with what is shown in the packet-listing window.

Q8. Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET?

→ No

Q9. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?

→ See the Photo below



Capturing from Wi-Fi: en0

Wireshark · Packet 1340 · Wi-Fi: en0

```
    \r\n
    [HTTP response 2/2]
    [Time since request: 0.250354000 seconds]
    [Prev request in frame: 1295]
    [Prev response in frame: 1300]
    [Request in frame: 1302]
    [Request URI: http://testphp.vulnweb.com/style.css]
    File Data: 5482 bytes
v Line-based text data: text/css (324 lines)
    body{\r\n
    \tfont-family: Arial,sans-serif;\r\n
    \tcolor: #333333;\r\n
    \tline-height: 1.166;\t\r\n
    \tmargin: 0px;\r\n
    \tpadding: 0px;\r\n
    }\r\n
    \r\n
    a:link, a:visited{\r\n
    \tcolor: #006699;\r\n
    \ttext-decoration: none;\r\n
    }\r\n
    \r\n
    a:hover {\r\n
    \ttext-decoration: underline;\r\n
    }\r\n
    \r\n
    h1, h2, h3, h4, h5, h6 {\r\n
    \tfont-family: Arial,sans-serif;\r\n
    \tmargin: 0px;\r\n
    \tpadding: 0px;\r\n
```

☑ Show packet bytes
Help
Close
Wi-Fi: en0: <live capture in progress>
Packets: 23308 · Displayed: 23308 (100.0%)
Profile: Default

Q10. Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE:” line in the HTTP GET? If so, what information follows the “IF-MODIFIED-SINCE:” header?

→No

Q11. What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

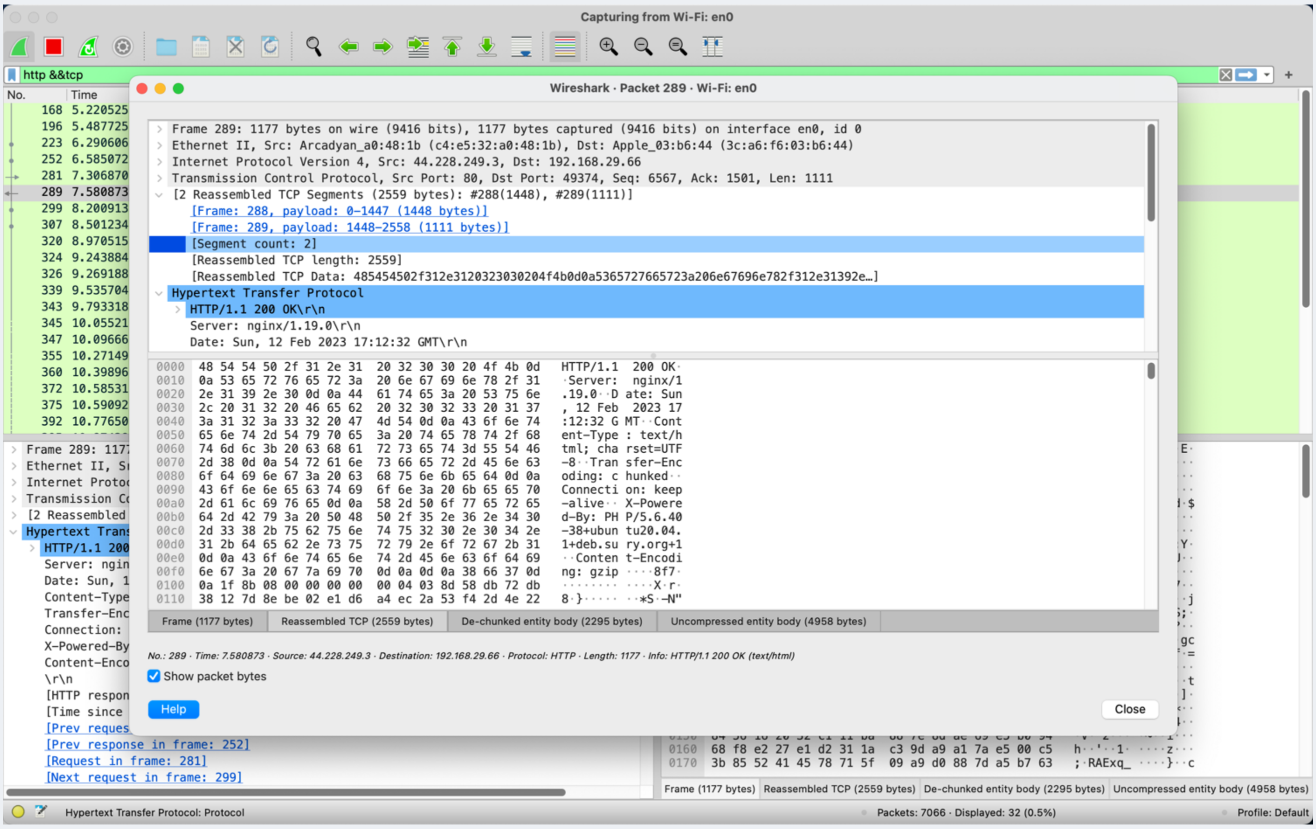
→No. did not get another status code other then 200

Q12. How many HTTP GET request messages were sent by your browser (again, ignoring requests for favicon.ico)?

→Just one

Q13. How many data-containing TCP segments were needed to carry the single HTTP response?

→ There are 2 TCP segments



Capturing from Wi-Fi: en0

http &&tcp

```
No.   Time
168   5.220525
196   5.487725
223   6.290606
252   6.585072
281   7.306870
289   7.580873
299   8.200913
307   8.501234
320   8.970515
324   9.243884
326   9.269188
339   9.535704
343   9.793318
345   10.05521
347   10.09666
355   10.27149
360   10.39896
372   10.58531
375   10.59092
392   10.77650
```

Wireshark · Packet 289 · Wi-Fi: en0

```
> Frame 289: 1177 bytes on wire (9416 bits), 1177 bytes captured (9416 bits) on interface en0, id 0
> Ethernet II, Src: Arcadyan_a0:48:1b (c4:e5:32:a0:48:1b), Dst: Apple_03:b6:44 (3c:a6:f6:03:b6:44)
> Internet Protocol Version 4, Src: 44.228.249.3, Dst: 192.168.29.66
> Transmission Control Protocol, Src Port: 80, Dst Port: 49374, Seq: 6567, Ack: 1501, Len: 1111
v [2 Reassembled TCP Segments (2559 bytes): #288(1448), #289(1111)]
    [Frame: 288, payload: 0-1447 (1448 bytes)]
    [Frame: 289, payload: 1448-2558 (1111 bytes)]
    [Segment count: 2]
    [Reassembled TCP length: 2559]
    [Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a5365727665723a206e67696e782f312e31392e…]
v Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Server: nginx/1.19.0\r\n
    Date: Sun, 12 Feb 2023 17:12:32 GMT\r\n
```

```
0000  48 54 54 50 2f 31 2e 31  20 32 30 30 20 4f 4b 0d   HTTP/1.1  200 OK·
0010  0a 53 65 72 76 65 72 3a  20 6e 67 69 6e 78 2f 31   ·Server:  nginx/1
0020  2e 31 39 2e 30 0d 0a 44  61 74 65 3a 20 53 75 6e   .19.0··D ate: Sun
0030  2c 20 31 32 20 46 65 62  20 32 30 32 33 20 31 37   , 12 Feb  2023 17
0040  3a 31 32 3a 33 32 20 47  4d 54 0d 0a 43 6f 6e 74   :12:32 G MT··Cont
0050  65 6e 74 2d 54 79 70 65  3a 20 74 65 78 74 2f 68   ent-Type : text/h
0060  74 6d 6c 3b 20 63 68 61  72 73 65 74 3d 55 54 46   tml; cha rset=UTF
0070  2d 38 0d 0a 54 72 61 6e  73 66 65 72 2d 45 6e 63   -8··Tran sfer-Enc
0080  6f 64 69 6e 67 3a 20 63  68 75 6e 6b 65 64 0d 0a   oding: c hunked··
0090  43 6f 6e 6e 65 63 74 69  6f 6e 3a 20 6b 65 65 70   Connecti on: keep
00a0  2d 61 6c 69 76 65 0d 0a  58 2d 50 6f 77 65 72 65   -alive·· X-Powere
00b0  64 2d 42 79 3a 20 50 48  50 2f 35 2e 36 2e 34 30   d-By: PH P/5.6.40
00c0  2d 33 38 2b 75 62 75 6e  74 75 32 30 2e 30 34 2e   -38+ubun tu20.04.
00d0  31 2b 64 65 62 2e 73 75  72 79 2e 6f 72 67 2b 31   1+deb.su ry.org+1
00e0  0d 0a 43 6f 6e 74 65 6e  74 2d 45 6e 63 6f 64 69   ··Conten t-Encodi
00f0  6e 67 3a 20 67 7a 69 70  0d 0a 0d 0a 38 66 37 0d   ng: gzip ····8f7·
0100  0a 1f 8b 08 00 00 00 00  00 04 03 8d 58 db 72 db   ········ ····X·r·
0110  38 12 7d 8e be 02 e1 d6  a4 ec 2a 53 f4 2d 4e 22   8·}····· ··*S·-N"
```

Frame (1177 bytes)
Reassembled TCP (2559 bytes)
De-chunked entity body (2295 bytes)
Uncompressed entity body (4958 bytes)

No.: 289 · Time: 7.580873 · Source: 44.228.249.3 · Destination: 192.168.29.66 · Protocol: HTTP · Length: 1177 · Info: HTTP/1.1 200 OK (text/html)

☑ Show packet bytes

```
> Frame 289: 117
> Ethernet II, S
> Internet Proto
> Transmission C
> [2 Reassembled
v Hypertext Trans
  > HTTP/1.1 200
    Server: ngin
    Date: Sun, 1
    Content-Type
    Transfer-Enc
    Connection:
    X-Powered-By
    Content-Enco
    \r\n
    [HTTP respon
    [Time since
    [Prev reques
    [Prev response in frame: 252]
    [Request in frame: 281]
    [Next request in frame: 299]
```

```
0160  68 f8 e2 27 e1 d2 31 1a  c3 9d a9 a1 7a e5 00 c5   h··'··1· ····z···
0170  3b 85 52 41 45 78 71 5f  09 a9 d0 88 7d a5 b7 63   ;·RAExq_ ····}··c
```

Frame (1177 bytes)
Reassembled TCP (2559 bytes)
De-chunked entity body (2295 bytes)
Uncompressed entity body (4958 bytes)

Hypertext Transfer Protocol: Protocol

Packets: 7066 · Displayed: 32 (0.5%)

Profile: Default

Q14. What is the status code and phrase associated with the response to the HTTP GET request?

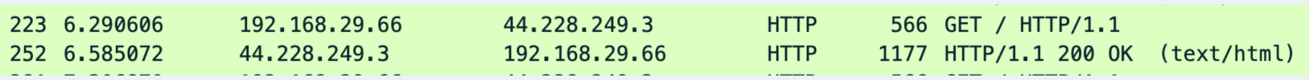
→ HTTP/1.1 200 OK\r\n

Q15. Are there any HTTP status lines in the transmitted data associated with a TCP- induced “Continuation”?

→ No

Q16. How many HTTP GET request messages were sent by your browser? To which Internet addresses were these GET requests sent?

→ One: 192.168.29.66



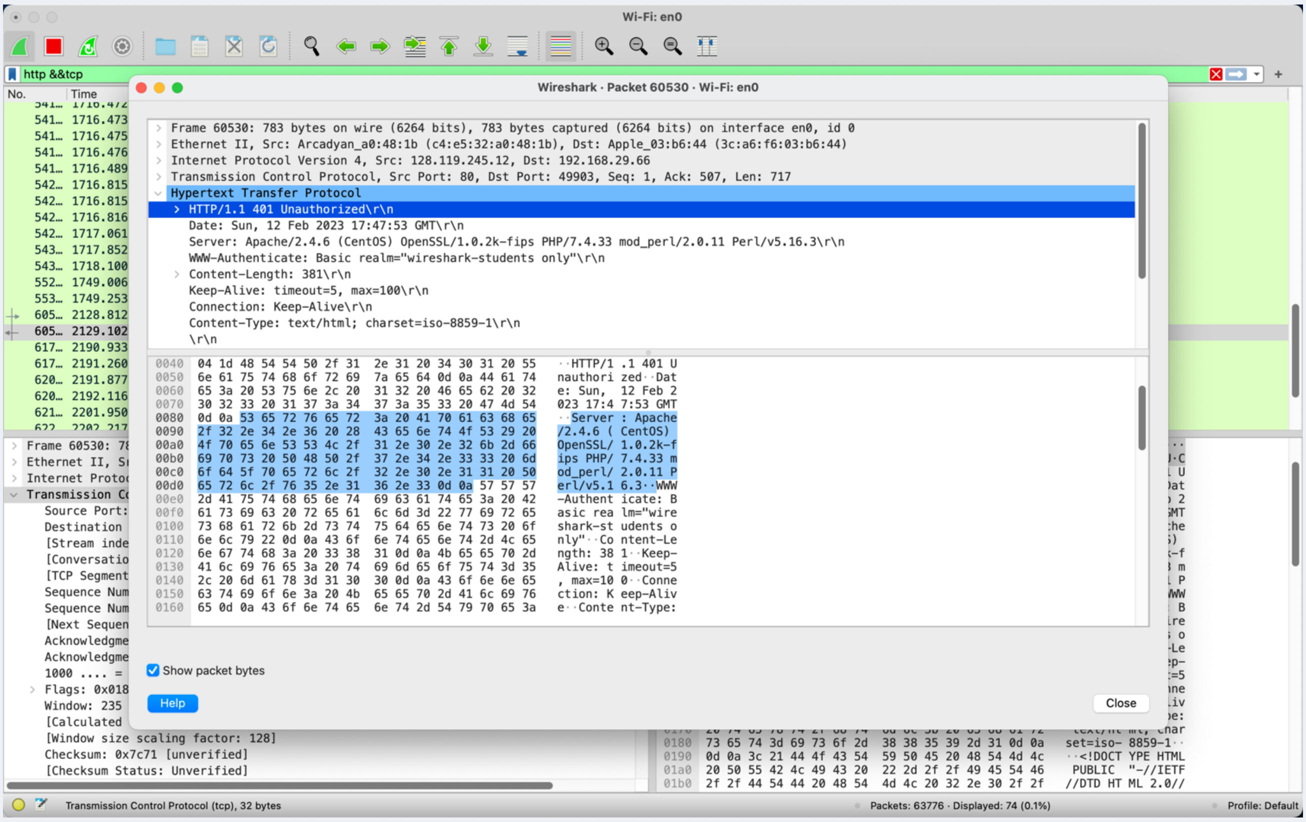
```
223  6.290606      192.168.29.66       44.228.249.3        HTTP       566  GET / HTTP/1.1
252  6.585072      44.228.249.3        192.168.29.66       HTTP      1177  HTTP/1.1 200 OK  (text/html)
```

Q17.Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.

→ Based on the timestamps, it appearsthe images were downloaded serially.

Q18. What is the server's response (status code and phrase) in response to the initial HTTP GET message from your browser?

→ HTTP/1.1 401 Unauthorized


Wi-Fi: en0
http &&tcp
No.
Time
Wireshark · Packet 60530 · Wi-Fi: en0
Show packet bytes
Help
Close
Transmission Control Protocol (tcp), 32 bytes
Packets: 63776 · Displayed: 74 (0.1%)
Profile: Default

```
541… 1716.473
541… 1716.475
541… 1716.476
541… 1716.489
542… 1716.815
542… 1716.815
542… 1716.816
542… 1717.061
543… 1717.852
543… 1718.100
552… 1749.006
553… 1749.253
605… 2128.812
605… 2129.102
617… 2190.933
617… 2191.260
620… 2191.877
620… 2192.116
621… 2201.950
```

```
> Frame 60530: 783 bytes on wire (6264 bits), 783 bytes captured (6264 bits) on interface en0, id 0
> Ethernet II, Src: Arcadyan_a0:48:1b (c4:e5:32:a0:48:1b), Dst: Apple_03:b6:44 (3c:a6:f6:03:b6:44)
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.29.66
> Transmission Control Protocol, Src Port: 80, Dst Port: 49903, Seq: 1, Ack: 507, Len: 717
v Hypertext Transfer Protocol
  > HTTP/1.1 401 Unauthorized\r\n
    Date: Sun, 12 Feb 2023 17:47:53 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3\r\n
    WWW-Authenticate: Basic realm="wireshark-students only"\r\n
  > Content-Length: 381\r\n
    Keep-Alive: timeout=5, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=iso-8859-1\r\n
    \r\n
```

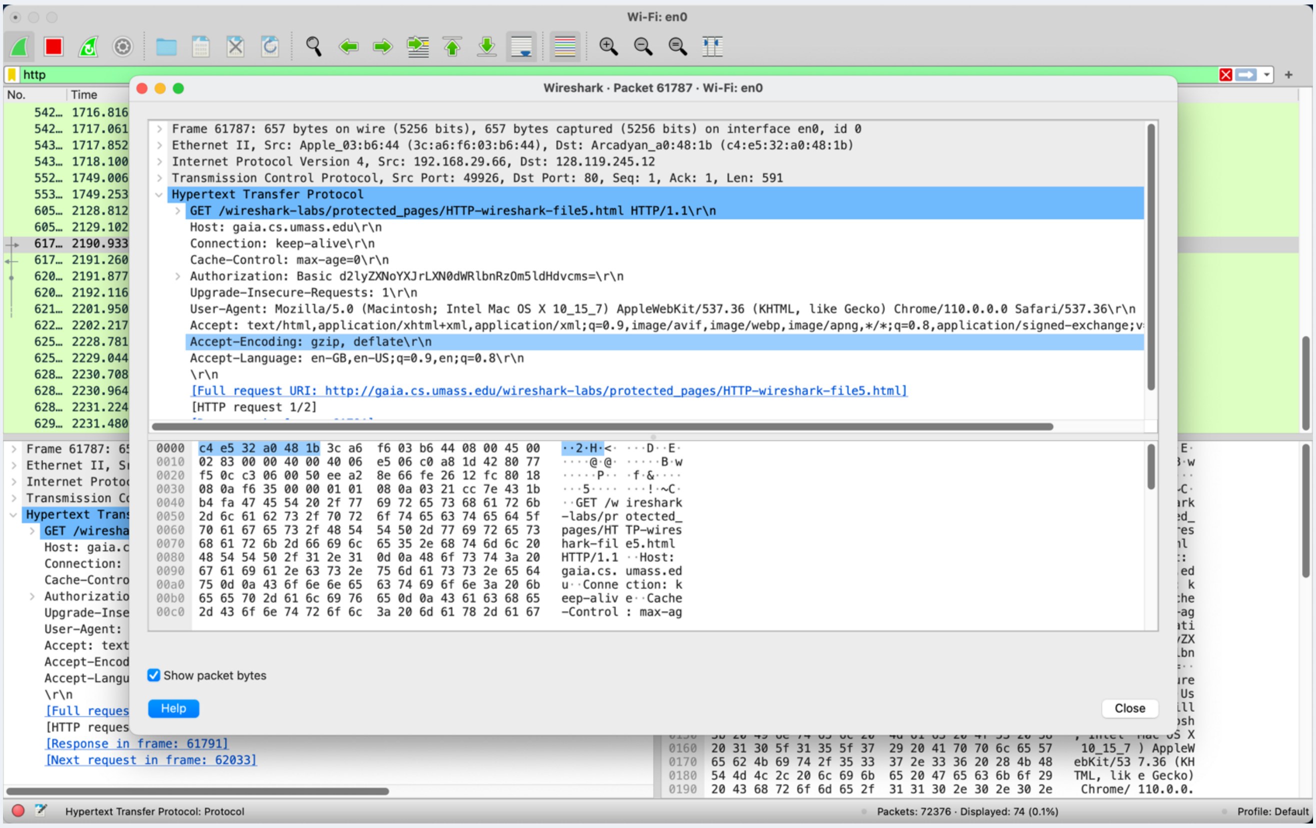
```
0040  04 1d 48 54 54 50 2f 31  2e 31 20 34 30 31 20 55   ··HTTP/1 .1 401 U
0050  6e 61 75 74 68 6f 72 69  7a 65 64 0d 0a 44 61 74   nauthori zed··Dat
0060  65 3a 20 53 75 6e 2c 20  31 32 20 46 65 62 20 32   e: Sun,  12 Feb 2
0070  30 32 33 20 31 37 3a 34  37 3a 35 33 20 47 4d 54   023 17:4 7:53 GMT
0080  0d 0a 53 65 72 76 65 72  3a 20 41 70 61 63 68 65   ··Server : Apache
0090  2f 32 2e 34 2e 36 20 28  43 65 6e 74 4f 53 29 20   /2.4.6 ( CentOS)
00a0  4f 70 65 6e 53 53 4c 2f  31 2e 30 2e 32 6b 2d 66   OpenSSL/ 1.0.2k-f
00b0  69 70 73 20 50 48 50 2f  37 2e 34 2e 33 33 20 6d   ips PHP/ 7.4.33 m
00c0  6f 64 5f 70 65 72 6c 2f  32 2e 30 2e 31 31 20 50   od_perl/ 2.0.11 P
00d0  65 72 6c 2f 76 35 2e 31  36 2e 33 0d 0a 57 57 57   erl/v5.1 6.3··WWW
00e0  2d 41 75 74 68 65 6e 74  69 63 61 74 65 3a 20 42   -Authent icate: B
00f0  61 73 69 63 20 72 65 61  6c 6d 3d 22 77 69 72 65   asic rea lm="wire
0100  73 68 61 72 6b 2d 73 74  75 64 65 6e 74 73 20 6f   shark-st udents o
0110  6e 6c 79 22 0d 0a 43 6f  6e 74 65 6e 74 2d 4c 65   nly"··Co ntent-Le
0120  6e 67 74 68 3a 20 33 38  31 0d 0a 4b 65 65 70 2d   ngth: 38 1··Keep-
0130  41 6c 69 76 65 3a 20 74  69 6d 65 6f 75 74 3d 35   Alive: t imeout=5
0140  2c 20 6d 61 78 3d 31 30  30 0d 0a 43 6f 6e 6e 65   , max=10 0··Conne
0150  63 74 69 6f 6e 3a 20 4b  65 65 70 2d 41 6c 69 76   ction: K eep-Aliv
0160  65 0d 0a 43 6f 6e 74 65  6e 74 2d 54 79 70 65 3a   e··Conte nt-Type:
```

```
> Frame 60530: 78
> Ethernet II, Sr
> Internet Protoc
v Transmission Co
    Source Port:
    Destination
    [Stream inde
    [Conversatio
    [TCP Segment
    Sequence Num
    Sequence Num
    [Next Sequen
    Acknowledgme
    Acknowledgme
    1000 .... =
  > Flags: 0x018
    Window: 235
    [Calculated
    [Window size scaling factor: 128]
    Checksum: 0x7c71 [unverified]
    [Checksum Status: Unverified]
```

```
0180  73 65 74 3d 69 73 6f 2d  38 38 35 39 2d 31 0d 0a   set=iso- 8859-1··
0190  0d 0a 3c 21 44 4f 43 54  59 50 45 20 48 54 4d 4c   ··<!DOCT YPE HTML
01a0  20 50 55 42 4c 49 43 20  22 2d 2f 2f 49 45 54 46    PUBLIC  "-//IETF
01b0  2f 2f 44 54 44 20 48 54  4d 4c 20 32 2e 30 2f 2f   //DTD HT ML 2.0//
```

Q19. When your browser sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

→ Authorization field.



Wireshark · Packet 61787 · Wi-Fi: en0

```
> Frame 61787: 657 bytes on wire (5256 bits), 657 bytes captured (5256 bits) on interface en0, id 0
> Ethernet II, Src: Apple_03:b6:44 (3c:a6:f6:03:b6:44), Dst: Arcadyan_a0:48:1b (c4:e5:32:a0:48:1b)
> Internet Protocol Version 4, Src: 192.168.29.66, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 49926, Dst Port: 80, Seq: 1, Ack: 1, Len: 591
v Hypertext Transfer Protocol
  > GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\r\n
  > Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-GB,en-US;q=0.9,en;q=0.8\r\n
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html]
    [HTTP request 1/2]
    [Response in frame: 61791]
    [Next request in frame: 62033]
```

```
0000  c4 e5 32 a0 48 1b 3c a6  f6 03 b6 44 08 00 45 00   ··2·H·<· ···D··E·
0010  02 83 00 00 40 00 40 06  e5 06 c0 a8 1d 42 80 77   ····@·@· ·····B·w
0020  f5 0c c3 06 00 50 ee a2  8e 66 fe 26 12 fc 80 18   ·····P·· ·f·&····
0030  08 0a f6 35 00 00 01 01  08 0a 03 21 cc 7e 43 1b   ···5···· ···!·~C·
0040  b4 fa 47 45 54 20 2f 77  69 72 65 73 68 61 72 6b   ··GET /w ireshark
0050  2d 6c 61 62 73 2f 70 72  6f 74 65 63 74 65 64 5f   -labs/pr otected_
0060  70 61 67 65 73 2f 48 54  54 50 2d 77 69 72 65 73   pages/HT TP-wires
0070  68 61 72 6b 2d 66 69 6c  65 35 2e 68 74 6d 6c 20   hark-fil e5.html
0080  48 54 54 50 2f 31 2e 31  0d 0a 48 6f 73 74 3a 20   HTTP/1.1 ··Host:
0090  67 61 69 61 2e 63 73 2e  75 6d 61 73 73 2e 65 64   gaia.cs. umass.ed
00a0  75 0d 0a 43 6f 6e 6e 65  63 74 69 6f 6e 3a 20 6b   u··Conne ction: k
00b0  65 65 70 2d 61 6c 69 76  65 0d 0a 43 61 63 68 65   eep-aliv e··Cache
00c0  2d 43 6f 6e 74 72 6f 6c  3a 20 6d 61 78 2d 61 67   -Control : max-ag
```

☑ Show packet bytes

Wi-Fi: en0
http
No.
Time
Help
Close
Hypertext Transfer Protocol: Protocol
Packets: 72376 · Displayed: 74 (0.1%)
Profile: Default

Q20. What does the "Connection: close" and "Connection: Keep-alive" header field imply in HTTP protocol? When should one be used over the other?

→ The Connection general header controls whether the network connection stays open after the current transaction finishes. If the value sent is keep-alive, the connection is persistent and not closed, allowing for subsequent requests to the same server to be done. Except for the standard hop-by-hop headers (Keep-Alive, Transfer-Encoding, TE, Connection, Trailer, Upgrade, Proxy-Authorization and Proxy-Authenticate), any hop-by-hop headers used by the message must be listed in the Connection header, so that the first proxy knows it has to consume them and not forward them further. Standard hop-by-hop headers are also required to be listed.

Conclusion

In this experiment I learned Wireshark and how to track a request made my us. I learned about things which goes to print website to our screen.