

We have some general notes on what happened and what we learned at the top, but wrote a more extensive walk through with screenshots in the pages below.

### General Notes & Sources

- 2 ports that did the handshake, only one sent a GET request, other closed connection immediately
- PSH is like the elevator button—pushing the current message to be immediately processed
- Sequence num becomes acknowledgement num
- First GET request has no authorization, server responds with WWW authorization header
- ARP translates between IP and MAC addresses
- Future GET requests to home page do not make HTTP requests, page is cached
- DNS lookup, A handles IPv4, AAAA handles IPv6 - only A returned an address
- ~600 DNS servers in the world
- We make request for favicon.ico from different port. It does not exist, and returns 404. Connection closes. We reset connection, and don't try to access it any more
- We try to connect from multiple ports sometimes, and when they fail, rather than retrying on the same port immediately, we just switch to another port and reconnect from there
- It seems that using multiple ports potentially speeds up communication
- Resets whenever it serves the resource
- Going back just retrieves cached html from browser
- Has to make new connection whenever we try to connect to a new link
- When we're not doing anything, it will make Keep-alive calls
- A lot of other stuff unrelated to what we are specifically doing happens in the background
- <https://datatracker.ietf.org/doc/html/rfc7617#section-4>

### Wireshark Walkthrough

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.60.128	192.168.60.2	DNS	80	Standard query 0x0ffa A cs338.jeffondich.com
2	0.000099003	192.168.60.128	192.168.60.2	DNS	80	Standard query 0x06a0 AAAA cs338.jeffondich.com
3	0.007473512	192.168.60.2	192.168.60.128	DNS	159	Standard query response 0x06a0 AAAA cs338.jeffondich.com SOA ns-cloud-d1...
4	0.007484912	192.168.60.2	192.168.60.128	DNS	390	Standard query response 0x0ffa A cs338.jeffondich.com A 45.79.89.123 NS n...
5	0.008717947	192.168.60.128	45.79.89.123	TCP	74	47306 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3669267...
6	0.008910052	192.168.60.128	45.79.89.123	TCP	74	47308 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3669267...
7	0.055603472	45.79.89.123	192.168.60.128	TCP	60	80 → 47306 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
8	0.055612972	45.79.89.123	192.168.60.128	TCP	60	80 → 47308 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
9	0.055724575	192.168.60.128	45.79.89.123	TCP	54	47306 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
10	0.055793677	192.168.60.128	45.79.89.123	TCP	54	47308 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
11	0.056290892	192.168.60.128	45.79.89.123	HTTP	395	GET /basicauth/ HTTP/1.1
12	0.056534298	45.79.89.123	192.168.60.128	TCP	60	80 → 47306 [ACK] Seq=1 Ack=342 Win=64240 Len=0
13	0.103991639	45.79.89.123	192.168.60.128	HTTP	457	HTTP/1.1 401 Unauthorized (text/html)
14	0.104031041	192.168.60.128	45.79.89.123	TCP	54	47306 → 80 [ACK] Seq=342 Ack=404 Win=63837 Len=0
15	2.787359431	192.168.60.128	72.21.21.29	TCP	54	58988 → 80 [ACK] Seq=1 Ack=1 Win=63920 Len=0
16	2.787699941	72.21.21.29	192.168.60.128	TCP	60	[TCP ACKed unseen segment] 80 → 58988 [ACK] Seq=1 Ack=2 Win=64240 Len=0
17	5.057292174	192.168.60.128	45.79.89.123	TCP	54	47308 → 80 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
18	5.057885391	45.79.89.123	192.168.60.128	TCP	60	80 → 47308 [ACK] Seq=1 Ack=2 Win=64239 Len=0

Frame 11: 395 bytes on wire (3160 bits), 395 bytes captured (3160 bits) on interface eth0, id 0  
 Ethernet II, Src: VMware\_59:80:5c (00:0c:29:59:80:5c), Dst: VMware\_f6:05:ab (00:50:56:f6:05:ab)  
 Internet Protocol Version 4, Src: 192.168.60.128, Dst: 45.79.89.123  
 Transmission Control Protocol, Src Port: 47306, Dst Port: 80, Seq: 1, Ack: 1, Len: 341  
 Hypertext Transfer Protocol  
 GET /basicauth/ HTTP/1.1  
 Host: cs338.jeffondich.com  
 User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:91.0) Gecko/20100101 Firefox/91.0  
 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8  
 Accept-Language: en-US,en;q=0.5  
 Accept-Encoding: gzip, deflate  
 Connection: keep-alive  
 Upgrade-Insecure-Requests: 1  
 Full request URI: http://cs338.jeffondich.com/basicauth/  
 HTTP request 1/1  
 Response in frame: 13

0000 00 50 56 f6 05 ab 00 0c 29 59 88 5c 08 00 45 00 PV .... )Y \ . E.  
 0010 01 7d 69 02 40 00 40 06 4c 86 c0 a8 3c 00 2d 4f }i @ @ L ... < - 0

wireshark eth0WAD1J1.pcapng Packets: 115 - Displayed: 115 (100.0%) - Dropped: 0 (0.0%) | Profile: Default

In this screenshot, several things are happening. First, look at the lines highlighted in blue (frames 1-4). The first two are requests to a DNS server for the IP address of the domain we just searched. There are two, because one is A type and the other is AAAA type—standing for IPv4 and IPv6 respectively. We get a response back from both, and the A type one gives us an IP address (the one used throughout the rest of the process).

Next, we connect to the server. Frames 5-10 are the three-way handshake. There are 6 messages because we try to connect from two different ports on our computer—both port 47306 and 47308. From what we can infer, this is to accelerate loading times on our computer, since we can have multiple channels open at once.

After making both connections, the first port to finish makes a GET request for /basicauth/. This fails, since it requires authentication.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.60.128	192.168.60.2	DNS	80	Standard query 0x0ffa A cs338.jeffondich.com
2	0.000000003	192.168.60.128	192.168.60.2	DNS	80	Standard query 0x06a0 AAAA cs338.jeffondich.com
3	0.007473512	192.168.60.2	192.168.60.128	DNS	159	Standard query response 0x06a0 AAAA cs338.jeffondich.com SOA ns-cloud-d1...
4	0.007484912	192.168.60.2	192.168.60.128	DNS	390	Standard query response 0x0ffa A cs338.jeffondich.com A 45.79.89.123 NS n...
5	0.008717947	192.168.60.128	45.79.89.123	TCP	74	47306 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3669267...
6	0.008910052	192.168.60.128	45.79.89.123	TCP	74	47308 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3669267...
7	0.055603472	45.79.89.123	192.168.60.128	TCP	60	80 → 47306 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
8	0.055612972	45.79.89.123	192.168.60.128	TCP	60	80 → 47308 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
9	0.055724575	192.168.60.128	45.79.89.123	TCP	54	47306 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
10	0.055793677	192.168.60.128	45.79.89.123	TCP	54	47308 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
11	0.056290892	192.168.60.128	45.79.89.123	HTTP	395	GET /basicauth/ HTTP/1.1
12	0.056534298	45.79.89.123	192.168.60.128	TCP	60	80 → 47306 [ACK] Seq=1 Ack=342 Win=64240 Len=0
13	0.103991639	45.79.89.123	192.168.60.128	HTTP	457	HTTP/1.1 401 Unauthorized (text/html)
14	0.104031041	192.168.60.128	45.79.89.123	TCP	54	47306 → 80 [ACK] Seq=342 Ack=404 Win=63837 Len=0
15	2.787359431	192.168.60.128	72.21.91.29	TCP	54	58988 → 80 [ACK] Seq=1 Ack=1 Win=63920 Len=0
16	2.787699941	72.21.91.29	192.168.60.128	TCP	60	[TCP ACKed unseen segment] 80 → 58988 [ACK] Seq=1 Ack=2 Win=64240 Len=0
17	5.057292174	192.168.60.128	45.79.89.123	TCP	54	47308 → 80 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
18	5.057885391	45.79.89.123	192.168.60.128	TCP	60	80 → 47308 [ACK] Seq=1 Ack=2 Win=64239 Len=0

Frame 13: 457 bytes on wire (3656 bits), 457 bytes captured (3656 bits) on interface eth0, id 0  
 Ethernet II, Src: VMware\_f6:05:ab (00:50:56:f6:05:ab), Dst: VMware\_59:80:5c (00:0c:29:59:80:5c)  
 Internet Protocol Version 4, Src: 45.79.89.123, Dst: 192.168.60.128  
 Transmission Control Protocol, Src Port: 80, Dst Port: 47306, Seq: 1, Ack: 342, Len: 403  
 Hypertext Transfer Protocol  
   HTTP/1.1 401 Unauthorized\r\n
   Server: nginx/1.18.0 (Ubuntu)\r\n
   Date: Thu, 07 Apr 2022 01:55:14 GMT\r\n
   Content-Type: text/html\r\n
   Content-Length: 188\r\n
   Connection: keep-alive\r\n
   WWW-Authenticate: Basic realm="Protected Area"\r\n
   \r\n
   [HTTP response 1/1]  
   [Time since request: 0.047700747 seconds]  
   [Request in frame: 11]  
   [Request URI: http://cs338.jeffondich.com/basicauth/]  
   File Data: 188 bytes  
 Line-based text data: text/html (7 lines)  
   <html>\r\n
   <head>\r\n
   <title>401 Unauthorized\r\n
   </title>\r\n
   </head>\r\n
   <body>\r\n
   <h1>401 Unauthorized\r\n
   </h1>\r\n
   </body>\r\n
   </html>\r\n

0000 00 0c 29 59 88 5c 00 50 56 f6 05 ab 08 00 45 00 ...Y \. P V...E...  
 0010 01 bb 09 d6 00 00 80 06 ab 74 2d 4f 59 7b c0 a8 .....t.OY...

wireshark\_ethOWAD1J1.pcapng      Packets: 115 - Displayed: 115 (100.0%) - Dropped: 0 (0.0%) | Profile: Default

This request fails, and sends back a 401 error, and gives us some html text telling us we can't access the requested files.

Note that in our response from our server, there is a HTTP header called WWW-Authenticate. This means that access to files in this region require username and password. After the header, it says that the authentication scheme is "Basic" and that this realm is "Protected Area". The "Basic" scheme tells the client how to format its authentication, and the region is just the portion of services protected by this authentication. If the region changed, we'd need to reauthenticate.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

No.	Time	Source	Destination	Protocol	Length	Info
67	66.074185508	72.21.91.29	192.168.60.128	TCP	60	80 → 58988 [FIN, PSH, ACK] Seq=1 Ack=3 Win=64239 Len=0
68	66.074165911	192.168.60.128	72.21.91.29	TCP	54	58988 → 80 [ACK] Seq=3 Ack=2 Win=63920 Len=0
69	94.409965400	192.168.60.128	45.79.89.123	TCP	74	47310 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3678797...
70	94.418486121	192.168.60.128	192.168.60.2	DNS	80	Standard query 0x8b2b A cs338.jeffondich.com
71	94.417579808	192.168.60.2	192.168.60.128	DNS	390	Standard query response 0x8b2b A cs338.jeffondich.com A 45.79.89.123 NS n...
72	94.469521839	45.79.89.123	192.168.60.128	TCP	60	80 → 47310 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
73	94.466039244	192.168.60.128	45.79.89.123	TCP	54	47310 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
74	94.461314371	192.168.60.128	45.79.89.123	HTTP	438	GET /basicauth/ HTTP/1.1
75	94.461079006	45.79.89.123	192.168.60.128	TCP	60	80 → 47310 [ACK] Seq=1 Ack=385 Win=64240 Len=0
76	94.511454393	45.79.89.123	192.168.60.128	HTTP	458	HTTP/1.1 200 OK (text/html)
77	94.511525697	192.168.60.128	45.79.89.123	TCP	54	47310 → 80 [ACK] Seq=385 Ack=405 Win=63836 Len=0
78	94.729391582	192.168.60.128	45.79.89.123	HTTP	355	GET /favicon.ico HTTP/1.1
79	94.738096710	45.79.89.123	192.168.60.128	TCP	60	80 → 47310 [ACK] Seq=405 Ack=686 Win=64240 Len=0
80	94.738238916	192.168.60.128	45.79.89.123	TCP	54	47310 → 80 [FIN, ACK] Seq=686 Ack=405 Win=63836 Len=0
81	94.738526328	45.79.89.123	192.168.60.128	TCP	60	80 → 47310 [ACK] Seq=405 Ack=687 Win=64239 Len=0
82	94.778698675	45.79.89.123	192.168.60.128	HTTP	383	HTTP/1.1 404 Not Found (text/html)
83	94.778733576	192.168.60.128	45.79.89.123	TCP	54	47310 → 80 [RST] Seq=687 Win=0 Len=0
84	95.256758210	192.168.60.128	45.79.89.123	TCP	74	47312 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3678792...

↳ Frame 74: 438 bytes on wire (3504 bits), 438 bytes captured (3504 bits) on interface eth0, id 0

↳ Ethernet II, Src: VMware 59:88:5c (00:0c:29:59:88:5c), Dst: VMware f6:05:ab (00:50:56:f6:05:ab)

↳ Internet Protocol Version 4, Src: 192.168.60.128, Dst: 45.79.89.123

↳ Transmission Control Protocol, Src Port: 47310, Dst Port: 80, Seq: 1, Ack: 1, Len: 384

↳ Hypertext Transfer Protocol

↳ GET /basicauth/ HTTP/1.1\r\n

Host: cs338.jeffondich.com\r\n

User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:91.0) Gecko/20100101 Firefox/91.0\r\n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8\r\n

Accept-Language: en-US,en;q=0.5\r\n

Accept-Encoding: gzip, deflate\r\n

Connection: keep-alive\r\n

Upgrade-Insecure-Requests: 1\r\n

↳ Authorization: Basic Y3MzMzZgcGFzc3dvcmQ=\r\n

Credentials: cs338:password

\r\n

[Full request URI: http://cs338.jeffondich.com/basicauth/]

[HTTP request 1/2]

[Response in frame 76]

[Next request in frame 78]

0030 fa f0 85 8d 00 90 47 45 54 20 2f 62 61 73 69 63 ..... GE T /basic
0040 61 75 74 68 2f 20 48 54 50 2f 31 2e 31 0d 0a ..... auth/ HT TP/1.1

Credentials (http.authbasic)
Packets: 115 · Displayed: 115 (100.0%) · Dropped: 0 (0.0%)
Profile: Default

Upon hitting submit on our authentication, we are allowed back in. Once again, we make a request, this time passing in our HTTP with a new header: "Authorization". This header includes our credentials, which are unencrypted (i.e. passed as plain text over the network). The server then verifies them, and grants us access if our credentials match.

After gaining access, we can now load the page. The server sends back the page with an OK message, and then attempts to fetch the icon for the tab. This request is made by the other port that we opened up, and we get a 404 page not found error, since there is no icon that exists for the page. We also send a reset (RST) message to the server, essentially asking to start over, since we've gotten what we've requested.

No.	Time	Source	Destination	Protocol	Length	Info
76	94.511454393	45.79.89.123	192.168.60.128	HTTP	458	HTTP/1.1 200 OK (text/html)
77	94.511525697	192.168.60.128	45.79.89.123	TCP	54	47310 → 80 [ACK] Seq=385 Ack=405 Win=63836 Len=0
78	94.729391582	192.168.60.128	45.79.89.123	HTTP	355	GET /favicon.ico HTTP/1.1
79	94.730096710	45.79.89.123	192.168.60.128	TCP	60	80 → 47310 [ACK] Seq=405 Ack=686 Win=64240 Len=0
80	94.730238916	192.168.60.128	45.79.89.123	TCP	54	47310 → 80 [FIN, ACK] Seq=686 Ack=405 Win=63836 Len=0
81	94.730526328	45.79.89.123	192.168.60.128	TCP	60	80 → 47310 [ACK] Seq=405 Ack=687 Win=64239 Len=0
82	94.778698675	45.79.89.123	192.168.60.128	HTTP	383	HTTP/1.1 404 Not Found (text/html)
83	94.778733576	192.168.60.128	45.79.89.123	TCP	54	47310 → 80 [RST] Seq=687 Win=0 Len=0
84	95.256758210	192.168.60.128	45.79.89.123	TCP	74	47312 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3678792...
85	95.303953318	45.79.89.123	192.168.60.128	TCP	60	80 → 47312 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
86	95.304072223	192.168.60.128	45.79.89.123	TCP	54	47312 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
87	95.965326794	192.168.60.128	45.79.89.123	HTTP	499	GET /basicauth/amateurs.txt HTTP/1.1
88	95.965700509	45.79.89.123	192.168.60.128	TCP	60	80 → 47312 [ACK] Seq=1 Ack=446 Win=64240 Len=0
89	95.966551744	192.168.60.128	45.79.89.123	TCP	54	47312 → 80 [FIN, ACK] Seq=446 Ack=1 Win=64240 Len=0
90	95.967102466	45.79.89.123	192.168.60.128	TCP	60	80 → 47312 [ACK] Seq=1 Ack=447 Win=64239 Len=0
91	96.014741599	45.79.89.123	192.168.60.128	HTTP	375	HTTP/1.1 200 OK (text/plain)
92	96.014818102	192.168.60.128	45.79.89.123	TCP	54	47312 → 80 [RST] Seq=447 Win=0 Len=0
93	98.696368707	192.168.60.128	45.79.89.123	TCP	74	47314 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3679136...

**Hypertext Transfer Protocol**

HTTP/1.1 200 OK\r\n

Server: nginx/1.13.0 (Ubuntu)\r\n

Date: Thu, 07 Apr 2022 01:56:50 GMT\r\n

Content-Type: text/plain\r\n

Content-Length: 75\r\n

Last-Modified: Mon, 04 Apr 2022 14:10:51 GMT\r\n

Connection: keep-alive\r\n

ETag: "624afc6b-4b"\r\n

Accept-Ranges: bytes\r\n

\r\n

[HTTP response 1/1]

[Time since request: 0.049414805 seconds]

[Request in frame: 87]

[Request URI: http://cs338.jeffondich.com/basicauth/amateurs.txt]

File Data: 75 bytes

Line-based text data: text/plain (3 lines)

"Amateurs hack systems, professionals hack people."\\n

\\n

-- Bruce Schneier\\n

0000 00 0c 29 59 88 5c 00 50 56 f6 05 ab 08 00 45 00 ..)Y.\P V....E  
0010 01 69 09 f4 00 00 80 06 ab a8 2d 4f 59 7b c0 a8 .i.....-OY(

After this, we can click on the pages linked in the doc. We have to reconnect, because of the RST, so we reconnect on a new port, and request the page. Since we're already authorized, we don't have to reauthorize ourselves. This is probably because we have been temporarily saved as a "trusted/authorized client" on the server.

No.	Time	Source	Destination	Protocol	Length	Info
98	99.275703403	192.168.60.128	45.79.89.123	TCP	54	47314 → 80 [FIN, ACK] Seq=450 Ack=1 Win=64240 Len=0
99	99.276225725	45.79.89.123	192.168.60.128	TCP	60	80 → 47314 [ACK] Seq=1 Ack=451 Win=64239 Len=0
100	99.321773288	45.79.89.123	192.168.60.128	HTTP	462	HTTP/1.1 200 OK (text/plain)
101	99.321821790	192.168.60.128	45.79.89.123	TCP	54	47314 → 80 [RST] Seq=451 Win=0 Len=0
102	99.321955495	45.79.89.123	192.168.60.128	TCP	60	80 → 47314 [FIN, PSH, ACK] Seq=409 Ack=451 Win=64239 Len=0
103	99.321972890	192.168.60.128	45.79.89.123	TCP	54	47314 → 80 [RST] Seq=451 Win=0 Len=0
104	101.685145905	192.168.60.128	45.79.89.123	TCP	74	47318 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3679435...
105	101.718045458	192.168.60.128	45.79.89.123	TCP	74	47320 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3679438...
106	101.736530420	45.79.89.123	192.168.60.128	TCP	60	80 → 47318 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
107	101.736641325	192.168.60.128	45.79.89.123	TCP	54	47318 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
108	101.765169402	45.79.89.123	192.168.60.128	TCP	60	80 → 47320 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
109	101.765340309	192.168.60.128	45.79.89.123	TCP	54	47320 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
110	102.415042713	192.168.60.128	45.79.89.123	HTTP	498	GET /basicauth/dancing.txt HTTP/1.1
111	102.415355225	45.79.89.123	192.168.60.128	TCP	60	80 → 47318 [ACK] Seq=1 Ack=445 Win=64240 Len=0
112	102.416431669	192.168.60.128	45.79.89.123	TCP	54	47318 → 80 [FIN, ACK] Seq=445 Ack=1 Win=64240 Len=0
113	102.416859987	45.79.89.123	192.168.60.128	TCP	60	80 → 47318 [ACK] Seq=1 Ack=446 Win=64239 Len=0
114	102.464634559	45.79.89.123	192.168.60.128	HTTP	528	HTTP/1.1 200 OK (text/plain)
115	102.464727202	192.168.60.128	45.79.89.123	TCP	54	47318 → 80 [RST] Seq=446 Win=0 Len=0

Future requests are as simple as making new connections, http requests, getting served the page, and then finishing our connection (FIN), and resetting. Note that when we go back to the previous page, it does not make a new request at all. This is because the previous page was cached entirely by our browser, and therefore *it* can serve us the previous page, rather than requesting it from the server. Sometimes multiple ports connect, sometimes not, though this seems to be largely random, based on our repeated tests.