

### The Basic Story (as we understand it):

The first thing that happens when you try to connect to this server is a DNS lookup. This DNS lookup finds the IP address of the server you are trying to contact, which is necessary for the next steps in the connection. Next, the three-way TCP handshake is completed. This handshake begins with a SYN, which is essentially your computer shouting out into the void “hey is anyone there at this address?”. The server then responds with a SYN, ACK, saying “yes I’m here, do you want to connect?”. Your computer then responds with an ACK, finalizing the connection by saying “Yes I want to connect”. After the TCP handshake is complete, both sides are connected, and are ready to pass requests or data between each other. The first data that is sent over by the server is some very basic HTML code that outlines a “401 Authorization Required” page, and a prompt for a username and password. Then, the user sends a request to the server for the rest of the page data, and in that query is the imputed username and password. The server checks if the credentials are correct, and if they are, it sends over the rest of the information contained in the secure site.

- What queries are sent from the browser, and what responses does it receive?
  - Line 60 shows the request, GET /basicauth which is the browser asking the server for the web page

The image shows a Wireshark packet capture of an HTTP transaction. The top pane displays a list of packets. Packet 60 is highlighted, showing a GET request for /basicauth. The bottom pane shows the details of the selected packet, including the Hypertext Transfer Protocol section. The request is a GET for /basicauth, and the response is a 401 Unauthorized message. The details pane also shows the full request URI and the response in frame 62.

No.	Time	Source	Destination	Protocol	Length	Info
58	0.168988103	172.233.221.124	192.168.64.2	TCP	66	80 → 41632 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
59	0.168996062	192.168.64.2	172.233.221.124	TCP	54	41632 → 80 [ACK] Seq=1 Ack=1 Win=32128 Len=0
60	0.169114520	192.168.64.2	172.233.221.124	HTTP	416	GET /basicauth HTTP/1.1
61	0.191932992	172.233.221.124	192.168.64.2	TCP	54	80 → 41632 [ACK] Seq=1 Ack=363 Win=64128 Len=0
62	0.191933034	172.233.221.124	192.168.64.2	HTTP	454	HTTP/1.1 301 Moved Permanently (text/html)
63	0.191949701	192.168.64.2	172.233.221.124	TCP	54	41632 → 80 [ACK] Seq=363 Ack=401 Win=31872 Len=0
64	0.193815035	192.168.64.2	172.233.221.124	TCP	74	35664 → 443 [SYN] Seq=0 Win=32120 Len=0 MSS=1460
65	0.217131133	172.233.221.124	192.168.64.2	TCP	66	443 → 35664 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0

```

Ethernet II, Src: ba:f3:c9:ae:35:41 (ba:f3:c9:ae:35:41), Dst: 1e:57:dc:d3:e1:64 (1e:57:dc:d3:e1:64)
Internet Protocol Version 4, Src: 192.168.64.2, Dst: 172.233.221.124
Transmission Control Protocol, Src Port: 41632, Dst Port: 80, Seq: 1, Ack: 1, Len: 362
Hypertext Transfer Protocol
  GET /basicauth HTTP/1.1\r\n
    Host: cs338.jeffondich.com\r\n
    User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0) Gecko/20100101 Firefox/115.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    DNT: 1\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://cs338.jeffondich.com/basicauth]
    [HTTP request 1/4]
    [Response in frame: 62]
    [Next request in frame: 75]
  
```

- Line 80 shows a 401 Unauthorized message, indicating that the client does not yet have access to the realm.
- Line 80 also has the header, Authenticate: Basic realm="Protected Area". Basic indicates the use of the HTTP Basic authentication scheme, and the rest of the text (namely the “Protected Area” string) is just an arbitrary label created by the developer.
- The server initially denied access to the “Protected Area” because the client had not yet imputed the required authentication credentials. Because of this we are

hit with a 401 unauthorized access code.

```

80 0.273092250 172.233.221.124 192.168.64.2 HTTP 457 HTTP/1.1 401 Unauthorized
81 0.273102250 192.168.64.2 172.233.221.124 TCP 54 41632 → 80 [ACK] Seq=720
82 10.158548215 192.168.64.2 23.64.114.200 TCP 54 [TCP Keep-Alive] 51486 →
83 10.158571590 192.168.64.2 23.64.114.200 TCP 54 [TCP Keep-Alive] 51480 →
84 10.184687815 23.64.114.200 192.168.64.2 TCP 54 [TCP Keep-Alive ACK] 80
85 10.184688232 23.64.114.200 192.168.64.2 TCP 54 [TCP Keep-Alive ACK] 80
86 10.414214629 192.168.64.2 172.233.221.124 TCP 54 [TCP Keep-Alive] 41632 →
87 10.439949520 172.233.221.124 192.168.64.2 TCP 54 [TCP Keep-Alive ACK] 80

HTTP/1.1 401 Unauthorized\r\n
  [Expert Info (Chat/Sequence): HTTP/1.1 401 Unauthorized\r\n]
  Response Version: HTTP/1.1
  Status Code: 401
  [Status Code Description: Unauthorized]
  Response Phrase: Unauthorized
  Server: nginx/1.18.0 (Ubuntu)\r\n
  Date: Tue, 24 Sep 2024 21:45:54 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 2/4]
  [Time since request: 0.023049389 seconds]
  [Prev request in frame: 60]
  [Prev response in frame: 62]
  [Request in frame: 75]
  [Response in frame: 76]
  HTTP WWW-Authenticate header (http.www_authenticate), 48 byte(s)
  Packets: 93 · Displayed: 93 (100.0%) · Dro

```

- Line 88 which was sent from the server to the browser contains the username and password we already put into the browser encoded in base 64, along with them in plaintext as well.
- After that, the server sends us all of the previously inaccessible data and html for the website.
- After the password is typed by the user, what sequence of queries and responses do you see?
  - Line 88 shows the credentials being sent to the server which is followed by line 89 which is the server sending the client the website's html and the secure sites data.



- We are pretty sure that this data is encoded, not encrypted, which means there is no encryption key.
- How does what you observe via Wireshark connect to the relevant sections of the HTTP and HTTP Basic Authentication specification documents?
  - As stated in the security considerations section of the Basic Authentication specification documents, the Basic authentication scheme is not secure. It seems that anyone monitoring a computer's network traffic would be able to intercept not only the encoded credentials, but also the human-comprehensible plaintext version, which is extremely insecure.
  - The Authentication header we found also lined up with what was expected from section 2 of the Basic Authentication specification documents, as we needed to authenticate ourselves in the eyes of the server in order to gain access to the protected space.