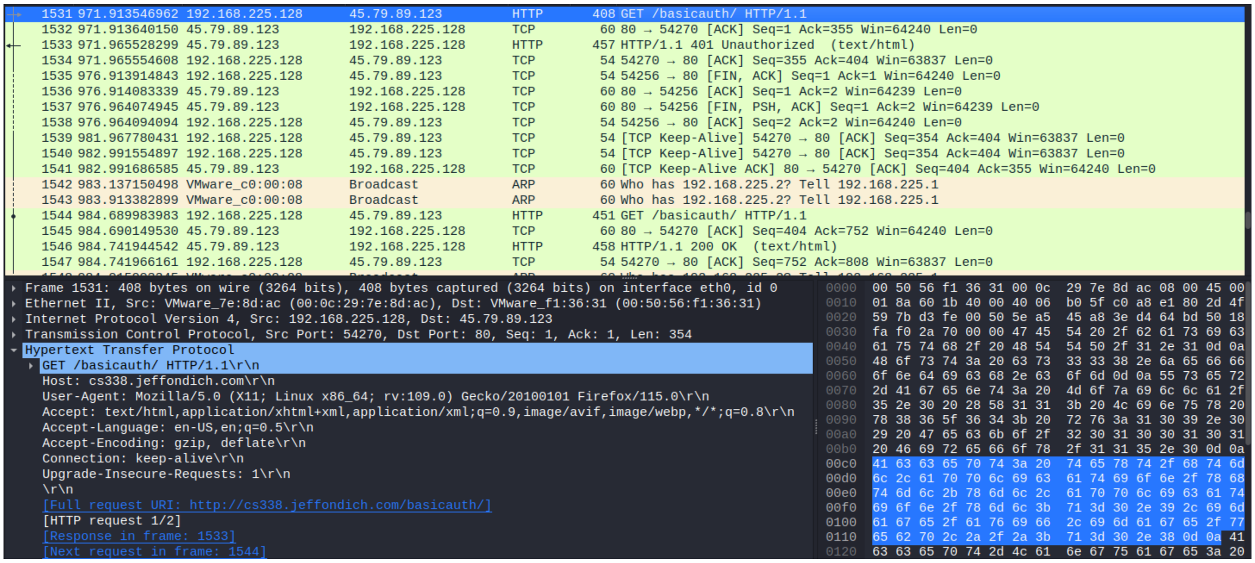
Jack Huffman
CS338 Computer Security
Jeff Ondich
9-19-2023

The authorization sequence which occurs when trying to access the super high tech webpage http://cs338.jeffondich.com/basicauth/ is surprisingly, relatively simple. As with most attempts to access a webpage, this interaction begins with a TCP handshake to establish a connection with the server. Following this the user's client sends a HTTP GET request to the host server, below is a wireshark screenshot of the frames associated with the interaction and the associated GET request.

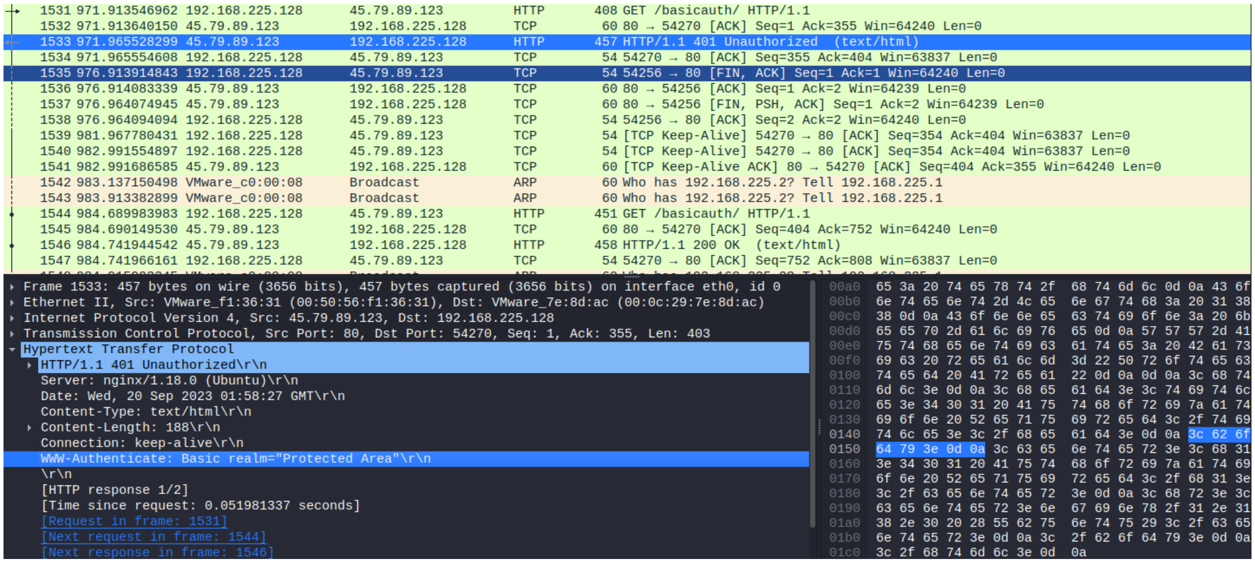


```
1531 971.913546962 192.168.225.128   45.79.89.123      HTTP   408 GET /basicauth/ HTTP/1.1
1532 971.913640150 45.79.89.123      192.168.225.128   TCP     60 80 → 54270 [ACK] Seq=1 Ack=355 Win=64240 Len=0
1533 971.965528299 45.79.89.123      192.168.225.128   HTTP   457 HTTP/1.1 401 Unauthorized  (text/html)
1534 971.965554608 192.168.225.128   45.79.89.123      TCP     54 54270 → 80 [ACK] Seq=355 Ack=404 Win=63837 Len=0
1535 976.913914843 192.168.225.128   45.79.89.123      TCP     54 54256 → 80 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
1536 976.914083339 45.79.89.123      192.168.225.128   TCP     60 80 → 54256 [ACK] Seq=1 Ack=2 Win=64239 Len=0
1537 976.964074945 45.79.89.123      192.168.225.128   TCP     60 80 → 54256 [FIN, PSH, ACK] Seq=1 Ack=2 Win=64239 Len=0
1538 976.964094094 192.168.225.128   45.79.89.123      TCP     54 54256 → 80 [ACK] Seq=2 Ack=2 Win=64240 Len=0
1539 981.967780431 192.168.225.128   45.79.89.123      TCP     54 [TCP Keep-Alive] 54270 → 80 [ACK] Seq=354 Ack=404 Win=63837 Len=0
1540 982.991554897 192.168.225.128   45.79.89.123      TCP     54 [TCP Keep-Alive] 54270 → 80 [ACK] Seq=354 Ack=404 Win=63837 Len=0
1541 982.991686585 45.79.89.123      192.168.225.128   TCP     60 [TCP Keep-Alive ACK] 80 → 54270 [ACK] Seq=404 Ack=355 Win=64240 Len=0
1542 983.137150498 VMware_c0:00:08   Broadcast         ARP     60 Who has 192.168.225.2? Tell 192.168.225.1
1543 983.913382899 VMware_c0:00:08   Broadcast         ARP     60 Who has 192.168.225.2? Tell 192.168.225.1
1544 984.689983983 192.168.225.128   45.79.89.123      HTTP   451 GET /basicauth/ HTTP/1.1
1545 984.690149530 45.79.89.123      192.168.225.128   TCP     60 80 → 54270 [ACK] Seq=404 Ack=752 Win=64240 Len=0
1546 984.741944542 45.79.89.123      192.168.225.128   HTTP   458 HTTP/1.1 200 OK  (text/html)
1547 984.741966161 192.168.225.128   45.79.89.123      TCP     54 54270 → 80 [ACK] Seq=752 Ack=808 Win=63837 Len=0
```

```
▸ Frame 1531: 408 bytes on wire (3264 bits), 408 bytes captured (3264 bits) on interface eth0, id 0
▸ Ethernet II, Src: VMware_7e:8d:ac (00:0c:29:7e:8d:ac), Dst: VMware_f1:36:31 (00:50:56:f1:36:31)
▸ Internet Protocol Version 4, Src: 192.168.225.128, Dst: 45.79.89.123
▸ Transmission Control Protocol, Src Port: 54270, Dst Port: 80, Seq: 1, Ack: 1, Len: 354
▾ 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: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
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://cs338.jeffondich.com/basicauth/]
    [HTTP request 1/2]
    [Response in frame: 1533]
    [Next request in frame: 1544]
```

```
0000  00 50 56 f1 36 31 00 0c  29 7e 8d ac 08 00 45 00
0010  01 8a 60 1b 40 00 40 06  b0 5f c0 a8 e1 80 2d 4f
0020  59 7b d3 fe 00 50 5e a5  45 a8 3e d4 64 bd 50 18
0030  fa f0 2a 70 00 00 47 45  54 20 2f 62 61 73 69 63
0040  61 75 74 68 2f 20 48 54  54 50 2f 31 2e 31 0d 0a
0050  48 6f 73 74 3a 20 63 73  33 33 38 2e 6a 65 66 66
0060  6f 6e 64 69 63 68 2e 63  6f 6d 0d 0a 55 73 65 72
0070  2d 41 67 65 6e 74 3a 20  4d 6f 7a 69 6c 6c 61 2f
0080  35 2e 30 20 28 58 31 31  3b 20 4c 69 6e 75 78 20
0090  78 38 36 5f 36 34 3b 20  72 76 3a 31 30 39 2e 30
00a0  29 20 47 65 63 6b 6f 2f  32 30 31 30 30 31 30 31
00b0  20 46 69 72 65 66 6f 78  2f 31 31 35 2e 30 0d 0a
00c0  41 63 63 65 70 74 3a 20  74 65 78 74 2f 68 74 6d
00d0  6c 2c 61 70 70 6c 69 63  61 74 69 6f 6e 2f 78 68
00e0  74 6d 6c 2b 78 6d 6c 2c  61 70 70 6c 69 63 61 74
00f0  69 6f 6e 2f 78 6d 6c 3b  71 3d 30 2e 39 2c 69 6d
0100  61 67 65 2f 61 76 69 66  2c 69 6d 61 67 65 2f 77
0110  65 62 70 2c 2a 2f 2a 3b  71 3d 30 2e 38 0d 0a 41
0120  63 63 65 70 74 2d 4c 61  6e 67 75 61 67 65 3a 20
```

Instead of being granted access, the server recognizes that the client is currently unauthorized to access the page and sends a 401 Unauthorized status code back to the client. Contained within this status code is a ‘WWW-Authenticate’ header that defines which HTTP Authentication scheme is to be used to access the server, this header can be seen in the screenshot below.

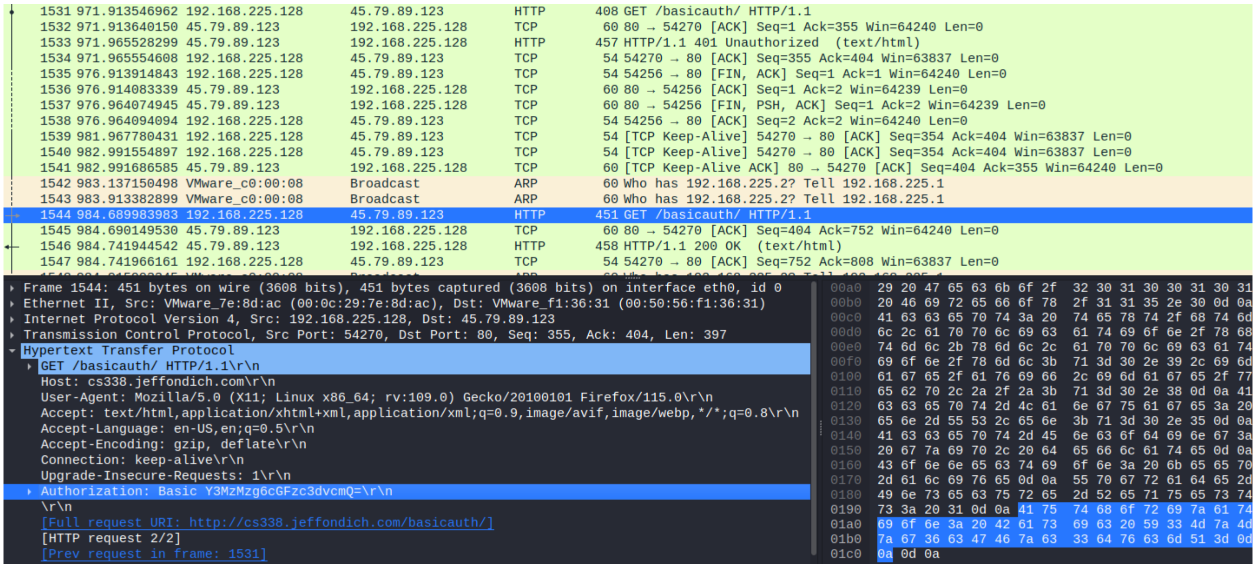


```
1531 971.913546962 192.168.225.128   45.79.89.123      HTTP   408 GET /basicauth/ HTTP/1.1
1532 971.913640150 45.79.89.123      192.168.225.128   TCP     60 80 → 54270 [ACK] Seq=1 Ack=355 Win=64240 Len=0
1533 971.965528299 45.79.89.123      192.168.225.128   HTTP   457 HTTP/1.1 401 Unauthorized  (text/html)
1534 971.965554608 192.168.225.128   45.79.89.123      TCP     54 54270 → 80 [ACK] Seq=355 Ack=404 Win=63837 Len=0
1535 976.913914843 192.168.225.128   45.79.89.123      TCP     54 54256 → 80 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
1536 976.914083339 45.79.89.123      192.168.225.128   TCP     60 80 → 54256 [ACK] Seq=1 Ack=2 Win=64239 Len=0
1537 976.964074945 45.79.89.123      192.168.225.128   TCP     60 80 → 54256 [FIN, PSH, ACK] Seq=1 Ack=2 Win=64239 Len=0
1538 976.964094094 192.168.225.128   45.79.89.123      TCP     54 54256 → 80 [ACK] Seq=2 Ack=2 Win=64240 Len=0
1539 981.967780431 192.168.225.128   45.79.89.123      TCP     54 [TCP Keep-Alive] 54270 → 80 [ACK] Seq=354 Ack=404 Win=63837 Len=0
1540 982.991554897 192.168.225.128   45.79.89.123      TCP     54 [TCP Keep-Alive] 54270 → 80 [ACK] Seq=354 Ack=404 Win=63837 Len=0
1541 982.991686585 45.79.89.123      192.168.225.128   TCP     60 [TCP Keep-Alive ACK] 80 → 54270 [ACK] Seq=404 Ack=355 Win=64240 Len=0
1542 983.137150498 VMware_c0:00:08   Broadcast         ARP     60 Who has 192.168.225.2? Tell 192.168.225.1
1543 983.913382899 VMware_c0:00:08   Broadcast         ARP     60 Who has 192.168.225.2? Tell 192.168.225.1
1544 984.689983983 192.168.225.128   45.79.89.123      HTTP   451 GET /basicauth/ HTTP/1.1
1545 984.690149530 45.79.89.123      192.168.225.128   TCP     60 80 → 54270 [ACK] Seq=404 Ack=752 Win=64240 Len=0
1546 984.741944542 45.79.89.123      192.168.225.128   HTTP   458 HTTP/1.1 200 OK  (text/html)
1547 984.741966161 192.168.225.128   45.79.89.123      TCP     54 54270 → 80 [ACK] Seq=752 Ack=808 Win=63837 Len=0
```

```
▸ Frame 1533: 457 bytes on wire (3656 bits), 457 bytes captured (3656 bits) on interface eth0, id 0
▸ Ethernet II, Src: VMware_f1:36:31 (00:50:56:f1:36:31), Dst: VMware_7e:8d:ac (00:0c:29:7e:8d:ac)
▸ Internet Protocol Version 4, Src: 45.79.89.123, Dst: 192.168.225.128
▸ Transmission Control Protocol, Src Port: 80, Dst Port: 54270, Seq: 1, Ack: 355, Len: 403
▾ Hypertext Transfer Protocol
  ▸ HTTP/1.1 401 Unauthorized\r\n
    Server: nginx/1.18.0 (Ubuntu)\r\n
    Date: Wed, 20 Sep 2023 01:58:27 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 1/2]
    [Time since request: 0.051981337 seconds]
    [Request in frame: 1531]
    [Next request in frame: 1544]
    [Next response in frame: 1546]
```

```
00a0  65 3a 20 74 65 78 74 2f  68 74 6d 6c 0d 0a 43 6f
00b0  6e 74 65 6e 74 2d 4c 65  6e 67 74 68 3a 20 31 38
00c0  38 0d 0a 43 6f 6e 6e 65  63 74 69 6f 6e 3a 20 6b
00d0  65 65 70 2d 61 6c 69 76  65 0d 0a 57 57 57 2d 41
00e0  75 74 68 65 6e 74 69 63  61 74 65 3a 20 42 61 73
00f0  69 63 20 72 65 61 6c 6d  3d 22 50 72 6f 74 65 63
0100  74 65 64 20 41 72 65 61  22 0d 0a 0d 0a 3c 68 74
0110  6d 6c 3e 0d 0a 3c 68 65  61 64 3e 3c 74 69 74 6c
0120  65 3e 34 30 31 20 41 75  74 68 6f 72 69 7a 61 74
0130  69 6f 6e 20 52 65 71 75  69 72 65 64 3c 2f 74 69
0140  74 6c 65 3e 3c 2f 68 65  61 64 3e 0d 0a 3c 62 6f
0150  64 79 3e 0d 0a 3c 63 65  6e 74 65 72 3e 3c 68 31
0160  3e 34 30 31 20 41 75 74  68 6f 72 69 7a 61 74 69
0170  6f 6e 20 52 65 71 75 69  72 65 64 3c 2f 68 31 3e
0180  3c 2f 63 65 6e 74 65 72  3e 0d 0a 3c 68 72 3e 3c
0190  63 65 6e 74 65 72 3e 6e  67 69 6e 78 2f 31 2e 31
01a0  38 2e 30 20 28 55 62 75  6e 74 75 29 3c 2f 63 65
01b0  6e 74 65 72 3e 0d 0a 3c  2f 62 6f 64 79 3e 0d 0a
01c0  3c 2f 68 74 6d 6c 3e 0d  0a
```

From the WWW-Authenticate header we can see that the server is communicating to the client that basic authentication is required to access the protected area. Once the client receives this 401 message it needs to gather authorization information from the user to pass on to the server, as such it will prompt the user to input a username and password. Once this information has been inputted the client takes the username and password in the form ‘username:password’ and encodes it in base64. It is important to note that this is not a form of encryption but instead simply encoding the data so information is preserved in transit. Now the client can resend its HTTP GET request to the server but this time with an authorization header, this request can be seen in the screenshot below.



```
1531 971.913546962 192.168.225.128   45.79.89.123      HTTP   408 GET /basicauth/ HTTP/1.1
1532 971.913640150 45.79.89.123      192.168.225.128   TCP     60 80 → 54270 [ACK] Seq=1 Ack=355 Win=64240 Len=0
1533 971.965528299 45.79.89.123      192.168.225.128   HTTP   457 HTTP/1.1 401 Unauthorized  (text/html)
1534 971.965554608 192.168.225.128   45.79.89.123      TCP     54 54270 → 80 [ACK] Seq=355 Ack=404 Win=63837 Len=0
1535 976.913914843 192.168.225.128   45.79.89.123      TCP     54 54256 → 80 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
1536 976.914083339 45.79.89.123      192.168.225.128   TCP     60 80 → 54256 [ACK] Seq=1 Ack=2 Win=64239 Len=0
1537 976.964074945 45.79.89.123      192.168.225.128   TCP     60 80 → 54256 [FIN, PSH, ACK] Seq=1 Ack=2 Win=64239 Len=0
1538 976.964094094 192.168.225.128   45.79.89.123      TCP     54 54256 → 80 [ACK] Seq=2 Ack=2 Win=64240 Len=0
1539 981.967780431 192.168.225.128   45.79.89.123      TCP     54 [TCP Keep-Alive] 54270 → 80 [ACK] Seq=354 Ack=404 Win=63837 Len=0
1540 982.991554897 192.168.225.128   45.79.89.123      TCP     54 [TCP Keep-Alive] 54270 → 80 [ACK] Seq=354 Ack=404 Win=63837 Len=0
1541 982.991686585 45.79.89.123      192.168.225.128   TCP     60 [TCP Keep-Alive ACK] 80 → 54270 [ACK] Seq=404 Ack=355 Win=64240 Len=0
1542 983.137150498 VMware_c0:00:08   Broadcast         ARP     60 Who has 192.168.225.2? Tell 192.168.225.1
1543 983.913382899 VMware_c0:00:08   Broadcast         ARP     60 Who has 192.168.225.2? Tell 192.168.225.1
1544 984.689983983 192.168.225.128   45.79.89.123      HTTP   451 GET /basicauth/ HTTP/1.1
1545 984.690149530 45.79.89.123      192.168.225.128   TCP     60 80 → 54270 [ACK] Seq=404 Ack=752 Win=64240 Len=0
1546 984.741944542 45.79.89.123      192.168.225.128   HTTP   458 HTTP/1.1 200 OK  (text/html)
1547 984.741966161 192.168.225.128   45.79.89.123      TCP     54 54270 → 80 [ACK] Seq=752 Ack=808 Win=63837 Len=0
```

```
Frame 1544: 451 bytes on wire (3608 bits), 451 bytes captured (3608 bits) on interface eth0, id 0
Ethernet II, Src: VMware_7e:8d:ac (00:0c:29:7e:8d:ac), Dst: VMware_f1:36:31 (00:50:56:f1:36:31)
Internet Protocol Version 4, Src: 192.168.225.128, Dst: 45.79.89.123
Transmission Control Protocol, Src Port: 54270, Dst Port: 80, Seq: 355, Ack: 404, Len: 397
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: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
  Connection: keep-alive\r\n
  Upgrade-Insecure-Requests: 1\r\n
  Authorization: Basic Y3MzMzg6cGFzc3dvcmQ=\r\n
  \r\n
  [Full request URI: http://cs338.jeffondich.com/basicauth/]
  [HTTP request 2/2]
  [Prev request in frame: 1531]
```

```
00a0  29 20 47 65 63 6b 6f 2f  32 30 31 30 30 31 30 31
00b0  20 46 69 72 65 66 6f 78  2f 31 31 35 2e 30 0d 0a
00c0  41 63 63 65 70 74 3a 20  74 65 78 74 2f 68 74 6d
00d0  6c 2c 61 70 70 6c 69 63  61 74 69 6f 6e 2f 78 68
00e0  74 6d 6c 2b 78 6d 6c 2c  61 70 70 6c 69 63 61 74
00f0  69 6f 6e 2f 78 6d 6c 3b  71 3d 30 2e 39 2c 69 6d
0100  61 67 65 2f 61 76 69 66  2c 69 6d 61 67 65 2f 77
0110  65 62 70 2c 2a 2f 2a 3b  71 3d 30 2e 38 0d 0a 41
0120  63 63 65 70 74 2d 4c 61  6e 67 75 61 67 65 3a 20
0130  65 6e 2d 55 53 2c 65 6e  3b 71 3d 30 2e 35 0d 0a
0140  41 63 63 65 70 74 2d 45  6e 63 6f 64 69 6e 67 3a
0150  20 67 7a 69 70 2c 20 64  65 66 6c 61 74 65 0d 0a
0160  43 6f 6e 6e 65 63 74 69  6f 6e 3a 20 6b 65 65 70
0170  2d 61 6c 69 76 65 0d 0a  55 70 67 72 61 64 65 2d
0180  49 6e 73 65 63 75 72 65  2d 52 65 71 75 65 73 74
0190  73 3a 20 31 0d 0a 41 75  74 68 6f 72 69 7a 61 74
01a0  69 6f 6e 3a 20 42 61 73  69 63 20 59 33 4d 7a 4d
01b0  7a 67 36 63 47 46 7a 63  33 64 76 63 6d 51 3d 0d
01c0  0a 0d 0a
```

This authorization header once again states that basic authentication is being used and following this is the base64 encoded version of our username and password (of the form ‘username:password’). This means that the browser is not authenticating the credentials but is instead sending the credentials to the server who will do the authentication. The server does this by decoding the base64 representation and checks the username-password combo against its database of acceptable inputs. If the credentials provided allow access to the requested material it will send that material to the browser, if the server failed to authenticate the credentials it will send another 401 message to the client, prompting them to reattempt the submission of credentials.

A few things to note:

- The browser does not check the validity of the credentials, but rather encodes it into base64 and sends it to the server who will do the checking
- The password is not encrypted, it is sent as clear text, meaning that if the message were to be intercepted it would be quite simple to deduce the username and password, this poses a serious threat to privacy and is a clear weakness of the HTTP basic authentication

- If this interaction between client and server were secured with HTTPS, then the entire server would be encrypted making it much more secure. During this process an encryption key would be generated during the TLS handshake.

Citations:

RFC 7616 - The 'Basic' HTTP Authentication Scheme
https://datatracker.ietf.org/doc/html/rfc7617