

Leo Moran

09/24/2024

Once I first entered the link, my browser went through the standard TCP Handshake (Figure 1, Packets 1-3) before requesting access to the basicauth page. The server (which I know due to the source of the IP) acknowledged the request and provided a 401 error sign stating how the client was not authorized (Figure 1, Packet 6), which led to the pop up asking for login details.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.197.128	172.233.221.124	TCP	74	34318 → 80 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSval=3454788807 TSecr=0 WS=128
2	0.023100270	172.233.221.124	192.168.197.128	TCP	60	80 → 34318 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
3	0.023163687	192.168.197.128	172.233.221.124	TCP	54	34318 → 80 [ACK] Seq=1 Ack=1 Win=32120 Len=0
4	0.023507710	192.168.197.128	172.233.221.124	HTTP	499	GET /basicauth/ HTTP/1.1
5	0.023825361	172.233.221.124	192.168.197.128	TCP	60	80 → 34318 [ACK] Seq=1 Ack=446 Win=64240 Len=0
6	0.046596309	172.233.221.124	192.168.197.128	HTTP	859	HTTP/1.1 401 Unauthorized (text/html)
7	0.046643300	192.168.197.128	172.233.221.124	TCP	54	34318 → 80 [ACK] Seq=446 Ack=806 Win=31395 Len=0

(Figure 1)

When looking over Burp Suite I can see how in the response of my browser's request for access, the webpage is considered as a Protected Area (Fig. 2, Response Line 7), which led me to a different webpage than the one I requested (Fig. 2, Response Lines 9-15), but it does not appear to be the login popup.

Request						Response				
Pretty Raw Hex						Pretty Raw Hex Render				
1	GET /basicauth/ HTTP/1.1					1	HTTP/1.1 401 Unauthorized			
2	Host: cs338.jeffondich.com					2	Server: nginx/1.18.0 (Ubuntu)			
3	Accept-Language: en-US					3	Date: Wed, 25 Sep 2024 01:19:15 GMT			
4	Upgrade-Insecure-Requests: 1					4	Content-Type: text/html			
5	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36					5	Content-Length: 590			
6	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7					6	Connection: keep-alive			
7	Accept-Encoding: gzip, deflate, br					7	WWW-Authenticate: Basic realm="Protected Area"			
8	Connection: keep-alive					8	<html>			
9						9	<head>			
10						10	<title>			
							401 Authorization Required			
							</title>			
							</head>			
						11	<body>			
						12	<center>			
							<h1>			
							401 Authorization Required			
							</h1>			
							</center>			
						13	<center>			
							nginx/1.18.0 (Ubuntu)			
							</center>			
						14	</body>			
						15	</html>			

(Figure 2)

Once I entered the login details and pressed entered, another request was sent, which is similar to my initial request, except now there is a new line that provides a form of authorization (Fig. 3, Request Line 4). Once the request was sent, the response returned with a 200 error sign (Fig. 3, Response Line 1) instead of a 401, allowing me access to the webpage. It is also worth noting how the Protected Area line previously is not seen within this new response.

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
1	GET /basicauth/ HTTP/1.1			1	HTTP/1.1 200 OK		
2	Host: cs338.jeffondsch.com			2	Server: nginx/1.18.0 (Ubuntu)		
3	Cache-Control: max-age=0			3	Date: Wed, 25 Sep 2024 01:19:29 GMT		
4	Authorization: Basic YSMzMzg6cGFzc3dvcwQ=			4	Content-Type: text/html		
5	Accept-Language: en-US			5	Connection: keep-alive		
6	Upgrade-Insecure-Requests: 1			6	Content-Length: 509		
7	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)			7			
8	Chrome/126.0.6478.127 Safari/537.36			8	<html>		
9	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7			9	<head>		
10	Accept-Encoding: gzip, deflate, br				<title>		
11	Connection: keep-alive				Index of /basicauth/		
					</title>		
					</head>		
					<body>		
					<h1>		
					Index of /basicauth/		
					</h1>		
					<hr>		
					<pre>		
							
					../		
							
							
					amateurs.txt		
							
							
					armed-guards.txt		
							

(Figure 3)

As part of the response for the accepted request, I can now see the webpage, and the list of links it had (Fig. 4 & 5, Lines 8-16), which were all links to different text files.

Response			
Pretty	Raw	Hex	Render
1	HTTP/1.1 200 OK		
2	Server: nginx/1.18.0 (Ubuntu)		
3	Date: Wed, 25 Sep 2024 01:19:29 GMT		
4	Content-Type: text/html		
5	Connection: keep-alive		
6	Content-Length: 509		
7			
8	<html>		
9	<head>		
	<title>		
	Index of /basicauth/		
	</title>		
	</head>		
10	<body>		
11	<h1>		
	Index of /basicauth/		
	</h1>		
	<hr>		
	<pre>		
			
	../		
			
12			
	amateurs.txt		
			
13			
	armed-guards.txt		
			

04-Apr-2022 14:10

75

(Figure 4)

		04-Apr-2022 14:10	161
14	<code> dancing.txt </code>		
		04-Apr-2022 14:10	227
15	<code></pre> <hr> </body></code>		
16	<code></html></code>		
17			

(Figure 5)

It should be noted that on Wire Shark, after entering the login details another request was seen on Wire Shark that was near identical to my initial one, except this one was accepted (Fig. 6, Package 18-20).

17	11.088178218	172.233.221.124	192.168.197.128	TCP	60 [TCP Keep-Alive ACK] 80 → 60898 [ACK] Seq=404 Ack=355 Win=64240 Len=0
18	12.411113444	192.168.197.128	172.233.221.124	HTTP	451 GET /basicauth/ HTTP/1.1
19	12.411555782	172.233.221.124	192.168.197.128	TCP	60 80 → 60898 [ACK] Seq=404 Ack=752 Win=64240 Len=0
20	12.435789204	172.233.221.124	192.168.197.128	HTTP	458 HTTP/1.1 200 OK (text/html)
21	12.435871351	192.168.197.128	172.233.221.124	TCP	54 60898 → 80 [ACK] Seq=752 Ack=808 Win=31717 Len=0
22	12.594296983	192.168.197.128	172.233.221.124	HTTP	368 GET /favicon.ico HTTP/1.1
23	12.594683062	172.233.221.124	192.168.197.128	TCP	60 80 → 60898 [ACK] Seq=808 Ack=1066 Win=64240 Len=0
24	12.596501137	192.168.197.128	172.233.221.124	TCP	54 60898 → 80 [FIN, ACK] Seq=1066 Ack=808 Win=31717 Len=0
25	12.596841669	172.233.221.124	192.168.197.128	TCP	60 80 → 60898 [ACK] Seq=808 Ack=1067 Win=64239 Len=0

(Figure 6)

I then ran through the same process again, except this time, I purposefully entered the wrong login details twice, the first time I used cs339 for the username, then for the second time I used passwords for the password. I've noticed how when I changed the username, in the center of the Authorization code had zk(Fig. 7, Line 4), and when I did the wrong password, the latter half of the code had Rz (Fig. 8, Line 4). When comparing the codes from the wrong logins to the correct login, I then realized that the login details are set on one string, and are encrypted before being sent to the server (Fig. 9, Line 4).

Request		Raw	Hex
	Pretty		
1	GET /basicauth/ HTTP/1.1		
2	Host: cs338.jeffondich.com		
3	Cache-Control: max-age=0		
4	Authorization: Basic Y3MzMzk6cGFzc3dvcmQ=		
5	Accept-Language: en-US		
6	Upgrade-Insecure-Requests: 1		
7	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36		
8	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7		
9	Accept-Encoding: gzip, deflate, br		
10	Connection: keep-alive		
11			
12			

(Figure 7)

Request

```
1 GET /basicauth/ HTTP/1.1
2 Host: cs338.jeffondich.com
3 Cache-Control: max-age=0
4 Authorization: Basic Y3MzMzk6cGFzc3dvcnRz
5 Accept-Language: en-US
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/126.0.6478.127 Safari/537.36
8 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
9 Accept-Encoding: gzip, deflate, br
10 Connection: keep-alive
11
12
```

(Figure 8)

Request

```
1 GET /basicauth/ HTTP/1.1
2 Host: cs338.jeffondich.com
3 Cache-Control: max-age=0
4 Authorization: Basic Y3MzMzg6cGFzc3dvcnQ=
5 Accept-Language: en-US
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/126.0.6478.127 Safari/537.36
8 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
9 Accept-Encoding: gzip, deflate, br
.0 Connection: keep-alive
.1
.2
```

(Figure 9)