

Emma Roskopf

Collaborators: Ashok Khare, Lily Haas (we're each submitting our own document but worked together to figure out what's going on) (we didn't all wanna boot up Kali)

## Introduction to nginx password protection

- Authentication is when the website is checking to make sure the user is who they say they are
  - Passwords are often the way that this works, enter the correct username and password and then the user is confirmed
- Authorization is checking that the user is allowed to access certain areas, information, folders, etc. Like a level of clearance
  - Here, the authorization is just being authenticated, certain users don't have more privileges than others

## Wireshark Sniffing at the website

- We used the capture filter "host 45.79.89.123" because the basic auth site is on the same IP address as the other parts of the site, which we confirmed by running "wget <http://cs338.jeffondich.com/basicauth/>" on the terminal and it gave back the same IP
- The goal here is to get all communications between our computer, port 40408 and the server that we are requesting info from, running on port 80

- |    |              |                |                |      |   |
|----|--------------|----------------|----------------|------|---|
| 17 | 46.165795275 | 172.16.168.129 | 45.79.89.123   | HTTP | 438 GET /basicauth/ HTTP/1.1                        |
| 18 | 46.166246457 | 45.79.89.123   | 172.16.168.129 | TCP  | 60 80 → 40408 [ACK] Seq=404 Ack=726 Win=64240 Len=0 |

- So here you can see we're getting the main page of the website

- |    |              |                |                |     |  |
|----|--------------|----------------|----------------|-----|--|
| 25 | 56.328520824 | 172.16.160.129 | 45.79.89.123   | TCP | 54 [TCP Keep-Alive] 40408 → 80 [ACK] Seq=1026 Ack=1137 Win=63837 Len=0     |
| 26 | 57.352554566 | 172.16.160.129 | 45.79.89.123   | TCP | 54 [TCP Keep-Alive] 40408 → 80 [ACK] Seq=1026 Ack=1137 Win=63837 Len=0     |
| 27 | 57.352957290 | 45.79.89.123   | 172.16.160.129 | TCP | 60 [TCP Keep-Alive ACK] 80 → 40408 [ACK] Seq=1137 Ack=1027 Win=64240 Len=0 |

- Here TCP is doing something, each frame has [TCP Keep-Alive]. We're thinking this is the client and server making sure each other is there
  - “Still alive over there” “Yep” “K”

- |    |              |                |                |      |   |
|----|--------------|----------------|----------------|------|---|
| 28 | 57.408900307 | 172.16.160.129 | 45.79.89.123   | HTTP | 498 GET /basicauth/dancing.txt HTTP/1.1               |
| 29 | 57.409182962 | 45.79.89.123   | 172.16.160.129 | TCP  | 60 80 → 40408 [ACK] Seq=1137 Ack=1471 Win=64240 Len=0 |
| 30 | 57.455453713 | 45.79.89.123   | 172.16.160.129 | HTTP | 528 HTTP/1.1 200 OK (text/plain)                      |
| 31 | 57.455475871 | 172.16.160.129 | 45.79.89.123   | TCP  | 54 40408 → 80 [ACK] Seq=1471 Ack=1611 Win=63837 Len=0 |

- Here we have gotten into the authentication password blocked area of the website
  - In Frame 28, we see the name of the file I accessed, /basicauth/dancing.txt
  - This seems wrong because this file and its name was blocked but by sniffing I can see what the client is looking at, and see that it is being rendered in plain text
- That was a pretty good first pass, so now we're going to try to mess with it instead of following the regular order that things should be done

## We found the HTML for the secure side of the website

6	0.117979410	45.79.89.123	172.16.160.129	TCP	54 80 → 40416 [ACK] Seq=1 Ack=385 Win=63836 Len=0
7	0.117979410	45.79.89.123	172.16.160.129	HTTP	458 HTTP/1.1 200 OK (text/html)
8	3.305220777	172.16.160.129	45.79.89.123	HTTP	499 GET /basicauth/amateurs.txt HTTP/1.1
9	3.305475822	45.79.89.123	172.16.160.129	TCP	60 80 → 40416 [ACK] Seq=405 Ack=830 Win=64240 Len=0
10	3.351617662	45.79.89.123	172.16.160.129	HTTP	375 HTTP/1.1 200 OK (text/plain)
11	3.351630600	172.16.160.129	45.79.89.123	TCP	54 40416 → 80 [ACK] Seq=830 Ack=726 Win=63836 Len=0

```

Content-encoded entity body (gzip): 205 bytes -> 509 bytes
File Data: 509 bytes
Line-based text data: text/html (9 lines)
<html>\r\n
<head><title>Index of /basicauth/</title></head>\r\n
<body>\r\n
<h1>Index of /basicauth/</h1><hr><pre><a href="..">../</a>\r\n
<a href="amateurs.txt">amateurs.txt</a>                                04-Apr-2022 14:10          75\r\n
<a href="armed-guards.txt">armed-guards.txt</a>                      04-Apr-2022 14:10          161\r\n
<a href="dancing.txt">dancing.txt</a>                                04-Apr-2022 14:10          227\r\n
</pre><hr></body>\r\n
</html>\r\n

```

- On the second pass, we realized that by expanding the HTTP messages we can get the line data, which tells us the HTML being rendered and gives the names of all the files
- These file names could be sensitive or private, and so being able to see them seems bad

**We found the password and credential by dropdown menu underneath ‘Authorization’**

```
4 0.071506925 172.16.160.129 45.79.89.123 HTTP 438 GET /basicauth/ HTTP/1.1
5 0.071826271 45.79.89.123 172.16.160.129 TCP 60 80 → 40416 [ACK] Seq=1 Ack=385 Win=64240 Len=0
6 0.117979410 45.79.89.123 172.16.160.129 HTTP 458 HTTP/1.1 200 OK (text/html)
7 0.117997032 172.16.160.129 45.79.89.123 TCP 54 40416 → 80 [ACK] Seq=385 Ack=405 Win=63836 Len=0
8 3.305220777 172.16.160.129 45.79.89.123 HTTP 499 GET /basicauth/amateurs.txt HTTP/1.1
9 3.305475822 45.79.89.123 172.16.160.129 TCP 60 80 → 40416 [ACK] Seq=405 Ack=830 Win=64240 Len=0
10 3.351617662 45.79.89.123 172.16.160.129 HTTP 375 HTTP/1.1 200 OK (text/plain)
11 3.351630690 172.16.160.129 45.79.89.123 TCP 54 40416 → 80 [ACK] Seq=830 Ack=726 Win=63836 Len=0

TCP payload (445 bytes)
Hypertext Transfer Protocol
  GET /basicauth/amateurs.txt 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
  Authorization: Basic Y3MzMzg6cGFzc3dvcmQ=\r\n
    Credentials: cs338:password
  Connection: keep-alive\r\n
  Referer: http://cs338.jeffondich.com/basicauth/\r\n
```

- “No one enters dropdown menus. They’ll never find it here”  
~the developers probably
- This seems distinctly BAD
- We just sniffed someone who had accessed the website and now we have their identity
  - [“Identity theft is not a joke Jim”](#)
  - This seems like a pretty major problem. By knowing the credentials we can now just go back in as this person, and look at all the information and no one is going to know (we’re gonna give them a name, this is Dwight). Unless Dwight has a particularly strong alibi for this exact moment where Jim hacked his account, now we just have a huge security breach that no one knows about. Also Jim signed up perfectly legally with Dwight’s credentials, so you can’t really trace this back to him at all.

**Even better, we now have the whole file without even signing in as Dwight**

```

14 11.711433072 45.79.89.123 172.16.160.129 HTTP 462 HTTP/1.1 200 OK (text/plain)
15 11.711451039 172.16.160.129 45.79.89.123 TCP 54 40416 → 80 [ACK] Seq=1279 Ack=1134 Win=63836 Len=0
16 21.854715709 172.16.160.129 45.79.89.123 TCP 54 [TCP Keep-Alive] 40416 → 80 [ACK] Seq=1278 Ack=1134 Win=63836 Len=0
17 22.879339857 172.16.160.129 45.79.89.123 TCP 54 [TCP Keep-Alive] 40416 → 80 [ACK] Seq=1278 Ack=1134 Win=63836 Len=0
18 22.879703281 45.79.89.123 172.16.160.129 TCP 60 [TCP Keep-Alive ACK] 80 → 40416 [ACK] Seq=1134 Ack=1279 Win=64240 Len=0

[HTTP response 3/3]
[Time since request: 0.046237628 seconds]
[Prev request in frame: 8]
[Prev response in frame: 10]
[Request in frame: 12]
[Request URI: http://cs338.jeffondich.com/basicauth/armed-guard.txt]
File Data: 161 bytes
- Line-based text data: text/plain (4 lines)
  "The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards."
  \n
  -- Gene Spafford\n
  \n

```

- Jim must be *thrilled* he doesn't even have to commit identity theft
  - He literally just needs to sniff Dwight's computer and the website that we know that Dwight is using
  - We may not get all of the files, like if Dwight didn't open them all, but we do just have the information without even signing in as another person

## Out of curiosity we went back to see what fake passwords look like

```

13 22.245500704 172.16.160.129 45.79.89.123 HTTP 446 GET /basicauth/ HTTP/1.1
14 22.245784978 45.79.89.123 172.16.160.129 TCP 60 80 → 40424 [ACK] Seq=404 Ack=734 Win=64240 Len=0

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 YmFkIHVzZXJ1YmV1OnBhc3N3b3Jk\r\n
Credentials: bad username:password
\r\n
[Full request URI: http://cs338.jeffondich.com/basicauth/]
[HTTP request 2/6]

```

- And yeah we get the fake passwords too, it's all stored in here

```

24 44.053018442 172.16.160.129 45.79.89.123 HTTP 438 GET /basicauth/ HTTP/1.1
25 44.053306798 45.79.89.123 172.16.160.129 TCP 60 80 → 40424 [ACK] Seq=1210 Ack=1510 Win=64240 Len=0
26 44.099832108 45.79.89.123 172.16.160.129 HTTP 458 HTTP/1.1 200 OK (text/html)
27 44.099849593 172.16.160.129 45.79.89.123 TCP 54 40424 → 80 [ACK] Seq=1510 Ack=1614 Win=63837 Len=0

Connection: keep-alive\r\n
Content-Encoding: gzip\r\n
\r\n
[HTTP response 4/6]
[Time since request: 0.046813666 seconds]
[Prev request in frame: 20]
[Prev response in frame: 22]
[Request in frame: 24]
[Next request in frame: 28]
[Next response in frame: 30]
[Request URI: http://cs338.jeffondich.com/basicauth/]

```

- Also even if there's a lot of fake passwords, once we get in Wireshark identifies where the correct credentials came from, and here we entered the correct username and

password at frame 24 and when we get in (get the message 200 OK, instead of a 400 Unauthorized) it references back to Frame 24, where we entered the correct password

### **Mastermind plan**

1. Break target's computer
2. Install wireshark on loaner computer and hide the icon
3. Get target to sign in to the website and use their credentials
4. If they happen to use the file you want, great, we have everything from picking up the HTML
5. Otherwise sign in at some other point as them, with their credentials, like during lunch break or something so it isn't outside of business hours
6. Congratulations you have now seen Dwight's top secret diary (that he keeps in plain text files, idk ask him about that) or his bank account, your priorities are your own