

# The Sequence of Events of HTTP's Basic Authentication

**Introduction:** I'm using Kali Linux. There's an app in Kali Linux called Burp Suite which freezes per frame to see the HTTP requests and responses, which makes it easier to visualize how these things work.

## The Story:

When I first typed in <http://cs338.jeffondich.com/basicauth> onto Burp Suite's web, my Kali machine's browser requested it using the GET HTTP/1.1 protocol. The host server (Jeff's website) responded with a message saying 301 Moved Permanently – which basically can mean that the original page URL was changed. If you see the URL title (header's row, fourth column) in the table of the image below, the URL changed from /basicauth to /basicauth/. There's an additional backslash redirect that happened. In simple terms, it's just a good samaritan who tells me that the office is here and not there, even though both office doors look so identical.

The screenshot displays the Burp Suite Community Edition v2024.6.3 interface. The top menu bar includes options like Burp, Project, Intruder, Repeater, View, and Help. Below the menu, there's a toolbar with various tools such as Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, and Settings. The main window is divided into several panes. The 'HTTP history' pane shows a list of requests and responses. The 'Request' pane shows the details of the selected request, and the 'Response' pane shows the details of the selected response. The 'Inspector' pane on the right shows the request and response headers and body.

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes
1	http://cs338.jeffondich.com	GET	/basicauth			301	400	HTML		301 Moved Permanently	
2	http://cs338.jeffondich.com	GET	/basicauth/			401	805	HTML		401 Authorization Req...	
3	http://cs338.jeffondich.com	GET	/basicauth/			200	666	HTML		Index of /basicauth/	
4	http://cs338.jeffondich.com	GET	/favicon.ico			404	728	HTML	ico	404 Not Found	
5	http://cs338.jeffondich.com	GET	/basicauth/								

**Request Details:**

```
1 GET /basicauth HTTP/1.1
2 Host: cs338.jeffondich.com
3 Accept-Language: en-US
4 Upgrade-Insecure-Requests: 1
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
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
7 Accept-Encoding: gzip, deflate, br
8 Connection: keep-alive
```

**Response Details:**

```
1 HTTP/1.1 301 Moved Permanently
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Tue, 24 Sep 2024 23:55:36 GMT
4 Content-Type: text/html
5 Content-Length: 178
6 Location: http://cs338.jeffondich.com/basicauth/
7 Connection: keep-alive
8
9 <html>
10 <head>
11 <title>
12 301 Moved Permanently
13 </title>
14 </head>
15 <body>
16 <center>
```

**Inspector Details:**

Request attributes: 2

Request headers: 7

Response headers: 6

But then security came along and stopped my request from entering `/basicauth/`.

The screenshot displays the Burp Suite Community Edition v2024.6.3 interface. The top menu bar includes Burp, Project, Intruder, Repeater, View, and Help. Below it, a toolbar shows various tools like Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, and Settings. The main workspace is divided into several panes. The 'HTTP history' pane shows a table of intercepted requests:

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes
1	http://cs338.jeffondich.com	GET	/basicauth			301	400	HTML		301 Moved Permanently	
2	http://cs338.jeffondich.com	GET	/basicauth/			401	805	HTML		401 Authorization Req...	
3	http://cs338.jeffondich.com	GET	/basicauth/			200	666	HTML		Index of /basicauth/	
4	http://cs338.jeffondich.com	GET	/favicon.ico			404	728	HTML	ico	404 Not Found	
5	http://cs338.jeffondich.com	GET	/basicauth/								

The 'Request' pane shows the details of the selected request (index 2):

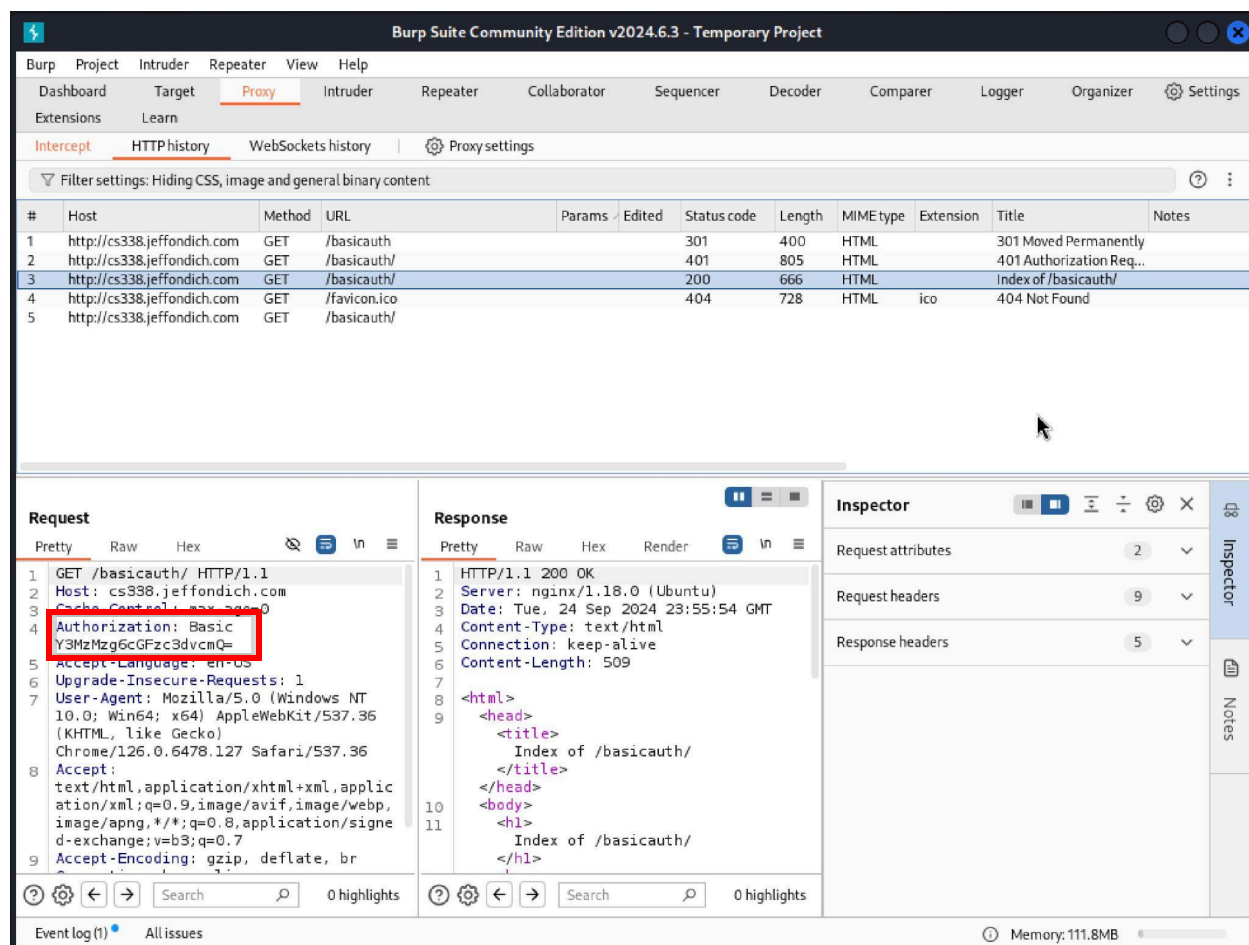
```
1 GET /basicauth/ HTTP/1.1
2 Host: cs338.jeffondich.com
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=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

The 'Response' pane shows the details of the selected response (index 2):

```
1 HTTP/1.1 401 Unauthorized
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Tue, 24 Sep 2024 23:55:41 GMT
4 Content-Type: text/html
5 Content-Length: 590
6 Connection: keep-alive
7 WWW-Authenticate: Basic realm="Protected Area"
8
9 <html>
10 <head>
11 <title>
12 401 Authorization Required
13 </title>
14 </head>
15 <body>
16 <center>
```

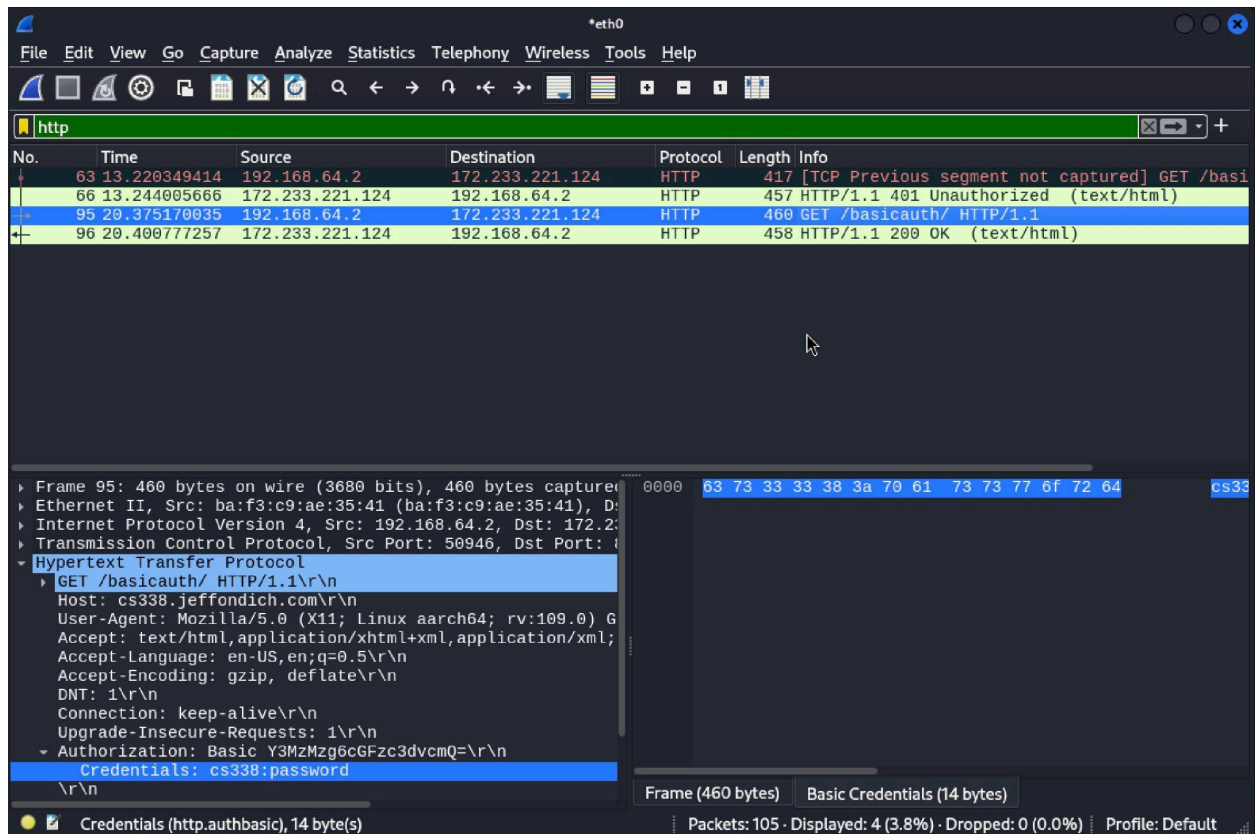
The 'Inspector' pane on the right shows the 'Request attributes' (2), 'Request headers' (7), and 'Response headers' (6) sections. The 'Response headers' section is expanded, showing the 'WWW-Authenticate: Basic realm="Protected Area"' header highlighted with a red box.

The security responded with 401 Unauthorized from my original request, and they also have an additional WWW-Authenticate header, basically asking my ID (in this case username and password) if I'm the right person to enter.



So I gave my ID by typing in the username `cs338` and password `password`, and with that information my browser sends another GET HTTP request to the server with an additional `Authorization` header containing the base64 encoded username and password, which is in the red square box on the image above. The security was happy and said 200 OK, meaning that I'm authorized to enter.

But the problem here is that maybe the good samaritan was eavesdropping when I said the username and password.



And if the good samaritan actually did eavesdrop, he basically knows my ID, both username and password by using WireShark. This is because the website was only using HTTP and not HTTPS.

MORAL OF THE STORY: Always be cautious when you do interactions that require usernames and passwords, because there may be a security breach when you input something in a website that is not secure.

## So What is Authorization Header?

In the words of Mozilla, “The HTTP Authorization request header can be used to provide credentials that authenticate a user agent with a server, allowing access to a protected resource.” In the case of HTTP Basic Authentication, the header has the base64 encryption for the username and password. If you see from the WireShark screenshot, it is formatted with `Authorization: Basic` and the base64 encryption, and the unencrypted username and password underneath is separated by a colon.

What this basically means in terms of the story is that I give the security guy my ID, he authenticates me, and if I’m on the list of authorized users, I’ll be let into the office.

Jeremy Gautama

Sources:

- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/301>
- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Authorization>