

Osareme Davis

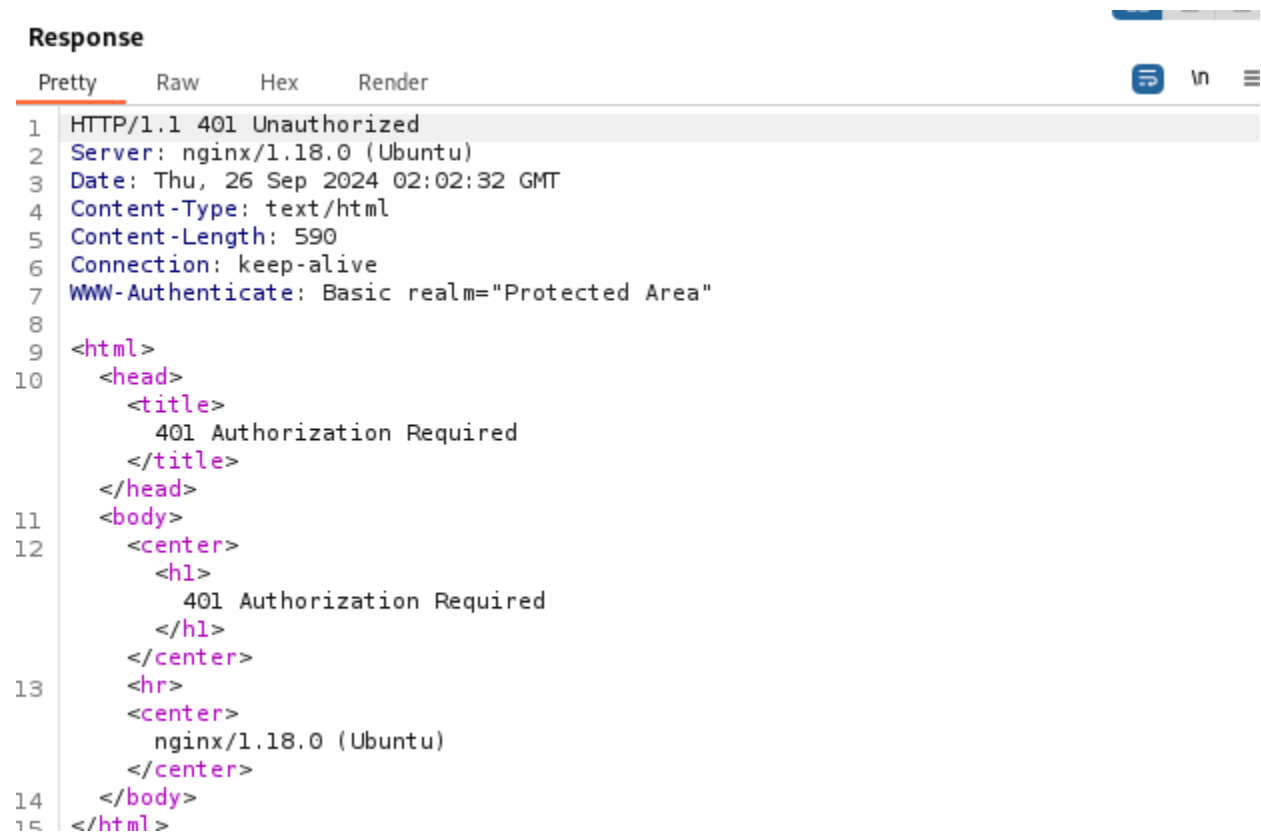
25 September 2024

CS 338 Computer Security

Professor Jeff Ondich

The Process of Basic Authentication over a HTTP Connection

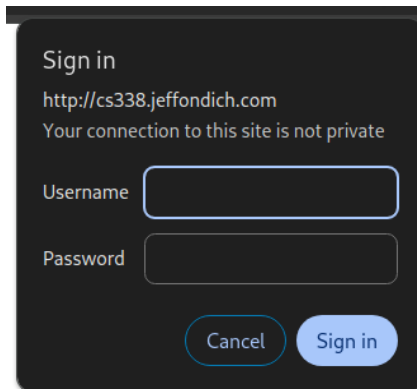
After the completion of the TCP handshake, the browser sends its usual GET request to retrieve the webpage. Unlike a regular HTTP request, however, the server sends a different response code: 401 Unauthorized.



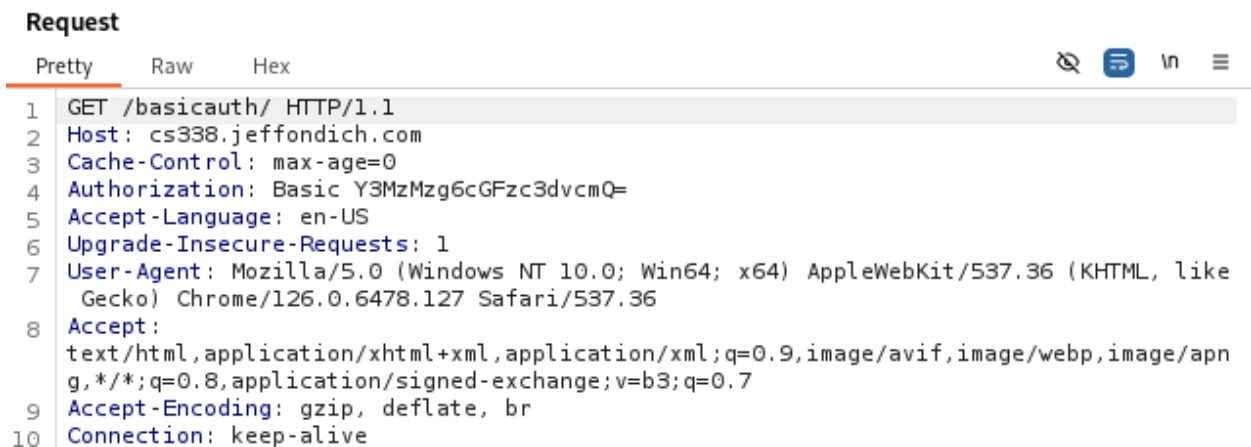
The screenshot shows a web browser's developer console with the 'Response' tab selected. The response is an HTTP 401 Unauthorized status. The headers include 'Server: nginx/1.18.0 (Ubuntu)', 'Date: Thu, 26 Sep 2024 02:02:32 GMT', 'Content-Type: text/html', 'Content-Length: 590', 'Connection: keep-alive', and 'WWW-Authenticate: Basic realm="Protected Area"'. The body of the response is an HTML document with a title '401 Authorization Required' and a message '401 Authorization Required' displayed in the center. The browser's user interface shows a blue address bar and a search icon.

```
Response
Pretty Raw Hex Render
1 HTTP/1.1 401 Unauthorized
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Thu, 26 Sep 2024 02:02:32 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>
17 <h1>
18 401 Authorization Required
19 </h1>
20 </center>
21 <hr>
22 <center>
23 nginx/1.18.0 (Ubuntu)
24 </center>
25 </body>
26 </html>
```

The server, having HTTP Basic Authentication set up for this particular resource, sends this response to prompt the client to send the request with the correct credentials. The browser then prompts the user for said credentials with a unique dialog:

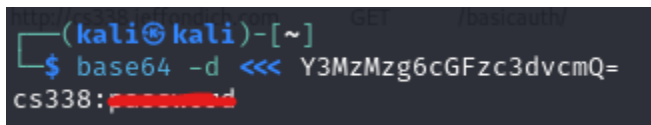


Once the credentials are entered, the browser sends another GET request, this time including a special header “Authorization”:



This header identifies the authentication protocol and includes a base64 encoded string.

Said string is actually the credentials that were entered, sent as a username:password pair:



(password for this example redacted for privacy)

The server now checks this request against a “password file” which contains all of the username:password pairs which it should allow

(source:

<https://docs.nginx.com/nginx/admin-guide/security-controls/configuring-http-basic-authentication/>).

Should the username:password pair not exist in the password file, another 401 response is sent. With the correct credentials, though, the server responds with the usual 200 OK response and serves the web content.