Ruben Boero and Vanessa Heynes

What queries are sent from the browser, and what responses does it receive?

Browser sends a query to the server, asking for the /basicauth/ page:

```
37    0.232054426    172.16.15.128    45.79.89.123    HTTP    417
GET /basicauth/ HTTP/1.1
```

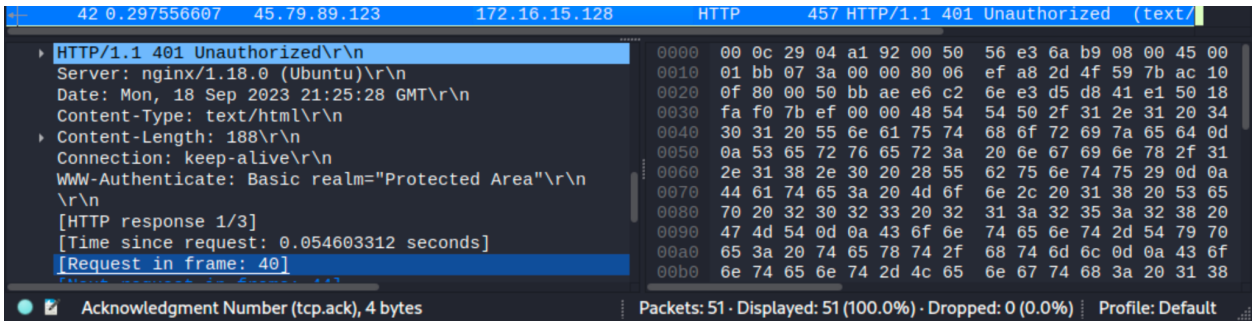
In response, server acknowledges the client's request:

```
41    0.243207591    45.79.89.123    172.16.15.128    TCP    60    80
→ 48046 [ACK] Seq=1 Ack=364 Win=64240 Len=0
```

But refuses to send the page without authorizing the user first:

```
42    0.297556607    45.79.89.123    172.16.15.128    HTTP    457
HTTP/1.1 401 Unauthorized  (text/html)
```

The 401 status code is how nginx communicates to the client that they have been denied authorization.



Here, the server is saying that the requested page is protected by Basic HTTP security

After the password is typed by the user, what sequence of queries and responses do you see?

The password is sent by the client to the server:

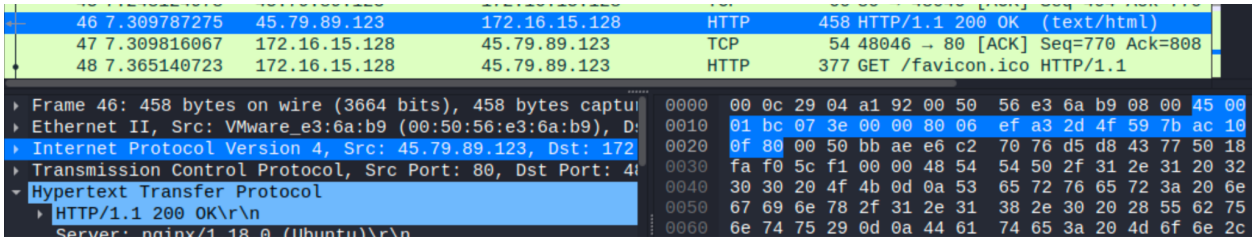
```
44    7.247785473    172.16.15.128    45.79.89.123    HTTP    460
GET /basicauth/ HTTP/1.1
```

The server acknowledges that the password has been received:

```
45    7.248124978    45.79.89.123    172.16.15.128    TCP    60    80
→ 48046 [ACK] Seq=404 Ack=770 Win=64240 Len=0
```

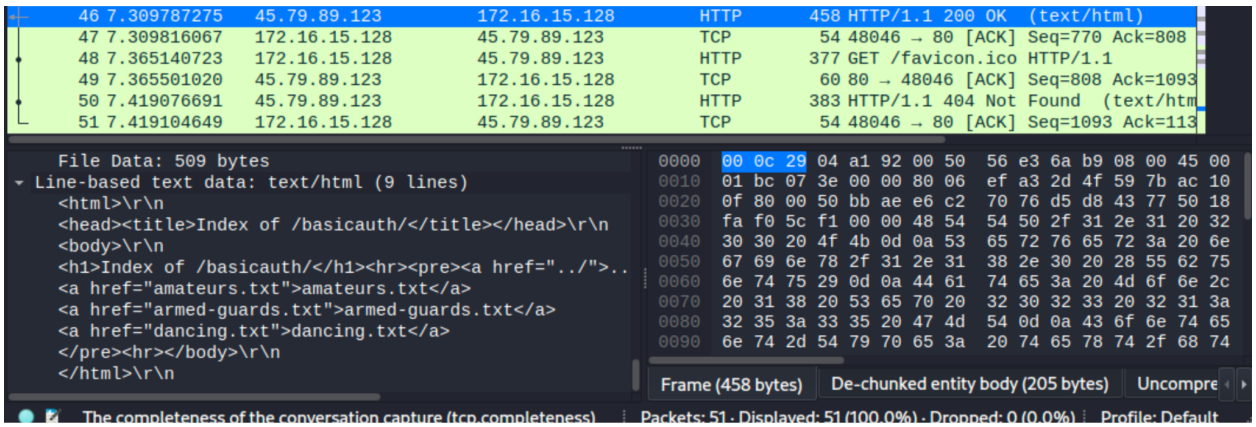
And that the password is correct:

```
46    7.309787275    45.79.89.123    172.16.15.128    HTTP    458
HTTP/1.1 200 OK  (text/html)
```



Notice that the header contains “OK”, rather than “Unauthorized” like in packet 42 shown above.

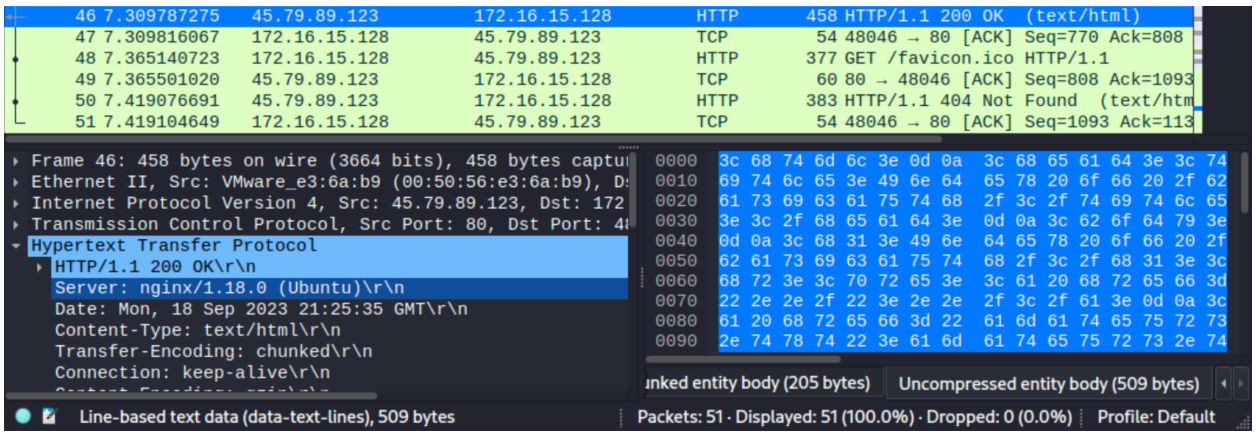
The server sends the previously requested /basicauth/ HTML to the client:



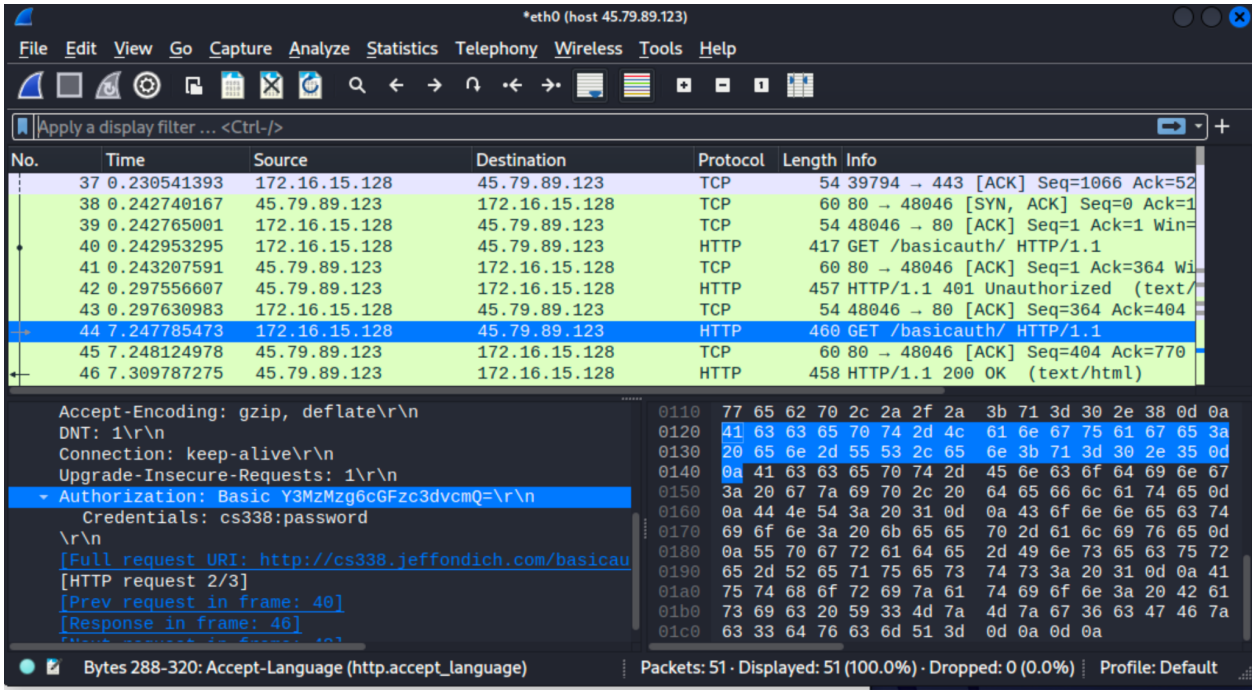
Finally, the server then sends the favicon of the page.

Is the password sent by the browser to the server, or does the browser somehow do the password checking itself?

The server—running nginx—is doing the checking:



We came to this conclusion because the OK message comes from the server, not from the client.



Screenshot shows the client sending the password to the server.

The password is sent by the client to the server in this packet:
`44 7.247785473 172.16.15.128 45.79.89.123 HTTP 460 GET /basicauth/ HTTP/1.1`

Additionally, we found some nginx documentation/examples that makes it seem like the checking is happening within the server.

If the former, is the password sent in clear text or is it encrypted or something else?

The password is sent in clear text encoded with base64. The username and password are separated by a colon character:

```
Authorization: Basic Y3MzMzg6cGFzc3dvcmQ=\r\n
      Credentials: cs338:password
```

If it's encrypted, where did the encryption key come from?

It's not encrypted.

How does what you observe via Wireshark connect to the relevant sections of the HTTP and HTTP Basic Authentication specification documents?

What we observed was directly connected to the section of the HTTP information under the 2.0 header.

- The username and password are sent unencrypted in plain text using base64 encoding.
- The username and password are sent as 1 string where the username and password are separated by a comma character.
- The HTTP headers were formatted in the same way as the documents describe.

Generally, the HTTP messages sent back and forth between the server and the client match the HTTP documentation.

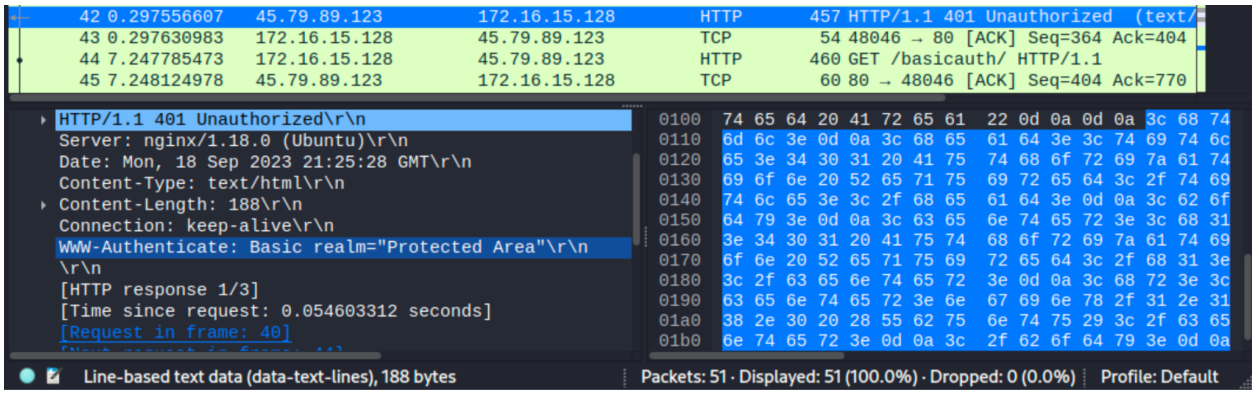
How is the password verified? By whom?
The password is verified by nginx from the server side.

Sources:

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

Where is the authorized header?

The authorization header is found in packet 42:



Here we can see that the server sends an HTTP response to the client. In this packet the authorized header can be found. The information in this packet tells the client that they are not yet an authorized user (status 401), as well as providing the client with information about the type of authentication that is being used (Basic)

Source:

- https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Authorization