Name: Bryan Yang

HTTP's Basic Authentication: A Story

What is happening when you sign in to http://cs338.jeffondich.com/basicauth/ ?

Your browser and a nginx server are communicating with each other. As the client, you want to go to this website. But the server will only give you the website if you have the right credentials.

Before even knowing that a username and password are needed, the first step is that the browser requests the page from the server: “GET /basicauth/ HTTP/1.1\r\n …”

This looks pretty similar as requesting from any other website that doesn’t require a password.

- But the response from the server is not the typical “200 OK”. The server responds with a “401 Unauthorized”.
- Furthermore, there is an interesting HTTP header:

  WWW-Authenticate: Basic realm="Protected Area"\r\n

  This is an HTTP response header that defines the specific methods to let people view the website. Importantly, it shows the encryption scheme for sending passwords to the server. You can read more about it here: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/WWW-Authenticate

So, overall, it seems like the authorization is working; the server is not giving the website to anyone that asks. In fact, let's look at the HTML in the server’s response and see what it contains:

The HTML data in the response looks like:

```
Unset
<html>
	<head><title>401 Authorization Required</title></head>
	<body>
		<center><h1>401 Authorization Required</h1></center>
		<hr><center>nginx/1.18.0 (Ubuntu)</center>
	</body>
</html>
```

So the content of the site behind the authentication is not visible in the server's initial response, all we see is "401 Authorization Required".

After receiving the 401 response, the client shows an authentication window, asking for credentials. After typing in the password and clicking submit, the client sends another GET request, almost exactly like the initial request we did earlier. However there are some differences between this request and the request before signing in. Most notably, after typing the credentials, the HTTP request contains an "*Authorization*" header. For me, the value was "Basic Y3MzMzg6cGFzc3dvcmQ=".

- According to Mozilla web docs, this Authorization header "indicates what authentication schemes can be used to access the resource". So for us, the authentication scheme is "Basic Y3MzMzg6cGFzc3dvcmQ=". For more info, check out https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Authorization

Let’s see what happens when we enter the wrong credentials:

For example, instead of the correct password, I’ll try some incorrect passwords

PW = 123:

- Authorization: Basic Y3MzMzg6MTIz\r\n

PW: bruh

- Authorization: Basic Y3MzMzg6YnJ1aA==\r\n

So when the client re-requests for the web page, the password that is typed in changes the authorization header. The password is sent from the browser to the server. However, as we can see above, it is encrypted in some way. The encryption is done via the scheme that we saw in the WWW-Authenticate header earlier.

In the HTTP Basic Authentication documentation, the security, or lack thereof of Basic authentication is highlighted.

- “The most serious flaw of Basic authentication is that it results in the cleartext transmission of the user's password over the physical network.”

We can see how the password is being transmitted from the user to the server. This means that if I were to go to this site and enter my password, someone else on the network could see that information.