Tuesday, 2024-09-24

Tux Adventure

The cold air pierces through a nervous figure, pacing between a sullen tree and a red phone booth. 11:58. Not yet, the figure thinks. Somewhere in the deserted streets, faint meowing cries for the elusive comfort of home. Nyan should be okay, the figure reassures himself. The figure, Tux, is not like your typical penguin. He loves to swim but in the sea of ones and zeros, not of water molecules. He eats fish but only those with questionable links and information, not with protein and omega-3 fatty acids. Tux's cat, Nyan cat, a magical creature who has Poptart as a body and farts rainbow, has gone missing during Tux's business trip. The clock tower booms in the distance, indicating the start of tomorrow. Finally, thinks Tux. His body begins to shimmer in ones and zeroes, and soon, Tux is a stream of ones and zeroes floating through the wifi modems and routers. His friend Ferris the crab has told Tux about the last sightings of his beloved cat. According to Ferris, the faint trails of rainbow lead to an obscure website called "http://cs338.jeffondich.com/basicauth/."

Nyan, a mischievous soul farting rainbow everywhere, has often escaped the .gif cage and flown off into various corners of the internet, so much so that it is difficult to find people who do not know about Nyan cat. Almost there, thinks Tux as he hurriedly spawns a Firefox to talk to the Nginx helpdesk for the TCP handshake. The helpdesk, delighted by the recent spike in SYN requests, eagerly acknowledges the request and returns the handshake, opening a TCP connection. Now, for those who do not know, in the internet world, if you are browsing websites, communications are done via HTTP. Tux asks Firefox for the website, which then sends a GET request to the Nginx helpdesk:

```
GET /basicauth/ HTTP/1.1
Host: cs338.jeffondich.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

Tux would never have known that the website is password protected, so he did not include the Authorization information. In return, Nginx helpdesk first acknowledges his request and responds with 401:

```
HTTP/1.1 401 Unauthorized
Server: nginx/1.18.0 (Ubuntu)
Date: Wed, 25 Sep 2024 01:19:55 GMT
Content-Type: text/html
Content-Length: 590
Connection: keep-alive
WWW-Authenticate: Basic realm="Protected Area"
```

Nginx helpdesk alerts Firefox about the protection via WWW-Authentication header, telling the browser that the authentication type is Basic and that the browser is about to access the Protected Area. Accepting the challenge, Firefox prompts Tux for a username and password. When one spends years on the internet, one builds an intuition for usernames. Passwords, on the other hand, are a little tricky to get right. So Tux guesses 'cs338' and 'password.' Then, Firefox concatenates the username and password and encodes it in Base64 before it sends another GET request but with the Authorization header that now has the newly encrypted id-password string:

```
GET /basicauth/ HTTP/1.1
Host: cs338.jeffondich.com
Cache-Control: max-age=0
Authorization: Basic Y3MzMzg6cGFzc3dvcmQ=
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.dd0
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

Now this is very unsafe, for anyone with some knowledge about Wireshark or BurpSuite can capture and decode the password using Base64. The Nginx helpdesk receives the new request and compares the id-password string with the real id and password and, suprisingly, they matched. The Nginx helpdesk wonders how everyone who visits the website is able to access the top secret realm. Maybe they should not use password 'password,' but this is not as important as the whereabouts of Nyan cat. The Nginx helpdesk returns a response:

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Wed, 25 Sep 2024 01:20:03 GMT
Content-Type: text/html
Connection: keep-alive
Content-Length: 509
```

Now, able to see the website, Tux looks for Nyan cat. But Nyan is nowhere to be seen. Thinking Nyan maybe playing in the favicon, Tux sends yet another GET request through Firefox:

```
GET /favicon.ico HTTP/1.1
Host: cs338.jeffondich.com
Accept-Language: en-US,en;q=0.9
User-Agent: Mozilla/5.0
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Referer: http://cs338.jeffondich.com/basicauth/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

Upon receiving the request, the Nginx helpdesk tries to looks for the favicon, but, oh no, it is no where to be seen. The Nginx helpdesk responses with 404:

```
HTTP/1.1 404 Not Found
Server: nginx/1.18.0 (Ubuntu)
Date: Wed, 25 Sep 2024 01:20:03 GMT
Content-Type: text/html
Connection: keep-alive
Content-Length: 564
```

When nothing shows up on favicon, Tux falls to his tiny knees, crying. Then, a familiar jingle begins to ring in his ears. Recognizing the song, Tux looks around and there it is. That mischievous Poptart cat, springing out of Tux's business bag as if nothing has happend.

Main Takeaways

The most crucial takeaway of this lab is how Basic authentication by itself is not a safe way to authenticate users over the internet, for anyone with a computer can decode Base64 with appropriate tools. TLS [ACK] occurs after any kind of data transfer to let the sender know that the data was received. It was interesting to observe how the client only sends Authorization Header upon receiving WWW-Authentication header from the server. The base assumption that websites won't have password protection is not only efficient but also clever. From the client side, it always looked like it was the client who "knew" whether the website was password protected or not. In reality, it sends a GET request without any Authorization header, receives 401, then resents a GET request with the Authorization header. The Authorization header consists of type of authentication method ("Basic") in our case and authorization parameters, including but not limited to, id-password string, etc.

Relevant Links

https://developer.mozilla.org/en-US/docs/Web/HTTP/Authentication (https://developer.mozilla.org/en-US/docs/Web/HTTP/Authentication) https://datatracker.ietf.org/doc/html/rfc7617 (https://datatracker.ietf.org/doc/html/rfc7617)