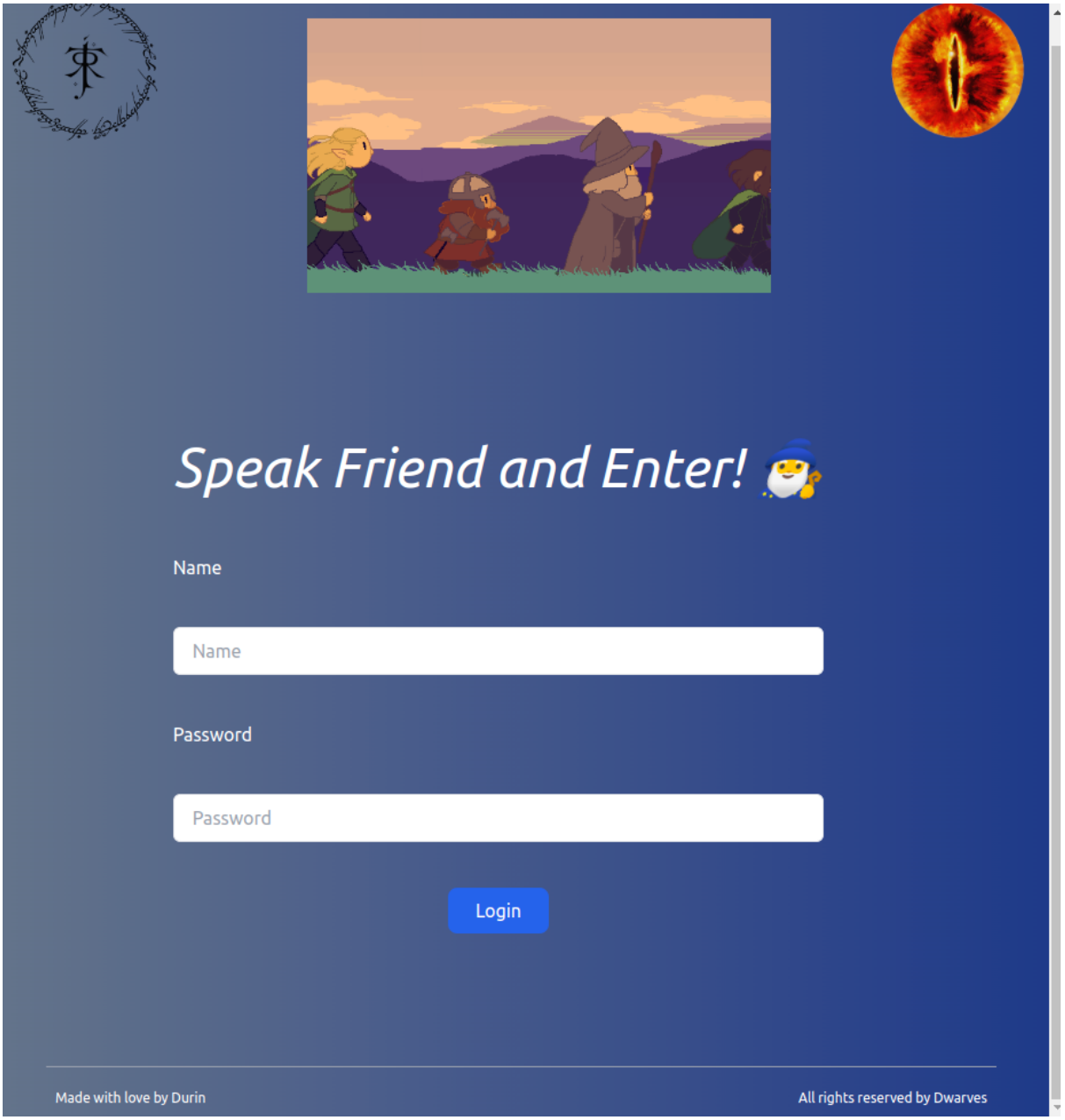
Description

The current challenge tries to address security problems of the Key ID parameter in JWT. Specifically, KID (Key ID) is an optional header on JWT which allows the developers to specify the key (and its directory) in order to verify the token. That being said, a malicious user is able to modify that parameter which can lead to vulnerabilities like Directory transversal, SQL injection and even OS injection. More about the KID issues can be found on hack tricks and on medium.

Walkthrough

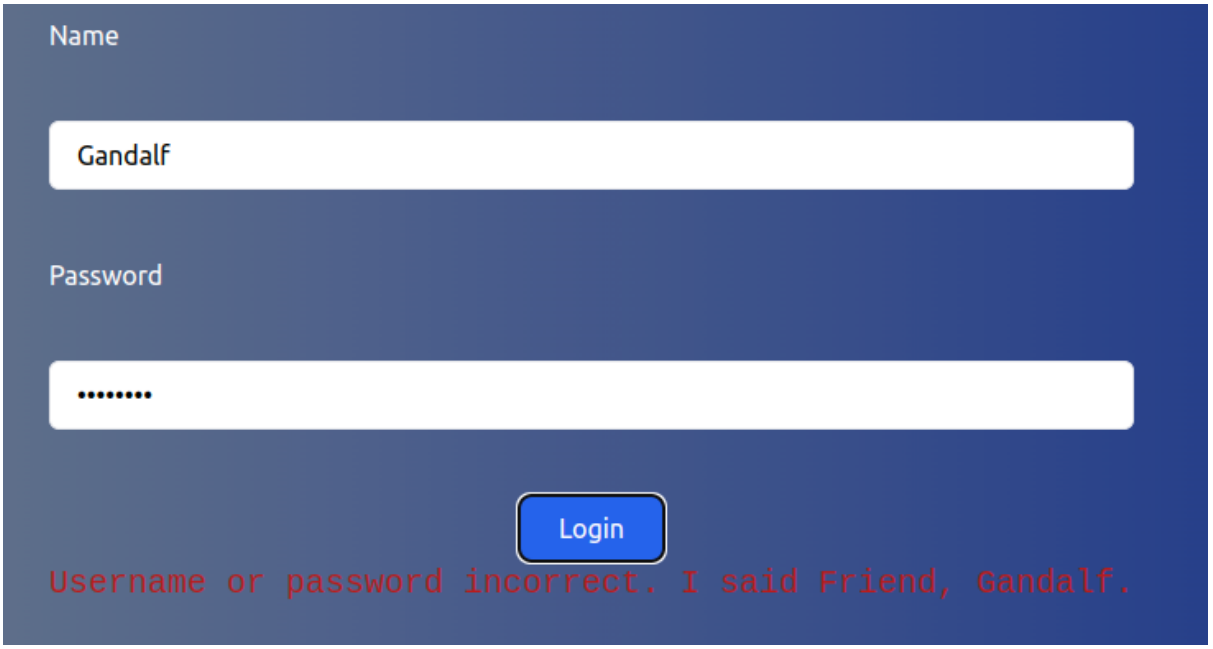
First of all by opening the challenge page we have a login form themed by Tolkien's Lord of the Rings.



In order to bypass the login mechanism the user should use the credentials
Username: **Gandalf**
Password: **Mellon**

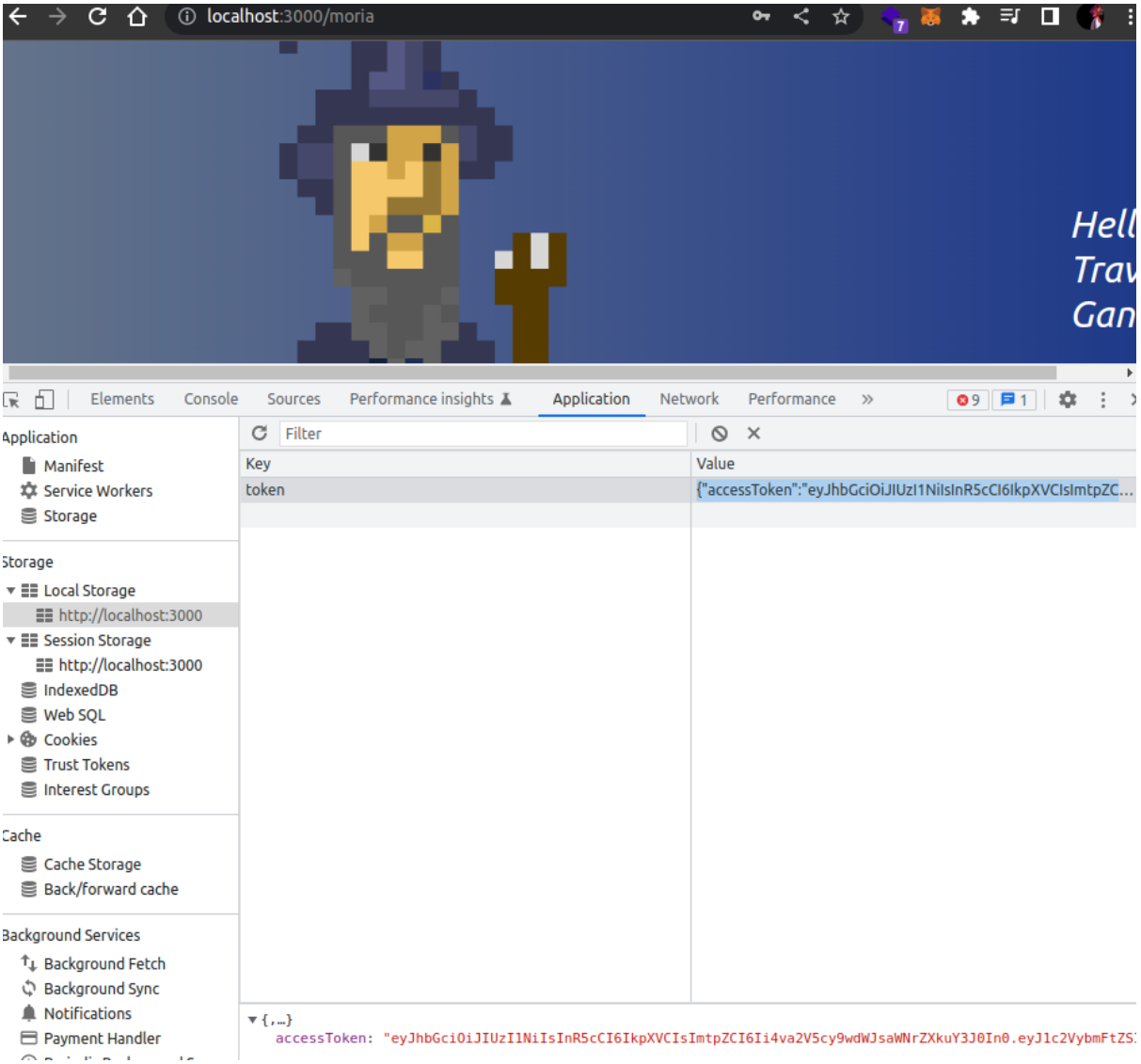
The credentials can be found by simply googling the famous phrase of Gandalf in the Gates of Moria the "Speak friend and enter". The answer is **Mellon** which means friend in Sindarin.

In the case of false credentials a message shows up which indicates that the credentials are wrong.

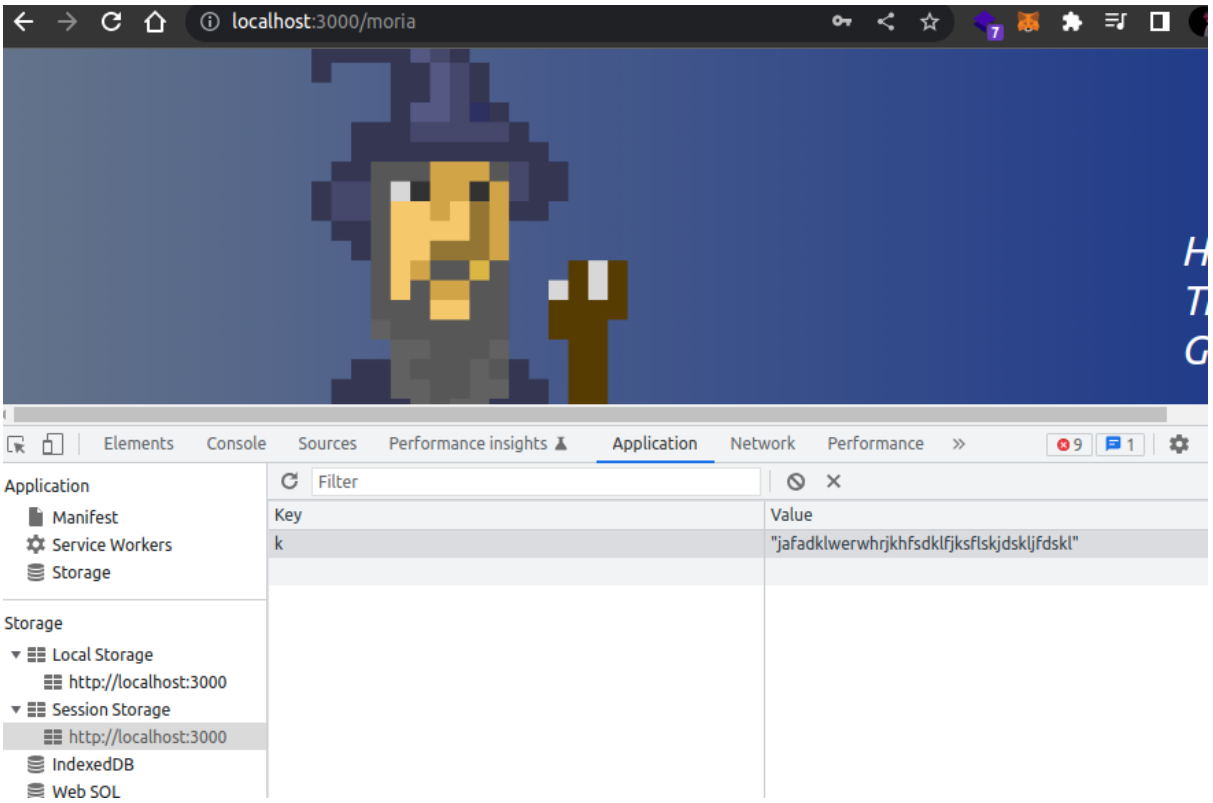


Next after the successful connection the user is prompted in the directory /moria.

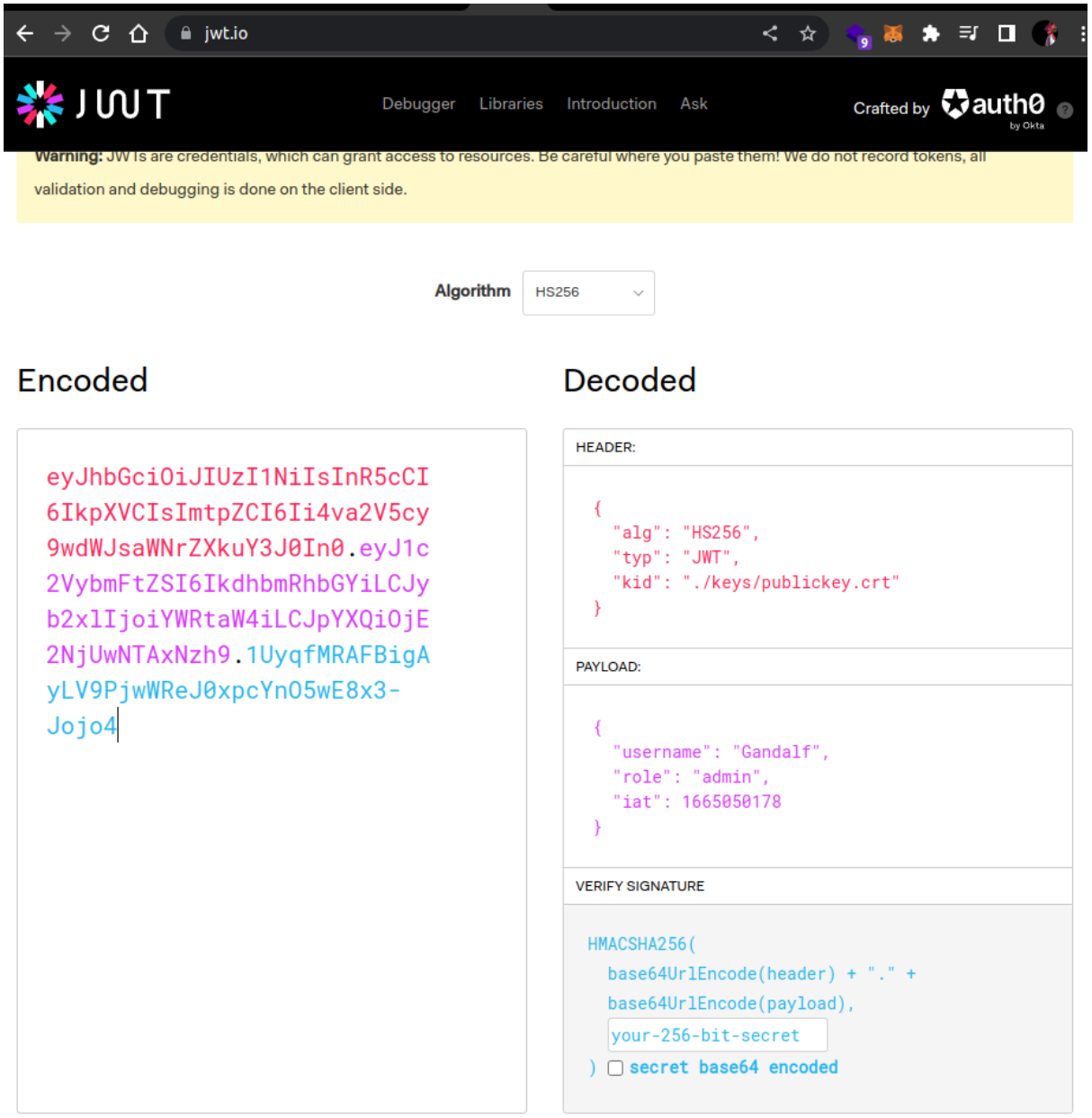
Here the user needs to check Local and Session storage of the Browser. On the **Local Storage** we have the JWT accessToken of the user Gandalf .



On the **Session Storage** we have the public key that is used to validate the user.



Then by using jwt.io we see the values of the Json Web token:



Since there exists the kid parameter we try to read files from the application and the results will be shown in the **Session Storage** “k” parameter. Based on that we modify the kid parameter to “kid”: “./keys/flag.txt”

Encoded


```
eyJhbGciOiJIUzI1NiIsInR5cCI
6IkpXVCIsImtpZCI6Ii4va2V5cy
9mbGFnLnR4dCJ9.eyJ1c2VybmFt
ZSI6IkdhbmRhbGYiLCJyb2xlIjo
iYWRtaW4iLCJpYXQiOjE2NjUwNT
AxNzh9.jhnxuQIFsoeVlHvunpw2
6FhHsaxwHU2Z6aKcE5Edz94
```


Decoded

HEADER:

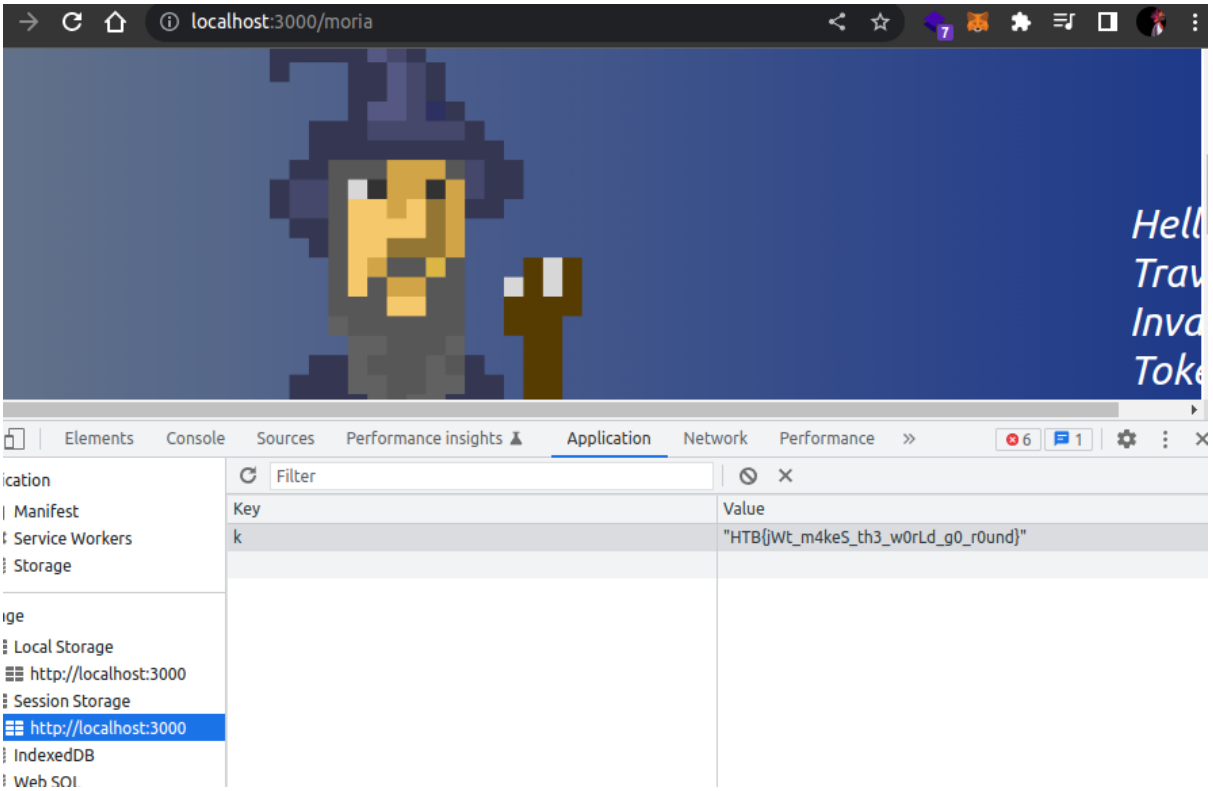

```
  "typ": "JWT",
  "kid": "./keys/flag.txt"
}
```


PAYLOAD:


```
{
  "username": "Gandalf",
  "role": "admin",
  "iat": 1665050178
}
```


VERIFY SIGNATURE

Then we use the generated token in the application, refresh the page and we see that we can read the flag.txt file in **Session Storage**.



Note: If we delete both of the **Local** and **Session Storage** tokens the app redirects us to the login page.

Misc

1. Regarding the challenge structure I tried to be as close as possible with past challenges that I played in the hack the box platform. That being said, I use a Dockerfile with a supervisord and a **run.sh** file to deploy the application.
2. A **docker_build.sh** file exists to build the image and deploy the docker container.
3. The front-end was built on React(with tailwind css) and the back-end is a very basic express server.
4. For the front-end I use the build version but you can also find the source code in the lotr folder.

Author: Mike Anastasiadis
Linkedin: https://www.linkedin.com/in/mike-anastasiadis
Twitter: @Thr3atRipper