**Challenge Description**

Our agents discovered COViD's admin panel! They also stole the credentials minion:banana, but it seems that the user isn't allowed in. Can you find another way?

**Solution**

Firstly, we are given an admin panel link, when you visit the website you are greeted by:

Graphical user interface

Description automatically generated

Inputting the username:password into the fields and pressing sign in gives us the following error.

Graphical user interface

Description automatically generated

Which meant that the credentials were correct, but the user was not an admin thus he couldn’t enter. I then proceeded to do a manual explore on OWASP ZAP and then tried to login with the same credentials.



The POST request sent to /login contained the username and password we entered and the response was an access token:

*{"accessToken":"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Im1pbmlvbiIsInJvbGUiOiJ1c2VyIiwiaWF0IjoxNjA3MjczNzgwfQ.by3\_9WqVZ3ChEgP9TN5xGfKCmT98lrKpSt0-aUfLbc-iWm7PYGk05oy1hndLMJjDDqxuQ3IzB-VD01Mb3YEajRIB\_G7ZxFlXeZoE\_91dXZvXaVum7uDLW9U-G710ec9QDKoZObosEvsAOvGkQGpLcClgKh-7IAp3yMb8UWx6As05FKqmYvB7nS9out1kFCV5LJ4WVYU8x0POSkaQEcOPeu3ve6eCaF-jfMX32aJdP16v3NaGJmaCUCXxtN67sk7CszcpE9ladpZjELkxR40ltWg20fn1qwUj0fPXRLUoyZYOCa12tZgHcDyQTAETTPDABNoYn7hdi42PLyskb0b\_twmE4mo4Bu\_FZYJjfsqogO\_G3Rd2TQoPeRnO9xTcjIIyfumz2\_oT-4B0JVtUC4YOuQL2WIvvIHePsNLRsXpajct9TrQ48sj4f2oTKiGZlb62VCoG1JUKNd6S9WlyWwhBtpFszaFFvfCzieIDFA4lecGH7\_hZXKlgTzDcQtgcKDP27vDm4z98K50jI57G2s\_3UPnpcC07Ri\_0h4HhSpJ\_Z26gGH3TqyuBU5R4WpLhP3u0rfH1bEjVEDc2VLp3JlUJbV52yhw82dGPbFPhvd-6Fj5c5LlN\_-kdNgrE\_6SQWAQgRaWzFd1UndtMy3GRLf5n3MRB7O8m9Y94wPwk5bpgmQA"}*

This accessToken was added into an Authorization header in a GET request sent to /unlock

Graphical user interface, text, application

Description automatically generated

So, from the 3 “.” located in the accessToken I realized that this was a JSON Web Token (JWT) and it was being used to authenticate the user.

JWT tokens contain 3 parts:

* Header (which contains the algorithm type and token type)
* Payload (data)
* Verify Signature

I used <https://jwt.io/> and inputted the accessToken which gave me the decoded base64url information about the JWT

Graphical user interface, text, application

Description automatically generated

So, this challenge required me to craft a JSON Web Token. However, the token used RS256 which was an asymmetrical algorithm and thus a private key was used to sign the token and I couldn’t find the private key anywhere on the website’s code.

Luckily, searching around the internet I found out that there is a well-known vulnerability where you can trick the server into accepting a symmetrically signed token. Firstly, I needed the public key which I found in the HTML code of the website in a comment

Text

Description automatically generated

So, I