Token Machine

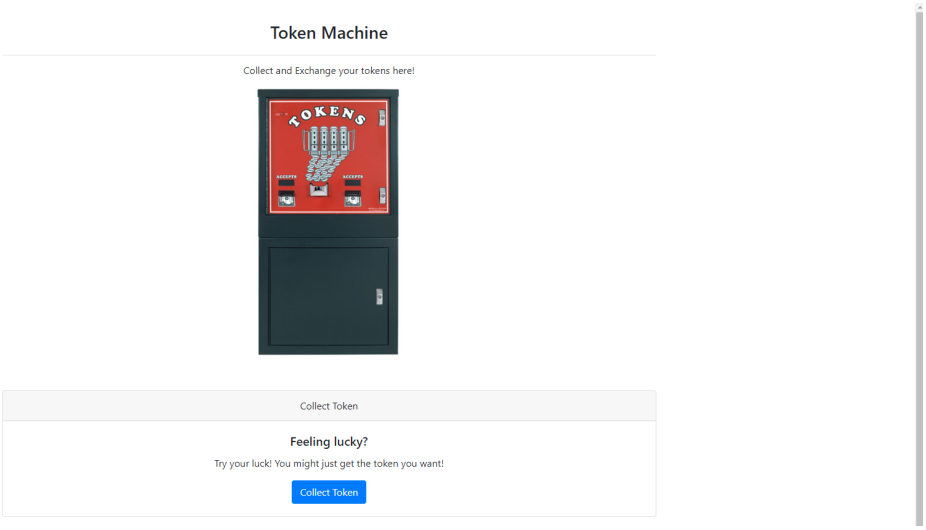
Problem Type

HS256 Token Vulnerability, Privilege Escalation, Authentication Bypass

---

Solution

This solution walkthrough does not involve Burp Suite, although it can be used as well.



Step 1 - Collect the JWT Token:

Obtain a valid JWT Token by clicking on the 'Collect Token' button on the home webpage. After selecting 'Collect Token', check the cookies to obtain the JWT token.

A sample token is:

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoidXNlciIsInRpbWUiOjE2MzczNDE5MDMuMzQ1ODY4fQ.vvUr
```

This token is a JSON Web Token - JSON Web Tokens consist of three parts(encrypted in base64) separated by dots (.), which are:

- Header
- Payload
- Signature

Header

```
  "typ": "JWT",
  "alg": "HS256"
}
```

Payload

```
{
  "role": "user",
  "time": 1637341903.345868
```

```
}
```

We want to change the value in the `role` field from `user` to `admin`. The current token, when exchanged, is of *no value*. An admin token would likely have *more value*. In the source code, we also find this small snippet of HTML commented:

```
<!-- <p> Admin tokens might be worth more than user tokens ... </p> -->
```

Signature The signature of the HS256 JWT is created with the following encryption algorithm:

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret)
```

Hence, without knowing the `secret`, we are unable to edit and provide an accurate signature.

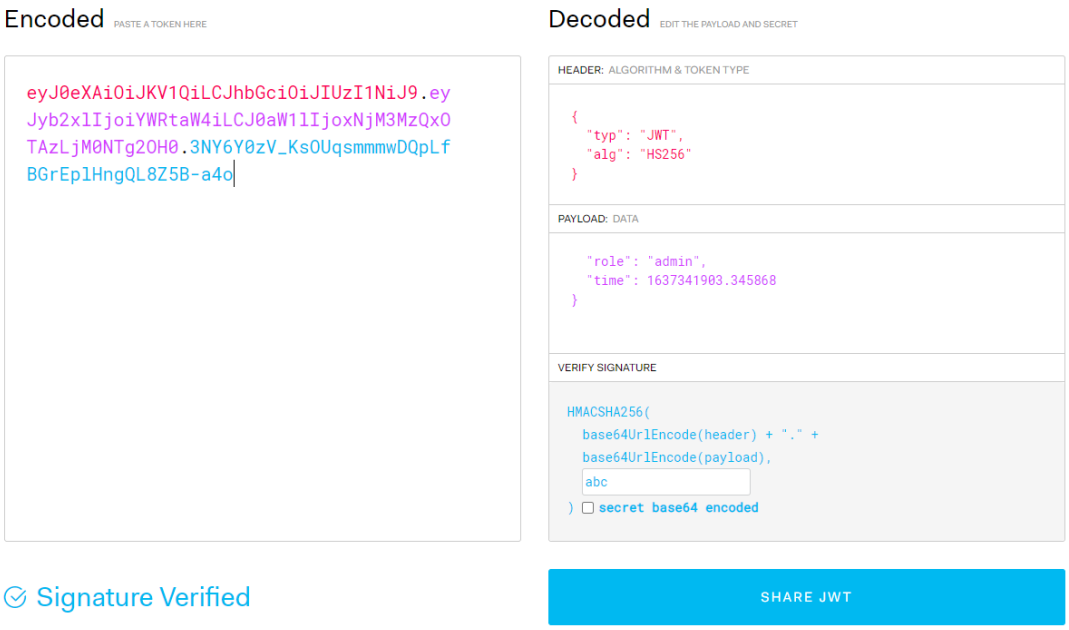
Step 2 - Cracking the JWT Token:

However, HS256 JWT Tokens are vulnerable to brute force attacks. An open source tool `c-jwt-cracker` (https://github.com/brendan-rius/c-jwt-cracker) can be used to brute force the secret, especially if the secret is poorly set.

Running c-jwt-cracker on the token gives us the secret: `abc`

```
┌──(kali㉿kali)-[~/Downloads/tools/c-jwt-cracker]
└─$ ./jwtcrack eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoidXNlciIsInRpbWUiOjE2MzczNDE5MDMuMzQ1ODY4fQ.vvUrhB4XbSyvjylEjqBmfHUKopQo6ZXy7QMMNHQC61Y
Secret is "abc"
```

We then craft our new JWT Token, with the role changed from `user` to `admin`, as well as mocking a valid signature with the secret `abc`.

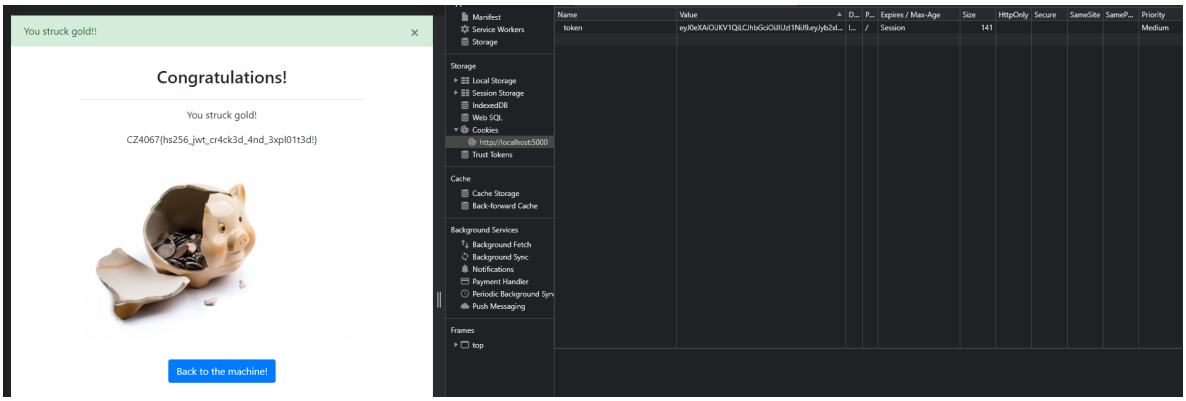


We obtain the new JWT token:

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiYWRtaW4iLCJ0aW1lIjoxNjM3MzQxOTAzLjM0NTg2OH0.3NY

Step 3 - Exchanging the JWT Token:

After editing the token cookie and clicking on the button 'Exchange Token', we are brought to the success page with the flag printed.



Flag

CZ4067{hs256_jwt_cr4ck3d_4nd_3xpl01t3d!}