Flags

1. no_clients_plz_ee2f24
2. not_this_again_39d025
3. p1c0_s3cr3t_ag3nt_7e9c671a
4. s0m3_SQL_93e76603
5. m0R3_SQL_plz_015815e2
6. 3v3n_m0r3_SQL_c2c37f5e
7. th3_c0nsp1r4cy_l1v3s_a03e3590
8. jawt_was_just_what_you_thought_d571d8aa3163f61c144f6d505ef2919b

Python Libraries Used

- requests - used for getting and posting requests to URLs
- subprocess - used for calling commands in the terminal
- jwt - used for creating a jwt token
- jsbeautifier - used to make the javascript look more readable

How to Run

1. Download and install john the ripper password cracker from: https://github.com/openwall/john in the same directory where you put my homework folder. (so when you perform an ls command there should be MyHwDirectory and john-1.9.0-jumbo-1)
2. Download rockyou.txt from https://www.kaggle.com/wjburns/common-password-list-rockyoutxt and put it in the root of my directory
3. In order to run each individual task do ./taskN.sh or to run all tasks at once do ./tasks.sh

Task 1

For this task I started by doing an inspect element on the website and in the body element I found the script in charge of handling the password verification. I realized that it basically split

the password into segments that are 4 characters long and performed checks on each segment to see if they match a string. Then I put all the strings together in order and got the flag.

Task 2

I used a similar approach for the second website where I did an inspect element on the website and once I saw the script I realized that it was very hard to read so I took all the script code into a text editor and changed all the obscure variable names into ones that made sense such as '0x5a46' became 'init'. Then I realized that there were similar checks being made as in task 1 and once I put them all together I got the flag.

Task 3

For this task when I got on the website I immediately pressed the flag button and was prompted with the error message that I am not picobrowser. My first instinct was to see if I can download picobrowser and then it would let me view the flag but the first result on google was the answer so I then googled the other part of the error message (Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0) and found out that this is the user agent for Firefox on my computer. Then I remembered that within Burp Suite I can change the user-agent by intercepting a client request and then changing the user-agent in that request to 'picobrowser' and then when I forwarded the new request I was given the flag.

Task 4

For this task I went into the login page and started by doing an inspect element but I didn't find anything noteworthy so I looked at the hint again and decided to go back to the slides and looked through the sql injection slides. After looking at the "using sql injection to log in" slide I decided to try using "' OR 1=1 --" in the username and put in a random password and once I clicked login I got the flag.

Task 5

For this task I went with the same approach as Task 4 but it told me that it detected SQLi so I decided to use Burp Suite to look into the request that the client was sending out and discovered that there is a debug flag within the request so I set it to 1 and it showed me the SQL query that the backend was going to run. I thought that I would try performing a query where I would guess the name in (SELECT * FROM users WHERE name='name' AND password='password') and comment out the rest and I did that by adding a comment after the name so that the password field would be ignored. I first tried Admin'-- as that was the capitalization of the word admin on the support page but that didn't work so I then tried admin'-- and this gave me the flag. I did this in the repeater tab of Burp Suite and changed the username until it gave me the flag.

Task 6

For this task I did the same thing as in task 5 where I set the debug flag to 1 and analyzed the SQL query that was printed out and I also noticed that the password that I had provided was being encrypted in the query. I then decided to figure out what cipher it was using by providing a bunch of strings and I realized that this was a caesar cipher of 13 so I decided to try doing SQL injection where I would encrypt (' OR 1=1--) which became (' BE 1=1--) and when I provided that the website gave me the flag.

Task 7

For this task I started out by inputting a random username and password and decided to look through the intercepted requests from the client and realized that in the 3rd and 4th request there was an admin flag in the cookie that was set to False and so I set it to True for both requests and forwarded it back to the site and once the site received both requests it gave me the flag.

Task 8

For this task I started out by researching what a Java Web Token is and once I finished I decided to try and see what would happen if I tried to login as admin and to my surprise it gave me an error message and there was no cookie in the client request as I expected so I tried to see

what would happen if I provided a random name and realize that with the random name there was a cookie sent in the second request that was in the format: (jwt=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyIjoiYnJ1aCJ9.VykR6GCaZ99NO_A0Pjf_aQ_sdKKTahZo2rn8xViX_mM) so I knew at that point that this was the Java Web Token and that I would have to determine its secret to be able to generate a JWT for the admin. I passed this token to the decoder in Burp Suite and had it decode it as Base64 and got back: {"typ":"JWT","alg":"HS256"}.{"user":"bruh"}.W)è`gßMO_Cãf_aQ_±ÒM¨Y£jçóbX_mM which confirmed that this was a JWT that used the HW256 encoding and it was for the user "bruh". I at this point started looking into how I would get the secret and from the hint I got the memo that I will need to brute force the secret and I stumbled on the John the Ripper password cracker tool that was linked in the website so I decided to try and use this tool to crack the secret. I then proceeded to install this password cracker on my machine and in the instructions I noticed that I should provide a wordlist for better results so I decided to use the rockyou password list since I remembered that we discussed it in one of our lectures. Then I put the Java Web Token that I got from the user "bruh" into a file called passwd and ran the command "./john passwd --wordlist=rockyou.txt" and then the command "./john passwd --show" to see the secret which was ilovepico. I then looked into seeing how I can encode this and found a nice tool online at https://jwt.io/ which allowed for decoding and encoding of a JWT if the secret was provided so I input the previous JWT for the user "bruh", changed the secret to "ilovepico" and then changed the user to "admin" which gave me the JWT: (eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyIjoiYWRtaW4ifQ.gtqDl4jVDvNbEe_JYEZTN19Vx6X9NNZtRVbKPBkhO-s). I then passed this jwt as a cookie in the client request, that was intercepted when I put in the user "admin", in the format: (Cookie: jwt=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyIjoiYWRtaW4ifQ.gtqDl4jVDvNbEe_JYEZTN19Vx6X9NNZtRVbKPBkhO-s) and when I forwarded it back I got the flag.