**Project Report for Practical Web Apps Hacking –**

**Vulnerability Assessment and Penetration Test(VAPT) –**

**Title – IDOR in password reset form**

**Description of the vulnerability -**

**Project 1 –**

**Website – picoCTF**

Project title – login

Description of the vulnerability – My dog-sitter's brother made this website but I can't get in; can you help?

Summary –

The vulnerability was that the login page did not have any server side involvement, the answer to this was to open the source page that was the index.js file and there was a function called btoa which converts base 64 encoded string to a normal string, then pasted the username and the password using the Unix shell with the command echo ‘username’ | base64 -d and echo ‘password’ | base64 -d respectively and then got the flag as the output and then copied the picoCTF flag and successfully completed the challenge.

Steps to reproduce –

1.open the source page that was the index.js file

2.paste the username and the password using the Unix shell with the command echo ‘username’ | base64 -d and echo ‘password’ | base64 -d respectively

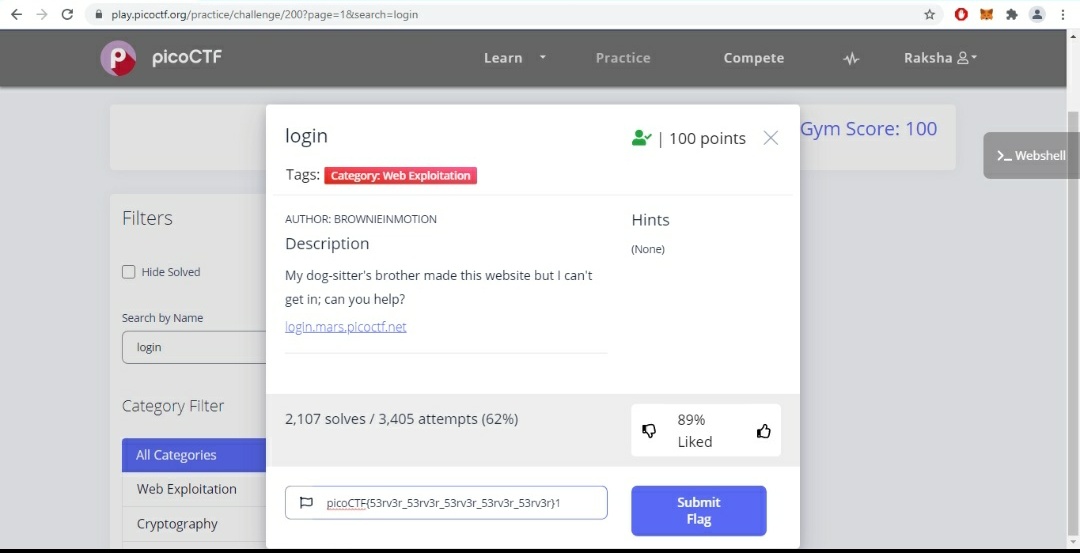
3. the flag as the output and then copy the picoCTF flag

4.successfully completed the challenge.

Impact –

Access to the account by copy pasting the username and password from the Unix shell after converting the 64 bit word to normal form which was present in the index.js file in the source file.

Got the output by the Unix commands that is echo ‘username’ | base64 -d and echo ‘password’ | base64 -d respectively and then got the picoCTF flag which then was pasted and successfully completed.



**Project 2 –**

**Website – picoCTF**

**Project title – Spelling-quiz**

Description of the vulnerability - I found the flag, but my brother wrote a program to encrypt all his text files. He has a spelling quiz study guide too, but I don't know if that helps.

Summary –

There was three files in the list when the ls command was executed in the Unix shell and one file named encrypt.py had an invalid syntax and then executed the command GNU nano 2.9.3 and it displayed random sets of jumbled words and then copy pasted the words to a cypher analyser in fox entry and then it was a monoalphabetic substitution cipher which then performed frequency analysis in the substitution analysis tool and auto solved it and then found the picoCTF flag there and successfully completed.

Steps to reproduce –

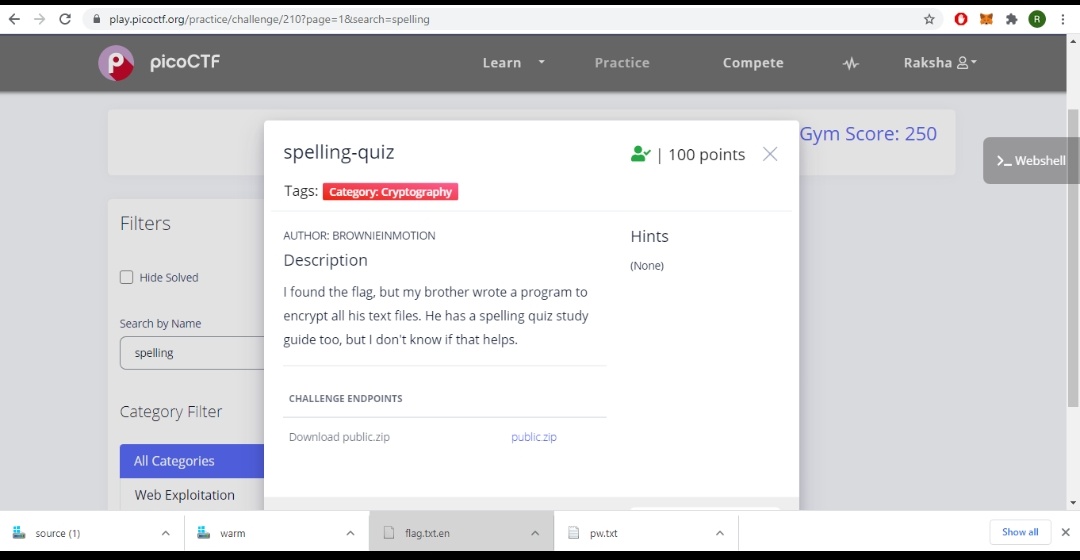
1.There was three files in the list when the ls command was executed in the Unix shell and one file named encrypt.py

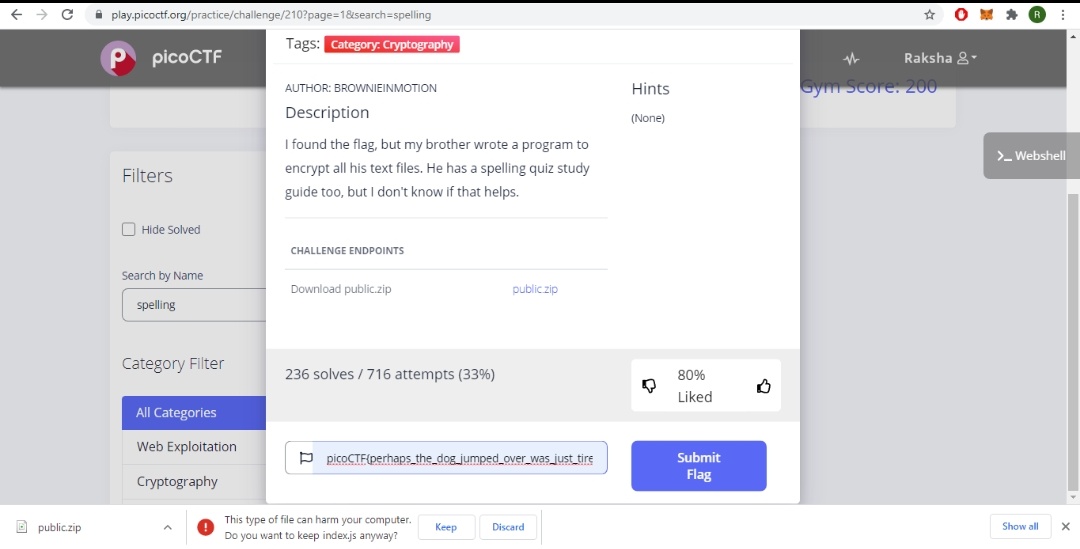
2.open the encrypt.py file

3.executed the command GNU nano 2.9.3 and it displayed random sets of jumbled words

4.then copy pasted the words to a cypher analyser in fox entry and then it was a monoalphabetic substitution cipher which then performed frequency analysis in the substitution analysis tool and auto solved it

5.found the picoCTF flag in the substitution analysis tool and successfully completed





**Project 3 –**

**Website – picoCTF**

**Project title – Shop**

Description of the vulnerability – Best Stuff – Cheap Stuff, Buy Buy Buy… Store Instance: source. The shop is open for business at nc mercury.picoctf.net 11371.

Summary –

Copy pasted the Unix command and then choose the all the options one by one and then that allowed to buy coins and also sell the items and then when it reached to 140 coins it displayed the picoCTF flag which was in decimal and then used an online tool to convert decimal to ASCII text and it generated the actual picoCTF flag and then copy pasted it to the shop file and submitted and successfully completed.

Steps to reproduce –

1.Copy pasted the Unix command and then choose the all the options one by one.

2.then that allowed to buy coins and also sell the items

3.then when it reached to 140 coins it displayed the picoCTF flag which was in decimal form

4.then used an online tool to convert decimal to ASCII text and it generated the actual picoCTF flag

5.then copy pasted it to the shop file and submitted and successfully completed.