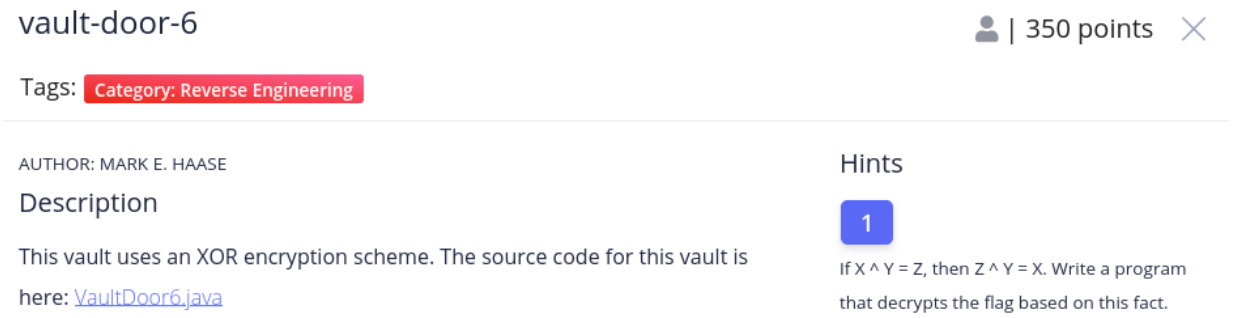
Mark Seliternikov
picoCTF - Vault Door 6 [350 points]



The challenge gives a source code in which the password is saved inside it. The attempt was to hide the password using XOR.

In this line you can see that the password is the flag and the contents inside the flag are hidden:

```
String userInput = scanner.next();
String input = userInput.substring("picoCTF{".length(),userInput.length()-1);
if (vaultDoor.checkPassword(input)) {
```

Later on there's a function checkPassword() that is being called in order to check if the input matches.

The function:

```
public boolean checkPassword(String password) {
    if (password.length() ≠ 32) {
        return false;
    }
    byte[] passBytes = password.getBytes();
    byte[] myBytes = {
        0x3b, 0x65, 0x21, 0xa , 0x38, 0x0 , 0x36, 0x1d,
        0xa , 0x3d, 0x61, 0x27, 0x11, 0x66, 0x27, 0xa ,
        0x21, 0x1d, 0x61, 0x3b, 0xa , 0x2d, 0x65, 0x27,
        0xa , 0x6c, 0x60, 0x37, 0x30, 0x60, 0x31, 0x36,
    };
    for (int i=0; i<32; i++) {
        if (((passBytes[i] ^ 0x55) - myBytes[i]) ≠ 0) {
            return false;
        }
    }
    return true;
}
```

The function checks whether the byte (char is 1 byte) of the string matches the byte of the list of myBytes in order.
The list contains hex values that are the result of input byte XOR 0x55.

```
if (((passBytes[i] ^ 0x55) - myBytes[i]) ≠ 0) {
    return false;
```

So in order to solve it I've made a python script that finds the correct input byte so it'd match the result.
Here is the script I wrote: [I've added the flag format in advance in order to get a pretty print :)]

```python
myBytes = [0x3b, 0x65, 0x21, 0xa , 0x38, 0x0 , 0x36, 0x1d, 0xa , 0x3d, 0x61, 0x27, 0x11, 0x66, 0
x27, 0xa ,0x21, 0x1d, 0x61, 0x3b, 0xa , 0x2d, 0x65, 0x27,0xa , 0x6c, 0x60, 0x37, 0x30, 0x60, 0x3
1, 0x36]

print("picoCTF{", end='')

for i in range(32):

    for char in range(0xff):

        if (char ^ 0x55) - myBytes[i] == 0:

            print(chr(char), end='')

print('}')
```

And here is the output:

```
┌──(kali㉿kali)-[~]
└─$ python3 solver.py
picoCTF{n0t_mUcH_h4rD3r_tH4n_x0r_95be5dc}
```

Conclusion:

The main takeaway from all the vault-door challenges is: **DO NOT STORE THE PASSWORD ON THE SOURCE CODE.**