# Challenge 23: PW Crack 3

## Description

Can you crack the password to get the flag? Download the password checker here and you'll need the encrypted flag and the hash in the same directory too. There are 7 potential passwords with 1 being correct. You can find these by examining the password checker script.

## Tags

`Beginner picoMini 2022` `General Skills` `password_cracking`

## Points

`100`

## Solution

In this challenge we are given a script and an encrypted flag. also we are given a hash file. The script is a simple password checker, if we entered the correct password it will print the flag. we can take a look at the script. It might has something useful to us.

```
import hashlib
```

```
### THIS FUNCTION WILL NOT HELP YOU FIND THE FLAG --LT ########################
def str_xor(secret, key):
    #extend key to secret length
    new_key = key
    i = 0
    while len(new_key) < len(secret):
        new_key = new_key + key[i]
        i = (i + 1) % len(key)
    return "".join([chr(ord(secret_c) ^ ord(new_key_c)) for (secret_c,new_key_
###############################################################################

flag_enc = open('level3.flag.txt.enc', 'rb').read()
correct_pw_hash = open('level3.hash.bin', 'rb').read()


def hash_pw(pw_str):
    pw_bytes = bytearray()
    pw_bytes.extend(pw_str.encode())
    m = hashlib.md5()
    m.update(pw_bytes)
    return m.digest()


def level_3_pw_check():
    user_pw = input("Please enter correct password for flag: ")
    user_pw_hash = hash_pw(user_pw)

    if( user_pw_hash == correct_pw_hash ):
        print("Welcome back... your flag, user:")
        decryption = str_xor(flag_enc.decode(), user_pw)
        print(decryption)
        return
    print("That password is incorrect")



level_3_pw_check()


# The strings below are 7 possibilities for the correct password.
#   (Only 1 is correct)
pos_pw_list = ["f09e", "4dcf", "87ab", "dba8", "752e", "3961", "f159"]
```

As we can see, the password is hashed and then compared to the hash file. We also have a list of possible passwords. We can try to crack the password using the list of possible passwords. We can use the `hashlib` library to hash the passwords and compare them to the hash file.

```python
import hashlib

correct_pw_hash = open('level3.hash.bin', 'rb').read()

def hash_pw(pw_str):
    pw_bytes = bytearray()
    pw_bytes.extend(pw_str.encode())
    m = hashlib.md5()
    m.update(pw_bytes)
    return m.digest()

pos_pw_list = ["f09e", "4dcf", "87ab", "dba8", "752e", "3961", "f159"]

for pw in pos_pw_list:
    if hash_pw(pw) == correct_pw_hash:
        print("The password is: " + pw)
```

If we run the script we just created, we will get the password.

```
$ python3 crack.py

The password is: dba8
```

Now we can use the password to decrypt the flag.

```
$ python3 level3.py
Please enter correct password for flag: dba8

Welcome back... your flag, user:
picoCTF{m45h_fl1ng1ng_cd6ed2eb}
```

## Flag

> picoCTF{m45h_fl1ng1ng_cd6ed2eb}