PW Crack 3

PicoCTF

Fares Zuleta

February 21, 2026

Contents

1 Description

Can you crack the password to get the flag? Download the password checker 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.

2 Files

After downloading the files, we can see three files in our directory.

```
[Zer0th@Arch PWCrack3]$ ls
level3.flag.txt.enc level3.hash.bin level3.py
```

When reading the script we can observe the following code.

3 Script

```
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_c
        ) in zip(secret,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 = ["6997", "3ac8", "f0ac", "4b17", "ec27", "4e66", "865e"]
```

4 Analysis

After analyzing the source code, we can extract the following line.

```
pos_pw_list = ["6997", "3ac8", "f0ac", "4b17", "ec27", "4e66", "865e"]
```

We can either try every password and eventually get the flag or we can create a script to compare the hash key with the real password instead of guessing.

5 Creating the script

Instead of manually testing each candidate, we can create a script to automate the hash comparison process. The challenge already gave us a .hash.bin.

```
level3.hash.bin
```

To read the file we have to execute the following command in our linux/OS.

```
hexedit level3.hash.bin
```

5.1 Hexedit

Then it displays the hexadecimal values of the hash key (MD5).

```
1B 18 E1 31 6F 92 18 CC 5B 05 3E 1C EA 28 E0 2E
```

5.2 Python

After reading the files we already have all the information to create our own script.

```
import hashlib

PW = ["6997", "3ac8", "f0ac", "4b17", "ec27", "4e66", "865e"]

key = "1b18e1316f9218cc5b053e1cea28e02e"

for pw in PW:
        hashed = hashlib.md5(pw.encode()).hexdigest()
        if hashed == key:
                print(pw)
                break
```

5.3 Explaining our script

The following line imports the hashlib module to compute MD5 hashes. Then we have the password array, adding our passwords. The key line is the hexadecimal value of the key we previously obtained, The hexadecimal hash was converted to lowercase to match the format returned by hexdigest().

```
import hashlib

PW = ["6997", "3ac8", "f0ac", "4b17", "ec27", "4e66", "865e"]
key = "1b18e1316f9218cc5b053e1cea28e02e"
```

The `for` loop iterates through each candidate password, computes its MD5 hash, and compares the result with the extracted hash value.

```
for pw in PW:
        hashed = hashlib.md5(pw.encode()).hexdigest()
        if hashed == key:
                print(pw)
                break
```

6 Executing our script

We now execute the script to determine the correct password without manually guessing.

```
[Zer0th@Arch PWCrack3]$ python3 unhash.py
865e
```

As shown above, the password obtained is ”865e”.

7 Executing the challenge’s script

After successfully obtaining the password, we must test it in the level3.py script to know its correctness.

```
[Zer0th@Arch PWCrack3]$ python3 level3.py
Please enter correct password for flag: 865e
Welcome back... your flag, user:
picoCTF{m45h_fl1ng1ng_2b072a90}
```

The flag was successfully obtained.