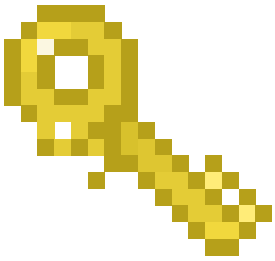


Jam



Type: Cryptography

Difficulty: Medium

Flag: `TFCCTF{PUmP_up_th3_j4m,PUMP_17_up!h4shpump_it!}`

Points: 490

Description

Found this amazing new game! It's called Jam! It's very hard, though, can you help me advance faster?

Solution

After looking at the source file of challenge, we notice a few interesting things, in the way the console is being initialised

```
def __init__(self):
  self.__secret = secret.encode("utf-8")
  self.__hash = hashlib.md5(self.__secret + b' 0').hexdigest()
  self.__coins = 0
```

We notice that the goal of the challenge is to get 6 coins. It's seemingly impossible, as after working for 5 coins, you become too tired to work for more!

```
if self.__coins >= 5:
  print("Can't work anymore! You're too tired!")
  return
```

In case we want to play later, we have a way to recover our session, using the hash we got after working n times!

MD5 Length Extension Attack

MD5 is a hashing algorithm, that's vulnerable to a length extension attack. That means, we can add some extra data at the end of a partly given hashed string, and figure out what it's hash will be!

Read more about it here:

> Length extension attack - Wikipedia
>
> In cryptography and computer security, a length extension attack is a type of attack where an attacker can use () and the length of to calculate Hash(‖) for an attacker-controlled , without needing to know the content of .
>
> 𝕎 https://en.wikipedia.org/wiki/Length_extension_attack

And this is exactly what we will use in order to *crack* this game. There are some tools online that can help us out with this task. One of them is hashpump, which has an available package for Python.

> https://github.com/bwall/HashPump

Solve Script

```
from pwn import *
import hashpumpy

h, t = hashpumpy.hashpump("7cbf6c7b09179cb964ef2e9589732f10", "1", "0", 33)

p = process("main.py")
print(p.recvuntil("----------"))
p.recvline()
p.sendline("3")
print(p.recvuntil("have?"))
```

```
p.sendline(t)
print(p.recvuntil("?"))
p.sendline(h)
print(p.recvuntil("-------"))
p.recvline()
p.sendline("2")
print(p.recvuntil("-------"))
p.sendline("1")
print(p.recvuntil(">"))
flag = p.recvline()
print(flag)
```