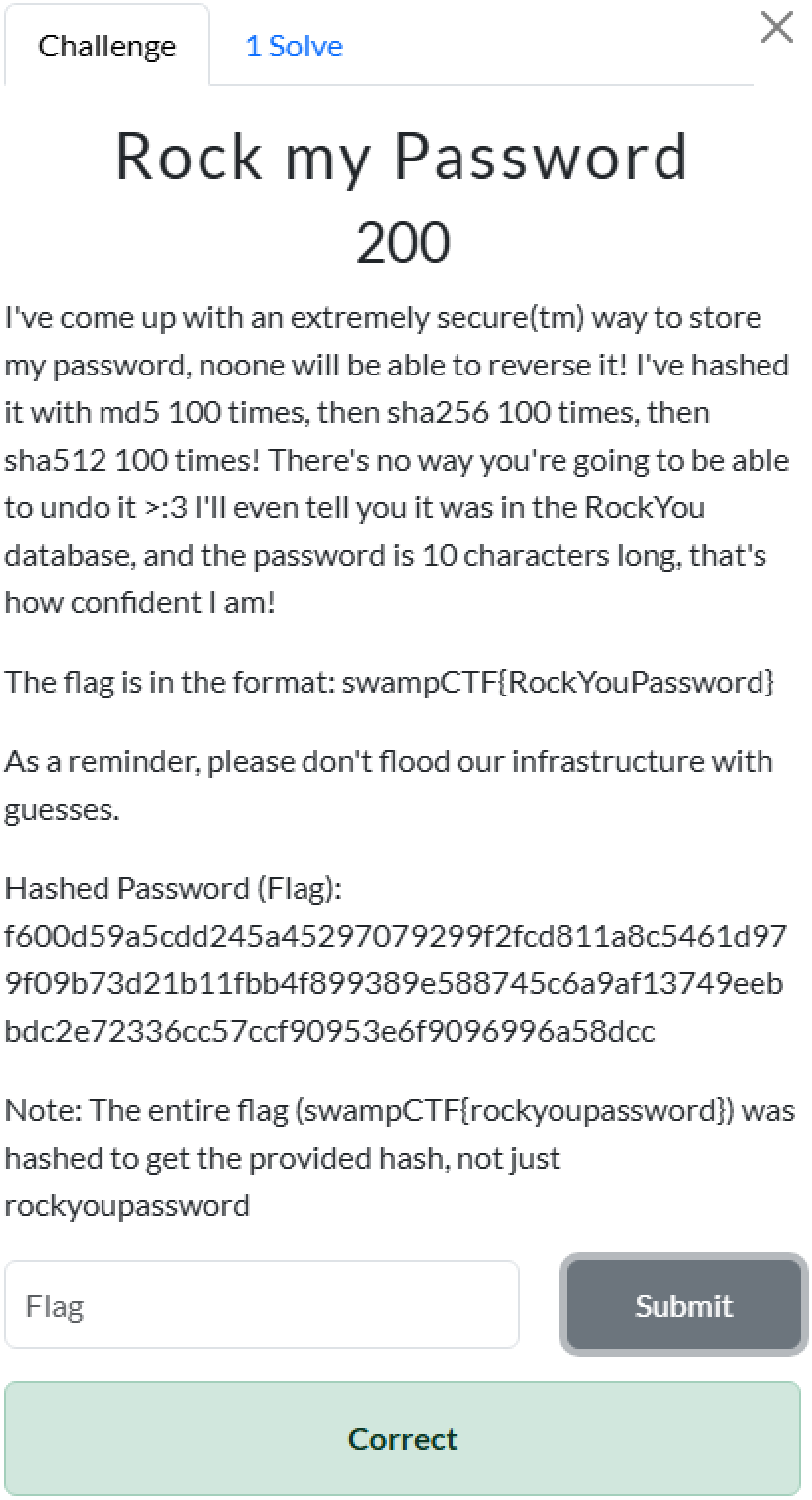

Rock my Password

200

I've come up with an extremely secure(tm) way to store my password, noone will be able to reverse it! I've hashed it with md5 100 times, then sha256 100 times, then sha512 100 times! There's no way you're going to be able to undo it >:3 I'll even tell you it was in the RockYou database, and the password is 10 characters long, that's how confident I am!

The flag is in the format: swampCTF{RockYouPassword}

As a reminder, please don't flood our infrastructure with guesses.

Hashed Password (Flag): f600d59a5cdd245a45297079299f2fcd811a8c5461d979f09b73d21b11fbb4f899389e588745c6a9af13749eebbdc2e72336cc57ccf90953e6f9096996a58dcc

Note: The entire flag (swampCTF{rockyoupassword}) was hashed to get the provided hash, not just rockyoupassword

Correct

The challenge is asking us to reverse a series of hashes applied to a password. The steps are:

Password format: The password is in the RockYou database and is 10 characters long. The flag format is swampCTF{RockYouPassword}.

- The entire flag (swampCTF{password}) is hashed multiple times.
- First, MD5 is applied 100 times.
- Then, SHA-256 is applied 100 times to the result of the MD5 hash.
- Finally, SHA-512 is applied 100 times to the result of the SHA-256 hash.
- The result after 300 hash iterations is compared with the provided hash.

Final Goal: We are given the final hash, and we need to find the 10-character password from the RockYou database that, when the flag is hashed in the described manner, results in the given hash.

Steps:

1. Download RockYou.txt
2. Python Script

```
import hashlib

# Path to your rockyou.txt file
ROCKYOU_PATH = "rockyou.txt"
# Final hash from the challenge
TARGET_HASH =
"f600d59a5cdd245a45297079299f2fcd811a8c5461d979f09b73d21b11fbb4f899389e
588745c6a9af13749eebbdc2e72336cc57ccf90953e6f9096996a58dcc"

def hash_100_times(algorithm, data):
   for _ in range(100):
      h = hashlib.new(algorithm)
      h.update(data)
      data = h.digest()
   return data

def process_rockyou():
   with open(ROCKYOU_PATH, "r", encoding="latin-1") as file:  # Changed to text mode
with latin-1 encoding
      for line in file:
         password = line.strip()
         if len(password) != 10:
            continue
      flag = b"swampCTF{" + password.encode() + b"}"  # Encoding the password to bytes for hashing

      md5_result = hash_100_times("md5", flag)
      sha256_result = hash_100_times("sha256", md5_result)
      sha512_result = hash_100_times("sha512", sha256_result)

      if sha512_result.hex() == TARGET_HASH:
        print(f"✅ Found it! Password: {password}")
        print(f"🚩 Flag: swampCTF{{{password}}}")
        return

  print("❌ No match found.")

if __name__ == "__main__":
  process_rockyou()
```

Flag : swampCTF{secretcode}