CPRE 2300f

HW 02

Assignments will be submitted in PDF format via Canvas.

Please submit your homework online through Canvas. Late homework will not be accepted.
Please ensure that you support all your answers with the correct screenshots showing your solutions.

1. In this “lab” problem, you have been tasked with accessing a locked server, and you will need to perform password cracking! You were given a line of password hash from (/etc/shadow) for the administrator of the Server (attached with the HW). Screenshot your progress and explain your process for the following steps.
    a. Determine the used hash type of the password.

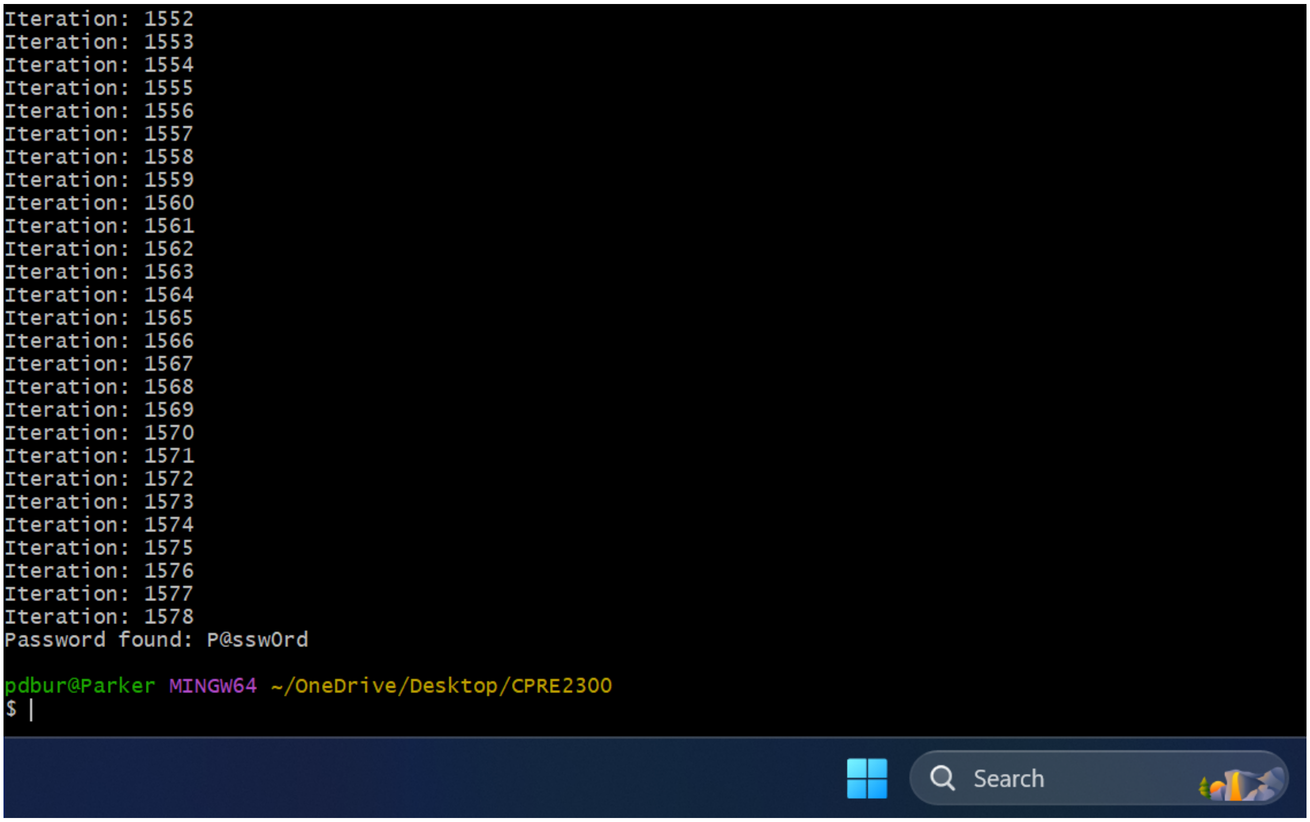
admin1:$6$xgLS35S6$2UjEq.dUhICPw9zgDVJXcQYQp/9ilLPQt/8Zgu0uwngI5mVvB1eKQG9SnVLjmOOfkB4Jjb5VSAXGXjY4Cf5k90:18169:0:99999:7:::

This is the admin password hash given to use in /etc/shadow. Each field in the hash is separated by a colon (:). The $6$ is the key point to understanding what hash is being used here. $6$ means that this hash is SHA-512. This is the default Linux password storage hash.

b. Determine the salt value of the password.

Password hashes are structued as $<hash-type>$<salt>$<hash-string>. Therefore looking at the hash string given to us above xgLS35S6 would be salt value of this specific password.

c. Crack the password of the server using OpenSSL tool.

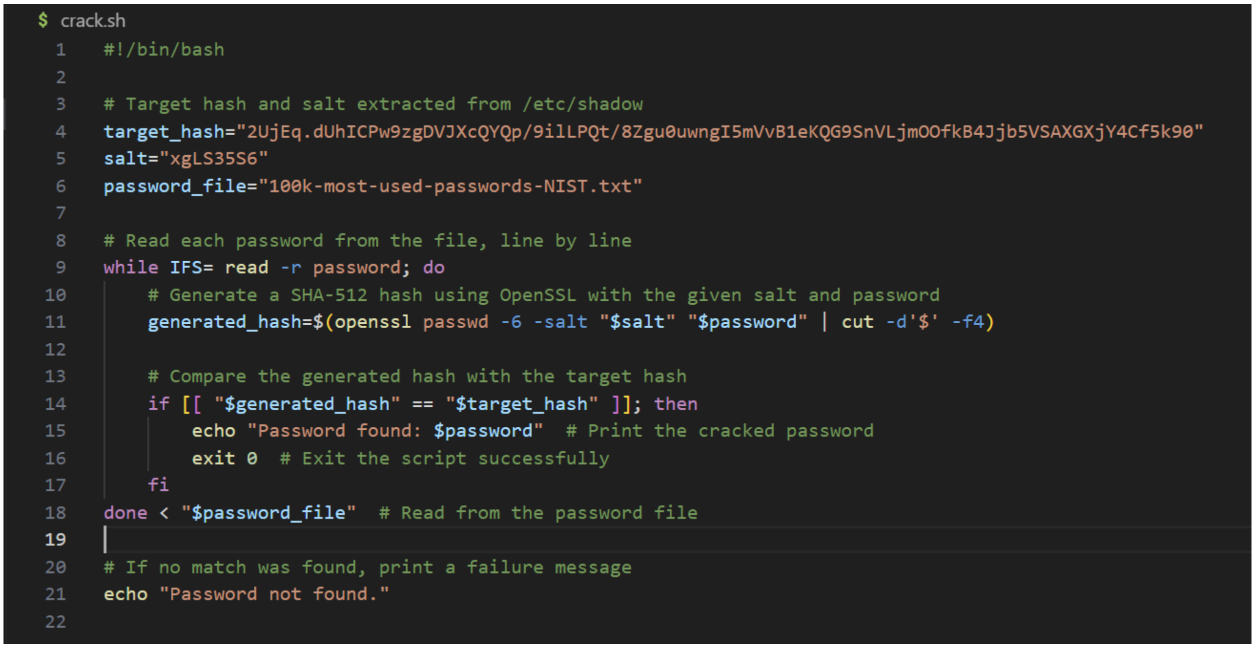


```
Iteration: 1552
Iteration: 1553
Iteration: 1554
Iteration: 1555
Iteration: 1556
Iteration: 1557
Iteration: 1558
Iteration: 1559
Iteration: 1560
Iteration: 1561
Iteration: 1562
Iteration: 1563
Iteration: 1564
Iteration: 1565
Iteration: 1566
Iteration: 1567
Iteration: 1568
Iteration: 1569
Iteration: 1570
Iteration: 1571
Iteration: 1572
Iteration: 1573
Iteration: 1574
Iteration: 1575
Iteration: 1576
Iteration: 1577
Iteration: 1578
Password found: P@ssw0rd

pdbur@Parker MINGW64 ~/OneDrive/Desktop/CPRE2300
$ |
```

I added iterations to my bash code to ensure that it was running and I could also track the progress. I removed the iterations for the final submission however as they were not needed. I used Git Bash to run the bash script.

d. Submit commented, working and compiled code/script which cracks the password



$ crack.sh

```bash
#!/bin/bash

# Target hash and salt extracted from /etc/shadow
target_hash="2UjEq.dUhICPw9zgDVJXcQYQp/9ilLPQt/8Zgu0uwngI5mVvB1eKQG9SnVLjmOOfkB4Jjb5VSAXGXjY4Cf5k90"
salt="xgLS35S6"
password_file="100k-most-used-passwords-NIST.txt"

# Read each password from the file, line by line
while IFS= read -r password; do
    # Generate a SHA-512 hash using OpenSSL with the given salt and password
    generated_hash=$(openssl passwd -6 -salt "$salt" "$password" | cut -d'$' -f4)

    # Compare the generated hash with the target hash
    if [[ "$generated_hash" == "$target_hash" ]]; then
        echo "Password found: $password"  # Print the cracked password
        exit 0  # Exit the script successfully
    fi
done < "$password_file"  # Read from the password file

# If no match was found, print a failure message
echo "Password not found."

```

e. Provide the password

The password given to us in the shadow:

P@ssw0rd

This is also shown in my screenshot of the compiled code running

What to submit?

- PDF containing your answers to the questions above
- Code, in your language of choice, cracking the password
- Submit in a zip file to Canvas

Hints:

- It would be useful if you search for Linux shadow file password format.
- You must use the latest OpenSSL version 1.1.1x for this problem. It would be helpful to read about creating Linux password hashes using OpenSSL.
- You can use a password list for your cracking. There is a password list of most used 100K passwords, according to NIST attached with the HW.
- You will need to write some code to iterate through the password list (feel free to use any language you prefer).

- In the case that you would prefer not to install OpenSSL on your personal machine, you can use vdi.engineering.iastate.edu to access the Iowa State University workstations or work on any computer in Coover as they have OpenSSL pre-installed.
- It is highly recommended to code it in C or Python or Bash Script
- If you use C, look up the crypt.h header files and the -lcrypt flag for compiling
- If you use Python, look up the crypt package/modules
- If you want to go with the OpenSSL route, it is easiest to write a bash script since it is basically running a terminal command. Experiment with one command first in terminal before turning it into a script
- passlib is a good option for this homework due to the rounds parameter. (Will work on windows)
- OpenSSL is suggested has to do with default parameters in Linux password generation
- If you use a hash algorithm or library, you have to make sure it does 5000 rounds or you will get a different output
- crypt will also work though the salt needs to be in the format AND you need to be in linux when you run the code (since it uses openssl as a backend) (\$6\$salt\$)
- Ensure that when parsing the file, you are using utf-8 encoding and that there are no extra characters at the end of the line such as \r