**Problem 2: Password Cracking**

Victoria Ng, Laraib Sarmad

Pace University

CIT 74409: Computer Security Overview

Professor Li-Chiou Chen

12/6/2021

**Contributions**

Both parties assisted each other with cracking the password. But Laraib took on the main leading role in it and writing down what additional methods that we used that either worked to solve the problem or failed when trial and error testing. Victoria took on the main role in writing the rest of the paper and putting everything together to make sure that everything flowed together and had the correct format. Both parties were evenly involved in each part of this project.

**Introduction**

The problem that we chose to investigate was the password cracking problem. In that problem we were given three different entries from a Kali Linux shadow file from users 1, 2, and 8. All three given entries were in hash form which was generated using SHA-512. The goal was for us to find out what the password for user 8 was. We were already given the password for user1 which was cit251cit251 with a hint that told us that the password for user8 was very similar. Additional material that was given was a list of 30 potential passwords for user8 that we had to test with using a bash script and hashcat. Bash scripts are used to automate tasks that are repetitive on the Linux filesystem like loops, functions, conditional constructs, and etc. Hashcat is a password cracking tool that is fast, efficient, and is a versatile hacking tool According to cso.com, hashcat guesses a password hashes it and then compares the resulting hash to the one it's trying to crack. If the hashes match, then we know the password. If it does not match, then it keeps on guessing and or applying. This relates to the bash scripts and cracking passwords because both could be used if a user wanted to crack a password.

**Problem Research**

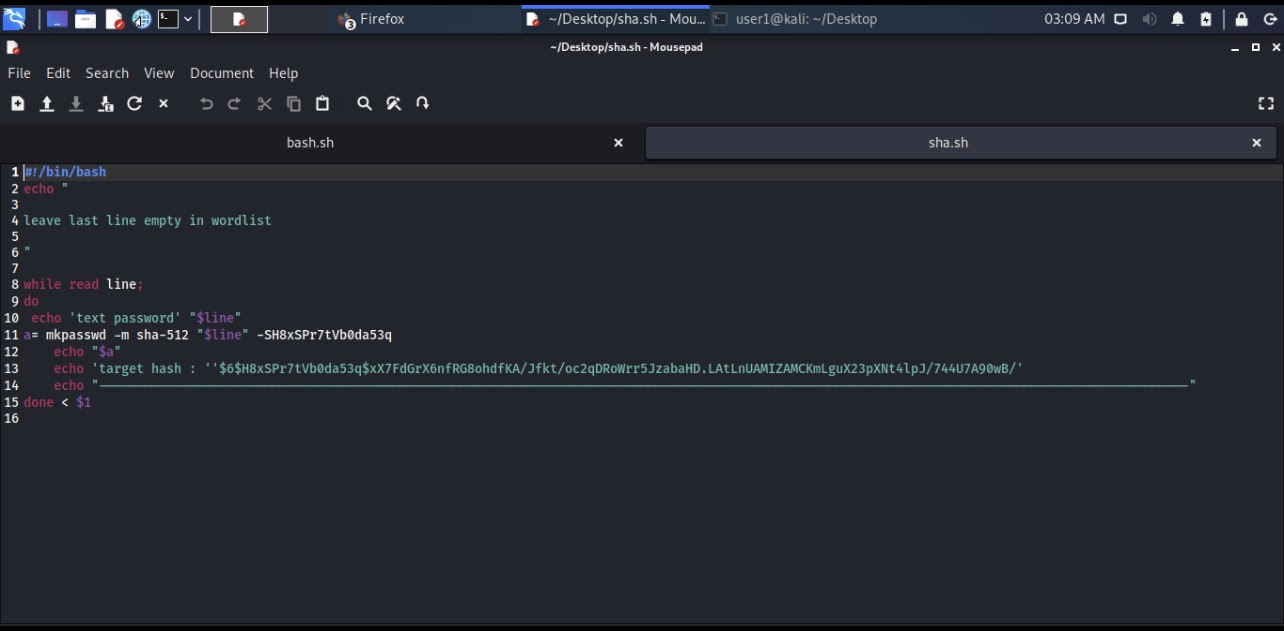
Here you describe the detail steps you took to investigate and solve the

problem, including the steps on using any software/commands to solve the problem,

screen shots of key steps, etc. You should provide evidence to show how you solve

the problems.

The method we used is usually called Dictionary attack. A dictionary attack is a systematic technique of guessing a password. Attackers try different but commonly used passwords. After the list we were provided we tried dictionary attack, by creating a list of the provided passwords called passwd.txt and sha.sh in which a script was written which includes the following:



**Commands we used to crack the password:**

We used Chmod to change the permission of the file called sha.sh, which has the bash script in order to make it work. +x represents that we have given this file the permission of execution. In the end we typed sha.sh, the file name for which we wanted to change the permission.

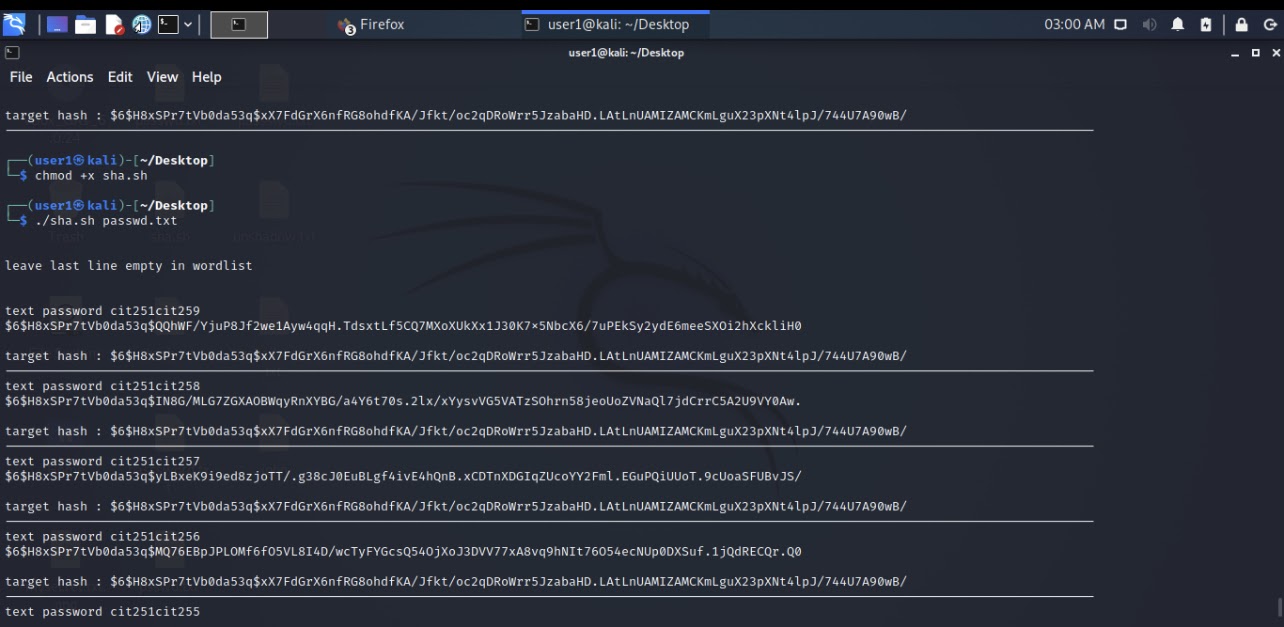
The command is as follow:

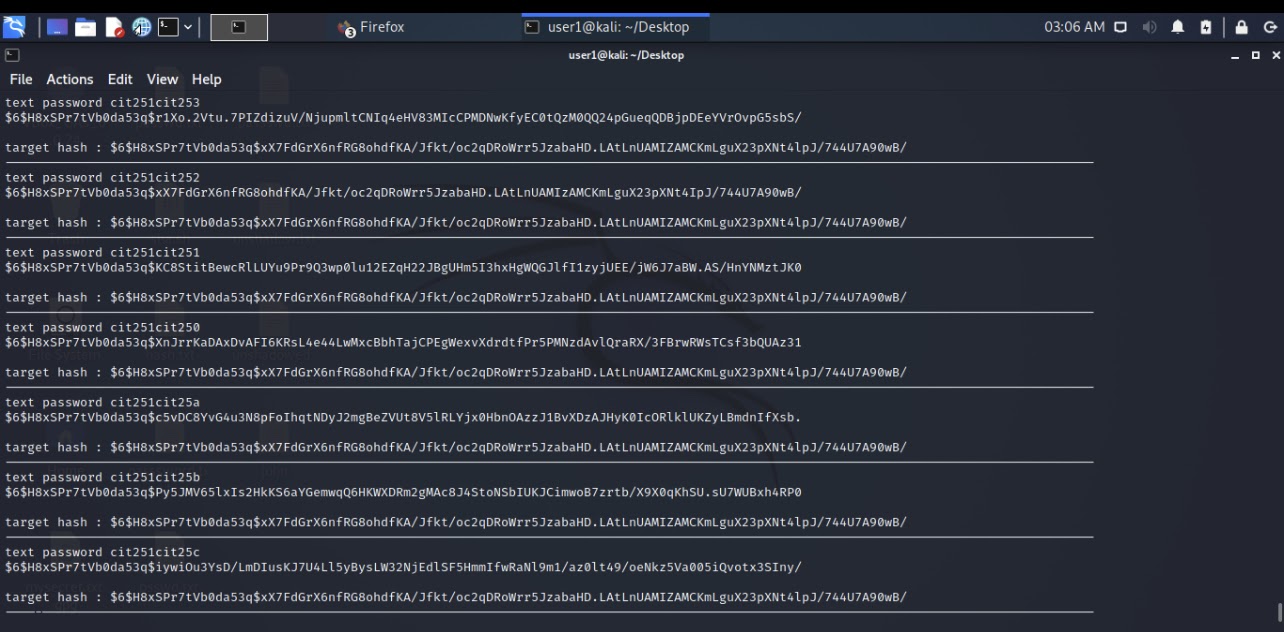
**chmod +x sha.sh**

In the next line, the command we used to get hashed passwords of every single entry from passwd.txt is

**./sha.sh passwd.txt**

In the bash script we wrote, in the end, there was a $1. That $1 will specify hashes of the entire passwd.txt file while using the command we used above.





**Results**

Describe and summarized the results to the problem from your research

After generating passwords for each of the entries, through the bash script. We compared the provided hash with every hash and in the end, it matched with one of the hashes. **cit251cit252** is the correct password because we matched the hashes. In the end, it matched with the provided one perfectly which is how we were able to obtain the correct password.

**Conclusion**

From lab 6 to this project, we all have learned about the world of passwords and hashing. We gained knowledge on what password shadow files, hashing, bash scripts, password cracking using shell scripts, and how to check through a list of possible passwords in Kali Linux using them. As we learned, passwords are not stored in plain texts. Instead, they are hashed using an algorithm such as SHA 512, Yescrypt, MD5, SHA1, and SHA256. However, hashes are not reversible, but that doesn’t mean that they aren’t crackable. An example of this is like the method that we used to crack the password for this project, dictionary attacks. But that isn’t the only method that is able to crack a hashed password. The one that we used is just one of the main commonly used ones. The password was not a common password that everyone uses which is why without a dictionary attack it would be really difficult to obtain it. Overall, this project helped us to better our understanding of how to carry out using a list of potential passwords and generate each one of their hashes to then later cross check everyone to see which one matched. Like in this case we were able to take the list of 30 possible passwords, run them using a shell script which would generate their hashed value. Then from there we took the new modified list and saw which one of the 30 passwords is the correct match for user8.

**References**

“How Does Hashcat Work?: Security Encyclopedia.” *HYPR*, https://www.hypr.com/hashcat/.

“Bash Scripting - Javatpoint.” *Www.javatpoint.com*, www.javatpoint.com/bash-scripting.

Stackflow.com William Pursell Aug 30 ‘16 at 5:13

Hashcat tutorial for password cracking, Ehacking Sep 3, 2021, Irfan

J.M. Porup Senior Writer, CSO | MAY 26, 2020 3:00 AM PDT