

Information Security Management Digital Assignment - 2

Name: Ayush Pathak
RegNo: 21BCE5322

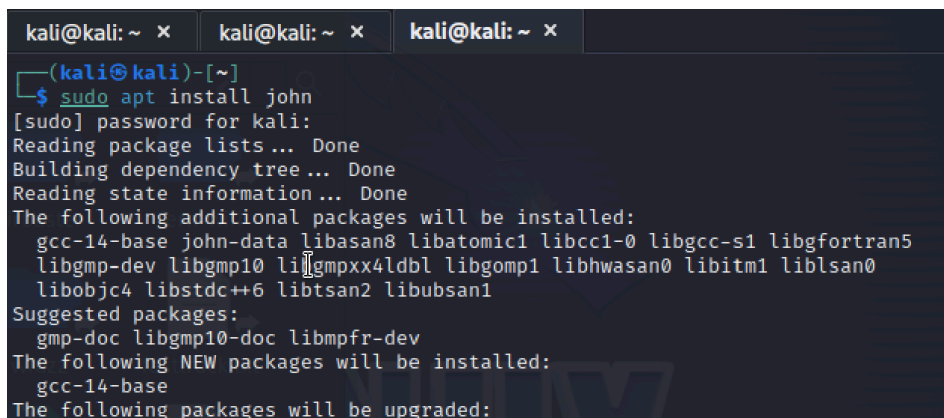
Password Attacks: John the Ripper

The objective is to crack password hashes and gain unauthorised access using John the Ripper. Follow the steps and provide detailed responses.

1. Install John the Ripper on your attacker machine.
We can install using the following command:

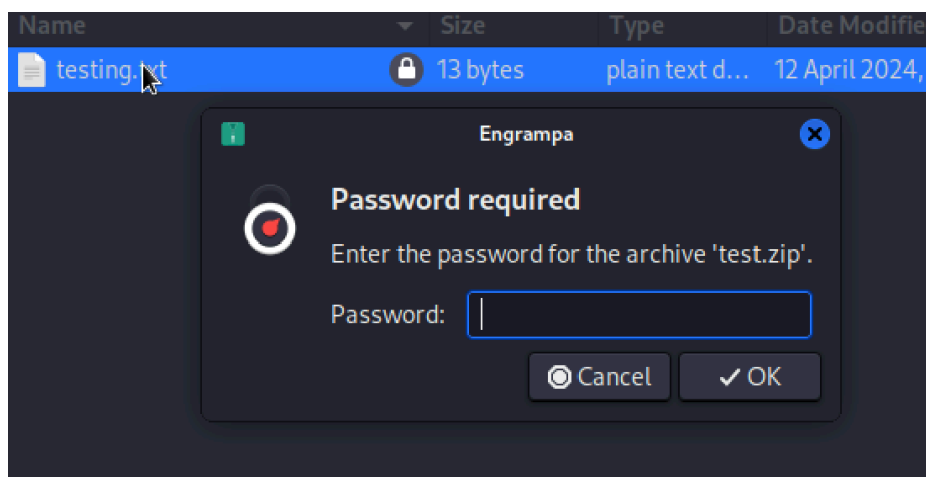
```
sudo apt-get install john
```

2. Use John the Ripper to perform a dictionary attack on a password-protected file containing hashed passwords.



```
kali@kali: ~ x kali@kali: ~ x kali@kali: ~ x
(kali@kali)-[~]
$ sudo apt install john
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  gcc-14-base john-data libasan8 libatomic1 libbcc1-0 libgcc-s1 libgfortran5
  libgmp-dev libgmp10 libgmpxx4ldbl libgomp1 libhwasan0 libitm1 liblsan0
  libobjc4 libstdc++6 libtsan2 libubsan1
Suggested packages:
  gmp-doc libgmp10-doc libmpfr-dev
The following NEW packages will be installed:
  gcc-14-base
The following packages will be upgraded:
```

1. We create a password protected file in for the start.



2. We convert this zip file into a text file.

```
(kali㉿kali)-[~]  
$ zip2john test.zip > test.txt  
ver 1.0 efh 5455 efh 7875 test.zip/testing.txt PKZIP Encr: 2b chk, TS_chk, cm  
plen=25, decmplen=13, crc=7B069D9A ts=7A71 cs=7a71 type=0
```

3. Using wordlist to crack password

```
(kali㉿kali)-[~]  
$ john --wordlist=rockyou.txt test.txt  
Using default input encoding: UTF-8  
Loaded 1 password hash (PKZIP [32/64])  
No password hashes left to crack (see FAQ)  
  
(kali㉿kali)-[~]  
$ john -show test.txt  
test.zip/testing.txt:password:testing.txt:test.zip::test.zip  
  
1 password hash cracked, 0 left  
  
(kali㉿kali)-[~]  
$
```

3. Demonstrate the use of rulesets to enhance password cracking efficiency.

By applying different changes to password guesses, rule sets in John the Ripper are utilised to increase the efficiency of password cracking. Rulesets are used for the following reasons:

Diversify Password Guesses: By implementing various modifications, such as appending or prepending characters, switching cases, adding numbers or symbols, etc., rulesets enable John the Ripper to provide a greater variety of password guesses. Because of their diversity, passwords that a straightforward dictionary-based assault would not be able to break are more likely to be cracked.

Cover popular Password Patterns: Rulesets may be made to specifically target popular password patterns and behaviours, such the usage of keyboard patterns (like "qwerty"), predictable replacements (like "P@ssw0rd"), or common additions of numbers and symbols (like "password123!").

Optimise Search Space: By giving priority to password guesses that are more likely to correspond with popular password generating practices, rulesets aid in the optimisation of the search space. Rulesets direct the cracking process towards

more likely guesses rather than exhaustively testing every potential combination, which cuts down on the amount of time and processing power needed for cracking.

Flexibility and Customisation: Users are able to design unique rule sets that are suited to their own requirements and intended contexts. This increases the overall success rate of the cracking effort by enabling them to modify the cracking technique to the features of the password hashes being targeted.

Boost Efficiency: Compared to brute-force or dictionary assaults alone, John the Ripper can break passwords more quickly when he uses rulesets. By using criteria specifically to decrease the number of guesses required to locate a successful password match, the cracking process may be accelerated considerably.

4. Analyze the cracked passwords and identify any patterns or weaknesses.

Common Words or Phrases: Many users choose passwords based on common words or phrases. Analyzing cracked passwords may reveal frequently used terms, such as "password," "123456," or "admin."

Dictionary Words with Substitutions: Users often modify dictionary words by substituting letters with similar-looking characters or symbols. For example, "password" might be written as "p@ssw0rd" or "pa\$\$w0rd."

Keyboard Patterns: Some users create passwords based on keyboard patterns, such as "qwerty," "asdf," or "zxcvbn."

Sequential Characters: Cracked passwords may contain sequential characters, such as "123456" or "abcdef."

Personal Information: Users often use easily discoverable personal information in their passwords, such as their name, username, or company name.

Common Patterns and Sequences: Cracked passwords may follow common patterns or sequences, such as "abcd1234" or "password123."

5. Perform a brute-force attack on a password-protected file with a simple password.

```
(kali@kali)-[~]
$ john --incremental test2.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:06:20 0g/s 20664Kp/s 20664Kc/s 20664KC/s mcmackr..mcenke8
ABC123 (test2.zip/testing.txt)
1g 0:00:06:25 DONE (2024-04-12 17:06) 0.002592g/s 20727Kp/s 20727Kc/s 20727KC
/s AS!5mw..ABCLF4
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

(kali@kali)-[~]
$ john --show test2.txt
test2.zip/testing.txt:ABC123:testing.txt:test2.zip::test2.zip

1 password hash cracked, 0 left
```

6. Compare the effectiveness of dictionary and brute-force attacks.

Speed: Dictionary attacks are generally faster than brute-force attacks due to their targeted nature, making them more practical for cracking passwords in real-world scenarios.

Success Rate: Dictionary attacks are highly effective against passwords based on common words or phrases, while brute-force attacks excel at cracking passwords with high complexity requirements.

Resource Requirements: Dictionary attacks require fewer computational resources compared to brute-force attacks, making them more accessible for attackers with limited resources.

Coverage: Brute-force attacks provide comprehensive coverage of the search space but are less efficient when dealing with passwords that can be cracked using dictionary-based methods.

7. Explain the significance of password attacks in the context of cybersecurity.

Password assaults are crucial in the context of cybersecurity because they are a prevalent and possibly catastrophic type of cyber threat. A password attack is when someone tries to guess or find out a user's password in order to access a device or account without authorisation. Such an assault may result in sensitive information theft, data breaches, and unauthorised access to vital systems, all of which might have detrimental effects on people and businesses.

It's crucial to create and maintain passwords according to standard practices in order to defend against password assaults. This include utilising multi-factor authentication whenever it is practical, staying away from popular terms and phrases, and creating strong, one-of-a-kind passwords for every account.

In order to stop password assaults and other cyberthreats, companies should also install security measures like firewalls, antivirus software, and patch management.