Penetration Testing and Red Teaming

Name: Sharvari Dubey

Group: A

Topic: Syngsong-Week 3 Assignment

Tutelr.

Index

Introduction - Red Teaming & Penetration Testing.

Red teaming and penetration testing (pen testing) are two critical practices in the field of cybersecurity used to assess the security posture of an organization or system. Both methods aim to identify vulnerabilities and weaknesses that malicious attackers could exploit but have different approaches and goals.

Red Teaming:

Red teaming is a comprehensive and strategic cybersecurity assessment that simulates real-world attack scenarios. It involves a team of skilled cybersecurity professionals, known as the "red team," who act as aggressive adversaries attempting to breach an organization's defenses. The red team's objective is to challenge and test the organization's security measures, policies, and preparedness.

Penetration Testing (Pentesting):

Pentesting, also known as a "penetration test" or "ethical hacking," is a targeted and tactical approach to evaluating the security of specific systems, networks, or applications. It involves authorized security experts, called "pen testers," who attempt to exploit vulnerabilities in a controlled environment. The primary goal of pen testing is to identify and fix security weaknesses before malicious hackers can exploit them.

Introduction - Syngsong

Synsong is a password-cracking tool. A password cracking tool is a cybersecurity software or script designed to guess or recover passwords used to secure user accounts, systems, or applications. These tools are used by security professionals and ethical hackers to assess the strength of passwords and identify weak or compromised credentials. However, they can also be misused by malicious actors for unauthorized access to sensitive information.

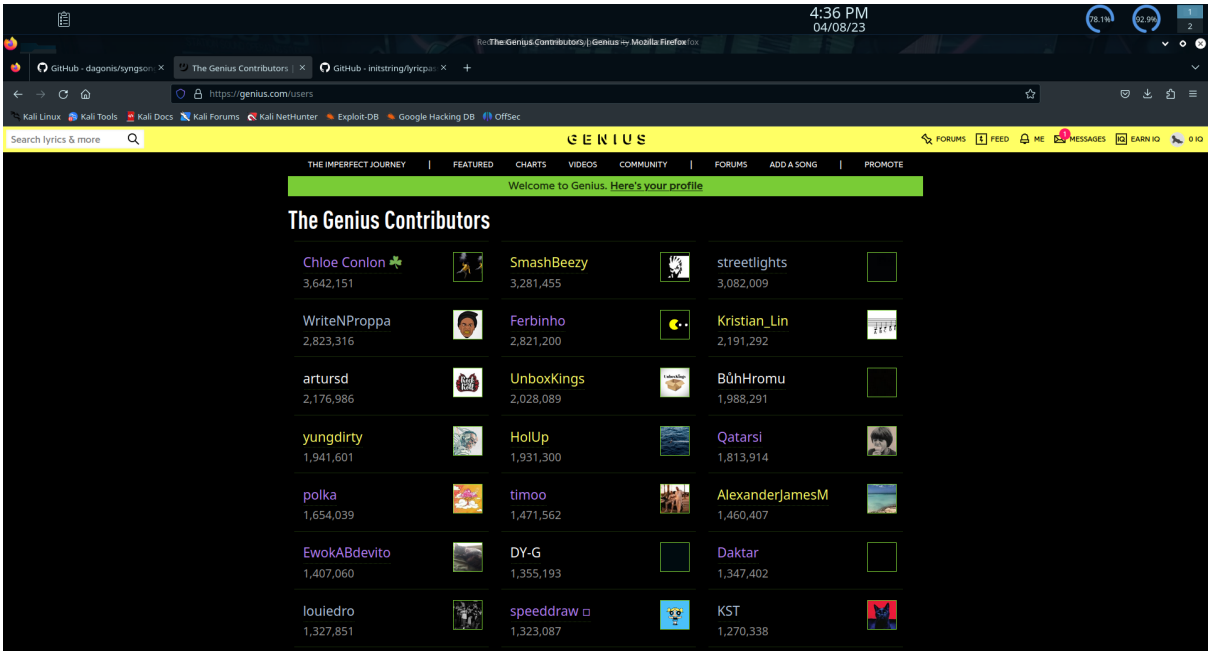
The two primary types of password-cracking techniques are:

1) Brute Force Attacks
2) Dictionary Attacks/Wordlist Attacks

Synsong is a tool that creates a password list for cracking lyrical passwords. It basically, generates a password list based on the lyrics of the song. So, we can say that it uses Dictionary Attack.

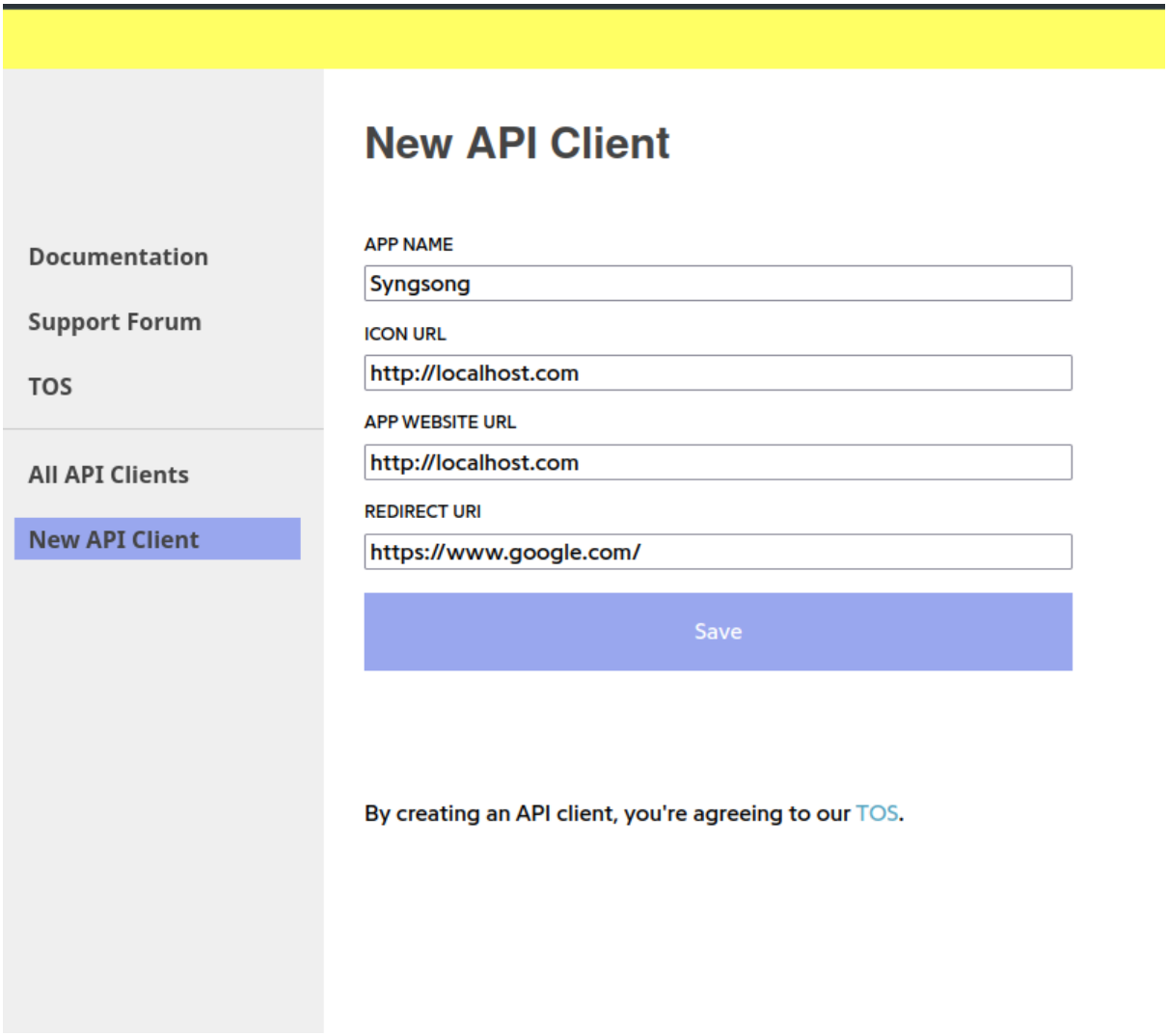
Installation Process

1) Go to its official GitHub page to install it - https://github.com/dagonis/syngsong
2) To generate password list lyrics, we will use the genius.com API. In order to obtain the API key, we must have an account. For an API Key, you need to create an account or login at https://genius.com/signup_or_login
3) The following page should be visible after you have logged in:



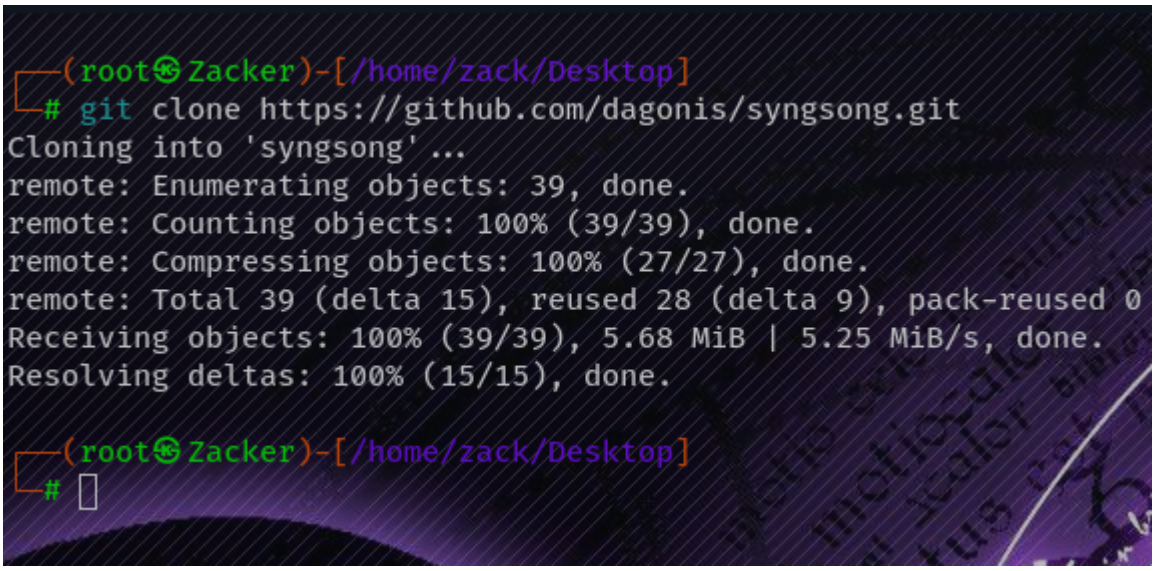
4) Now go to - https://genius.com/api-clients

5) And fill in the details and click on save to generate an access token.



6) After generating the access token copy and save the creds in your text editor for future use.

7) Now open your terminal. Clone the git repo at the desired location in your system.



8) Navigate to the newly cloned directory.

9) For installing requirements run the following command:

```
python3 -m pip install -r requirements.txt
```

10) Within the directory, you will find another folder called "syngsong".

11) Give the following command to make the file executable:

```
chmod 777 __main__.py
```

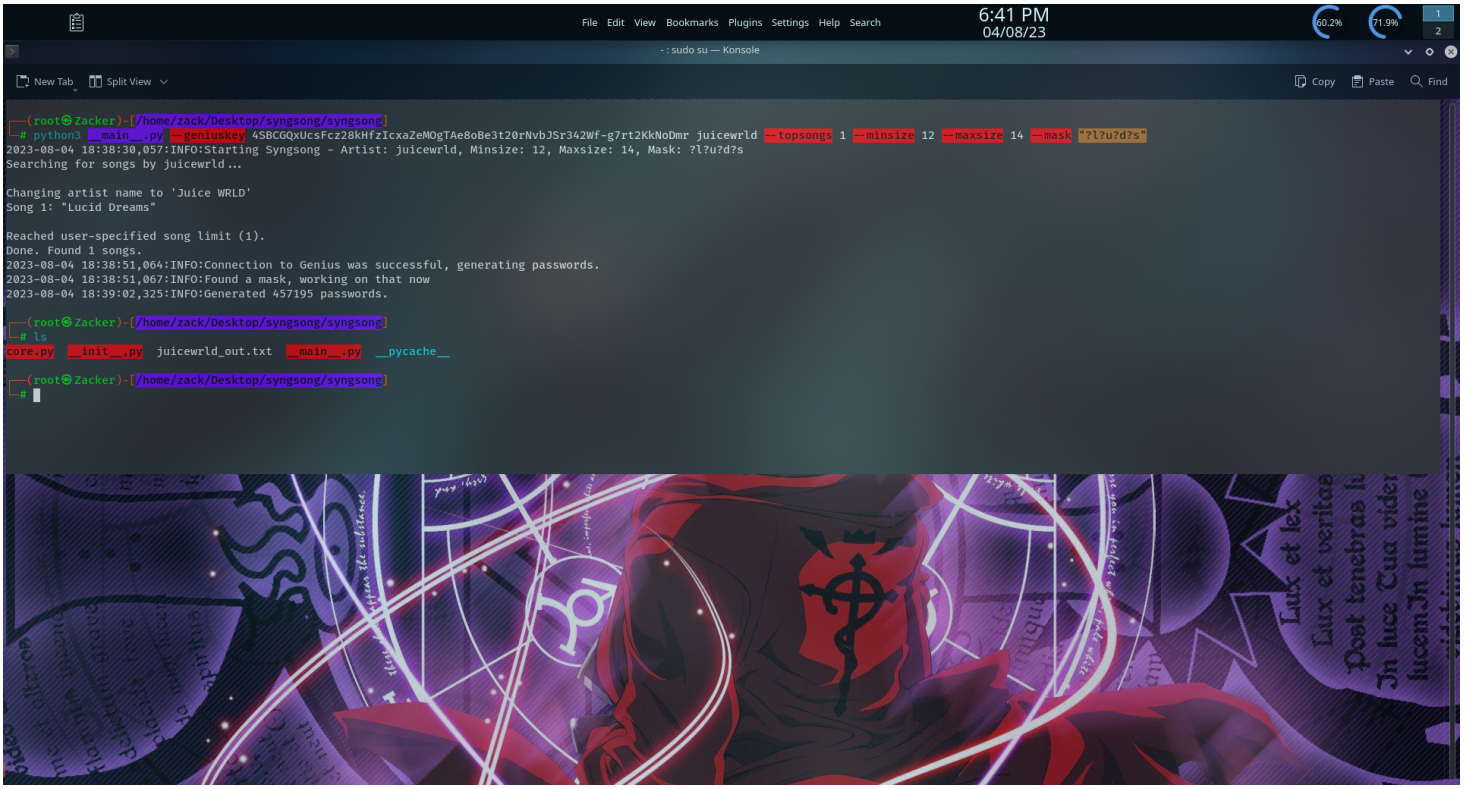
12) Now enter the following command:

```
python3 __main__.py --geniuskey {key} juicewrld --topsongs 1
--minsize 12 --maxsize 14 --mask "?l?u?d?s"
```

Note: Instead of key paste the access token that you have generated through Genius.

13) This command generates a password list based on the lyrics of the song by Juice Wrld that is trending on the internet. This will create a text file containing all combinations of passwords.

Note: It's Juice Wrld in my case because I gave the name of my favorite artist you can enter any name of the artist that you like to generate the list.



14) Now we will save our password in encrypted format in a file and try to crack it with the wordlist that we have just generated.

15) Let's suppose we will use "ManWhatHahaiB9" as a password. We will use md5 hash to encrypt it with the following command:

```
echo –n ManWhatHahaiB9" | md5sum | tr –d "–"››password
```

The above command will save our password in encrypted form in a file named "password".

Hashcat

Hashcat is a widely used and powerful open-source password-cracking tool in the field of cybersecurity. It is designed to perform advanced password recovery and hash-cracking attacks, helping security professionals, penetration testers, and ethical hackers assess the strength of passwords and improve overall security.

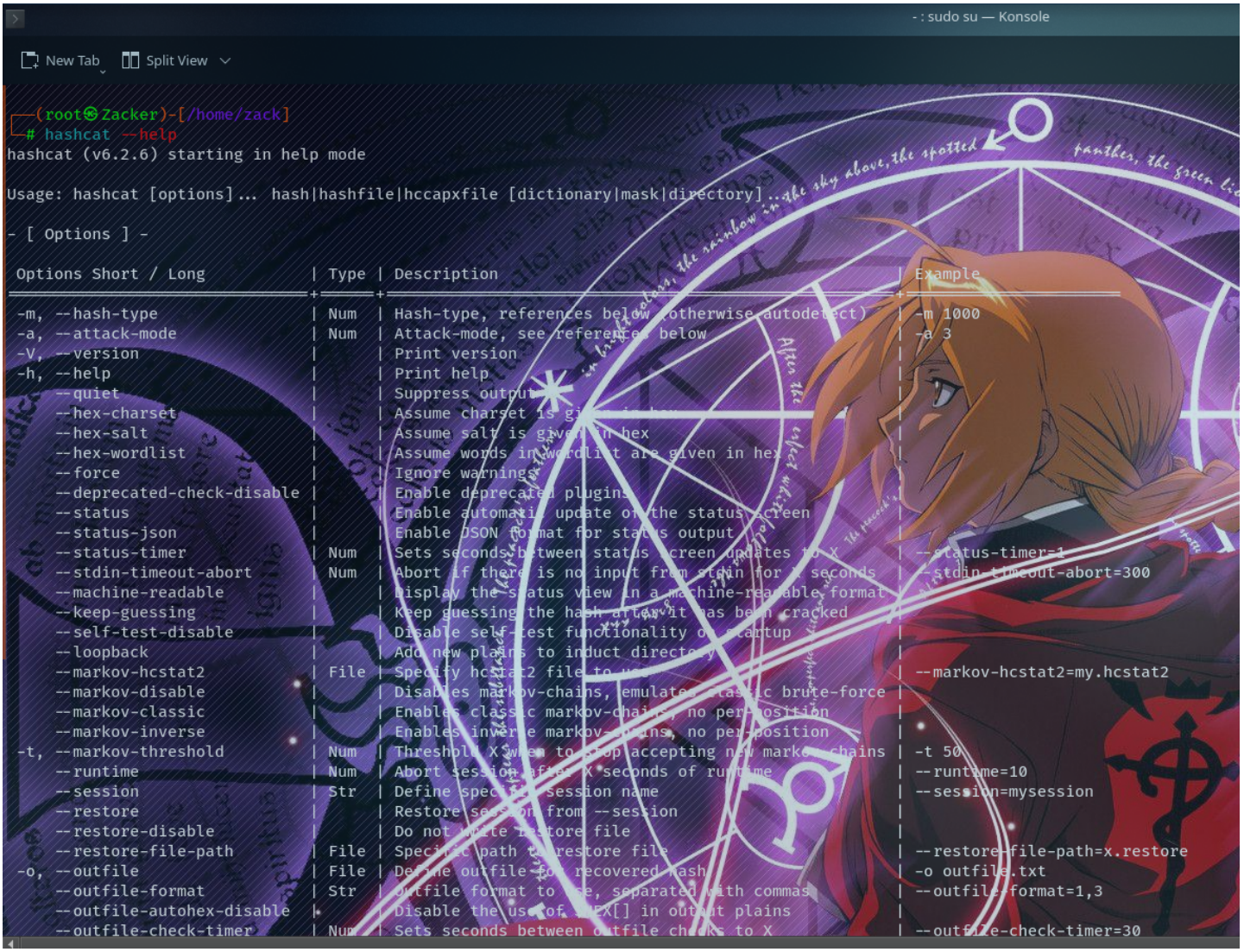
Key features of Hashcat include:

- ➔ Hash Cracking
- ➔ Performance Optimization
- ➔ Rule-Based Attacks
- ➔ Wordlist Generation
- ➔ Distributed Computing
- ➔ Comprehensive Platform Support

We are going to use this tool and try to crack the password that we stored in encrypted form.

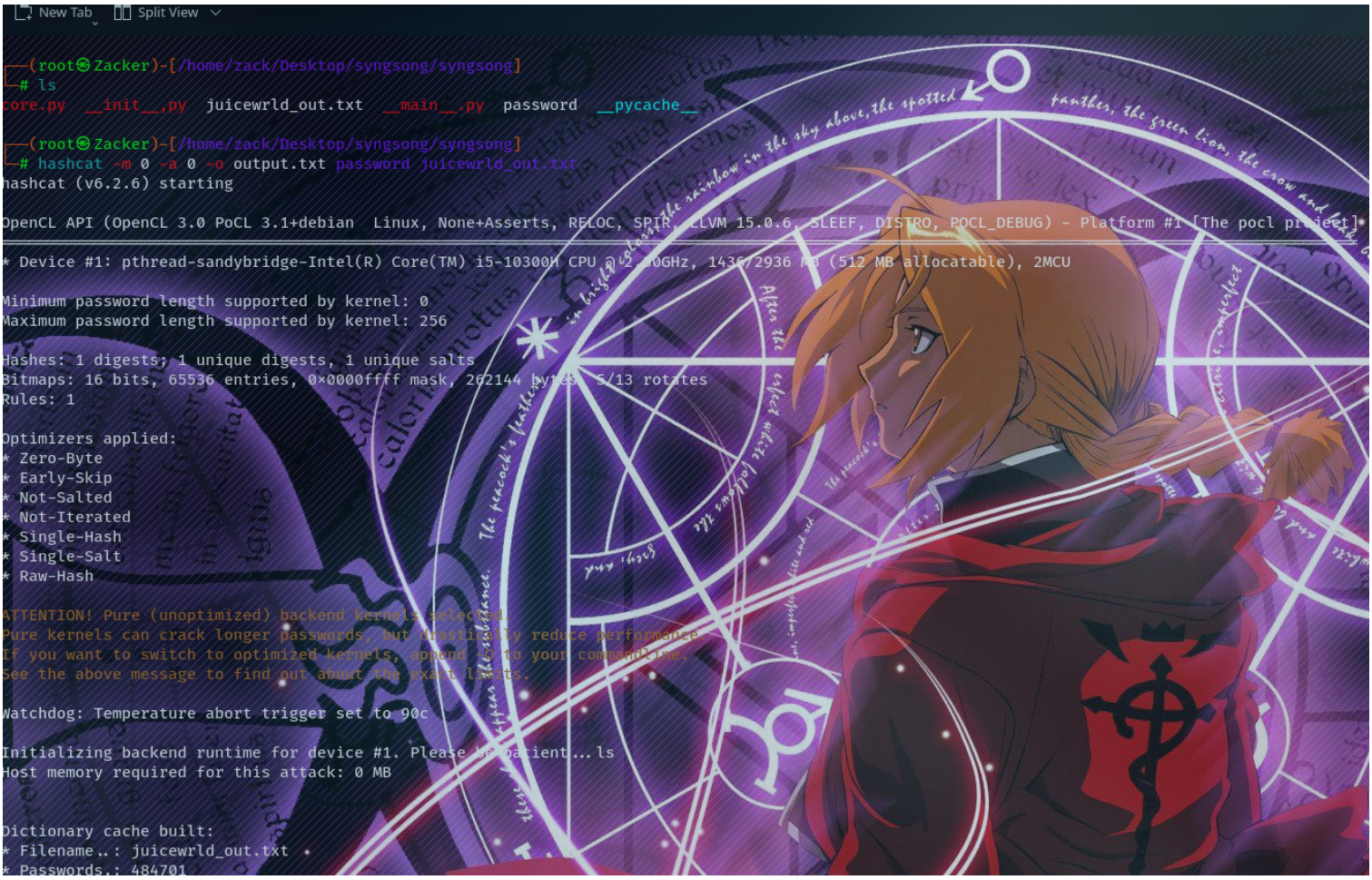
1) In Kali Linux hashcat comes preinstalled. I'll just run the command - `hashcat –help`

and the following interface will show up:

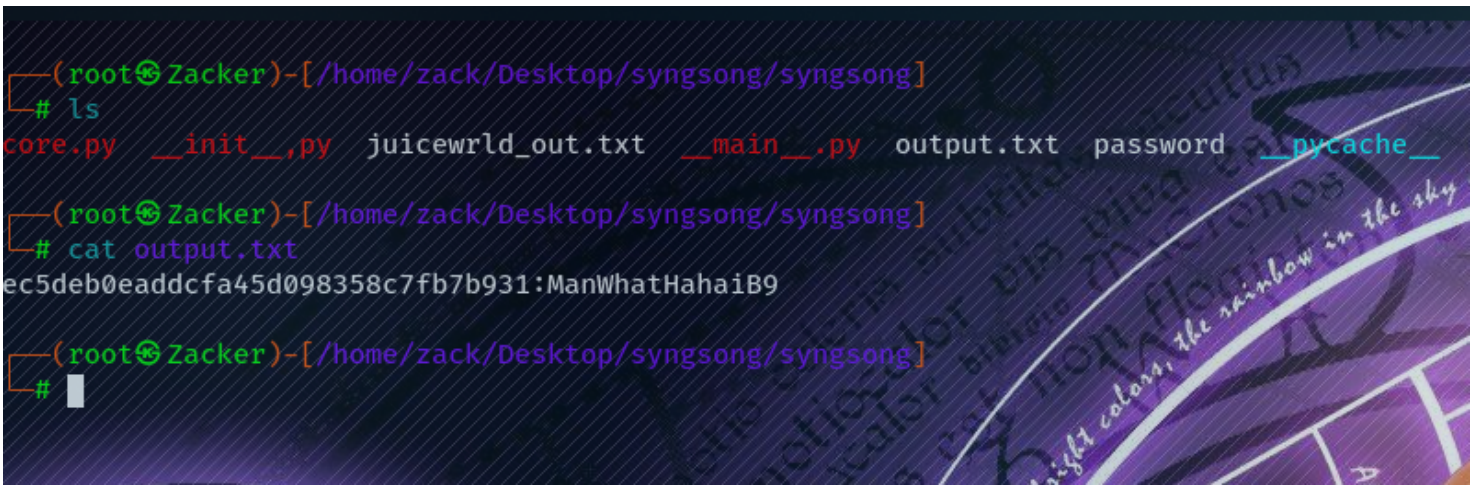


2) Our next step will be to run the command to use hashcat to crack the password with a wordlist named “juicewrld_out.txt” and save the password in a file called “output.txt”. The command is:

```
hashcat -m 0 -a 0 -o output.txt password juicewrld_out.txt
```



3) It has shown our output which is our password.



Password Analysis Tool - zxcvbn

Password complexity analysis tools, such as zxcvbn (pronounced "zxcvbn" like the first few keys on a keyboard), are designed to evaluate the strength and complexity of passwords. These tools use algorithms to estimate how resistant a password is to various types of attacks, such as brute force attacks, dictionary attacks, and more.

One of the popular password complexity analysis tools is zxcvbn, which was developed by Dropbox. It is available as an open-source library and has been integrated into various applications and websites to help users create stronger passwords. Here's how zxcvbn works:

- ➔ Scoring System: zxcvbn assigns a score to each password based on its estimated strength. The score is usually represented as a number or a rating, typically from 0 to 4, with 0 being very weak and 4 being very strong.

- ➔ Dictionary Checks: zxcvbn checks if the password appears in common password dictionaries, which contain a list of commonly used and easily guessable passwords. Passwords found in such dictionaries receive lower scores as they are more susceptible to dictionary attacks.

- ➔ Pattern Matching: zxcvbn identifies patterns and sequences in the password that are easy to guess. For example, passwords like "123456" or "qwerty" have predictable patterns and would receive low scores.

- ➔ Entropy Calculation: zxcvbn estimates the password's entropy, which is a measure of the password's randomness and unpredictability. Passwords with higher entropy are generally stronger.

- ➔ Feedback and Suggestions: zxcvbn provides feedback to users about the strength of their passwords and offers suggestions on how to improve them. This feedback can include tips to avoid common patterns, use a mix of character types, and increase password length.

By using zxcvbn or similar password complexity analysis tools, users can make more informed decisions when creating passwords. The goal is to encourage the use of stronger and more secure passwords that are resistant to various types of attacks.

Now we will analyze the strength of our password through this tool.

1) Open the terminal and install the zxcvbn tool if it is not installed: `pip install zxcvbn`
2) Type "zxcvbn" in the terminal and then type the password in our case it is "ManWhatHahaiB9" and press enter, following output would be showcased:


"guesses_log10": 1.6989700043360185
},
{
"pattern": "bruteforce",
"token": "HahaiB9",
"i": 7,
"j": 13,
"guesses": 10000000,
"guesses_log10": 7.0
}
],
"calc_time": "0:00:00.005496",
"crack_times_seconds": {
"online_throttling_100_per_hour": "1901160000000000.0105535580275"
"online_no_throttling_10_per_second": "52810000000",
"offline_slow_hashing_1e4_per_second": "52810000",
"offline_fast_hashing_1e10_per_second": "52.81"
},
"crack_times_display": {
"online_throttling_100_per_hour": "centuries",
"online_no_throttling_10_per_second": "centuries",
"offline_slow_hashing_1e4_per_second": "2 years",
"offline_fast_hashing_1e10_per_second": "53 seconds"
},
"score": 4,
"feedback": {
"warning": "",
"suggestions": []
}
}
(root㉿Zacker)-[/home/Zack]
#

Conclusion

From the above screenshot we can conclude:

- A password with a score of 4, as shown by password complexity analysis tools like zxcvbn, is considered a strong password. In most cases, a score of 4 indicates that the password is highly resistant to various types of attacks and is considered secure.

- Password complexity analysis tools typically use a scoring system to help users understand the strength of their passwords. The scoring can vary slightly depending on the tool, but generally, a score of 4 is the highest possible score and represents a password that is:

- Long: The password is sufficiently long, typically 12 characters or more, which increases its resistance to brute force attacks.

- Random and Unpredictable: The password contains a mix of character types, such as lowercase letters, uppercase letters, numbers, and special characters, making it difficult to guess.

- Not Found in Dictionaries: The password does not appear in common password dictionaries, which means it is not a common or easily guessable password.

- Not Following Predictable Patterns: The password does not follow common patterns (e.g., sequential numbers or keyboard patterns) that can be easily guessed.

- A password with a score of 4 is generally considered secure and suitable for protecting sensitive accounts and data. However, it's essential to remember that no password is completely invulnerable, and security best practices recommend using other measures, such as multi-factor authentication (MFA), to further enhance security.

- While a strong password is a critical component of cybersecurity, it's also essential to follow other security guidelines, such as avoiding password reuse across multiple accounts, updating passwords regularly, and using a password manager to securely store and manage passwords.

Assessment Answers

1. What is Syngsong?

A. A tool to create song lyrics

B. A tool to generate password guesses based on song lyrics via the Genius API

C. A tool to download songs from the internet

D. A tool to remix songs

2. How does Syngsong generate password guesses?

A. By randomly guessing words from song lyrics

B. By using a dictionary attack

C. By using the Genius API to extract lyrics and then applying Hashcat style masking

D. By brute-forcing the password

3. What is Hashcat-style masking?

A. A technique used to hash passwords

B. A technique used to generate random passwords

C. A technique used to mask certain parts of a password

D. A technique used to crack passwords

4. Can Syngsong generate passphrases that meet password complexity requirements?

A. Yes

B. No

5. Which API does Syngsong use to extract song lyrics?

A. Spotify API

B. Apple Music API

C. Genius API

D. Soundcloud API

6. How can Syngsong be useful for password cracking?

A. By generating a list of potential passwords based on song lyrics

B. By brute forcing passwords

C. By using social engineering techniques

D. By hacking into a system

7. What are some password complexity requirements that Syngsong can generate?

A. Length requirements

B. Character set requirements

C. Combination of uppercase and lowercase letters, numbers, and symbols

D. All of the above

8. Is Syngsong a legal tool?

A. Yes

B. No

9. How can Syngsong be used ethically?

A. To test the strength of one's own passwords

B. To test the strength of passwords for others with their consent

C. To crack passwords without authorization

D. To steal personal information

10. What are some potential risks associated with using Syngsong?

A. It can generate weak passwords

B. It can violate the terms of service of the Genius API

C. It can be used for illegal activities

D. All of the above