MIT World Peace University

Security Management and Cyber Laws
Third Year B. Tech, Semester 5

Tools and Techniques used in Ethical Hacking

SMCL Graded Tutorial for Unit 5
John the Ripper

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 10

November 28, 2023

Contents

1 Aim

To study and use John the Ripper to crack passwords.

2 Objectives

1. To study the different features of John the Ripper.
2. To study the different ways to crack passwords.
3. To study the need of John the Ripper.

3 What is John the Ripper?

John the Ripper is a widely used open-source password cracking tool. It is designed to identify weak passwords by employing various password cracking techniques, including dictionary attacks, brute force attacks, and more. John the Ripper is highly flexible and supports multiple hash algorithms.

3.1 Features

John the Ripper offers the following key features:

- Support for various password hash algorithms such as DES, MD5, SHA-1, SHA-256, etc.
- Configurability for different attack modes and methods.
- Compatibility with different platforms, including Unix, Windows, and more.
- Multiple attack modes, including dictionary attacks, brute force attacks, and hybrid attacks.
- Availability of community-contributed patches and enhancements.

3.2 Ways to Crack Passwords

1. **Dictionary Attack:** Uses a wordlist to try potential passwords.
2. **Brute Force Attack:** Tries all possible combinations of characters systematically.
3. **Hybrid Attack:** Combines dictionary and brute force attacks for increased effectiveness.
4. **Rainbow Table Attack:** Uses precomputed tables of hashes for quick password lookup.
5. **Rule-based Attack:** Applies a set of rules to generate passwords.
6. **Mask Attack:** Uses a predefined mask to generate passwords.
7. **Permutation Attack:** Generates passwords using permutations of characters.
8. **Markov Attack:** Uses Markov chains for intelligent password guessing.
9. **PRINCE Attack:** Utilizes a PRINCE algorithm for password generation.

3.3 Need of John the Ripper

John the Ripper is crucial for:

- Assessing password security by identifying weak or easily guessable passwords.
- Conducting security audits to uncover vulnerabilities in password policies.
- Educating users about the importance of strong and secure passwords.
- Strengthening overall cybersecurity by proactively addressing password weaknesses.

3.4 Installation

To install John the Ripper on your system, follow these steps:

1. Download the latest version from the official website.
2. Extract the downloaded archive.
3. Navigate to the extracted directory.
4. Run the installation command appropriate for your platform.

4 Usage

We will now explore different methods to crack passwords using John the Ripper.

4.1 Dictionary Attack

4.1.1 JTR Command

```
john --wordlist=dictionary.txt hash_file
```

4.1.2 Pros

1. Fast and efficient for passwords based on dictionary words.
2. Utilizes a predefined list of potential passwords.
3. Suitable for users who might use common words as passwords.

4.1.3 Cons

1. Less effective against passwords with complex structures or variations.
2. Limited by the contents of the chosen wordlist.
3. Might not be successful against passwords with additional characters or patterns.

4.1.4 Steps

1. Replace `hash_file` with the actual name of your password hash file.
2. Ensure that `dictionary.txt` contains a comprehensive wordlist.
3. Run the provided JTR command in your terminal.

4.1.5 Screenshots

4.2 Brute Force Attack

4.2.1 JTR Command

```
john --incremental hash_file
```

4.2.2 Pros

1. Exhaustive and covers all possible combinations.
2. Guarantees success given enough time and resources.

4.2.3 Cons

1. Time-consuming, especially for complex passwords.
2. Resource-intensive, requires substantial computing power.

4.2.4 Steps

1. Replace `hash_file` with the actual name of your password hash file.
2. Execute the provided JTR command in your terminal.

4.2.5 Screenshots

4.3 Hybrid Attack

4.3.1 JTR Command

```
john --incremental --wordlist=dictionary.txt hash_file
```

4.3.2 Pros

1. Combines the thoroughness of brute force with the efficiency of a wordlist.
2. More time-effective than a pure brute force approach.

4.3.3 Cons

1. Success highly depends on the quality of the wordlist.
2. May not be as fast as dictionary attacks alone.

4.3.4 Steps

1. Replace `hash_file` with the actual name of your password hash file.
2. Provide a comprehensive wordlist in `dictionary.txt`.
3. Run the specified JTR command.

4.3.5 Screenshots

4.4 Rainbow Table Attack

4.4.1 JTR Command

```
john --format=raw-md5 --external=rainbow_table.rt hash_file
```

4.4.2 Pros

1. Rapid lookup for precomputed hashes.
2. Efficient for commonly used passwords.

4.4.3 Cons

1. Limited to precomputed tables.
2. Requires significant storage for extensive tables.

4.4.4 Steps

1. Replace `hash_file` with the actual name of your password hash file.
2. Use an appropriate precomputed rainbow table (`rainbow_table.rt`).
3. Execute the provided JTR command.

4.4.5 Screenshots

4.5 Rule-based Attack

4.5.1 JTR Command

```
john --wordlist=dictionary.txt --rules hash_file
```

4.5.2 Pros

1. Dynamically applies rules for password generation.
2. Efficient for complex password structures.

4.5.3 Cons

1. Success depends on the effectiveness of the rule set.
2. May require fine-tuning for optimal results.

4.5.4 Steps

1. Replace `hash_file` with the actual name of your password hash file.
2. Customize and provide rules for password generation.
3. Run the specified JTR command.

4.5.5 Screenshots

4.6 Mask Attack

4.6.1 JTR Command

```
john --mask=?l?d?u hash_file
```

4.6.2 Pros

1. Allows customization based on known patterns.
2. Useful for structured password cracking.

4.6.3 Cons

1. Resource-intensive for complex mask patterns.
2. Success depends on accurately defining the password structure.

4.6.4 Steps

1. Replace `hash_file` with the actual name of your password hash file.
2. Adjust the mask pattern (`?l`, `?d`, `?u` for lowercase, digit, uppercase) based on the expected password structure.
3. Execute the specified JTR command.

4.6.5 Screenshots

4.7 Permutation Attack

4.7.1 JTR Command

```
john --incremental=perm hash_file
```

4.7.2 Pros

1. Comprehensive by trying all permutations of characters.
2. Useful for cases where the password structure is not well-defined.

4.7.3 Cons

1. Highly resource-intensive and time-consuming.
2. Success depends on the length and complexity of the password.

4.7.4 Steps

1. Replace `hash_file` with the actual name of your password hash file.
2. Execute the specified JTR command.

4.7.5 Screenshots

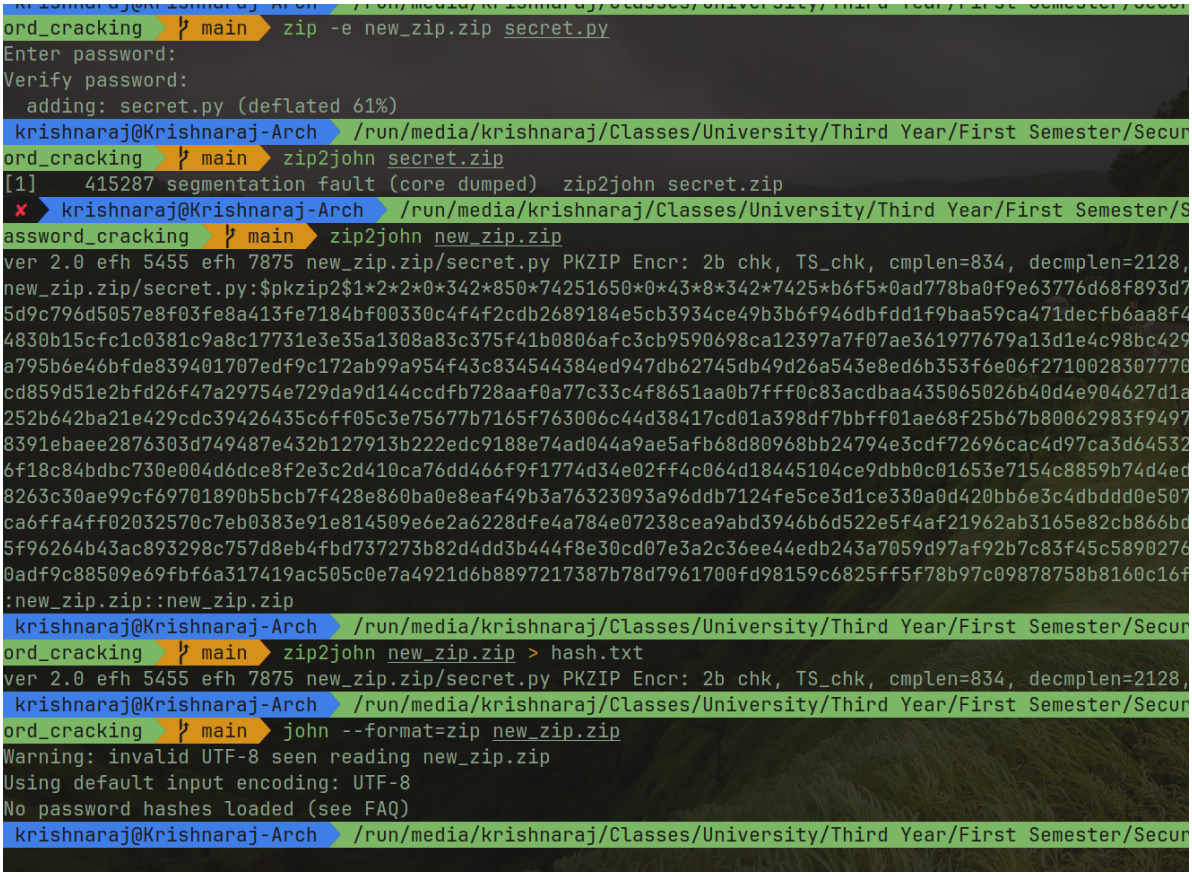

Figure 1: Making a Zip password

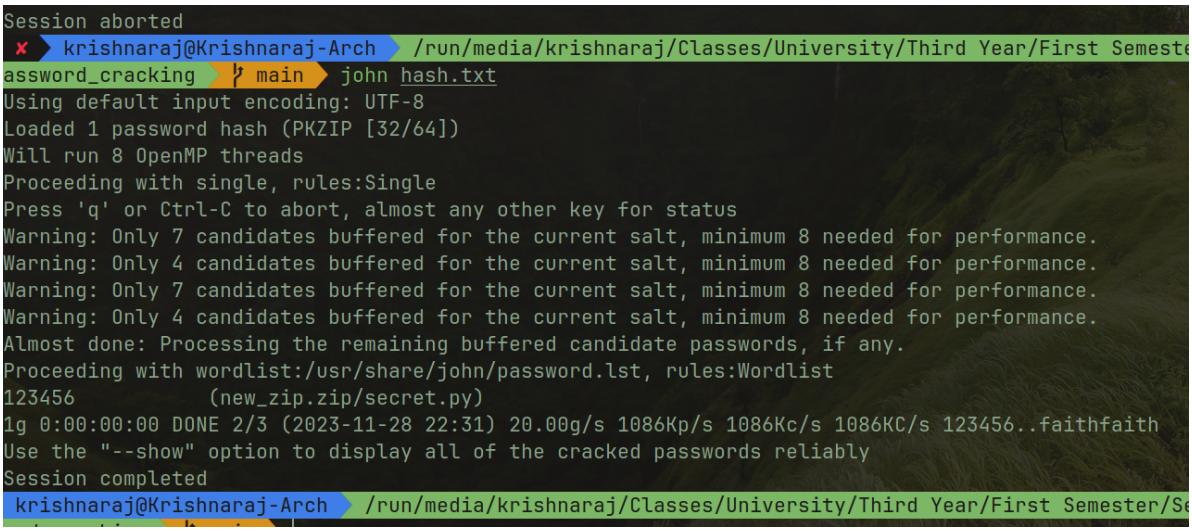

Figure 2: Crackign a Zip password with JTR

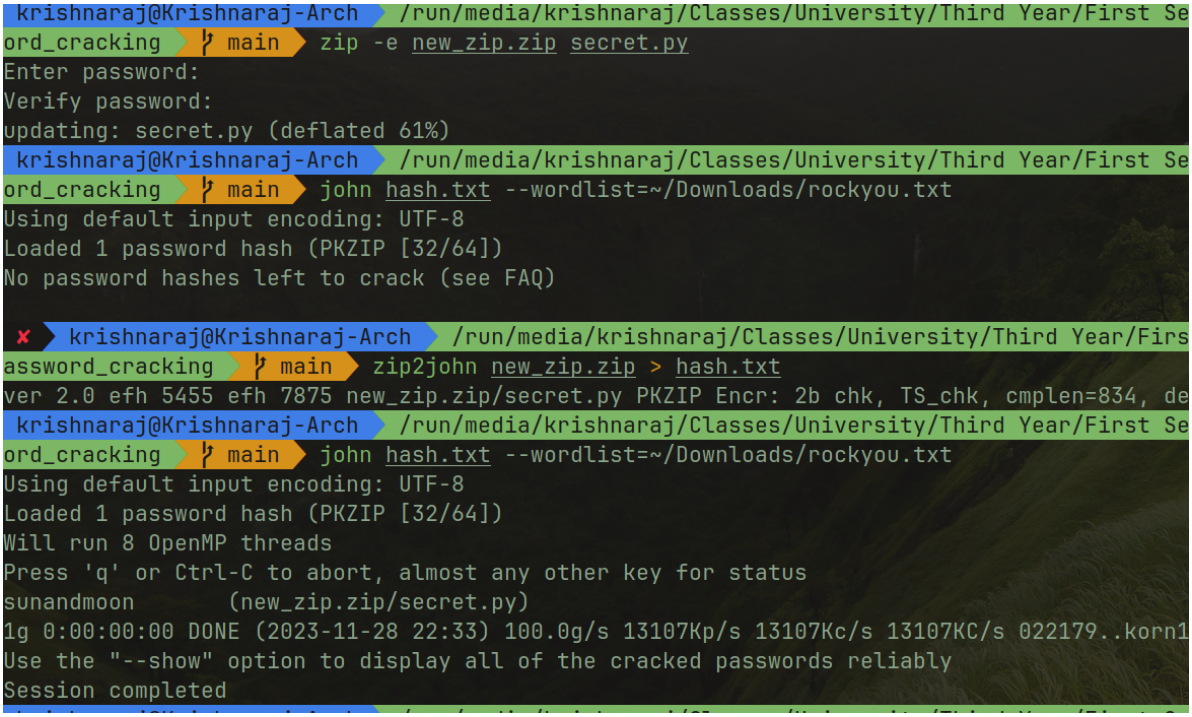

Figure 3: New Zip Password Cracked with Dictionary Attack

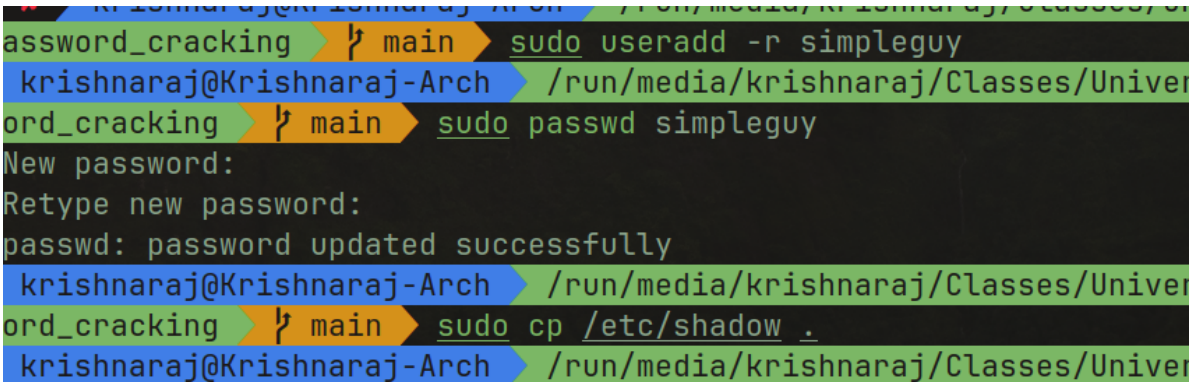

Figure 4: Making a new user with a simple password

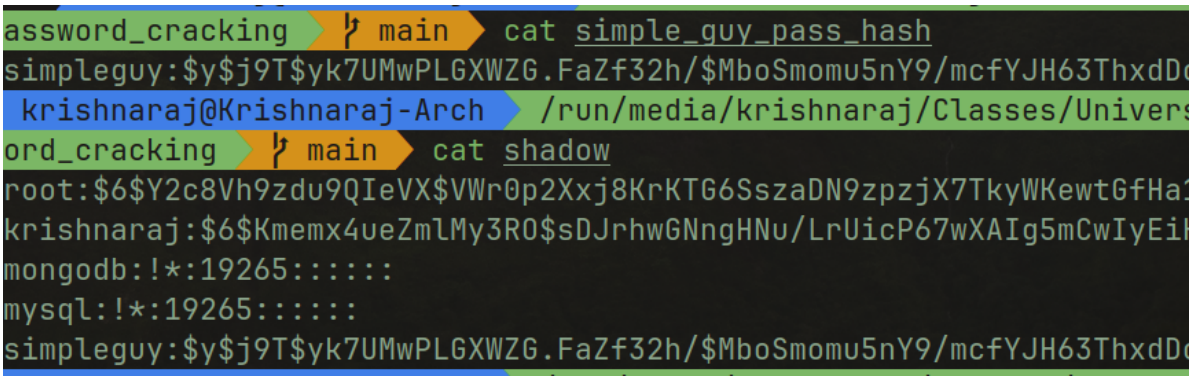

Figure 5: Getting the shadow file.

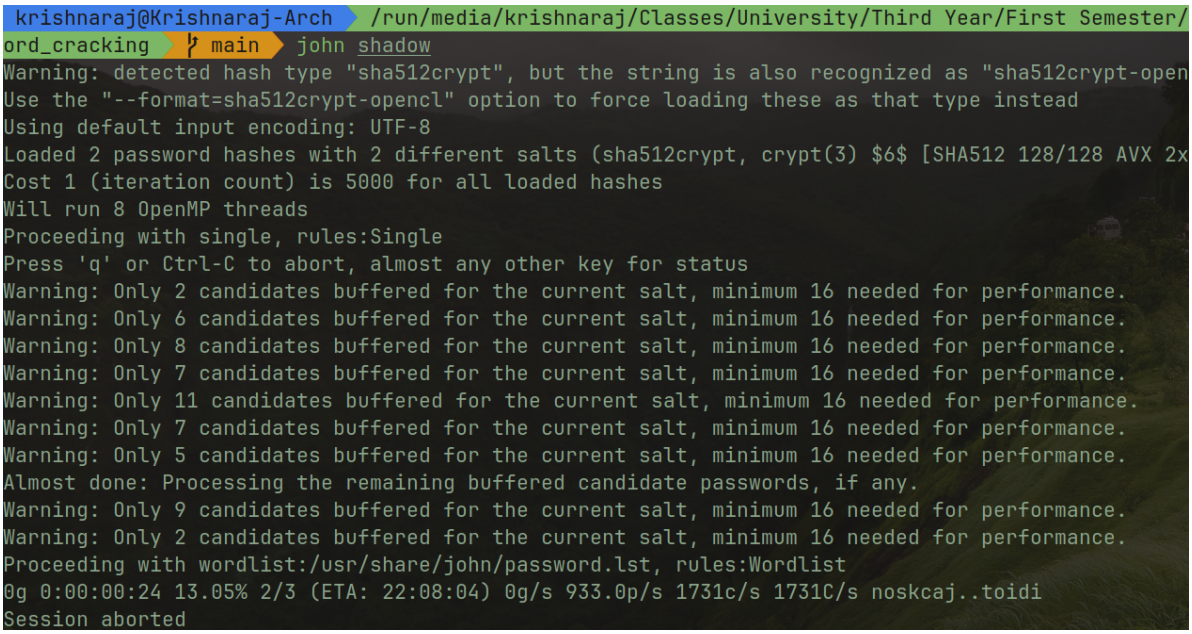

Figure 6: Trying to crack the file with Jtr

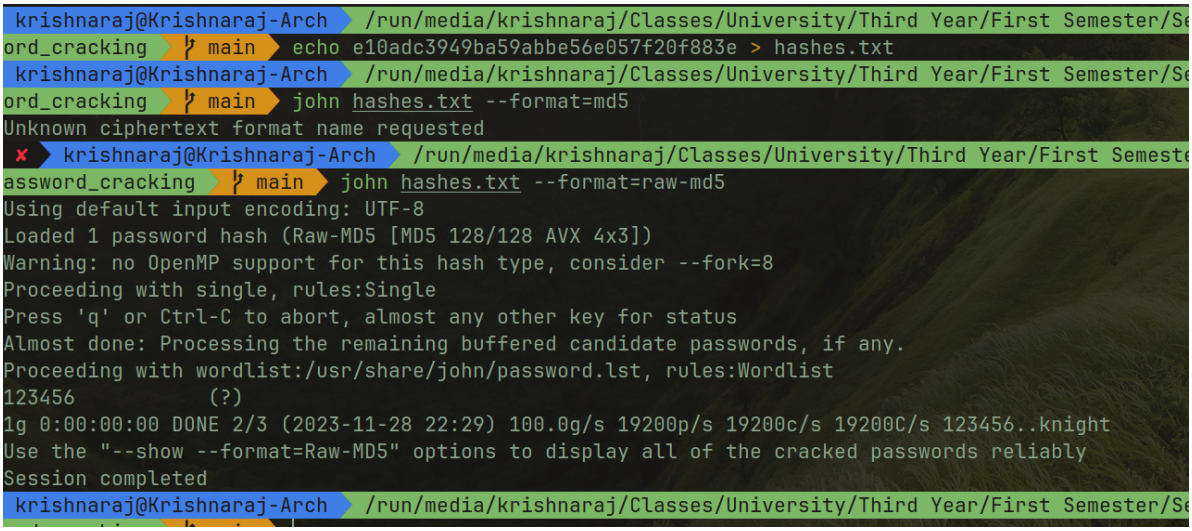

Figure 7: Cracking MD5 Hashes with JTR

5 Code

```python
import hashlib


def hash_password(password):
    # Hash the password using SHA-1
    hashed_password = hashlib.sha1(password.encode()).hexdigest()
    return hashed_password


# List of common passwords (for educational purposes only)
common_passwords = [
    "password",
    "123456",
    "qwerty",
    "letmein",
    "admin",
    "1234",
    "welcome",
    "tigger",
    "sunshine",
    "chocolate",
    "password1",
    "soccer",
    "anthony",
    "friends",
    "butterfly",
    "purple",
    "shadow",
    "melissa",
    "eminem",
    "matthew",
    "robert",
    "danielle",
    "forever",
    "family",
```

```
    "jonathan",
    "987654321",
    "computer",
    "whatever",
    "dragon",
    "vanessa",
    "cookie",
    "naruto",
    "summer",
    "sweety",
    "spongebob",
    "joseph",
    "junior",
    "softball",
    "taylor",
    "yellow",
    "daniela",
    "lauren",
    "mickey",
    "princesa",
    "alexandra",
    "alexis",
    "estrella",
    "miguel",
    "william",
    "thomas",
    "beautiful",
    "mylove",
    "angela",
    "poohbear",
    "patrick",
    "iloveme",
    "sakura",
    "adrian",
    "alexander",
    "destiny",
    "christian",
    "121212",
    "sayang",
    "america",
    "dancer",
    "monica",
    "richard",
    "112233",
    "princess1",
    "555555",
    "diamond",
    "carolina",
    "steven",
    "rangers",
    "louise",
    "orange",
    "789456",
    "999999",
    "shorty",
    "nathan",
    "snoopy",
    "gabriel",
    "hunter",
```

```
    "cherry",
    "killer",
    "sandra",
    "alejandro",
    "buster",
    "george",
    "brittany",
    "alejandra",
    "patricia",
    "rachel",
    "tequiero",
    "7777777",
    "cheese",
    "159753",
    "arsenal",
    "dolphin",
    "antonio",
    "heather",
    "ginger",
]

# Create a text file to store hashed passwords
with open("hashed_passwords.txt", "w") as file:
    for password in common_passwords:
        # Hash each password and write it to the file
        hashed_password = hash_password(password)
        file.write(f"{password}: {hashed_password}\n")

print("Hashed passwords are stored in 'hashed_passwords.txt'")
```

Listing 1: Script to make hashes of passwords.

6 Platform

Operating System: Arch Linux x86-64
IDEs or Text Editors Used: Visual Studio Code
Compilers or Interpreters: Python 3.10.1

7 Conclusion

Thus, we have successfully installed and used John the Ripper to crack passwords, and also learnt about the need of John the Ripper.

References

[1] John the Ripper.
Website: https://www.openwall.com/john/

[2] Password Cracking Techniques.
Website: https://www.cybersecurity-insiders.com/password-cracking-techniques/

[3] Dictionary Attack.
Website: https://www.techopedia.com/definition/1779/dictionary-attack

[4] Brute Force Attack.
Website: https://www.imperva.com/learn/application-security/brute-force-attack/

[5] Hybrid Attack.
Website: https://www.techopedia.com/definition/1779/dictionary-attack

[6] Rainbow Table Attack.
Website: https://www.techopedia.com/definition/1779/dictionary-attack

[7] Rule-based Attack.
Website: https://www.techopedia.com/definition/1779/dictionary-attack

[8] Mask Attack.
Website: https://www.techopedia.com/definition/1779/dictionary-attack

[9] Permutation Attack.
Website: https://www.techopedia.com/definition/1779/dictionary-attack

[10] MIT World Peace University.
Website: https://mitwpu.edu.in/

[11] Security Management and Cyber Laws.
Website: https://mitwpu.edu.in/course/b-tech-cyber-security/