IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, April 2025
Impact Factor: 7.67

Non-Hashed Passwords Cracking

Chakradhar Mhaske, Rohit Khade, Piyush Bhojane, Prof. Harihar
Navsahyadri Education Society's Group of Institutions, Polytechnic, Pune, Maharashtra, India

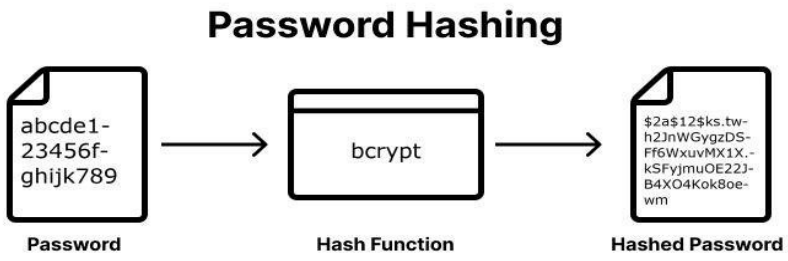
Abstract: *In the realm of cybersecurity, password protection plays a vital role in securing digital assets and personal data. While hashing is a widely accepted standard for storing passwords securely, many systems—due to poor design, legacy support, or misconfiguration—still store passwords in plaintext or reversible formats. This paper investigates the vulnerabilities and exploitation techniques associated with non-hashed (plaintext or symmetrically encrypted) password storage. We explore common scenarios where non-hashed passwords are exposed, analyze methods for retrieving them through system breaches, memory scraping, and forensic analysis, and demonstrate real-world case studies where such practices led to significant data breaches. The study also highlights the ease with which attackers can leverage these weaknesses using basic tools and scripting, emphasizing the critical need for secure password management practices. Recommendations are provided for detecting insecure storage methods and enforcing industry-standard hashing protocols to mitigate risks and enhance overall security posture. Non-hashed passwords stored in systems pose significant security risks, making them vulnerable to brute force and dictionary attacks. The objective of this project is to demonstrate and understand these vulnerabilities and how to effectively crack such passwords*

Keywords: cybersecurity

I. INTRODUCTION

Password security is a critical aspect of cybersecurity. Many systems still store passwords in non-hashed formats, making them susceptible to various attacks.

This project explores techniques to crack non-hashed passwords using brute force and dictionary attacks, emphasizing the importance of proper password hashing and encryption.



EXISTING SYSTEM

Current methods for password storage often involve simple text files without hashing.

Common solutions for cracking non-hashed passwords include:

1. Rainbow tables: Precomputed tables for reversing cryptographic hash functions.
2. Password cracking tools: Software designed to guess passwords (e.g., John the Ripper).
3. Hybrid attacks: Combining dictionary and brute force attacks to increase efficiency.

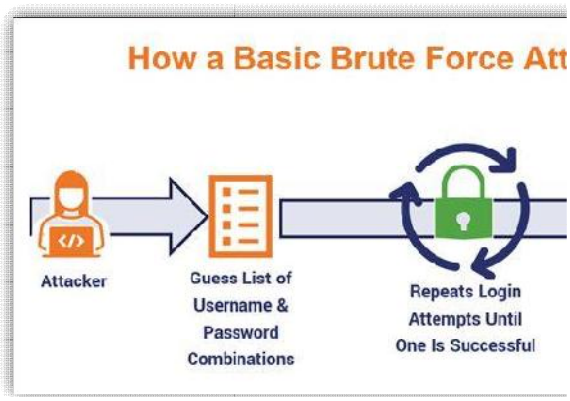
PROPOSED SYSTEM

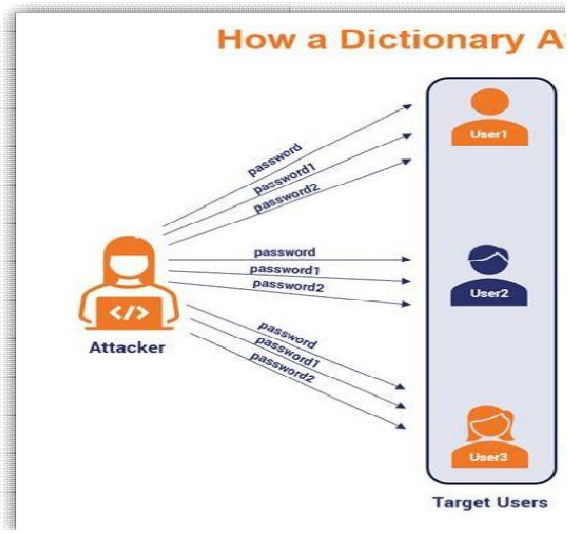
This project proposes using two main techniques to crack non-hashed passwords:

DOI: 10.48175/IJARSCT-25471

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, April 2025
Impact Factor: 7.67

1. Brute Force Attack: Trying all possible combinations until the correct password is found.
2. Dictionary Attack: Using a precompiled list of possible passwords to find matches.





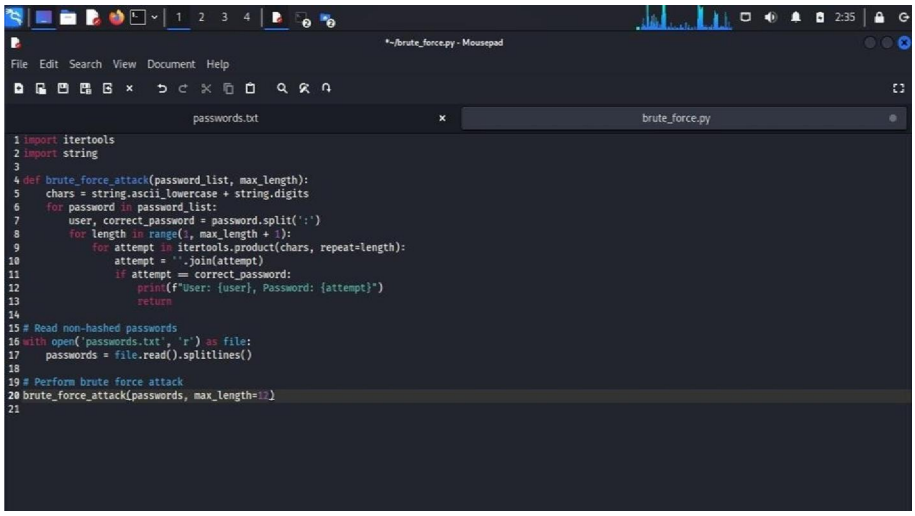
SOFTWARE & HARDWARE REQUIREMENTS

Software Requirements:
Kali Linux: A Linux distribution used for penetration testing.
Python: Programming language for writing custom scripts.
Libraries: Relevant Python libraries (e.g., hashlib, itertools). Hardware Requirements:
Processor: 1GHz CPU (32-bit)
RAM: 1GB
Display: 1024x768 resolution used for better experience
Sufficient storage for password lists and results.

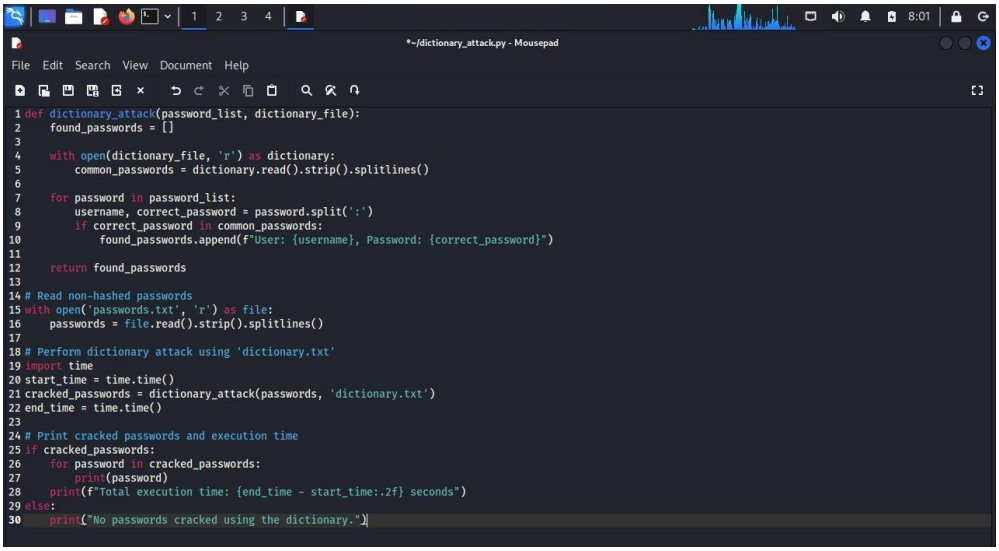
MODULES INVOLVED

1. Setup and configuration of Kali Linux environment.
2. Implementation of brute force attack using Python.
3. Implementation of dictionary attack using Python.
4. Analysis and documentation of the results obtained from the attacks.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, April 2025
Impact Factor: 7.67



Brute force attack's Python script



Dictionary attack's Python Script

WORKFLOW ARCHITECTURE

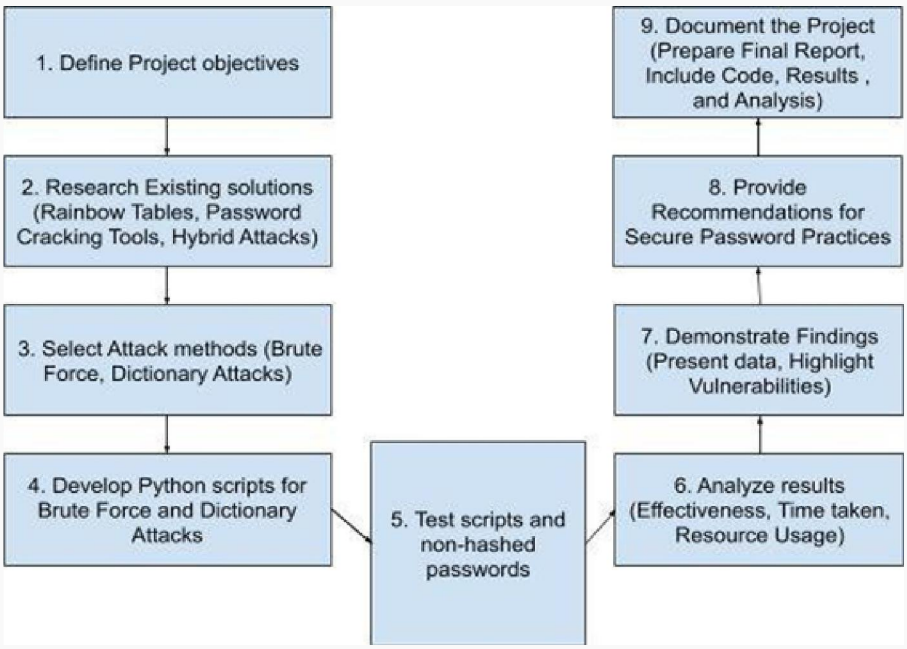
The architecture of the project involves:

Input: List of target non-hashed passwords.

Processing: Brute force and dictionary attacks implemented in Python.

Output: Successfully cracked passwords and the time taken to crack them.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, April 2025
Impact Factor: 7.67



Flow of events involved

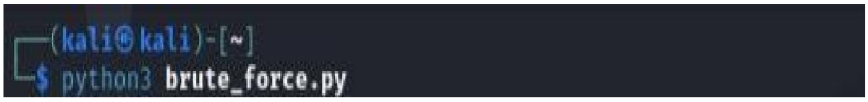
TESTING

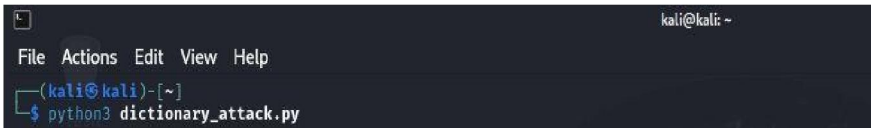
Testing Methodology: To evaluate the effectiveness of the proposed cracking methods, we conducted comprehensive tests using both brute force and dictionary attacks on a variety of non-hashed passwords.

Simple Passwords:

Length: 4-6 characters Composition: Lowercase letters Examples: admin, 1234

Moderate Complexity Passwords: Length: 8-12 characters Composition: Lowercase letters Examples: admin123, password123

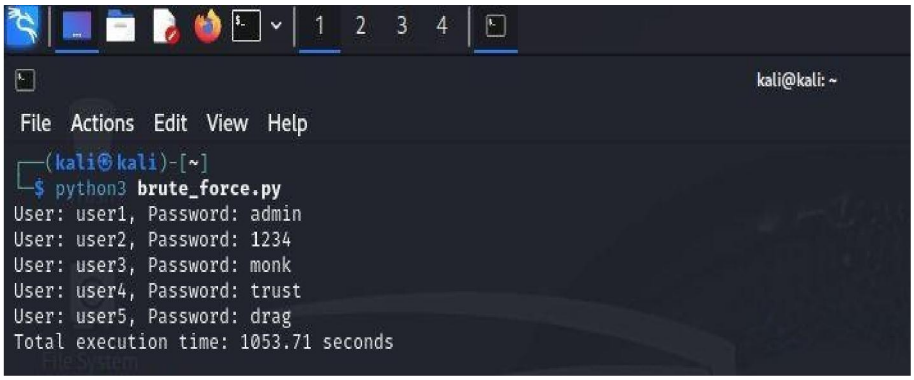




IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, April 2025
Impact Factor: 7.67

OUTPUT

BRUTE FORCE ATTACK

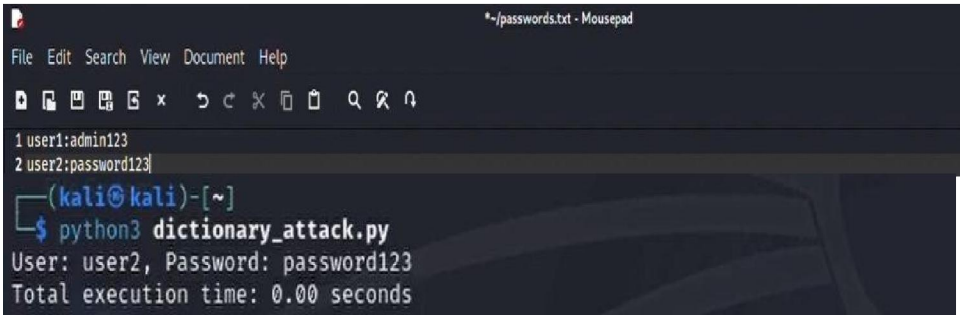


Effective for Shorter Passwords: Up to 6 characters.
Time-Consuming for Longer Passwords: Exponential increase in combinations.
Successful Cracks: Simple passwords like admin, trust, hello1.

OBSERVATIONS:

1. Passwords longer than 6 characters took impractical time.
2. Demonstration used passwords of 4-5 characters for manageability.
3. Successfully cracked all 5 passwords of 5 users
(less than 6 characters).

Output 1



Output 2

Output

Efficient for Common Passwords: Included in the dictionary.
Limited by Dictionary Thoroughness:Successful cracks: 123456, admin123, password123.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, April 2025
Impact Factor: 7.67

Observations:

1. Success depends on the comprehensiveness of the dictionary file.
2. Not dependent on password length, but on the presence in the dictionary.
3. Cracked only one longer password (password123) due to its inclusion in the dictionary file.

II. CONCLUSION

- This project successfully demonstrated the vulnerabilities of non-hashed passwords to both brute force and dictionary attacks.
- Brute force attacks, while guaranteed, are time-consuming for complex passwords. Dictionary attacks are faster but rely on the quality of the wordlist used.
- These findings highlight the critical need for implementing strong hashing mechanisms and using complex, unique passwords to enhance security.
- Future work should focus on exploring more sophisticated attack methods and developing better defense strategies to improve password security.

REFERENCES

[1]. OWASP (Open Web Application Security Project): Website: https://owasp.org/
[2]. Kali Linux Documentation: Website: https://www.kali.org/docs/
[3]. Python Documentation: Website: https://docs.python.org/3/
[4]. "Python for Offensive PenTest: A Practical Guide" by Hussam Khrais: A comprehensive guide on using Python for penetration testing and security analysis.
[5]. "The Hacker Playbook 3: Practical Guide To Penetration Testing" by Peter Kim: A practical guide covering various penetration testing techniques, including password cracking.
[6]. "Hacking: The Art of Exploitation" by Jon Erickson: A deep dive into the technical aspects of hacking, including password cracking methodologies.
[7]. "Violent Python: A Cookbook for Hackers, Forensic Analysts, Penetration Testers and Security Engineers" by TJ O'Connor: A collection of practical Python scripts for offensive and defensive security operations.
[8]. John the Ripper Documentation: Website: https://www.openwall.com/john/
[9]. Hashcat Documentation: Website: https://hashcat.net/wiki/
[10]. NIST (National Institute of Standards and Technology) Password Guidelines: Website: https://pages.nist.gov/800-63-3/sp800-63b.html