Brute-Force Attacks in Web Applications: Offensive Testing and Defensive Strategies

Jayson Sam Fong

Free-lance research

June 2025

Abstract

This paper explores brute-force attacks targeting web applications and demonstrates the methodology used to conduct a controlled offensive security test using tools such as Hydra, Crunch, CeWL, and SecLists in Kali Linux. A test environment was set up to simulate real-world conditions, focusing on identifying vulnerable login forms and applying custom and dictionary-based word lists. The paper highlights the effectiveness of different brute-force strategies and analyzes their outcomes. Defensive countermeasures are also discussed, including multi-factor authentication, account lockout policies, rate limiting, and user education. The research aims to bridge academic knowledge with practical hands-on experience, supporting the need for layered defense mechanisms in modern web applications.

Introduction

Brute-force attacks remain a persistent threat in the cybersecurity landscape, often used by attackers to gain unauthorized access to user accounts or administrative systems by systematically guessing login credentials. Despite the simplicity of brute-force techniques, they are alarmingly effective, particularly against systems lacking proper security configurations. With the growth of web-based services, understanding these attack vectors and their mitigation is essential for cybersecurity professionals.

This paper presents a practical exploration of brute-force attacks in a controlled environment using Kali Linux. By utilizing tools such as Hydra, CeWL, Crunch, and various precompiled wordlists, the research replicates an attack against a practice website designed for ethical hacking. The purpose is not only to understand the mechanics of such attacks but also to develop and evaluate effective defensive strategies. The methodology demonstrates the researcher's applied experience beyond academic coursework, contributing to a broader understanding of offensive and defensive security principles in web application contexts.

Methodology

To simulate a brute-force attack in a controlled environment, a penetration testing lab was configured using Kali Linux and a target web application hosted at ctf.techskyhub.com. The following tools and procedures were used to perform the attack, generate password wordlists, and analyze the results:

- • Tools Used:
    - - Hydra: A parallelized login cracker supporting various protocols.
    - - Crunch: A custom wordlist generator.
    - - CeWL: A custom dictionary generator that crawls websites for wordlists.

- - rockyou.txt and SecLists: Precompiled breach-based wordlists included in Kali Linux.

- • Environment Setup:

  - - Kali Linux was updated using `sudo apt update && sudo apt upgrade`.
  - - Target login form on ctf.techskyhub.com was examined using browser developer tools to locate input fields.

- • Wordlist Creation:

  - - rockyou.txt was extracted using `sudo gunzip rockyou.txt.gz`.
  - - crunch was used to create a password list: `crunch 6 8 abc123! -o pass.txt`.
  - - CeWL was used to generate targeted wordlists: `cewl -d 2 -w words.txt https://ctftechskyhub.com`.

- • Hydra Command Construction:

  - - The login form was identified as using HTTPS with POST method.
  - - Hydra was used to perform the attack:
  - `hydra -L users.txt -P pass.txt ctf.techskyhub.com https-post-form "/index.php:username=^USER^&password=^PASS^:Invalid username or password. Please try again." -V -o credentials.txt`

Results

The brute-force test was executed using multiple wordlists. Hydra successfully attempted thousands of login combinations against the practice target. The result file (credentials.txt) recorded all valid login attempts. The attack highlighted the importance of robust input validation and authentication security in web applications.

Defensive Strategies

To protect against brute-force attacks, organizations should consider implementing the following:

- • Strong Password Policies

  - - Enforce complex, non-dictionary-based passwords and periodic password changes.

- • Multi-Factor Authentication (MFA)

- - Requires an additional authentication factor to mitigate password guessing risks.

- • Account Lockout Policies
  - - Temporarily lock accounts after a predefined number of failed login attempts.
- • Rate Limiting and CAPTCHA
  - - Throttle requests and introduce CAPTCHA to detect and prevent automated attacks.
- • Monitoring and Alerts
  - - Use SIEM solutions to detect unusual login patterns and rapid attempts.
- • Web Application Firewalls (WAFs)
  - - Inspect and filter malicious HTTP traffic patterns in real time.

Conclusion

This research demonstrates the execution and detection of brute-force attacks using freely available tools in Kali Linux. The test environment provided insight into the effectiveness of automated password cracking and the importance of layered defensive mechanisms in modern web application design. As demonstrated, brute-force attacks are both simple and potentially damaging when proper security measures are absent. Organizations must implement a proactive, layered defense strategy to safeguard their digital assets.

References

OWASP. (2023). Brute Force Attack. https://owasp.org/www-community/attacks/Brute_force_attack

Offensive Security. (2022). Hydra - Tool Documentation. https://tools.kali.org/password-attacks/hydra

MITRE ATT&CK. (2023). Brute Force - T1110. https://attack.mitre.org/techniques/T1110/

NIST. (2017). Digital Identity Guidelines. NIST SP 800-63B. https://doi.org/10.6028/NIST.SP.800-63b

Dhanjani, N., Rios, B., & Hardin, B. (2009). Hacking: The Next Generation. O'Reilly Media.