Password Cracking Lab Report — John the Ripper

Prepared by:

Team2

1- Mohammed Almaghrabi
2- Abdulmajeed Ali Alghamdi
3- Khaled Albalawi

Date: 2025-11-03
Source: ~/Downloads/hashes_for_crack.txt (Juice Shop export)

Executive Summary

This report documents the results of a password-cracking session executed with John the Ripper against a hash export from an OWASP Juice Shop instance. The session loaded 22 Raw-MD5 hashes and successfully recovered plaintext credentials for 4 accounts. The recovered credentials indicate the use of weak, easily guessable passwords present in common wordlists. Immediate remediation and follow-up investigations are recommended to mitigate potential account compromise and credential reuse risks.

Scope & Objectives

Scope:

- - Target file: ~/Downloads/hashes_for_crack.txt (hash export from Juice Shop)
  - Hash format: Raw-MD5
  - Total hashes processed: 22
  Objectives:
- - Recover plaintext passwords where possible to assess exposure.
  - Provide remediation and prioritised action items.

Methodology

Tools and commands used during the session are listed below. The approach followed a practical workflow: dump hashes → confirm format → wordlist + rules → check cracked → incremental if needed.

- Primary command executed:

1. john --fork=4 --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt ~/Downloads/hashes_for_crack.txt

Technical Findings

Results:

Account	Recovered Password	Notes
admin@juice-sh.op	admin123	Common weak password
jim@juice-sh.op	ncc-1701	Simple alphanumeric pattern
demo	demo	Default/placeholder password
ethereum@juice-sh.op	private	Common word; likely reused

Session Summary:

- - Total hashes loaded: 22
- - Total cracked: 4
- - Remaining: 18

Analysis

1. Password Strength: The cracked accounts use weak, common passwords (e.g., 'admin123', 'demo'), which are present in standard wordlists such as rockyou. This suggests users employed trivial or default credentials.

2. Likelihood of Compromise: Accounts with recovered passwords are at high risk of unauthorized access. If any of these passwords are reused on other systems, the risk extends beyond the Juice Shop instance.

3. Attack Efficiency: Using a widely-available wordlist with basic rules and 4 process forks recovered 4/22 hashes quickly, demonstrating high efficiency against weak password selections.

Impact Assessment

Confidentiality: High for accounts with administrative privileges (e.g., admin@juice-sh.op).
Integrity: Potentially compromised if attackers modify data or settings.
Availability: Could be affected if attackers lock accounts or trigger incident responses.
Business Impact: For production systems, similar weak passwords would represent a significant security risk, leading to data breaches, unauthorized transactions, or lateral movement within the environment.

Immediate Remediation (Action Items)

- Execute the following actions within 24 hours:

2. Force password reset for the affected accounts listed above and invalidate active sessions.
3. Rotate any keys or tokens associated with affected accounts.
4. Review authentication logs for suspicious access (timestamps, source IPs, geolocation).
5. Notify affected users and the security/incident response team.

Short-term Measures (72 hours)

- Enforce password complexity and minimum length policies (e.g., 12+ characters, passphrases).
- Enable Multi-Factor Authentication (MFA) for all privileged and user accounts.
- Search for password reuse across other systems (credential stuffing risk) using internal discovery tools.

Long-term Recommendations

- Implement centralized authentication (SSO) and enforce strong password policies at the identity provider.
- Deploy continuous monitoring for anomalous login behavior and integrate with SIEM/SOAR.
- Include password hygiene and secure credential management in user training and onboarding.
- Consider replacing MD5-based storage with modern, salted, and slow hashing algorithms (e.g., bcrypt, Argon2).

Evidence & Reproducibility

- Commands executed (capture exact command lines and shell history).
- Wordlist used: /usr/share/wordlists/rockyou.txt (record checksum if needed).
- John version and system details (CPU cores used: 4 forks configured).
- Final cracked output saved from `john --show --format=Raw-MD5 ~/Downloads/hashes_for_crack.txt`.

Appendix A — Raw John Output (selected)

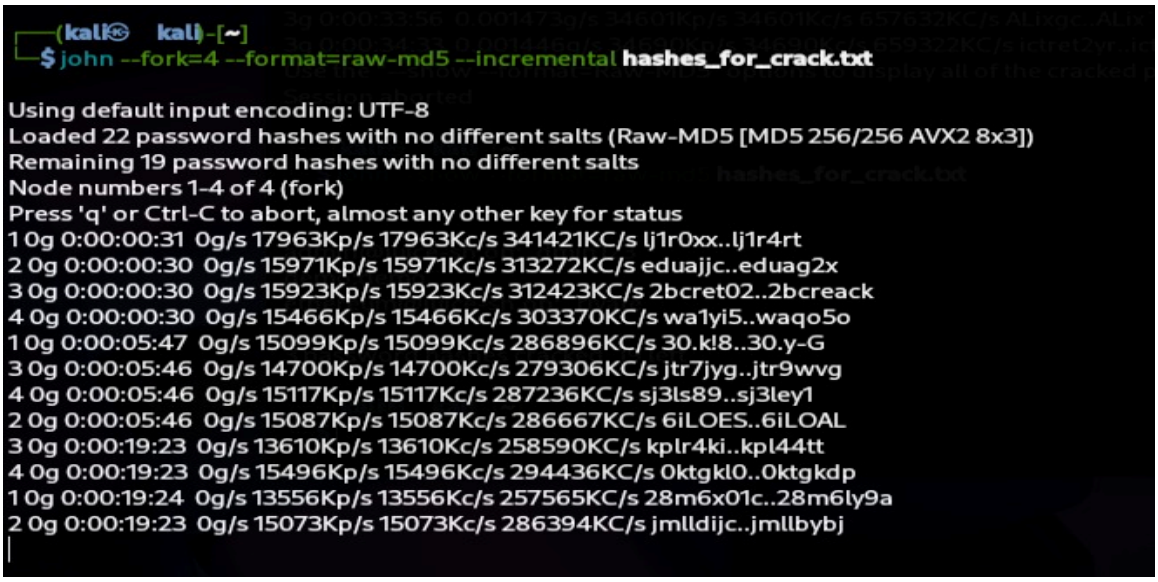
Using default input encoding: UTF-8
Loaded 22 password hashes with no different salts (Raw-MD5 [MD5 128/128 AVX 4x3])
Node numbers 1-4 of 4 (fork)

Sample candidates processed and performance metrics (truncated):
2 1g 0:00:00:00 DONE (2025-11-01 04:06) 2.439g/s 8745Kp/s ... benzbenz.abygurl69
3 0g 0:00:00:00 DONE (2025-11-01 04:06) 0g/s 9194Kp/s ... god143.a6_123
4 1g 0:00:00:00 DONE (2025-11-01 04:06) 2.439g/s 8745Kp/s ... fabian..*7¡Vamos!
1 2g 0:00:00:00 DONE (2025-11-01 04:06) 4.878g/s 8745Kp/s ... joefeher.ie168

Final show output:
admin@juice-sh.op:admin123
jim@juice-sh.op:ncc-1701
demo:demo
ethereum@juice-sh.op:private

Appendix B — Recommended Commands

- john --format=raw-md5 --incremental ~/Downloads/hashes_for_crack.txt



- john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt --rules --mask='?d?d?d' ~/Downloads/hashes_for_crack.txt

Report Approval

Prepared by:

Team2

1- Mohammed Almaghrabi
2- Abdulmajeed Ali Alghamdi
3- Khaled Albalawi

Date: 2025-11-03