Password Cracking & Authentication Security – Practical Study

1. How Passwords Are Stored (Hashing vs Encryption)

- **Hashing**
  - One-way function (cannot be reversed).
  - Same input → same output.
  - Used for storing passwords.
  - Examples: **MD5, SHA-1, SHA-256, bcrypt**
- **Encryption**
  - Two-way process (can be decrypted).
  - Uses a key.
  - Used for data transmission, not password storage.

✅ **Best Practice:** Passwords should always be **hashed + salted**, never encrypted.

---

2. Identifying Different Hash Types

Hash Type	Length	Security Level	Notes
MD5	32 hex	❌ Weak	Fast, easily cracked
SHA-1	40 hex	❌ Weak	Deprecated
SHA-256	64 hex	⚠️ Medium	Strong but fast
bcrypt	Variable	✅ Strong	Slow, salted
NTLM	32 hex	❌ Weak	Windows legacy

Tools to identify hashes:

- hashid
- hash-identifier
- Online hash identifiers

---

3. Generating Password Hashes

Linux examples:

```
echo -n "password123" | md5sum
echo -n "password123" | sha1sum
echo -n "password123" | sha256sum
```

Using John the Ripper:

```
john --test
```

---

4. Cracking Weak Hashes (Wordlist Attack)

Using Hashcat

```
hashcat -m 0 hash.txt rockyou.txt
```

- -m 0 → MD5
- rockyou.txt → Common password list

Using John the Ripper

```
john --wordlist=rockyou.txt hashes.txt
```

📌 **Weak passwords are cracked quickly because they exist in wordlists.**

---

5. Brute Force vs Dictionary Attacks

Attack Type	Description	Speed
Dictionary	Uses common passwords	Fast
Brute Force	Tries all combinations	Slow
Hybrid	Dictionary + patterns	Medium

Example hybrid rule:

password → Password@123

---

6. Why Weak Passwords Fail

- Short length
- Common words
- No symbols or numbers
- Reused passwords
- Predictable patterns

Example cracked passwords:

- admin
- 123456
- password@123

---

7. Importance of Multi-Factor Authentication (MFA)

MFA adds an extra layer:

- Something you **know** (password)
- Something you **have** (OTP, phone)
- Something you **are** (biometrics)

✅ Even if a password is cracked, MFA **blocks unauthorized access**.

---

8. Recommendations for Strong Authentication

- Use **12–16 character** passwords
- Enable **MFA everywhere**
- Use **password managers**
- Prefer **bcrypt / Argon2**
- Avoid password reuse
- Monitor login attempts
- Enforce account lockout policies