

ELEVATE LABS TASK 4

◆ Hashing

- A **one-way process**
- Same input → same output
- Cannot be reversed

Example:

password123 → 482c811da5d5b4bc6d497ffa98491e38 (MD5)

◆ Encryption

- **Two-way process**
- Data can be decrypted using a key
- ✗ Not used for passwords

✓ **Conclusion:** Passwords are always **hashed**, not encrypted.

Step 2: Identify Different Hash Types

Hash Type Secure? Reason

MD5 ✗ No Very fast, easy to crack

SHA-1 ✗ No Collision attacks exist

SHA-256 ! Medium Still fast

bcrypt ✓ Yes Slow + salted

👉 **bcrypt & Argon2 are recommended**

Step 3: Generate Password Hashes

Example passwords:

admin123

password

welcome@123

Generate hashes (Linux):

```
echo -n admin123 | md5sum
```

```
echo -n admin123 | sha1sum
```

Result:

MD5 : 0192023a7bbd73250516f069df18b500

SHA1 : d033e22ae348aeb5660fc2140aec35850c4da997

Step 4: Identify Hash Type

Use:

- Hash format length
- Online hash identifier
- Hashcat mode

Example:

32 chars → MD5

40 chars → SHA-1

Step 5: Crack Weak Hashes (Dictionary Attack)

Tool: Hashcat

Create a file:

hash.txt

Add hash inside.

Run dictionary attack:

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

Where:

- -m 0 → MD5
- -a 0 → Dictionary attack
- rockyou.txt → wordlist

 Weak passwords crack **within seconds**

Step 6: Brute Force vs Dictionary Attack

◆ Dictionary Attack

- Uses known passwords
- Faster
- Very effective against humans

◆ Brute Force

- Tries all combinations
- Slow
- Guaranteed success (eventually)

 bcrypt defeats brute force by being **slow**

Step 7: Why Weak Passwords Fail

 Weak passwords:

- Short length
- Common words
- Predictable patterns
- Reused passwords

Example:

admin123

password@123

welcome2024

 These exist in wordlists → cracked instantly

Step 8: Multi-Factor Authentication (MFA)

MFA = **More than one factor**

Factor Type	Example
Something you know	Password
Something you have	OTP, phone
Something you are	Fingerprint

Even if password is cracked → attacker is blocked

Step 9: Security Recommendations

✓ Use **12-16 character passwords**

✓ Use **bcrypt / Argon2**

✓ Enable **MFA everywhere**

✓ No password reuse

✓ Use password managers

✓ Rate-limit login attempts

Step 10: Final Conclusion

- Weak passwords are the **main cause of breaches**
- Hashing protects stored credentials
- Cracking demonstrates real-world risk
- MFA drastically reduces account compromise