Report Submission

To

Elevate Labs

Task 6: Password Strength Creation and Evaluation

Name	Yash Javiya
Submission to	Elevated Labs

Task 6: Password Strength Creation and Evaluation

Task 6: Password Strength Creation and Evaluation Report

Topic: Password Security & Best Practices
Objective: Understand the characteristics of strong passwords and evaluate them using online tools.

1. Introduction

In the modern digital landscape, password security is a foundational aspect of protecting personal and organizational data. Weak passwords can lead to unauthorized access, data breaches, and identity theft. This task focuses on experimenting with different password formats, testing their strength, and extracting insights on how to create secure passwords.

2. Tools Used

- **Online Password Strength Checkers**:
  - Password Meter
  - password monster

3. Passwords Tested

S.No	Password Tested	Complexity Level	Score (%)	Strength Category	Notes
1	password123	Weak	25%	Weak	Common, easily guessable
2	P@ssw0rd!	Medium	65%	Moderate	Slightly better with special characters
3	MyDog$Is@Brave2024	Strong	90%	Strong	Passphrase-style with symbols & numbers
4	123456	Very Weak	10%	Very Weak	Extremely common
5	L#9d$gT@3hV!u@7%	Very Strong (Random)	100%	Very Strong	High entropy and unpredictability

4. Key Learnings from Evaluation

- **Length matters**: Longer passwords significantly increase resistance to brute-force attacks.
- **Variety in character types**: Use of uppercase, lowercase, numbers, and symbols makes passwords harder to crack.
- **Avoid dictionary words**: Even if a password includes symbols, if it contains predictable words (like “password”), it's still weak.
- **Passphrases are powerful**: Sentences or phrases with modifications can be both memorable and strong.
- **Avoid repetition and patterns**: Sequences like 123456 or abcabc are easily cracked.

5. Common Password Attacks

Attack Type	Description
Brute Force	Tries all possible combinations until the correct one is found.
Dictionary Attack	Tries common words and phrases from a pre-compiled list.
Credential Stuffing	Reuses leaked credentials from previous data breaches.
Phishing	Tricking users into revealing passwords through fake websites or messages.

Task 6: Password Strength Creation and Evaluation

Here’s a detailed explanation of the four types of password attacks which is listed, including methodology, tools, targets, prevention, and real-world examples:

1. Brute Force Attack

Description:
A brute force attack systematically tries every possible combination of characters (letters, numbers, symbols) until the correct password is found.

Methodology:

- Exhaustive search technique: Starts with “a”, then “aa”, “ab”, etc.
- Attacks both online (e.g., login forms) and offline (e.g., hashed passwords).
- Time-consuming for strong passwords.

Common Tools:

- Hydra
- John the Ripper
- Medusa
- THC Hydra

Target Environments:

- SSH/FTP/HTTP login pages
- Encrypted archives
- Login forms without rate-limiting

Prevention:

- Use complex, long passwords
- Limit login attempts (rate limiting)
- Enable CAPTCHA
- Use multi-factor authentication (MFA)

Real-world Example:

- Attackers target admin login panel of a CMS using brute-force tools to guess the password.

---

2. Dictionary Attack

Description:
This attack uses a predefined list (dictionary) of commonly used passwords and phrases, attempting each one in succession.

Methodology:

- Based on human password habits (e.g., "password123", "admin", "qwerty").
- Faster than brute force but less comprehensive.

- Can be enhanced with rule sets or mutations (e.g., leetspeak: “p@ssw0rd”).

Common Tools:

- John the Ripper
- Hashcat
- Hydra
- CeWL (to create custom dictionaries)

Target Environments:

- Any system where users manually create passwords
- Offline cracking of password hashes

Prevention:

- Enforce strong password policies
- Block common dictionary passwords
- Salting and hashing passwords properly

Real-world Example:

- Attacker uses a wordlist of 10,000 most common passwords to try logging into WordPress admin accounts.

---

3. Credential Stuffing

Description:
An automated attack where attackers use stolen username-password pairs (from data breaches) to try and log into multiple unrelated services.

Methodology:

- Takes advantage of password reuse across platforms.
- Uses automation to try credentials at scale.
- Targets **APIs, websites, and mobile apps.**

Common Tools:

- Sentry MBA
- Snipr
- OpenBullet
- Modlishka (for session hijacking)

Target Environments:

- Web applications
- Online services (banking, e-commerce, social media)

Prevention:

- Detect unusual login patterns (e.g., IP geolocation, velocity rules)
- Implement MFA
- Monitor for credential leaks (dark web scanning)
- Use bot detection tools (e.g., reCAPTCHA, Cloudflare)

Real-world Example:

- 2019 Disney+ credential stuffing incident: Leaked login data from other sites was used to access thousands of Disney+ accounts shortly after launch.

---

4. Phishing

Description:
A social engineering attack where the attacker tricks users into giving up sensitive information (e.g., usernames, passwords, OTPs) by impersonating legitimate sources.

Methodology:

- Uses emails, fake websites, or SMS to lure victims.
- Often mimics legitimate brands (e.g., banks, Microsoft, Google).
- May include fake login forms or malicious attachments.

Types of Phishing:

- Email phishing
- Spear phishing (targeted)
- Smishing (SMS-based)
- Vishing (voice-based)

Common Tools:

- SET (Social Engineering Toolkit)
- Evilginx2
- Gophish
- King Phisher

Target Environments:

- Corporate email users
- Financial platforms
- Remote workers
- Cloud service users (e.g., Office 365, Google Workspace)

Prevention:

- Employee training and awareness
- Email filtering and spoof detection (SPF/DKIM/DMARC)
- Secure email gateways
- URL inspection and real-time phishing detection

Real-world Example:

- Google Docs phishing scam (2017): Attackers sent fake Google Docs invites that led to phishing pages, affecting thousands of users.

---

6. Tips for Creating Strong Passwords

1. Use at least **12–16 characters**.
2. Combine **uppercase, lowercase, digits, and symbols**.
3. Avoid using names, birthdates, or predictable patterns.
4. Prefer **passphrases** over single words (e.g., Sky@Is$Blue!23).
5. Don’t reuse passwords across platforms.
6. Consider using a **password manager** to store complex passwords securely.

---

7. Multi-Factor Authentication (MFA) – In Detail

What is MFA?

Multi-Factor Authentication (MFA) is a security mechanism that requires **two or more distinct authentication factors** to verify a user's identity before granting access to a system, application, or network.

The core idea is:

“Don’t rely on just passwords – add more layers of protection.”

The 3 Categories of Authentication Factors

Factor Type	**Description**	**Examples**
Something You Know	Information only the user knows	Passwords, PINs, answers to questions
Something You Have	Physical items the user possesses	OTP devices, smartphone apps, smartcards
Something You Are	Biometric traits of the user	Fingerprint, face, retina, voice, behavior

MFA = Combination of at least two of these.

Common MFA Methods & Examples

Factor Type	**Method**	**Description**

Task 6: Password Strength Creation and Evaluation

Knowledge (1st factor)	Password / PIN	Traditional credentials known only to the user
Possession (2nd factor)	SMS/Email OTP	One-time password sent via mobile or email
	Authenticator App (TOTP)	Time-based codes from apps like Google Authenticator, Authy, Microsoft Authenticator
	Push Notification	Approve login request with a single tap (e.g., Duo Push, Okta Verify)
	Hardware Tokens (HOTP/TOTP)	Devices like YubiKey or RSA SecurID generate OTPs
	Smart Cards / USB keys	Physical authentication devices (used in banking, defense, etc.)
Inherence (3rd factor)	Biometrics	Face ID, fingerprint scan, iris recognition, voice pattern, behavioral biometrics

---

How MFA Works – Workflow Example

1. **User enters their username and password** (something they know).
2. The system prompts for a second factor:
   - e.g., enters a code from an authenticator app (something they have).
3. If both checks pass → access granted.

Benefits of MFA

Benefit	**Explanation**
Stronger Security	Even if one factor (e.g., password) is compromised, attackers can't proceed.
Prevents Credential Stuffing	Makes stolen credentials from breaches ineffective without the second factor.
Regulatory Compliance	Many standards (GDPR, HIPAA, PCI-DSS) require MFA for sensitive access.
Protects Remote Access	Essential for VPNs, cloud services, and remote work.

Weaknesses & Limitations

Weakness	**Details**
SIM Swapping Attacks	SMS-based MFA can be hijacked by attackers gaining control of your SIM.

Phishing-Resistant MFA Needed	Standard TOTP codes can still be phished. Prefer FIDO2/WebAuthn or push-based MFA.
User Inconvenience	Some users may find it cumbersome or confusing to set up MFA.

Popular MFA Tools & Services

Tool/Service	**Type**	**Use Case**
Google Authenticator	TOTP (mobile app)	Website and app 2FA
Authy	TOTP (multi-device sync)	Cloud backup of MFA codes
Microsoft Authenticator	TOTP + Push	Enterprise use with Azure AD
Duo Security	Push + TOTP + biometrics	Enterprise SSO & VPN access
YubiKey	Hardware token	FIDO2, U2F, OTP, PGP encryption
Okta / OneLogin	Cloud MFA + SSO	Cloud-based identity management
FIDO2 / WebAuthn	Passwordless MFA	Secure, phishing-resistant MFA

Common Real-World Implementations

Scenario	**MFA Setup Example**
Corporate VPN Access	Username + Password + Duo Push notification
Online Banking	Password + SMS OTP or Smart Card
Cloud Email (e.g., O365)	Password + Authenticator App or FIDO2 Key
GitHub / GitLab	Password + TOTP or Security Key
Social Media (e.g., Instagram, Facebook)	Password + SMS or App-based OTP

MFA vs 2FA vs Passwordless

Term	**Description**
2FA (Two-Factor Auth)	Uses exactly **2** factors from different categories.
MFA (Multi-Factor Auth)	Uses **2 or more** factors. More general term.
Passwordless Authentication	Uses possession or biometrics without a password (e.g., FIDO2).

Phishing-Resistant MFA – Why It's Important

Attackers can phish OTPs from authenticator apps via fake login pages.

Phishing-resistant MFA like **FIDO2/WebAuthn** uses **public-private key cryptography** and cannot be phished.
E.g., **YubiKey + fingerprint + browser login**.

Best Practices for MFA Implementation

1. **Avoid SMS OTPs** for critical services – use apps or hardware tokens instead.
2. **Enable MFA everywhere**, not just on admin or sensitive accounts.
3. **Backup recovery codes** to prevent lockout.
4. **Train users** to recognize MFA phishing attacks (e.g., fake push requests).
5. **Monitor login behavior** – unusual MFA patterns may indicate attack attempts.

---

8. Outcome

- How password complexity affects strength and crackability.
- Tools used to measure and validate password quality.
- Real-world methods attackers use to compromise weak credentials.
- Best practices to follow for secure password creation and storage.

Conclusion

Creating strong passwords is not just a checkbox—it's a key defense mechanism against cyber threats. Through this task, I've gained hands-on knowledge and developed habits that improve both personal and organizational security hygiene.