**Software Engineering Virtual Experience Program**

**Hashing Algorithm:** The organization used the MD5 hashing algorithm to protect passwords. As mentioned earlier, MD5 is not recommended for password hashing due to its weaknesses.

**Protection Level:** MD5 provides weak protection for passwords. It is relatively fast to compute and susceptible to various attacks, making it easier for attackers to crack passwords.

**Controls for Harder Cracking:** To make password cracking harder for attackers in the event of a password database leak, the organization should implement the following controls:

* **Upgrade Hashing Algorithm:** Switch from MD5 to a strong and modern hashing algorithm like bcrypt, Argon2, or SHA-256 with a strong random salt. These algorithms are designed to be computationally intensive and slow down password cracking attempts significantly.
* **Salted Hashing:** Ensure that a unique and random salt is used for each password before hashing. Salting prevents attackers from using precomputed tables (rainbow tables) for password cracking.
* **Password Complexity Requirements:** Enforce password complexity rules, such as a minimum length and the use of a mix of uppercase, lowercase, digits, and special characters, to increase the key space and make brute-force attacks more difficult.
* **Password Expiry and Reset Policies:** Implement password expiry and regular password resets to reduce the window of opportunity for attackers to use compromised passwords.

**Organization's Password Policy:** Without specific information about the organization's password policy, such as password length or key space, we cannot provide details about it.

**Changes to Enhance Security:** To make breaking passwords harder, the organization should:

* **Upgrade Hashing Algorithm:** Move away from MD5 and adopt a more secure hashing algorithm.
* **Increase Password Length:** Encourage longer passwords, as longer passwords generally increase the complexity and make them harder to crack through brute force.
* **Password Complexity Requirements:** Enforce a mix of uppercase, lowercase, digits, and special characters to increase the key space.
* **Implement Salting:** Use random salts for each password before hashing to prevent the use of precomputed tables.
* **Enforce Regular Password Changes:** Although controversial, enforcing regular password changes can reduce the exposure window in case of a password breach. However, balance this with user convenience to avoid weaker passwords due to frequent changes.

Cracked Passwords

experthead:e10adc3949ba59abbe56e057f20f883e

interestec:25f9e794323b453885f5181f1b624d0b- **123456**

ortspoon:d8578edf8458ce06fbc5bb76a58c5ca4- **123456789**

reallychel:5f4dcc3b5aa765d61d8327deb882cf99- **qwerty**

simmson56:96e79218965eb72c92a549dd5a330112- **password**

bookma:25d55ad283aa400af464c76d713c07ad- **111111**

popularkiya7:e99a18c428cb38d5f260853678922e03- **12345678**

eatingcake1994:fcea920f7412b5da7be0cf42b8c93759- **abc123**

heroanhart:7c6a180b36896a0a8c02787eeafb0e4c- **1234567**

edi\_tesla89:6c569aabbf7775ef8fc570e228c16b98- **password1**

liveltekah:3f230640b78d7e71ac5514e57935eb69- **password!**

blikimore:917eb5e9d6d6bca820922a0c6f7cc28b- **qazxw**

by- Urvija Jha