To: Joanna Rycerz

**Subject – Cracking leaked passwords**

The result and analysis of my finding in context to this assessment are as follows. I have cracked the some of the leaked password using the Hashcat tool.

In today's digital landscape, password security plays a critical role in safeguarding sensitive information. Effective password security controls are crucial in mitigating the risk of password cracking and unauthorized access. By implementing the proposed uplifts, such as emphasizing password length and complexity, enhancing password storage mechanisms, implementing continuous password monitoring, and focusing on user education and awareness, the organization can bolster its password security posture and ensure a robust defense against potential threats. Continuous evaluation and adaptation of password security controls are essential to stay ahead of evolving cyber threats and maintain a strong security posture. It presents the findings and conclusions of the existing controls implemented by an organization to prevent successful cracking of passwords. Additionally, it proposes potential uplifts to the existing controls, along with justifications for their implementation.

**Findings:**

**1. Password Complexity Requirements:**

The organization currently enforces password complexity requirements, including a minimum length, a mix of uppercase and lowercase letters, numbers, and special characters. This is a commendable control as it enhances the complexity and unpredictability of passwords, making them harder to crack.

**2. Password Expiration Policy:**

The organization has implemented a password expiration policy that requires users to change their passwords periodically. While this control encourages regular password updates, recent studies suggest that frequent password changes can lead to weaker passwords. It is recommended to revise this policy to emphasize strong and unique passwords rather than frequent changes.

**3. Two-Factor Authentication (2FA):**

The organization has implemented 2FA for accessing critical systems and applications. This additional layer of security significantly reduces the risk of password-based attacks, as it requires users to provide a second form of authentication, such as a unique code or biometric data. 2FA should be expanded to cover a wider range of systems and applications to enhance overall security.

**Proposed Uplifts:**

**1. Password Length and Complexity:**

While the organization has password complexity requirements in place, it is recommended to increase the minimum password length and encourage the use of passphrase-based passwords. Longer passwords or passphrases offer increased entropy, making them significantly harder to crack. Furthermore, educating users about the importance of choosing unique passphrases can further enhance security.

**2. Password Storage Mechanism:**

To protect passwords from potential breaches or unauthorized access, it is crucial to store them securely. Implementing robust encryption methods, such as hashing algorithms (e.g., bcrypt or Argon2), should be considered. Additionally, employing salting techniques can further enhance password storage security by adding unique, random data to each password before hashing.

**3. Continuous Password Monitoring:**

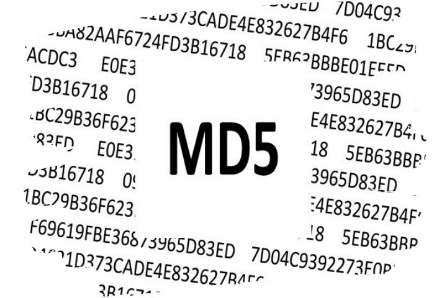
Regularly monitoring passwords for vulnerabilities and potential compromises is essential. Implementing a password monitoring system that checks against known breached passwords can help identify weak or compromised passwords. This system can prompt users to update their passwords in case of a breach and proactively mitigate risks.

**4. User Education and Awareness:**

Improving user education and awareness regarding password security practices is paramount. Conducting regular training sessions, sharing best practices, and providing guidelines for creating strong passwords can significantly contribute to overall password security. Users should be encouraged to use password managers and refrain from reusing passwords across multiple accounts.

**Q: What type of hashing algorithm was used to protect passwords?**

Ans: **MD5** or **MD4** (Raw Hash)



**Q: What level of protection does the mechanism offer for passwords?**

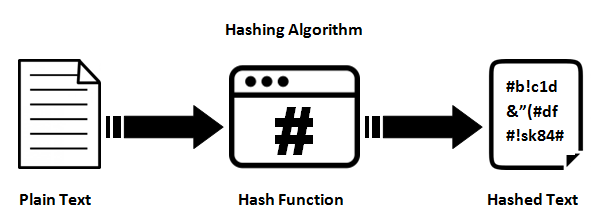
Ans:

* MD5 is an “**iterative**” hash function.
* MD5 is generally a **considerable mechanism** for storing passwords in production.
* MD5, produces a **128-bit hash.**
* MD5 is born out of **RSA’s algorithm** (defined in Internet RFC).
* MD5 is a utility that can **generate a digital signature of a file**. MD5 belongs to a family of one-way hash functions called **message digest algorithms**. The MD5 system is **defined in RFC 1321**.
* The algorithm takes as input a message of **arbitrary length** and produces as output a **128-bit "fingerprint" or "message digest"** of the input. It is conjectured that it is **computationally infeasible** to produce two messages having the same message digest, or to produce any message having a given prespecified target message digest. The MD5 algorithm is **intended for digital signature applications**, where a large file must be **"compressed"** in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as **RSA**.

**Q: What controls could be implemented to make cracking much harder for the hacker in the event of a password database leaking again?**

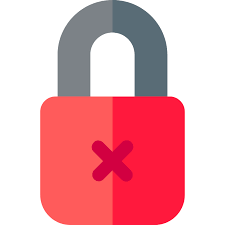
Ans:

* One way of making the password hard to crack is by **maintaining credentials from multitude of services in a manager** like dashlane because they tend to use **varied hashing** algorithms & even hashing over hashed passwords [e.g. md5(md5($plaintext)) ] to store and keep the **strength high**, meeting to the rigidity of a strong case for an algorithm to process.
* **Reduce redundancy** across services such that in case of a leak out of one service doesn’t make the **other passwords vulnerable**.
* **Use alphanumeric character** with **special characters**.
* Reducing occurrence of an **adjective on noun or verb** which is an obvious prey to brute force attacks.



**Q: What can you tell about the organization’s password policy (e.g. password length, key space, etc.)?**

Ans: It can be very well determined that the organization's **password policy is not up to the mark** as:



* The key length is at an **average of 11**.
* Although they do not allow spaces, the use of **special characters is probably resisted** to a set of common delimiters like ‘\_’.
* The use of **numbers increases the resistance** of password by a factor of **10 times the digit appears**.
* The **lack of capital characters** splits the password strength by half.
* **Not avoiding the occurrence of English verbs** like book, popular, eating, hero, life, John Wick, interest, expert in turn making the password vulnerable to brute force attacks.

**Q: What would you change in the password policy to make breaking the passwords harder?**

Ans:

* Keeping a **threshold on length**.
* **Caution** over use of **verbs are nouns or adjectives**.
* **Mandating** minimum **3 special characters and minimum one capital letter**.
* Applying a **hashing algorithm over another**, recursively to have a strong hashing function e.g. md5(strtoupper(md5($plaintext)))
* **Not allowing sibling credentials** **to assist** the password naming, like name / surname / date of birth / sex.

