 To: **Joanna Rycerz**

10th February 2023

Warsaw, Poland

**Subject – Cracking leaked passwords.**

Dear Mam,

Following an attempt to decode all the leaked hashes, I discovered multiple weaknesses in your password protocol. This email encompasses all my discoveries and proposals for enhancing your password protocol. The outcome and evaluation of my investigation in relation to this examination are detailed below. I successfully deciphered a few of the leaked passwords using the Hashcat and CrackStation tools.

**Question 1: What type of hashing algorithm was used to protect passwords?**  
Hashing Algorithm used was MD5 or MD4 (Raw Hash). Secure Hash Algorithm (SHA) and Message Digest (MD5) are the standard cryptographic hash functions to provide data security for authentication.

**Question 2: What level of protection does the mechanism offer for passwords?**All the password which are compromised were using MD5 which is a weaker hash algorithm and is prone to collisions. MD5 (message digest algorithm) is a bad password hashing algorithm because it is too fast and memory conserving. Attacker can compute the hash of large number of passwords per second. It was very easy to crack with CrackStation and rockyou.txt wordlist via terminal and web browsers. I would suggest that you use a very strong password encryption mechanism to create hashes for the password based on SHA.

* 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**.

**Question 3: What controls could be implemented to make cracking much harder for the hacker in the event of a password database leaking again?**

Try using better algorithm in place of MD5. Eg. SHA256 - Always use salts with hashes where feasible. - for better security use slow algorithm like bcrypt. Which make harder for attacker because it requires more CPU cycles to authenticate user.

* 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.

**Question 4: What can you tell about the organization’s password policy (e.g. password length, key space, etc.)?**After cracking the passwords, we find the following things about organisation’s password policy and conclude that the organization's **password policy is not up to the mark** as:

* Weak hash functions used with no salting.
* There is no specific requirement for the password creation. Users can use any combination of word and letters to create a password. Common passwords are used which can be easily guessed and cracked.
* No use of capital letters, numbers, and special symbols together. Although they do not allow spaces, the use of **special characters is probably resisted** to a set of common delimiters like ‘\_’.
* Minimum length for password is set to 6. The key length is at an **average of 11**.
* 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.

**Question 5: What would you change in the password policy to make breaking the passwords harder?**You can include several new things in your password policy. Check your password security with password strength checker tools and websites.

* Keeping a **threshold on length**. We can increase the password length to 12 because less character’s length it becomes easy for hacker to crack the password using brute force attack. Longer passwords are better, 8 characters is a starting point.
* **Caution** over use of **verbs are nouns or adjectives**. Avoid common words and character combinations in your password. Don’t use common phrase as password.
* **Mandating** minimum **3 special characters and minimum one capital letter**. Use of mix characters. Include special character, Capital and Small letters, numbers in your password.
* 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. Don’t let users include their username, actual name, date of birth and other personal information while creating a password.
* Train your users to follow these policies to keep their passwords safe.