Report:

To whom it may concern,

For this task, I utilized an open-source tool, “Hashcat”, for cracking the provided hashes.

My strategy for the attack was to first iteratively use different hashing algorithms to launch straight attacks using a wordlist of 10 million commonly used passwords. After this, I would attempt a full-on brute-force attack.

Upon my first attempt, however, using the one-way hashing algorithm, MD5, I was able to crack 100% of all passwords in the dump file. The MD5 is a relatively weak form of encryption and is easily defeated by modern computers. It is also vulnerable to "rainbow table" attacks, as well as hash collisions - which means that two different input passwords could produce the same output hash.

To make cracking much harder for the hacker in the event of a password database leaking again, the following controls could be implemented:

1. Use a stronger encryption algorithm: Instead of using MD5, organizations should consider using stronger encryption algorithms such as bcrypt, scrypt, or Argon2. These algorithms are specifically designed to be more resistant to cracking.
2. Use of Salt: Adding a randomly generated "salt" to the password before encrypting it can make cracking much harder for a potential hacker. A salt is a random value that is added to the password before it is hashed. This makes it much harder for the hacker to use precomputed tables or wordlists (like I did) to crack the password.

The passwords recovered include:

* "111111" and "123456", "123456789", "12345678", and "1234567" which are simple numeric patterns that are easily guessed and quickly brute-forced using modern computers.
* "abc123", "bluered", "password1", and "password" are very common words or patterns that are easily found in a dictionary attack.
* "qwerty" and "qazxsw" are common pattern based on the keyboard layout, which can also be easily guessed and are present in many wordlists.

It appears that there is no specific stringent password policy in place at this organization. More constraints should be implemented such as increasing the minimum length of passwords, and requiring the use of a combination of uppercase and lowercase letters, numbers, and special characters.

Best regards,

Destiny Uche