Dear Sir/Madam,

On my endeavor to crack the given passwords, I observed that the given hashes were generated using the MD5 algorithm (Message Digest). The algorithm works by dividing a given message into blocks of 512 bits and creating a 128 bit digest(typically, 32 Hexadecimal digits).
The passwords were weak and made it fairly easy to crack using Hashcat and md5decrypt.net.
I would like to bring to your attention my findings in relation to controls used by the organization and suggestions for improvement in password creation policy.

SHA-2 (Secure Hash Algorithm 2) is a set of cryptographic hash functions designed by the United States National Security Agency (NSA).
I would recommend this as an alternative to MD5 and particularly the Secure Hash Algorithm (SHA-256), due to the following reason:

- It is the standard cryptographic hash function used to provide data security for authentication.
- A major problem of the MD5 is that it is very fast. This means that hackers can find their way into the system by trying many password possibilities. This implies that with the MD5, brute force attacks are faster than SHA-256.
- The output size is longer and the probability of collisions is lower.

Now the question arises, is there any additional process to make the hashing more secure. Yes the solution is salting. Salting is appending random strings to the password and then hashing it. In this method if data is leaked then without salt string hacker can not do anything with users data. Another solution would be compression after salting and hashing as it makes it even more difficult to crack.

Some password policies for preventing data breach due to password decryption are.

- Standardize Password Length and Combinations (At Least 10 words in password).
- Mandatory at least one special character, number and capital letter in password.
- Avoid common words and character combinations in your password.
- Don’t reuse your passwords.
- Don’t let users include their name, address, date of birth and other important & public information while creating a password.
- Companies can also generate random and strong passwords for users.

- Need to periodically reset your password within limited days.
- Don't use the same password for different websites.
- Provide multi-factor authentication to strengthen power on the user side.
- Atlast, random passwords are the strongest.

These are some of my recommendations on the password hashing and updated password policies.

Regards,
Alvine Moses.

Observations:

Type of hashing algorithm was used to protect passwords:

Md5(Message Digest). This is a one-way cryptographic function that accepts a message of any length as input and returns as output a fixed-length digest value to be used for authenticating the original message.

Level of protection the mechanism offers passwords:

MD5 is insecure and provides a very low level of protection and should not be used in any application. Using salted md5 for passwords is worse because it's fast. This means that an attacker can try billions of candidate passwords per second on a single GPU. It concludes that the MD5 provides very low security for password

Controls that could be implemented to make cracking much harder for the hacker in the event of a password database leaking again:

A min-length password rule should be implemented.
Passwords that contain some special characters,numbers,lowercase alphabets as well as upper case alphabets.
iii)Using a hashing algorithm which provides a high level of protection.
Example:SHA-256 and SHA-3.
iv)Concept of password salting must be used.

After cracking the passwords, what I noted about the organization's password policy:

- I'm not convinced there's a rule regarding the minimum length of the password, although the minimum length given the cipher-texts given seems to be 6(**short passwords**).

- There is no specific requirement for password creation. Users can use any combination of numbers, special characters and letters to create a password.
- The user can also use usernames as passwords.

What would you change in the password policy to make breaking the passwords harder

- I would increase the minimum password length requirement to 10 characters. This will increase the computational effort required to crack the password.
- I would discourage using memorable keyboard paths. Users should avoid using sequential letters and numbers, amd should further try avoiding sequential keyboard paths e.g qwerty. These are among the first to be guessed.
- I would set form validation to require a minimum threshold of a combination of alpha-numeric and special characters to be used in the password. Eg Minimum 2 special characters (/,#,*,... etc), letters in different cases and numbers.
- Prohibit users from using their date-of-birth, phone numbers and usernames as part of their password as such password combinations are again easy to crack.
- Educate users on the benefit of using password managers. Having a password manager will allow having very long and random passwords(EG: u5@8ekoFq$%gmdosA21wE) without the need to remember them.
- Encourage use of two-factor authentication. Even hackers that have cracked passwords aren't going to easily access user accounts if users follow this tip. Two-factor(2FA)/ Three-factor authentication(3-FA). These require you to know something (your password), and to have something e.g a one-time password(otp), fingerprint, phone prompt or a combination of some of these.
- Encourage users to employ unique passwords that don't follow a similar pattern. Eg. if previous password was **%J3nn1eI5Bomb** next password can't be **%J3nn1eI5Bomb3rr**
- Awareness should spread among the users of the different tools being used and how passwords are being cracked so easily so that there is transparency and the users themselves be careful
- Train the users to follow the password policies and create safe and easy to remember passwords that are strong and not so easy to crack.

Cracked Passwords:

	Hash	Hashing Algo	Hashing Decrypt
experthead:	e10adc3949ba59abbe56e057f20f883e	MD5	123456
interestec:	25f9e794323b453885f5181f1b624d0b	MD5	123456789
ortspoon:	d8578edf8458ce06fbc5bb76a58c5ca4	MD5	qwerty
reallychel:	5f4dcc3b5aa765d61d8327deb882cf99	MD5	password
simmson56:	96e79218965eb72c92a549dd5a330112	MD5	111111
bookma:	25d55ad283aa400af464c76d713c07ad	MD5	12345678
popularkiya7:	e99a18c428cb38d5f260853678922e03	MD5	abc123
eatingcake1994:	fcea920f7412b5da7be0cf42b8c93759	MD5	1234567
heroanhart:	7c6a180b36896a0a8c02787eeafb0e4c	MD5	password1
edi_tesla89:	6c569aabbf7775ef8fc570e228c16b98	MD5	password!
liveltekah:	3f230640b78d7e71ac5514e57935eb69	MD5	qazxsw
blikimore:	917eb5e9d6d6bca820922a0c6f7cc28b	MD5	Pa$$word1
johnwick007:	f6a0cb102c62879d397b12b62c092c06	MD5	bluered
flamesbria2001:	9b3b269ad0a208090309f091b3aba9db	MD5	Flamesbria2001
oranolio:	16ced47d3fc931483e24933665cded6d	MD5	Oranolio1994
spuffyffet:	1f5c5683982d7c3814d4d9e6d749b21e	MD5	Spuffyffet12
moodie:	8d763385e0476ae208f21bc63956f748	MD5	moodie00
nabox:	defebde7b6ab6f24d5824682a16c3ae4	MD5	nAbox!1
bandalls:	bdda5f03128bcbdfa78d8934529048cf	MD5	Banda11s