Dear Sir / Madam,

I was assigned the task of assessing the security of your company and looking for ways to minimize the probability of computer network breaches. I have found some major issues with regard to your password management system and protocol.

The password hashes provided to me for assessment were of poor quality indeed. I was able to crack thirteen out of the nineteen passwords using the **hashcat** *"legacy"* tool, and the once popular one hundred and thirty-two ega-bytes-sized **rockyou.txt** word-list file, which contains some thirty-two million routine passwords. The other six passwords seemed relatively stronger and safer.

For re-assurance, I tried another test, but this time using an ever larger word-list file, the thirteen giga-bytes-sized **Rocktastic** file provided by **Nettitude Labs**. This file contains just over one billion words, which are more than enough to attempt to decrypt any so-called *"weak"* passwords. The results were the same: the same thirteen out of nineteen passwords were cracked once again.

Now, if possible, those six uncracked passwords should form a preliminary basis to guide the clients in choosing their passwords appropriately. Regarding the cracked passwords, the problem rests on both sides: that of the client and the host. On the client's part, the passwords chosen were retro and hackneyed; very easy to remember for the individual, but also very easy to guess for the hacker. On the host side, at least the cracked passwords were verifiably hashed using the **unsalted MD5 hashing algorithm**, which, though not outdated, is most certainly not recommended for safe-keeping sensitive information. Its use is restricted to verifying the integrity of files transferred over a network.

The company seems to have no password policy implemented in place. Based on observing the cracked passwords (as shown in the enclosed appendix), and complying with standard professional norms, the following points could guide the clients and restrict them to use a strong password:

1. Setting a minimum password length restriction (eight to fifteen characters)
2. Prohibiting more than two consecutive instances of a character.
3. Requiring at least one lower-case, upper-case, numeric, and special character, each.
4. Periodically changing the password, and ensuring that the old passwords are never reused.
5. Prohibiting the use of personal details in the password. * (name, address, contact, birthday, family, etc.)
6. Advising the clients against using ordinary words in their passwords, perhaps by providing the client with a list, or, even better, by running the client's password through a validation process using one of the above word list files in order to determine the password strength.

* If possible, it would be best for the company to generate appropriate and random passwords for the clients, just as banks generate and provide credit/debit card PINs to their customers.

The host, too, could rely on the modern **SHA-256 salted hashing algorithm**. **SHA-512** is even better as the reproduced hash is twice the number of bytes compared to SHA-256, which would take a hacker an even exponentially longer time to crack the passwords. At the cost of time complexity, some of the most secure hashing algorithms include **BCRYPT**, **SCRYPT**, and **Argon2**. Though in the end, the emphasis is mainly on the client in choosing uncommon and unusual passwords.

Sincerely,

Melwyn Francis Carlo
(Governance Analyst)

APPENDIX : The Thirteen Decrypted Passwords

Sr. No.	Encrypted MD5 Hash	Decrypted Password
1.	E10ADC3949BA59ABBE56E057F20F883E	123456
2.	25F9E794323B453885F5181F1B624D0B	123456789
3.	5F4DCC3B5AA765D61D8327DEB882CF99	password
4.	FCEA920F7412B5DA7BE0CF42B8C93759	1234567
5.	25D55AD283AA400AF464C76D713C07AD	12345678
6.	E99A18C428CB38D5F260853678922E03	abc123
7.	D8578EDF8458CE06FBC5BB76A58C5CA4	qwerty
8.	96E79218965EB72C92A549DD5A330112	111111
9.	7C6A180B36896A0A8C02787EEAFB0E4C	password1
10.	6C569AABBF7775EF8FC570E228C16B98	password!
11.	3F230640B78D7E71AC5514E57935EB69	qazxsw
12.	F6A0CB102C62879D397B12B62C092C06	bluered
13.	917EB5E9D6D6BCA820922A0C6F7CC28B	Pa$$word1