Dear colleague.

The following is my finding for the company password cracking case. By using the Hashcat Tool, I have cracked some of the leaked password. The password is generated from these salt value.

```
e10adc3949ba59abbe56e057f20f883e : 123456
25f9e794323b453885f5181f1b624d0b : 123456789
d8578edf8458ce06fbc5bb76a58c5ca4 : qwerty
5f4dcc3b5aa765d61d8327deb882cf99 : password
e99a18c428cb38d5f260853678922e03 : abc123
fcea920f7412b5da7be0cf42b8c93759 : 1234567
3f230640b78d7e71ac5514e57935eb69 : qazxsw
f6a0cb102c62879d397b12b62c092c06 : bluered
8d763385e0476ae208f21bc63956f748 : moodie00
```

To tackle the task, I have used the MD 5 Hashing Algorithm. The MD 5 message digest algorithm is a relatively basic password hashing algorithm due to a rapid speed and memory conserving. This would lead to the fact that the attacker can compute the hashing process in a short period of time, leading the risk of the privacy of our company.

To solve this potential concern, a more advanced algorithm is highly recommended. For instance, SHA256 is a better tool for algorithm. Slower algorithm such as bcrypt is also encouraged to use due to a better security from a slower algorithm. Also, I would recommend using salts with hashes when it is applicable.

As for the organization password policy, I could notice that there is a usage weak hash function with no salting protection. Besides, passwords are too commonly used and easy to be cracked due to its simplicity. To enhance the protection, some capital letters, special symbols are highly recommended to be used. Therefore, I would recommend we can make some amendment for the password policy. For instance, we could extend the password length to 10 digits, prohibit over-simplified password patterns.

Please feel free to contact me if you need any further information and help.

Best Regards,
Jade