Dear Sir/Ma'am

Greetings,
After trying to crack all the leaked hashes, I found several vulnerabilities in your password policy and this report concludes all the findings and suggestions to improve your password policy.

1) What type of hashing algorithm was used to protect passwords?
   All the passwords were hashed using the MD5 algorithm.

2) What level of protection does the mechanism offer for passwords?
   All the passwords which were compromised were using MD5 which is a weaker hash algorithm and is prone to collisions with low level of protection. Therefore, making it very easy to crack with Hashcat.com and rockyou.txt wordlist via terminal and web browsers which has been demonstrated below. Hence, MD5 is not recommended to be used in any application.

3) What controls could be implemented to make cracking much harder for the hacker in the event of a password database leaking again?
   It is suggested to use a very strong password encryption mechanism to create hashes for the password based on SHA. Secure Hash Algorithm (SHA) and Message Digest (MD5) are the standard cryptographic hash functions to provide data security for authentication. There are several versions of SHA such as SHA-3 and SHA-256, all of them highly suitable for providing a high level of protection.

4) What can you tell about the organization's password policy (e.g. password length, key space, etc.)?

   After cracking the passwords, the following conclusion about organization's password policy were found:

   • Minimum length for password is set to 6.

   • There is no specific requirement for password creation i.e. the users can use any combination of word and letters to create a password.

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

   To make a better and optimized password policy the following set of rules can be mandated. The recommendations are:

   • Avoid common words and character combinations in the password.

   • Longer passwords are better, 8 characters is a starting point.

• Disable the use of previously used passwords.

• Inclusion of special characters, spaces, uppercase and lowercase letters, and numbers in password, all being optional to include except for one special character.

• Disable users to use their username, actual name, date of birth and other personal information while creating a password

• Train users to follow these policies to keep their passwords safe.

• Concept of password salting must be used.

• An external Api based tool which checks for password strength should show that the used password is strong.

- It must contain between 7 and 12 characters. Use only characters from the following set: ! # $ % & ( ) * + , - . / 0123456789 : ; < = > ? @ ABCDEFGHIJKLMNOPQRSTUVWXYZ [\ ] _ ` abcdefghijklmnopqrstuvwxyz { | } ~

-

- It must contain at least 1 lowercase letter(s) (abcdefghijklmnopqrstuvwxyz).

-

- It must contain at least 1 capital letter(s) (ABCDEFGHIJKLMNOPQRSTUVWXYZ).

-

- It must contain at least 1 numeric character(s) (0123456789).

-

- It must contain at least 1 character(s) from the following set: ! # $ % & ( ) * + , - . / : ; < = > ? @ [\ ] _ ` { | } ~

-

- It must not contain more than 2 identical consecutive characters (AAA, iiii, $$$$$ ...).

-

- It must not contain your user name.

-

- It must not contain your e-mail address.

-

- It must not contain your first name.

-

- It must not contain your last name.

Thank you.

Regards,
Mahi Prasad
B.Tech Computer Science and Engineering

Security Algorithms used:
```
experthead:e10adc3949ba59abbe56e057f20f883e - MD5
interestec:25f9e794323b453885f5181f1b624d0b - MD5
ortspoon:d8578edf8458ce06fbc5bb76a58c5ca4 -MD5
reallychel:5f4dcc3b5aa765d61d8327deb882cf99 -MD5
simmson56:96e79218965eb72c92a549dd5a330112 - MD5
bookma:25d55ad283aa400af464c76d713c07ad - MD5
popularkiya7:e99a18c428cb38d5f260853678922e03 - MD5
eatingcake1994:fcea920f7412b5da7be0cf42b8c93759 - MD5
heroanhart:7c6a180b36896a0a8c02787eeafb0e4c - MD5
edi_tesla89:6c569aabbf7775ef8fc570e228c16b98 - MD5
liveltekah:3f230640b78d7e71ac5514e57935eb69 - MD5
blikimore:917eb5e9d6d6bca820922a0c6f7cc28b - MD5
johnwick007:f6a0cb102c62879d397b12b62c092c06 - MD5
flamesbria2001:9b3b269ad0a208090309f091b3aba9db - MD5
oranolio:16ced47d3fc931483e24933665cded6d - MD5
spuffyffet:1f5c5683982d7c3814d4d9e6d749b21e - MD5
moodie:8d763385e0476ae208f21bc63956f748 - MD5
nabox:defebde7b6ab6f24d5824682a16c3ae4 - MD5
bandalls:bdda5f03128bcbdfa78d8934529048cf - MD5
```

In the terminal, run the command:
```
$ sudo hashcat -a 0 -m 0 hashes.txt /usr/share/wordlists/rockyou.txt -o
cracked.txt
```
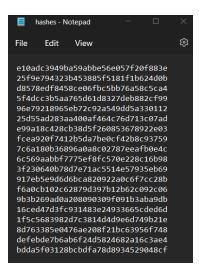
Command reference:
(-a) Attack mode: 0 (for dictionary attack)
(-m) Hash type: 0 (for MD5)
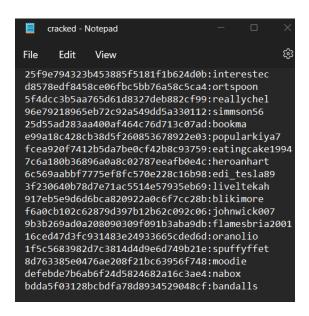(-o) File to store: crackedpass.txt (cracked passwords are stored here)

i) stored hash of passwords in `hashes.txt` file:

hashes - Notepad

File Edit View

```
e10adc3949ba59abbe56e057f20f883e
25f9e794323b453885f5181f1b624d0b
d8578edf8458ce06fbc5bb76a58c5ca4
5f4dcc3b5aa765d61d8327deb882cf99
96e79218965eb72c92a549dd5a330112
25d55ad283aa400af464c76d713c07ad
e99a18c428cb38d5f260853678922e03
fcea920f7412b5da7be0cf42b8c93759
7c6a180b36896a0a8c02787eeafb0e4c
6c569aabbf7775ef8fc570e228c16b98
3f230640b78d7e71ac5514e57935eb69
917eb5e9d6d6bca820922a0c6f7cc28b
f6a0cb102c62879d397b12b62c092c06
9b3b269ad0a208090309f091b3aba9db
16ced47d3fc931483e24933665cded6d
1f5c5683982d7c3814d4d9e6d749b21e
8d763385e0476ae208f21bc63956f748
defebde7b6ab6f24d5824682a16c3ae4
bdda5f03128bcbdfa78d8934529048cf
```

ii) ran the following hashcat command in the terminal:

```
└$ hashcat -m 0 -a 0 hashes.txt /usr/share/wordlists/rockyou.txt -o cracked.txt
hashcat (v6.2.5) starting
```

iii) results stored in cracked.txt file as following:

cracked - Notepad

File Edit View

```
25f9e794323b453885f5181f1b624d0b:interestec
d8578edf8458ce06fbc5bb76a58c5ca4:ortspoon
5f4dcc3b5aa765d61d8327deb882cf99:reallychel
96e79218965eb72c92a549dd5a330112:simmson56
25d55ad283aa400af464c76d713c07ad:bookma
e99a18c428cb38d5f260853678922e03:popularkiya7
fcea920f7412b5da7be0cf42b8c93759:eatingcake1994
7c6a180b36896a0a8c02787eeafb0e4c:heroanhart
6c569aabbf7775ef8fc570e228c16b98:edi_tesla89
3f230640b78d7e71ac5514e57935eb69:liveltekah
917eb5e9d6d6bca820922a0c6f7cc28b:blikimore
f6a0cb102c62879d397b12b62c092c06:johnwick007
9b3b269ad0a208090309f091b3aba9db:flamesbria2001
16ced47d3fc931483e24933665cded6d:oranolio
1f5c5683982d7c3814d4d9e6d749b21e:spuffyffet
8d763385e0476ae208f21bc63956f748:moodie
defebde7b6ab6f24d5824682a16c3ae4:nabox
bdda5f03128bcbdfa78d8934529048cf:bandalls
```

References:

[1]https://arstechnica.com/information-technology/2013/05/how-crackers-make-minced-meat-out-of-your-passwords/

[2] https://en.wikipedia.org/wiki/Salt_(cryptography)

[3] https://en.wikipedia.org/wiki/Cryptographic_hash_function

[4] https://en.wikipedia.org/wiki/Password_cracking#Software

[5] https://howsecureismypassword.net/

[6] https://hashcat.net/hashcat/

[7] https://md5decrypt.net/en/#answer