1. e10adc3949ba59abbe56e057f20f883e:123456
2. e99a18c428cb38d5f260853678922e03:abc123
3. 96e79218965eb72c92a549dd5a330112:111111
4. d8578edf8458ce06fbc5bb76a58c5ca4:qwerty
5. 3f230640b78d7e71ac5514e57935eb69:qazxsw
6. fcea920f7412b5da7be0cf42b8c93759:1234567
7. f6a0cb102c62879d397b12b62c092c06:bluered
8. 5f4dcc3b5aa765d61d8327deb882cf99:password
9. 25d55ad283aa400af464c76d713c07ad:12345678
10. 8d763385e0476ae208f21bc63956f748:moodie00
11. 25f9e794323b453885f5181f1b624d0b:123456789
12. 7c6a180b36896a0a8c02787eeafb0e4c:password1
13. 6c569aabbf7775ef8fc570e228c16b98:password!
14. 917eb5e9d6d6bca820922a0c6f7cc28b:Pa$$word1

* **What type of hashing algorithm was used to protect passwords?**

MD5 (MD5 Message Digest Algorithm)

* **What level of protection does the mechanism offer for passwords?**

Very little protection in the event of a password database leaking.

* **What controls could be implemented to make cracking much harder for the hacker in the event of a password database leaking again?**

1. Use Argon2id with a minimum configuration of 15 MiB of memory, an iteration count of 2, and 1 degree of parallelism.
2. If Argon2id is not available, use bcrypt with a work factor of 10 or more and with a password limit of 72 bytes.
3. For legacy systems using scrypt, use a minimum CPU/memory cost parameter of (2^16), a minimum block size of 8 (1024 bytes), and a parallelization parameter of 1.
4. If FIPS-140 compliance is required, use PBKDF2 with a work factor of 310,000 or more and set with an internal hash function of HMAC-SHA-256.
5. Consider using a pepper to provide additional defence in depth (though alone, it provides no additional secure characteristics).

Strong passwords stored with modern hashing algorithms and using hashing best practices should be effectively impossible for an attacker to crack.

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

From the cracked passwords and their usernames, it can be determined that the organization allows users to keep passwords short (6 characters), that too without a combination of numbers and letters, and also reuse usernames as part of passwords.

E.g. qwerty, 111111, and moodie00

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

1. Salting can be implemented to prevent usage of rainbow tables to speed up cracking.
2. Increasing the minimum password length requirement to 8 or more will increase the computational effort required to crack password.
3. Prevent passwords to be similar to usernames and implement using a combination of letters (upper and lower case), numbers, and special characters in the password.
4. Substitute some letters for numbers or special characters.

E.g. ‘E’ -> 3, ‘a’ -> @, ‘S’ -> $, etc

5. Random letter capitalization in the password makes it difficult to crack the password.

6. Since remembering such long passwords can be tricky, ensuring that the password translates to a paraphrase can be helpful in remembering them.

7. Do not reuse passwords or put personal information in them.