|  |  |  |  |
| --- | --- | --- | --- |
| **Cracked Leaked Passwords Using Hashcat** | | | |
| **Username** | **Hash** | **Type** | **Password** |
| 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** |

**I determined the following queries while I was cracking the passwords by using the MD5 hash algorithm:**

Q1: What type of hashing algorithm was used to protect passwords?

A: **MD5** or **MD4** (Raw Hash) are used to protect the passwords.

Q2: What level of protection does the mechanism offer for passwords?

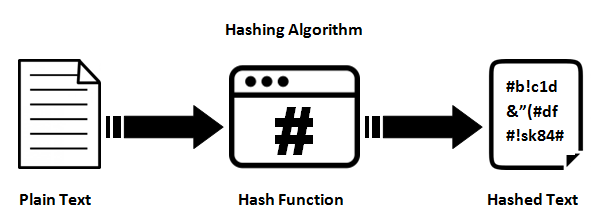
A:

* MD5 is an “**iterative**” hash function.
* MD5 is generally a **considerable mechanism** for storing passwords in production.
* MD5, produces a **128-bit hash.**
* MD5 is born out of **RSA’s algorithm** (defined in Internet RFC).
* MD5 is a utility that can **generate a digital signature of a file**. MD5 belongs to a family of one-way hash functions called **message digest algorithms**. The MD5 system is **defined in RFC 1321**.
* The algorithm takes as input a message of **arbitrary length** and produces as output a **128-bit "fingerprint" or "message digest"** of the input. It is conjectured that it is **computationally infeasible** to produce two messages having the same message digest, or to produce any message having a given prespecified target message digest. The MD5 algorithm is **intended for digital signature applications**, where a large file must be **"compressed"** in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as **RSA**.

Q3: What controls could be implemented to make cracking much harder for the hacker in the event of a password database leaking again?

A:

* One way of making the password hard to crack is by **maintaining credentials from multitude of services in a manager** like dashlane because they tend to use **varied hashing** algorithms & even hashing over hashed passwords [e.g. md5(md5($plaintext)) ] to store and keep the **strength high**, meeting to the rigidity of a strong case for an algorithm to process.
* **Reduce redundancy** across services such that in case of a leak out of one service doesn’t make the **other passwords vulnerable**.
* **Use alphanumeric character** with **special characters**.
* Reducing occurrence of an **adjective on noun or verb** which is an obvious prey to brute force attacks.



Q4: What can you tell about the organization’s password policy (e.g. password length, key space, etc.)?

A: It can be very well determined that the organization's **password policy is not up to the mark** as:

* The key length is at an **average of 11 or more**.
* Although they do not allow spaces, the use of **special characters is probably resisted** to a set of common delimiters like ‘\_’.
* The use of **numbers increases the resistance** of password by a factor of **10 times the digit appears**.
* The **lack of capital characters** splits the password strength by half.
* **Not avoiding the occurrence of English verbs** like book, popular, eating, hero, life, interest, expert in turn making the password vulnerable to brute force attacks.

Q5: What would you change in the password policy to make breaking the passwords harder?

A:

* Keeping a **threshold on length**.
* **Caution** over use of **verbs are nouns or adjectives**.
* **Mandating** minimum **3 special characters and minimum one capital letter**.
* Applying a **hashing algorithm over another**, recursively to have a strong hashing function e.g. md5(strtoupper(md5($plaintext)))
* **Not allowing sibling credentials** **to assist** the password naming, like name / surname / date of birth / sex.

I cracked some passwords using the MD5 hash algorithm :

happy123 --- 648605c11bdb955e16e7febc8db54103

hello@321 --- 31e1e408794fa74e3c81d7bedc653d2b

loveu@143 --- 2912e42b4006ccb983637573e2ba159a

mike$$hai --- 085fd7933d7d77016a4574cae7454d8d

mikeloves@u --- f486571d89f9748d46947cdc5f0ae86b

sachsandme --- 22e2efe8c2dd190741ba44012af48628

mahii@cool --- 4b12fcf0a015b60b877947a8f654fa0f

iloveyoumylove--- aa80fcd4b5dbf0b8f67ae53e36b8ccc3

hellopassword --- 62908bf72c21a3d8eaa23a55dec98e4b

hai123456 --- be8a6e19c968ff48e71b4e19c7d5ac98

smith@543 --- 9179c06a9246af417643e3b58cfd681c

georgeandme --- 9cda78653a67ae3fd5e2f8c507c586f8

smithgeorge --- 36c7e8b61f23af801d81ed0201af2ec8

kelvinandtemp --- 3296e0781297d71c94bff098a50132fd

software@me --- daf27c5167fe9b4a672907069d36889e

enginner123 --- f872b81d2e229d2290aec345ec2383ab

robert@scientist --- 71f1144aa32464ca9fb107ff556fb4e9

nick@doctor --- 52e100c1297866eec307845ad46c26d5

priyalovesme --- 68c09b6925f9008e8e2020b0756172a6

pradeep@rkp --- ff69513c9a019f0f2a28dc005b11f6c9

vinay1234 --- 73753d232df6ff960f2c84c5ecb2ae57

ushodaya@2015 --- ef50efd61c9e06d1c2fff0283cf7cbf0

singareni@mines --- 724db7c2de47d33ea01685cd29944f94

waited!@#jmet --- f6e127495d083b7e253c2d6b22bbe285

mining2015 --- b5ecce501e1dcdc6112cfdaff6e40865

scppolytechnic@2018 --- 93ab3cb1a6fc594a5d0eab74f9219031

iitbombay@2021 --- a2ce59fc53c4aef87aabed0e9c0b4089

gate2022@kharagpur --- a51bb51a00f4f3e64b30e39fee488d85

dream&&1234 --- 8faae79546b9d98f0ce62c38f0777910

246810@abc --- 14659bf6c36af5e39bc39577a11f9efb

24816@!cxa --- 1a616e51878623342ce7153133f155df

Loveyou\*\*\*321 --- a0d2c1c9a1ba8368e3127de14d2aa023

cracking!@#$%byjoe --- e005836b34e76ed818897a9dc9762d0f

lovecricket23 --- c08638727804bfd3c0a70df05d877af3

khokhogame --- 41d9dc8aea6c5662ab5aad6b86e66035

lepanga@kabadi --- 29d79ebc52820ab3e3c9b1551dc320a6

goodboy --- 57e7f266bb0dc62f2cb0f25976c14e93

yellowcolor --- b2df65ee5446b7e98d196c0aa8ec8e5c

blue@ok --- d9fa6315b3d0ff3118b613da79c6038b

kalpokisd --- 01a546b821c3ae1f4e955429ec5b3e78

bhhsvwytvv%7! --- 1839cb3e96c6c479803e8b92d6b25d1c

bvsdewq<>{}123 --- 9d8a7491481d1fe9be191aa580521b38

pinkyfrom#9 --- 9914d7394a207c5927394f34609ead66

bosslikesjohn&& --- f53d9de3e2da6857742f33a42ec2ca96

pollard@height --- 61863d15d5b6d689e3a2a586dd4a9f40

kingkhan987 --- 9e274a23b0b810b6e1adec24007b4219

price&profit --- c146e7ca3e3f1bdf928a80d4437515a5

time@distance --- 810c1d258c7eafd43d2c66c87d9f25d9

reasons1234 --- 6fd22b0354ea363a6cb8f55235f54857

crybefore%% --- 12b3629e914b8e937a33ec6a71ce4beb

livehappy --- f4e2ce32361e16998d22aace0ca153fe

marvelsdone --- 2be62453f1a86206e74df82acf32acc5

going2study --- 961783c294025ee9aa31382dee73ca2e

ridespeed7safe --- a7167ad87dd40b8003f7da49a26a23ae

armstrong@space --- 57f9600105135bf3439fc7cb93a210bc

petercollege --- 819d161e9f56c0dc8eeb92a621ec1dba

goal6754 --- 5037a04f9b9dd10a772ad15630653636

henrypot --- d8ae9b18b7c876a07812b52d39cd6fe7