What is a Rainbow Table? (MD5 Decryption Strategy)

infosecscout.com/what-is-a-rainbow-table

Patrick Fromaget

The MD5 algorithm is a one-way hash function, it's not reversible, so there is no way to decrypt a MD5 hash "automatically". However, current technologies allow us to use different strategies to crack MD5 hashes and find the original word. Using a rainbow table is one of them, and that's what I'll introduce in this article.

A rainbow table is a pre-generated file that is optimized for fast password cracking. It contains all the words like a dictionary, but also the hash equivalent. They take more disk space but are faster to use than other attack methods.

In this article, we'll see what why it's a good strategy, how it works and how you can generate them.

What are rainbow tables used for?

Rainbow tables are computed files containing hashes and their password equivalents. Using a rainbow table is a common attack used by hackers to crack passwords and find the clear text version of them from a hashed value stored in a database.

As a reminder, passwords are generally not stored in clear text in website databases. Most of the time, developers put some security layers in place to avoid any major issue if the database is stolen.

MD5 was one of the most common hash function used in the past to do this. Other algorithms are now being used, as MD5 is no longer safe to do this, but the idea is the same.

Hackers can use different strategies to crack password (dictionary, brute force and rainbow tables are the most common ones). In this article, we'll focus on the rainbow table attack, but you can find explanation about the other ones on this website if you are interested too.

How does a rainbow table attack work?

A rainbow table attack uses a pre-generated file containing hashes and their plain text equivalents to crack passwords stored in a database. If there is a match between a hash in the database and one in the rainbow table, the authentication is now possible, the password has been cracked.

Tools like RainbowCrack are often used to generate and use that kind of table (see next question). The generation process is complicated, but these tools will help a lot and using the tables once generated is straightforward, in short:

- A hash is identified in the website database
- The hacker run a search command to see if the hash is present in the rainbow table
- If there is a match, the hacker can now access the user account

The bigger the rainbow table is, the more chance there is to have a match.

Watch Video At:

https://youtu.be/O9E1PVZneqg

How to generate a rainbow table?

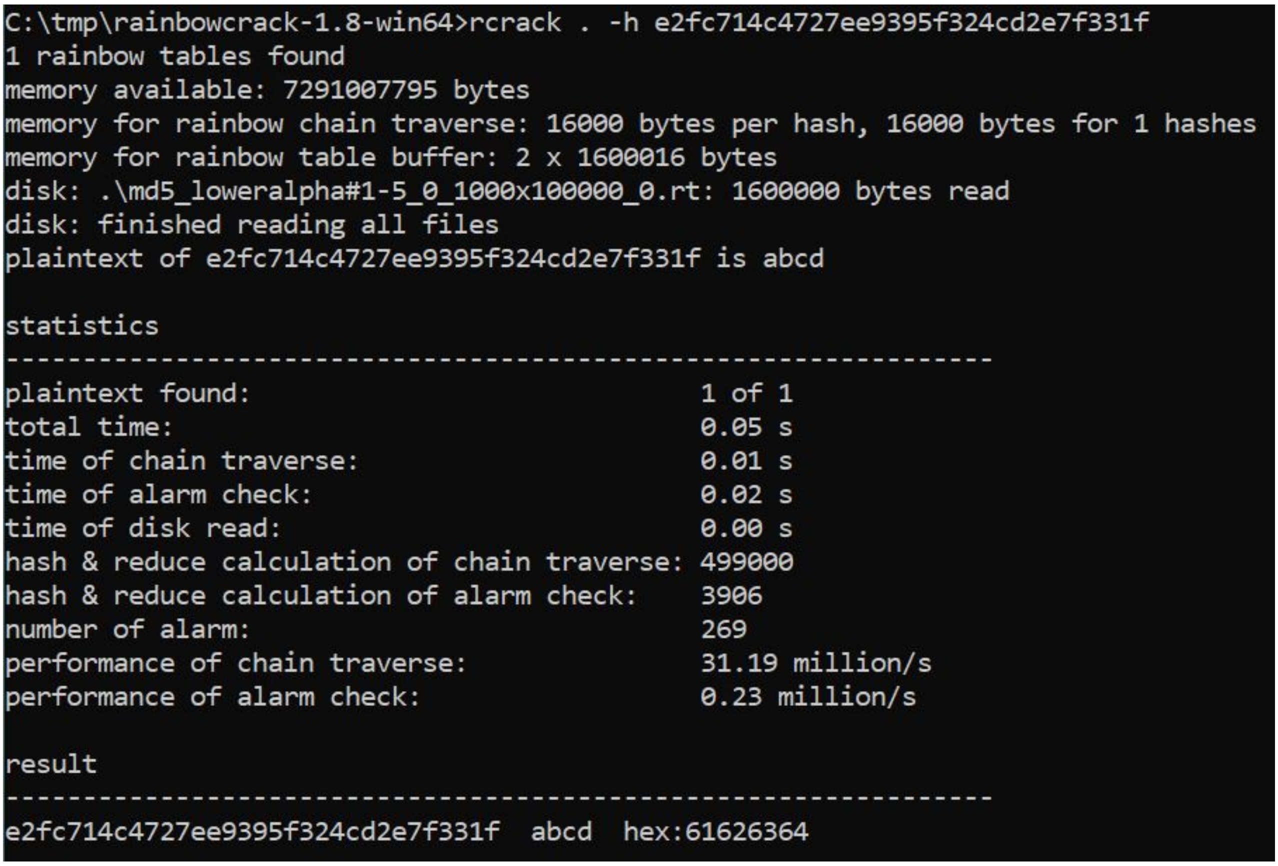
The easiest way to generate a rainbow table is to use a tool name RainbowCrack. It’s available on Windows and Linux and can generate tables for different hash algorithms (including MD5, SHA1 and SHA256).

To generate a table, you need to use the rtgen command. There are a few parameters required:

- **The algorithm**: for example MD5 is a common used algorithm in rainbow tables
- **The charset**: what kind of word it should generate in the file (digits, alpha-numeric in lower case, special characters, etc.).
- **The words length** (min and max).
- Table generation functions
- And the last parameter is to tell if you want to split the file in several small files or not

If you are interested in trying this, I explain everything in my book “The Secrets of MD5 Decryption“. I highly recommend reading it if you want to learn everything about the rainbow table strategy, but also the other listed previously (brute force and dictionary, for example).

The generation will take more or less time depending on these parameters. Once the rainbow table generated, you can use another command included in RainbowCrack (rcrack) to check if a specific hash is present in the table.
Here is an example with an MD5 hash:



```
C:\tmp\rainbowcrack-1.8-win64>rcrack . -h e2fc714c4727ee9395f324cd2e7f331f
1 rainbow tables found
memory available: 7291007795 bytes
memory for rainbow chain traverse: 16000 bytes per hash, 16000 bytes for 1 hashes
memory for rainbow table buffer: 2 x 1600016 bytes
disk: .\md5_loweralpha#1-5_0_1000x100000_0.rt: 1600000 bytes read
disk: finished reading all files
plaintext of e2fc714c4727ee9395f324cd2e7f331f is abcd

statistics
----------------------------------------------------------------
plaintext found:                             1 of 1
total time:                                  0.05 s
time of chain traverse:                      0.01 s
time of alarm check:                         0.02 s
time of disk read:                           0.00 s
hash & reduce calculation of chain traverse: 499000
hash & reduce calculation of alarm check:    3906
number of alarm:                             269
performance of chain traverse:               31.19 million/s
performance of alarm check:                  0.23 million/s

result
----------------------------------------------------------------
e2fc714c4727ee9395f324cd2e7f331f  abcd  hex:61626364
```

As you can see, my basic password (“abcd”) has been found in my rainbow table, the cracking is a success in this case, and the tool stop automatically.

How big is a rainbow table?

The size of a rainbow table will be between a few megabytes and several petabytes, depending on the character set used and passwords length. For example, a 7-characters lowercase only table will take 35 MB, while an 8-characters alphanumeric table will require 1 TB.

By using the different options in RainbowCrack, it’s possible to adjust the result to make sure it fits on your hard drive or server. But basically, make sure you have large disks if you are interested in testing this at a sufficient scale. If you don’t have space for more than 7-characters passwords, a brute force attack is probably a better strategy.

Tobtu.com has a tool that can help you to better estimate the file size you’ll generate with RainbowCrack. It’s not perfect, but it should give you a decent guess.

What is the best defense against rainbow table attacks?

As a general rule, the best defense against any attack, including rainbow tables, it to use long passwords, with a minimum complexity level and store them in a database by using a strong algorithm and salt. The longer the password are, the harder it will be to crack them.

Even if it helps, complexity is not really the most important factor to slow down hackers, length is. As explained in this article, there is a major difference in the attack duration for an 8-characters passwords and a 10-charaters password.

Users need to use longer passwords, even passphrase are pretty good if not too obvious, and developers must use salt to limit any major problem if a hacker get access to the database (on top of other security measures obviously).