Cracking hashes with Rainbow Tables by using RainbowCrack

Brennan Ormes

October 20, 2017

1 Introduction

A common thing for password crackers (aka "brute-force" tools) is that they usually make heavy use of the resources of the computer, using the processing power of the CPU to crack hashes. Where RainbowCrack differentiates from other brute-force programs is that it uses pre-computed tables (Referred to as "Rainbow Tables") that have both the plain text and hash of the password next to each other. I will go into more detail how these tables work further on in the report.

RainbowCrack was developed by Zhu Shuanglei and implements the improved version of the time-memory trade-off cryptanalysis attack which originally came from Phillipe Oechslin's "Ophcrack".

2 Context

RainbowCrack is used as a brute force tool, other tools would have to be used to get the hash (Such as password dump aka "pwdump" or other similar tools). Rainbow crack also has the following features:

2.1 Vulnerabilities

The vulnerabilities that this tool addresses is that both weak and strong passwords, depending on the hash, can be cracked easily. The tool shares similar functionalities to other password cracking tools in terms of what it is used for. The main reason why this tool is more useful than other password crackers is because of how the rainbow tables work, it is more feasible to crack complicated or longer passwords. It can be extremely effective against passwords that don't have a salt applied to them, however, depending on the strength of the salt, the password could still be cracked. A scenario could occur where the methods of salting the password do not use a random number to salt the hash with. However, there are limitations to this software, depending on character size on the complexity of passwords that are being cracked then the storage space to store these passwords can't be affordable or nonexistent.

2.2 Types of Attacks

The types of attacks that the tool carries out is brute-force attacks on hashes. The type of hashes the program can effectively crack are: lm, NTLM, md5, sha1, sha256. Word lists are replaced by rainbow tables for these attacks. Depending on the hashing algorithm used, determines the success rate of the table. On the developers' website, they have a list of tables and the effectiveness of each table listed beside it.

2.3 What other tools can be used with it?

2.3.1 pwdump

A common tool that can be used along with RainbowCrack is Password Dump (also known as pwdump) is a program that was created by Jeremy Allison. pwdump has gone through many iterations as it is a popular tool for breaking into Windows XP and 7 accounts. The tool can be used to output the LM and NTLM password hashes of local users accounts from SAM (Security Account Manager). The downside of using this tool for an attack is that the user must be on site and have access to the victim's computer and have an admin in order to use the tool.

2.3.2 HashID

HashID can be used to identify what type of hashes are used which can be useful to identify what rainbow tables you need for an attack.

3 Tool in detail

3.1 Core Functionality

The core functionality of the tool is for cracking hashes and using Rainbow tables to crack these hashes by using a time-memory trade-off (Less time, but more storage space is used in the preparation of these tables). The program has two main parts to it, "rcrack" and "rtgen". "rtgen" is used for generating the rainbow tables and "rcrack" is used to crack the hashes. There is another part to it called rtsort, which just arranges the rainbow table so it's quicker to go through the different chains and can remove "duplicate" chains that have the same final value as other chains.

Character sets can be defined inside the tools configuration files, which can allow more specific tables to be created if needed. For example, if you knew a password only consisted of the letter A to F and had the numbers 2 to 9 with a length of 10 (An example of BTHub2 passwords), then you can specify the table to create hashes only for this character list.

3.1.1 What are rainbow tables?

Rainbow tables are pre-computed tables that are used to break these hashes. The tables are made by using character lists that are pre-defined and can be created by the user. A very important feature of rainbow tables is the reduction function. The reduction function maps hashes to plaintexts. However, it is noteworthy that it does not reverse hash functions, but creates new. Basically, if you take the take the hash of a plain text that you are encrypting, and then take the reduction of that hash, it will give us another plain text.

An example of a reduction is that say we had a character set [0,1,2,3,4,5,6,7,8,9] and the password length is six, and the hashing algorithm MD5, we can get the following hash for the number 300466 which is bfae48de3cd88f762e6dffa44a6be3a2. So the reduction function for this, for example, could be taking the first six numbers of this hash which will be 483887 and from there, we hash that text and then apply the reduction function to it again. This process is called chaining, where it goes through the iterative steps:

- We get a plain text
- Apply a hash function to it
- From that hash apply the reduction function to it
- Apply the hash to the new plain text
- Rinse and repeat

The chaining will end at a certain hash. Once we have all the chains completed we will then be able to use the table to crack the hashes. So we use the table on a certain hash with a plain text that is unknown with it, and we want to check to see if it is inside the table (inside the chains we generated). Basically, the algorithm for checking the chains is:

- We look for hash in the list of our final reduced hashes, if it is there then break out of the loop
- If the hash is not there, then reduce the hash into another plain text (by using the reduction function as mentioned before) and then hash that new plain text
- Go back to the start
- If the hash matches one of our final hashes (That are on the end of the chains) then the chain for which the hash will match will be the chain that contains the original hash

The diagram figure 1 might make it easier to understand:

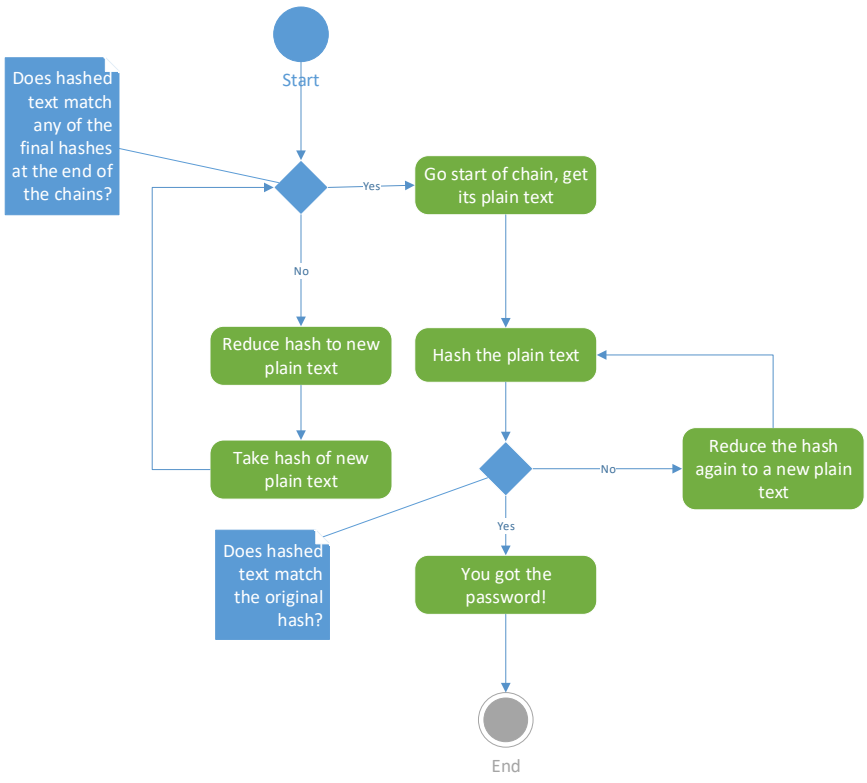

Figure 1: Algorithm for breaking hashes

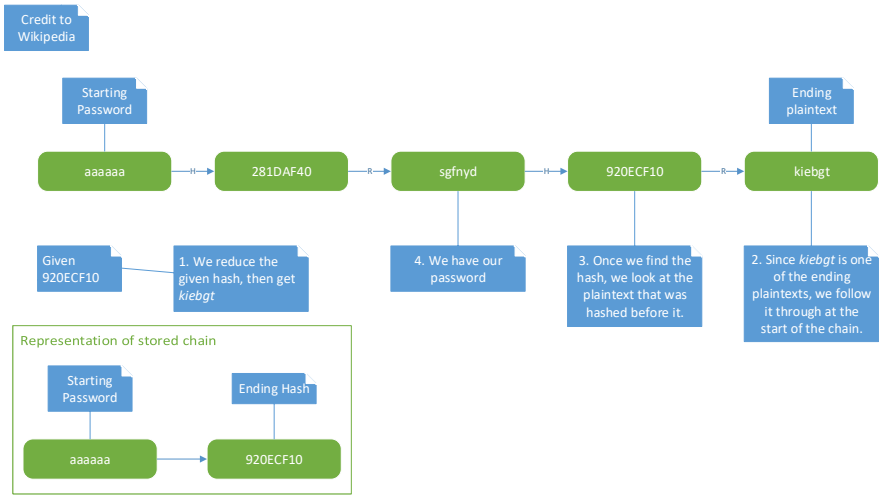

Figure 2: An example rainbow chain

The main goal of a chain within a rainbow table is that it decreases the amount of space required as each chain is represented by the final value hash. Basically, if a match is found, then we take the starting password of that chain and follow it to the point where it reaches the hash, so the plain text that was reduced for that hash will be the password.

Rainbow tables can also contain a salt for hashes which is extremely useful as a salt can be incredibly hard to crack and will take more time for a normal password cracker, but if the salt is solved already ahead of time, then there is a better chance of recovering a hash password.

3.1.2 Addressing collisions

Rainbow tables can solve the collision problem with normal hash chains by having multiple related reduction functions instead of a single reduction function. Basically for two chains to collide they would have to hit the same value on the same iteration. However, this can cause identical final values across multiple chains.

3.2 Walkthrough

The tool came pre-installed on my Kali Linux distribution, but you can get it at project-rainbowcrack.com

3.2.1 Generating a table

To generate a table we type in the following command

```
rtgen md5 loweralpha-numeric 1 7 0 3800 33554423 0
```

So we are telling the software to generate a table that uses the MD5 hash algorithm and use the loweralpha-numeric character set (a-z and 1-9, no symbols) and the length of the plaintexts must be between one and seven characters long. The first zero is the table index, this is to choose what reduction function is to be used. 3800 is our chain length, the longer the length the more plain texts that can be generated per chain but this will require more time to generate. 33554423 is the number of chains we are going to generate, the size of each rainbow chain is 16 bytes. And finally, the last zero represents the part index, which is for generating rainbow tables into smaller files. Now we will have to create the other tables:

```
rtgen md5 loweralpha-numeric 1 7 2 3800 33554432 0
rtgen md5 loweralpha-numeric 1 7 3 3800 33554432 0
rtgen md5 loweralpha-numeric 1 7 4 3800 33554432 0
rtgen md5 loweralpha-numeric 1 7 5 3800 33554432 0
```

Once these tables are finished generating, we will have six tables in total. The next step is to sort the tables (The rtgen program on my Kali Linux had the tables stored in the folder of the program, which was in /usr/share/rainbowcrack/). Once I was inside the folder where my tables were generated, I then used the rtsort command by simply typing:

rtsort .

This sorts all the tables within the current folder, it is extremely important to note that it is ill-advised to interrupt this process, as it can severely damage and corrupt the tables. Once our tables have been sorted we can now start using them to break hashes.

3.2.2 Breaking hashes with rcrack

So we have a text file called "secret.txt" which has the following hashes inside

fcaf8cb5751b2995c95f6c8021584eff
f061b402cb9fa744fba28ad4b5e0ad9c
f38bcf6b096ab2f078333b8b6a7b4ab6
0cf81f9038402e85910cfad17d0051b3
f5f0602cad64e18d5c6333f75979597e
2ccf6aa28316c337684fab8667192506
5bcb96fdfe4c211d97d545c780287869
6de0ea332a813e0f1fff0ac3c759b22b

We can crack each hash one by one by typing into the terminal:

rcrack . -h fcaf8cb5751b2995c95f6c8021584eff

or

rcrack . -l secret.txt (For cracking multiple hashes)

So once we have cracked the file we will get the following back: "h3ll0 w0rld th1s 1s f0r y0ur ey3s 0nly". We cracked these hashes very quickly, which would of taken longer for a normal brute force tool to do. Although this is a poor representation of the tools capabilities I do not possess the storage to create tables that could crack harder passwords. The syntax of the command follow:

rcrack <Path-to-table> <-h (single hash) or -l (text file with multiple hashes)>

In the above example, I just used a period instead of using a path to a specific table, as the period means that it will use any available tables inside the folder. From here, we can see that the tool can pick up how many hashes are within the text file and will start to crack each hash and output the plaintext for each hash. As seen, the tool is very fast for cracking multiple hashes.

4 Tool Evaluation

4.1 What advantages does the tool give?

The tool has a time-memory trade-off (Less time, but more memory is used, however) which means that because all the hashes are already made before the brute force happens, then they don't need to hash the word list as they are comparing the hashes from the password you are trying to crack.

The memory trade-off is because of the rainbow tables. Rainbow tables can range from thirty gigabytes or so to well over one hundred gigabytes, the table stores both the plain text and the hash together which can cause the huge sizes of these tables. If the table is already generated before the attack has taken place (Say, for instance, the program and relevant table is carried) hashes can then be broken on the spot regardless of computer resources. There will be some differences when using a slower computer, but they will get a huge advantage over using other brute force tools, ones that require the plaintext to be hashed and then used which can be very CPU intensive.

The main advantage of these huge tables is that they can be reused (If the hashes are same). Unfortunately, if you were to store these tables you would need huge amounts of hard drive space to store these tables. Rainbow Tables take up fewer hard drive space than other conventional hash tables because of the chaining, which can take up a considerable amount of space. That being said, the size of rainbow tables are still huge. The reason for rainbow tables being smaller than hash tables is that the chains can cover more than one hash, without all the hashes being stored. Because the way rainbow crack goes through the chains as previously explained in this document.

If the tool is used on windows, the graphics processing units can be used to accelerate the process considerably by offloading most of the runtime computation to NVIDIA or AMD cards, which overall improves the hash cracking and table generation performance.

4.2 Advantages compared to other tools

As mentioned before, because the tool uses a time-memory trade-off it can be considerably faster than other password cracker programs. If you are able to get a hash from a computer, and if you already have the tables on a portable hard disk drive then you can crack the hash there and then instead of having to take it to your system, which can make attacks much faster.

5 References

- Business Insider. Rainbow Tables: How to Create & Use Them to Crack Passwords.
  [online] Available at: http://www.businessinsider.com/rainbow-tables-how-to-create-and-use-them-to-crack-passwords-2011-11?IR=T
- En.wikipedia.org. Rainbow table.
  [online] Available at: https://en.wikipedia.org/wiki/Rainbow_table
- Kestas.kuliukas.com. How Rainbow Tables work.
  [online] Available at: http://kestas.kuliukas.com/RainbowTables/
- Project-rainbowcrack.com. RainbowCrack - Crack Hashes with Rainbow Tables.
  [online] Available at: http://project-rainbowcrack.com
- Security.stackexchange.com. What are rainbow tables and how are they used?.
  [online] Available at: https://security.stackexchange.com/questions/379/what-are-rainbow-tables-and-how-are-they-used
- Terry Zink: Security Talk. How rainbow tables work.
  [online] Available at: https://blogs.msdn.microsoft.com/tzink/2012/08/29/how-rainbow-tables-work/
- YouTube. Password Cracking 201: Rainbow Tables.
  [online] Available at: https://www.youtube.com/watch?v=Vxui5ypbo3I
- YouTube. Rainbow Table: How it works.
  [online] Available at: https://www.youtube.com/watch?v=rv06bwwAQqM
- YouTube. Rainbow Tables - Web Development.
  [online] Available at: https://www.youtube.com/watch?v=SOV0AeHuHaQ