IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

BRUTE FORCE, DICTIONARY AND RAINBOW TABLE ATTACK ON HASHED PASSWORDS

Priyanshu Patel[1], Parul Goswami[2] Ankit Mishra[3], Sameera Khan[4], Ashutosh Choudhary [5]
Department of CSE, Amity University Chhattisgarh
Department of CSE, Amity University Chhattisgarh
Assistant Prof, Department of Electrical and Electronics, Amity University Chhattisgarh
Assistant Prof, Department of CSE, Amity University Chhattisgarh
Assistant Prof, Department of CSE, Amity University Chhattisgarh

Abstract – Hash Cracking is a tremendous hardware demanding job, Cracking hashes is not that easy, it can take hours, days, months even years but after those long computation attempts, we left empty handed. But with the right method and analogy we can crack those hashes in less amount of time. In this paper we are going to dive deeper in hashes and password cracking methods to find out which method and practice is best for Hash Cracking.

Keyword – Hashed Password, Digest algorithm, Dictionary Attack

I. Introduction

"Password based authentication" is till now the most common way of granting access due to its simplicity, despite many well documented flaws being proposed in this single step authentication. While progress has been made in storing passwords and maintaining the hashes from a technical point of view, humans remain the weakest factor among all. System always performs what it is said to do. This was already pointed out nearly 43 years ago in 1978, when Morris and Thompson addressed the issue of UNIX password security, identifying numerous issues and proposing several countermeasures. parulgoswami510@gmail.comIt was too fast, which.

made them prone to brute-force attacks or a general cryptanalytic approach called *key search*. To put things in picture, back then cracking a **6-lowercase** character (26 lowercase alphabet) password took 107 hours and **5 alphanumeric** (26 lowercase alphabet + 26 uppercase alphabet + 10 letters) passwords took 318 hours using an average PDP-11 (16-bit minicomputers). While today it takes hardly a minute to crack those hashes.[1]

Zviran and Haga observed that passwords which are created by users do not change much with time in this rapidly growing era of the Internet. Passwords are short in length and are easy to guess, because users always try to use common patterns to remember and are easy to recall like pet name, date of birth, surname, etc.

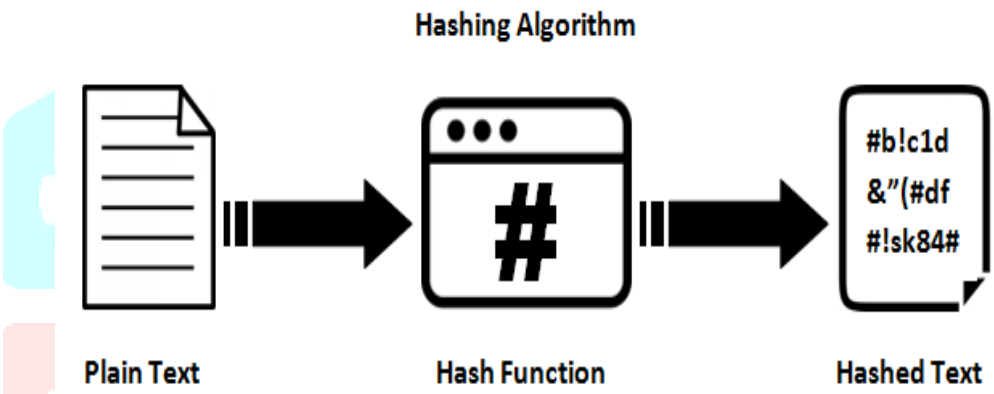
Most importantly, the level of data importance or sensitivity does not affect password composition. A set of guidelines for user-selected passwords

were adopted and mechanisms that monitor their implementation was also proposed.[2].

The aim of this study is to adopt a best password cracking method for password recovery and cryptanalysis as well. We are going to dive deeper to find out which is better, **Brute-force attack** or **rainbow table attack**.

1 - TYPES OF HASHES

Hashes are one-way functions which cannot be reversed.



A. MD4 HASH

MD4 (Message-digest algorithm) is a cryptographic hash function developed back in 1990 by Ronald Rivest. MD4 produces a 128-bit length digest. Security of MD4 hash has been severely compromised. The report of the 1st full collision attack against MD4 was published in 1995 and many newer attacks on MD4 have been published since then. As a research paper published by Yu Sasak back in 2007, an attack can generate collisions in less than2 MD4 hash operations. A theoretical preimage attack also exists against MD4.

MD4 hashes are also used to compute NTLM password-derivate key digest on MIcrosoft Windows NT, xp, Vista, 7, 8, 10. [3]

Example -

MD4("Priyanshu Parul")

=

5db3556dc4654cb7e3a44ac9cbf290da(128-bit long and 32 Chars Hexadecimal)

B. MD5 HASH

MD5 (message-digest algorithm) is a widely used hashing algorithm till date. It's hashing function produces a 128-bit hash value. Although the motive behind the MD5 was initi+ally designed to be used as a cryptographic hash function. The MD5 algorithm has also proven issues within its cryptographic method, A collision is when two words have the same hash generated. For example , if you know that "abc" and "def" have the same generated hash (just an example) You can say that "123abc" and "123def" have also the same hash generated.And this is a bad property for a cryptographic hash function as you can guess a lot of derived words, but it can still be useful to check the integrity of a subject against unintentional corruption. But it is also used to store passwords in the database since it is a one-way (hashes produced by it can't be reverse) hashing algorithm. [4]

Example

MD5("Priyanshu Parul")

=320c91d66cf6449acaff9314aaf65fee(128-bit long and 32 Chars Hexadecimal)

C. SHA-1

SHA-1 (**Secure Hash Algorithm 1**) was first published in1995 by NSA US. SHA-1 is an irreversible cryptographic hash function which takes any length of input and produces a digest of 160-bit hash value or 40 digit long. SHA-1 Produces a message digest based onprinciples similar to those of MD4 and MD5 message digest algorithms but SHA-1 generates 160-bit digest. And MD5 is comparatively slower than SHA-1.

In Practical world SHA-1 has many branches. It used in verifying the integrity of message or data (in the process of data transmission there may be the a chances data corruption or data may be changed by someone intentionally so to avoid this SHA-1 hash before transmission and after transmission can be compared), Git also uses this and easily able to find the single variable change in million or billion lines of code, SHA-1 also uses for password verification(Databases use to store passwords in the form of hashes, when the password is entered, comparison between hashes are performed). [5]

D. SHA-256

SHA-256 was designed by NSA in 2001. SHA-256 (Secure Hash Algorithm). SHA-256, which became the successor of SHA-1, bears another name-SHA-2. It's not much harder to encode than SHA-1, and its 256-bit key has never been compromised so far. It is a mathematical operation run on digital data and produces a digest of 64 char. In length. The SHA-2 hash function is implemented in some widely used security applications and protocols, like TLS and SSL, PGP, SSH, S/MIME, and IPsec.SHA-256 partakes in the process of authenticating Debian software packages and in the DKIM message signing standard; SHA-512 is part of a system to authenticate archival video from the International Criminal Tribunal of the Rwandan genocide. SHA-256 and SHA-512 are proposed for use in DNSSEC.] Unix and Linux vendors are moving to using 256-bit and 512-bit SHA-2 for secure password hashing. [6]

E. SHA-512

SHA-512, or Secure Hash Algorithm 512, is a hashing algorithm used to convert text of any length into a fixed-size string of 512 bits (64 bytes). Originally published in 2001, SHA-512 was developed by the US Government's National Security Agency (NSA). SHA-256 is a function with up to 2^{64} bits of input, which are broken into 512-bit blocks, and with a 256-bit output. It is conjectured to be collision-, preimage-, and second-preimage resistant at (resp.) 128-, 256-, and 256-bit security levels, but it has the unfortunate property that givenSHA-256(m) but not m , it is relatively easy to compute,SHA-256($m \parallel p \parallel m'$) for an arbitrary suffix m' , where p depends only on the length of m. SHA-512 is a function with up to 2^{128} bits of input, which are broken into 1024-bit blocks, and with a 512-bit output. It is conjectured to be collision-, preimage-, and second-preimage resistant at (resp.) 256-, 512-, and 512-bit security levels, but it has the unfortunate property that given SHA-512(m) but not m , it is relatively easy to compute SHA-512($m \parallel p \parallel m'$) for an arbitrary

suffix *m*', where *p* depends only on the length of *m*. [7]

3. Execution of attack to adopt best practice.

Attacks can be of various types, but the aim is to extract the password out of it (hash). Ideal Hashes are one-way functions which are irreversible by nature. For example if we divide 10 by 3 (10 / 3) we get 3.33, but if we want to reverse it by multiplying 3 to it (3.33 * 3) we get 9.99, we never get 10 back in place. So these are irreversible hash functions. As mentioned above MD4, MD5, SHA-1, SHA-256 and SHA-256 are irreversible hash functions.

For our purpose we are looking at traditional Brute-force approach and Rainbow table attack

- Brute-Force Attack

Brute-Force attack is an attack in which uses a predefined set of values to attack a target and analyze the response until he/she succeeds. Success depends on the set of predefined values. If it's larger, it'll take longer , but there's a far better probability of success.A brute force attack, also mentioned as an exhaustive search, could also be a cryptographic hack that relies on guessing possible combinations of a targeted password until the proper password is discovered. The longer the password, the more combinations which can need to be tested. A brute force attack is usually time consuming, difficult to perform if methods like data obfuscation are used, and sometimes downright impossible. However, if the password is weak it could merely take seconds with hardly any effort. Weak passwords are like shooting fish during a barrel for attackers, which is why all organizations should enforce a strong password policy across all users and systems.

In a traditional brute force attack, the attacker just tries the mixture of letters and numbers to get a password sequentially. However, this traditional technique will take longer when the password is long enough. These attacks can take several minutes to many hours or several years, counting on the system used and length of password.

To prevent password cracking from brute force attacks, one should use long and sophisticated passwords. This makes it hard for attackers to guess the password, and brute force attacks will take an excessive amount of time.Account lockout is different to stop the attacker from performing brute force attacks on web applications. However, for offline software, things aren't as easy to secure.

Brute force is additionally used to crack the hash and guess a password from a given hash. In this, the hash is generated from random passwords and then this hash is matched with a target hash until the attacker finds the correct one. Therefore, the higher the sort of encryption (64-bit, 128-bit or 256-bit encryption) wants to encrypt the password, the longer it can fancy break.

Brute force attacks can also be an extremely useful way for IT professionals to test the security of their networks. Indeed, one of the measures of a system's encryption strength is how long it would take for an attacker to be successful in a brute force attempt. Although often used by criminals for illegal purposes brute force can offer a backup option for password recovery if other methods have been exhausted.[10]

II. DICTIONARY ATTACK

Length of password	Only Number	Mixed lower and upper case alphabets	Mixed Number, Lower and Upper case alphabets	Mixed Number, Lower, Upper case alphabets and Symbols
3	> 1sec	> 1sec	> 1 sec	> 1 sec
4	> 1 sec	> 1 sec	> 1 sec	> 1 sec
5	> 1 sec	> 1 sec	3 sec	10 sec
6	> 1 sec	8 secs	3 mins	13 mins
7	> 1 sec	5 mins	10 hours	17 hours
8	> 1 sec	3 hours	10 days	57 days
9	4 secs	4 days	153 days	12 years
10	40 secs	169 days	1 years	928 years
11	6 mins	16 years	106 years	71K years
12	1 hours	600 years	6K years	5M years
13	11 hours	21K years	108K years	423M years
14	4 days	778K years	25M years	5Bn years
15	46 days	28M years	1Bn years	2Tn years
16	1 year	1Bn years	97Bn years	193Tn years

A Dictionary attack is a sub-Brute-force attack method where a person or attacker runs through common words and phrases, such as those from a dictionary, to guess passwords. The basic idea behind it is that people using simple, easy-to-remember passwords across multiple accounts means dictionary attacks can be successful while requiring fewer resources to execute. Dictionary attack in hash cracking is useful. Assume we had a wordlist or a dictionary of common passphrases or passwords and a hash file of hashed passwords. The idea behind it is to convert text from wordlist or Dictionary to the cracking hash and compare it with the hashes if they are matched then we are successfully able to crack the hash. Else it's time taking and dependent upon the wordlist. [11]

Best case scenario is when the hashes are matched at the 1st attempt or 1st iterative loop. [12]

Worst case scenario is when the password is not present in the wordlist. The iterative loop runs for the number of lines in a wordlist and is unable to find the hashes.Using a dictionary attack we are trying to crack hashes such as MD4, MD5, SHA-1, SHA-256, SHA-512 with **hashcat(A GPU hash cracking software)**[9] using rockyou.txt wordlist. Rockyou.txt is a list of leaked passwords from a database of real users between 2005 and 2006.[13]

 # Performing this task with a GeForce GT 710 DDr3 2gb video card. Finding "jazzpanget" password in hashed form and "jazzpanget"(7122196th line) is located exactly at the center of rockyou.txt wordlist.

Hash Type	Hashed password ("jazzpanget")	Time taken	Hash Cracking speed	GPU Temp.	Wordlist
MD4	3368fd8658c095c23a34d47aa01045f2	7 secs	1212.5 Kh/s	47 ℃	rockyou.txt
MD5	443ee45a5cb5c9a88f24ce0c2bfe1d43	9 secs	1196.2 Kh/s	48 ℃	rockyou.txt
SHA-1	1edd0ef91939b40aa913e43b1ab7d9bbbcfc454f	10 secs	1193.1 Kh/s	48 ℃	rockyou.txt
SHA-256	2CB673853F1CF5BD771D573CF4418C2BBE06F7F0741EAE4E87EE105E11CC4EB9	14 secs	1144.2 Kh/s	48 ℃	rockyou.txt
SHA-512	D4AD606D50D7C200A5B2779C6C8566A3F2150095DF64BAA2A8A1304E4CE0F951002208373E8CBDDB0817B66815A43768BB55CFCF93AA1630BF864AC2	17 secs	837.3 Kh/s	48 ℃	rockyou.txt

	5A562C7B				

*1Kh/s = 1000 (One thousand) Hashes per second

III. RAINBOW TABLE ATTACK

A Rainbow table can be defined as a database of precomputed dictionaries of plaintext passwords and their corresponding hash values. The aim is to find out what plaintext password produces a particular hash value. Since databases and wherever hashes are used, they are stored in the form of hashes(MD4, MD5, SHA-1, SHA-256, SHA-512). The hashes are compared and verified to authenticate the user. Since quite one text can produce an equivalent hash, it's not important to understand what the first password really was, as long because it produces an equivalent hash.[14]

A rainbow table works by doing a cryptanalysis very quickly and effectively. Unlike bruteforce attack, which works by calculating the hash function of each string present with them, calculating their hash value then comparing it with the one within the computer, at every step.[15][16]. A rainbow table attack eliminates this need by already computing hashes of the massive set of obtainable strings. Underneath Rainbow table is brute forced by hashes in a sorted order [17][18][19]

Formation of Rainbow table.

Software: RainbowCrack 1.8

GPU: AMD Radeon RX 5700 XT

IV. Conclusion

Even though additional sophisticated approaches could be employed, the results of the executed attack speak for themselves. Not only was the alarming amount of the passwords cracked, but the attack was also completed in a relatively short time period. Furthermore, the available wordlists grow larger by the day while the computers get faster, allowing for these attacks to become even easier over time.

For dictionary-based attack we consider that the password is already present in the used dictionary (Wordlist). Since dictionary/wordlist-based attacks are user defined dataset search probability attacks. If hash matches, we get the desired result else not, so for comparative analysis we take Brute-Force attack and Rainbow table attack into consideration.

V. REFERENCE

[1] R. Morris and K. Thompson, " Password Security: A Case History," Commun. ACM, vol. 22, no. 1 1, p p. 5 9 4–5 97, Nov. 1979.

[2] Zviran, M., & Haga, W. J. (1999). Password security: An empirical study. Journal of Management Information Systems, 15(4), 161 (125 pages).

[3] Yu Sasaki, Lei Wang, Kazuo Ohta and Noboru Kunihiro: New Message Difference for MD4, The University of Electro-Communications(20 pages)

[4] https://en.wikipedia.org/wiki/MD5

[5] https://en.wikipedia.org/wiki/SHA-1

[6] https://en.bitcoinwiki.org/wiki/SHA-256

[7] https://en.wikipedia.org/wiki/SHA-2

[8] https://qph.fs.quoracdn.net/main-qimg-e977ecac3bbe0b8a7535b1fe19e8c428

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

[10] https://www.geeksforgeeks.org/understanding-rainbow-table-attack/

[11] https://project-rainbowcrack.com/table.htm

[12] Ayushi. "Article: A Symmetric Key Cryptographic Algorithm." International Journal of Computer Applications 2010; 1(14):1–4, DOI: 10.5120/331-502.

[13] Ekta Agrawal, Dr. Parashu Ram Pal, "A Secure and Fast Approach for Encryption and Decryption of Message Communication," International Journal of Engineering Science and Computing. Volume 7 Issue No. 5

[14] Abhishek Joshi, Mohammad Wazid, R.H. Goudar, An Efficient Cryptographic Scheme for Text Message Protection Against Brute Force and Cryptanalytic Attacks, Procedia Computer Science, Volume 48, 2015, Pages 360-366, ISSN 1877-0509.

[15] Suyash Verma, Rajnish Choubey, Roopali soni (2012): "An Efficient Developed New Symmetric Key Cryptography Algorithm for Information Security," International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 7, July 2012) 18.

[16] Mahajan, Prerna and Abhishek Sachdeva. "A Study of Encryption Algorithms AES, DES and RSA for security." Global journal of computer science and technology 13 (2013)

[17] Dr. Sandeep Tayal, Dr. Nipin Gupta, Dr. Pankaj Gupta, Deepak Goyal, Monika Goyal, "A Review paper on Network Security and Cryptography," Advances in Computational Sciences and Technology ISSN 0973-6107 Volume 10, Number 5 (2017) pp. 763-770

[18] T.Saravanan, Dr. S.Venkatesh Kumar, "A Review Paper on Cryptography-Science of Secure Communication," International Journal of Computer Science Trends and Technology (IJCST) – Volume 6 Issue 4, Jul-Aug 2018

[19] A. M. Qadir and N. Varol, "A Review Paper on Cryptography," 2019 7th International Symposium on Digital Forensics and Security (ISDFS) Barcelos, Portugal, 2019, pp. 1-6, DOI: 10.1109/ISDFS.2019.8757514.