IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Quantifying Password Security Estimating The Cost Of Cracking

Dr. S. Vinayagapriya[1], Ashwin S [2], Charan MS [3]

[1]Department of Electronics and Communication Engineering, St. Joseph's College of Engineering, TN, India,

[2]Department of Electronics and Communication Engineering, St. Joseph's College of Engineering, TN, India,

[3]Department of Electronics and Communication Engineering, St. Joseph's College of Engineering, TN, India,

Abstract— We try to estimate the costs that an offline password cracker would take if they wish to undertake an attack lasting L days and successfully crack all the passwords under the given password space hashed by algorithm A. This research will involve analysing the various factors that contribute to the overall cost of password cracking, such as computing power, time, and resources. We say that using cost to crack as password strength is better than time to crack a password.

.

Keywords— Password cracking, Prediction, Estimating, Cloud services, Offline Cracking, Costs. Entropy

I. INTRODUCTION

Password hashing algorithms are a critical last line of defense against an offline attacker who has stolen password hash values from an authentication server. An attacker who has stolen a user's password hash value can attempt to crack each user's password offline by comparing the hashes of likely password guesses with the stolen hash value. Because the attacker can check each guess offline it is no longer possible to lockout the adversary after several incorrect guesses.

An offline attacker is limited only by the cost of computing the hash function. Ideally, the password hashing algorithm should be moderately expensive to compute so that it is prohibitively expensive for an offline attacker to crack most user passwords e.g., by checking millions, billions or even trillions of password guesses for each user.

In general, users of computer systems and networks use their user names and passwords to authenticate to these systems. It is a common practice to store hash values of the passwords along with the respective user names in the password file of a computer system. Otherwise, an attacker who can access the password file can see the passwords and use them to access the systems.

Storing hash values would prevent an attacker from doing this trivial "password look up." However, storing hash values of passwords in the password file does not offer protection against some attacks [4], [5], [6], [10]as outlined below:

An attacker can do a bruteforce attack by guessing all possible combinations of characters from a given character up to a given length[6].

Moore's Law, formulated in 1965 by Gordon Moore, co-founder of Intel, states that the number of transistors on a microchip doubles roughly every two years. This observation, while not a scientific law, has held true for decades, leading to an exponential increase in computing power..
The adversary willing to spend or find people in similar field can work together and reduce the time all together.
Our study aims to estimate the costs of cracking passwords on a cloud instance and show that time is not the best measure of calculating the safety of a password.

II. Time vs Money

The time value can change by how much someone is willing to spend. So, if you focus on just the cost to crack the password, you get a more universal answer that is far easier to understand. Consider your Netflix account, which is worth about $23 a month, so having a password that would take $8,000 to crack might be a little overkill. But if you have millions in the bank, then having a bank password that takes billions of dollars to crack might be worth it. Using time as a metric involves estimating how long it would take to crack a password, while using money as a metric involves estimating the cost to crack a password. The pros of using time as a metric are that it is a more universal measure and can put into context what is being protected. The cons are that the time value can change depending on how much someone is willing to spend.

When we use money as a metric, it becomes easier to understand and can be more accurate, as it considers the cost of computing power and other resources needed to crack a password. A password that takes a lot to crack should be that much rewarding to the adversary. It brings into perspective the economics behind password cracking i.e. the actual value of passwords being hashed and protected. In this business-driven world where money plays a big role, A CEO will more likely to listen to security advise if we told them that their password can be cracked with just "X" dollars and can result in "X^N" USD in losses and scandal rather than it takes X days to crack it. Cybersecurity is very important in a world where with advancement in technology, the people depend on technology to make their life easier, it also makes hacking easier with the number of free resources available. With just $12.00/hr cloud instances with twelve RTX 4090 vhGPU are available, which is more than sufficient power to crack few passwords.

III. Estimating Costs

Passwords can be compromised through various methods, depending on how they are stored—whether in plain text (in which case, cracking is unnecessary), using hash functions, or encryption. The chosen method depends on the attacker's approach. Numerous reputable organizations have experienced breaches, often involving internal data leaks, compromised usernames and passwords, phone numbers, or partial leaks. These breaches result from critical security flaws or negligence. Upon obtaining a list, the attacker needs to identify where the passwords are stored and organize the data and. If passwords are found to be stored in a has the adversary devises a suitable method to crack them.

There are several softwares available with extensive capabilities and support for password cracking such as Hashcat, JackTheRipper, Ophcrack; each with their own set of advantages and disadvantages. They are used by many users to to employ variety of attacks with rules and some understanding of human made passwords. The next step would be choosing a suitable method to crack the password list. Dictionary attacks are useful against passwords which are very small or have already been leaked. This attack uses pre-existing list of commonly used password. These lists of passwords may be from previous breaches, leaks, word data bases, commonly used passwords etc…

If you have been using password that were in a breach (can be found by using haveibeenpwned) , uses simple words , or u reuse it between sites, then adversary is able to quickly find out your password if they were to be provided with a power hardware.

For our test we use Hashcat to find the cost it would take to crack passwords generated by password generators. A hardware dedicated to password cracking has High performance GPU and CPU , due to the need for parallelization inorder to compute the passwords quickly.

We find that the Amazon EC2 instance p4.d24x costs with 8 A100 tensor core gpu has hourly costs of 36$ and has a Hashcat benchmark at 1.MD5 - 556.8 GH/s and 2. Bcrypt - 1069.2 kH/s.

After Hashing our passwords with the Hash function (MD5 and Bcrypt) we are testing , we proceed to use Hashcat to bruteforce and find the time it takes for cracking the passwords. We repeat the steps for various character set and find out the max time it takes to crack the passwords.

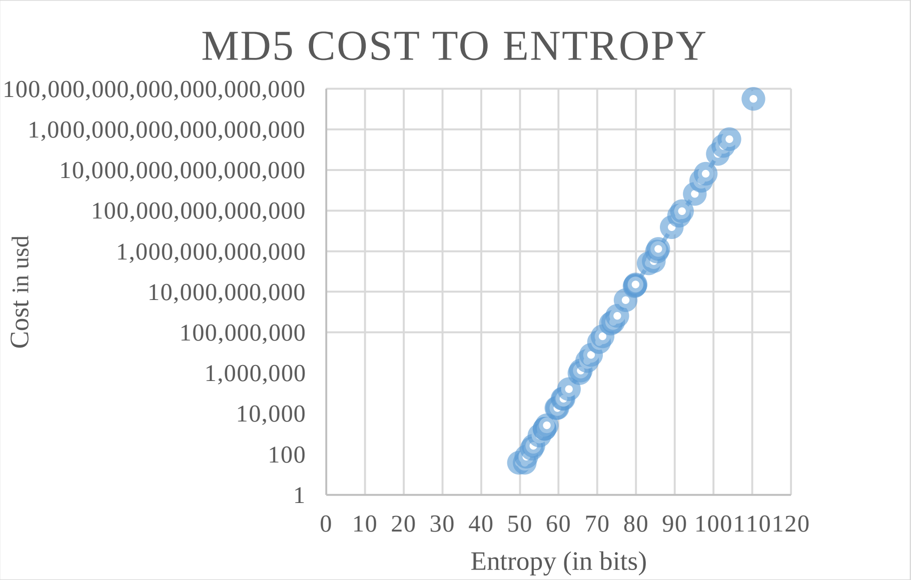
With the timings knows, we can calculate the costs for the attack using the formula C (in usd) = T x L x N

N = number of machines (in our case instances)

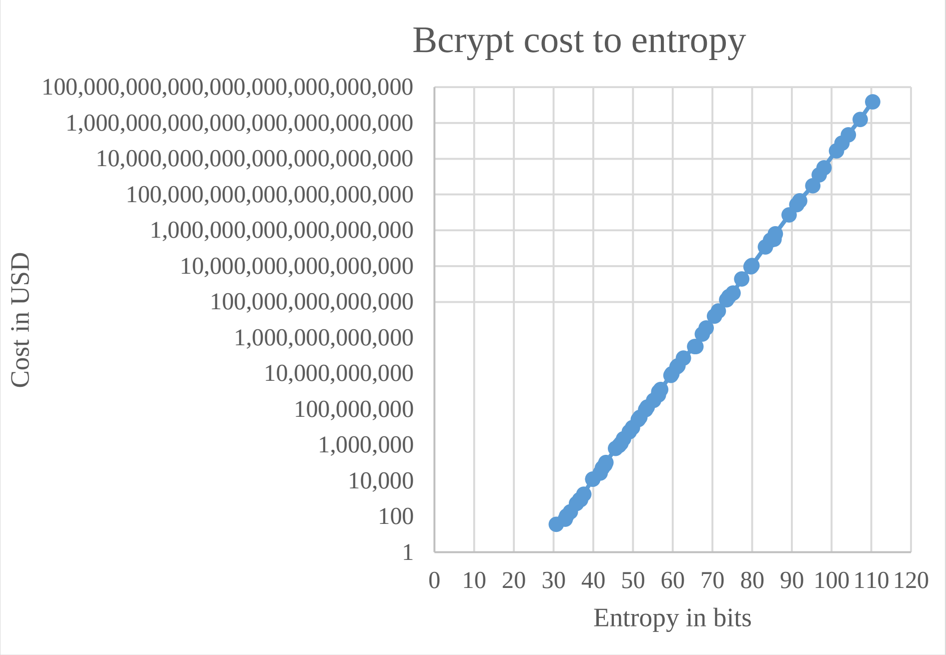
T = time in hours

L = cost per hour

IV. RESULT AND DISCUSSION



We find that cost to time increase exponentially but we also notice that there is a clear trend when we compare the costs to their corresponding entropy. There is a clear case of costs having an increase of 100% to 200% (doubling or tripling) for every bit increase in entropy , as shown in the graphs. Compared to the exponential graph of time to length, the linear graph of cost to entropy gives a much better insight into the capital required for password cracking. When we find the cost it takes to crack a certain entropy of password the costs for the rest of the variations become easy to predict with this revelation. This encourages more organizations to follow better security policies and Hashing standard, people will also learn better password practices to protect their accounts. The recoup from this situation for the adversary would be Cost recovery = total revenue - product costs. This cost should be in favour of the adversary if he wishes to undertake an attack thus allowing the defensive sides to work harder to improve algorithms and improve security measures and better estimate costs for the future.



VI .CONCLUSION

This research suggest that cost is a better parameter to test the strengths of passwords. No rational adversary would spend hours of time on cracking a password with no value. Since these are the maximum costs that would take an hacker and that these passwords are randomly generated by a password generator, If we were to consider the fact that most people don't use reputed password managers and just make up random passwords and reuse them the true costs would be even lower. Using cloud for Hacking can drastically improve the efficiency of the adversary. There are a lot of organizations still using old hash functions like MD5 which has been proven to be unsafe. Thus a lot of passwords in the world are not protected with standard methods. This study helps us understand the economics behind password cracking using cloud. We thus advise the use of memory hard functions like argon2 provide meaningful protection against offline attacks.

REFERENCES

1] Ankit Kumar Jain, Rohit Jones, Puru Joshi. "Survey of Cryptographic Hashing Algorithms for Message Signing", IJCST Vol. 8, Issue 2, April - June 2017

[2] Preneel, Bart., "Cryptographic hash functions", European Transactions on Telecommunications 5.4, pp. 431-448, 1994

[3] Mithilesh M Nimbalkar.,"Password cracking with brute force algorithm and dictionary attack using parallel programing" , JETIR 2023, Volume 10 , Issue 12.

[4] Jin Hong and Sunghwan Moon. A Comparison of Cryptanalytic Tradeoff Algorithms. Journal of Cryptology, 2012. Online version of this article is accessible at http://rd.springer. com/article/10.1007/s00145-012-9128-3.

[5] Antoine Joux. Algorithmic Cryptanalysis, chapter 5, pages 155–184. Chapman & Hall CRC Cryptography and Network Security Series. CRC Press, 2009

[6] Karen Scarfone and Murugiah Souppaya. Guide to Enterprise Password Management (Draft). NIST Special Publication 800-118 (Draft), 2009.

[7] National Institute for Standards and Technology. Federal Information Processing Standard (FIPS PUB 180-3) Secure Hash Standard, 2008.

[8] Quynh H. Dang, Elaine B. Barker "Secure Hash Standard, August 2002". NIST. FIPS PUB 180-2

[9] Ronald Rivest. The MD5 Message-Digest Algorithm. Internet Request for Comment RFC 1321, Internet Engineering Task Force, 1992

[10] James Michael Stewart. Comp TIA Security + Review Guide, chapter 6. John Wiley & Sons, 2011.