IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Issue 8, August 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.569

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 8, August 2021 ||

| DOI:10.15680/IJIRSET.2021.1008100 |

Implementation of Honeyword System for Security

Miss. Vrushali Thite, Prof. Dr.Mininath Nighot

Department of Computer Engineering, D. Y. Patil College of Engineering Ambi, Talegaon, Pune, India

ABSTRACT: in recent years, all of the activities of countries around the world have been carried out digitally, and all information or data has been shared over the network, increasing the speed and efficiency of data. Nevertheless, this transformation of data over the digital network poses a security risk, namely the loss of user data by third-party unauthorized persons or attackers. In order to preserve the security, every user is given a unique identifier or password, but the attacker still robs the password using several techniques to avoid this risk we use “honey words”. For detecting password database breaches, we propose a system called "Advanced Honey words System". The plan is to create several honey words, or false passwords, and store them alongside the real password. Because legitimate users are not required to recognize the honey words corresponding to their passwords, any login attempt with honey words is flagged as a password database compromise. Their concept includes honey word production, typo-safety measures to prevent false alarms, alarm policy upon detection, and checking the system's robustness against various attacks. If hacker trying to access user account and enter 3 times wrong password then hacker will get decoy file, also for each wrong password notification will go to admin and user. This will provide security. For each wrong password user and admin get notification.

KEYWORDS: Password, Honeywords, Password hash breach, Detection technique, Authentication, Security.

I. INTRODUCTION

Many businesses and software sectors store their data in databases such as ORACLE or Mysql, or other databases. Thus, in the database, the entry point of a system that needs the username and password is encrypted. Once a password file has been stolen, it is easy to capture most plaintext passwords using the password cracking technology. There are two issues that must be addressed in order to avoid this: The first passwords must be secured with the appropriate algorithm and protected. The second point is that the entry of unauthorized users into the system should be recognized by a secure system.

In the system we are concentrating on fake passwords and accounts. If any of the honey pot passwords is used, you can easily detect the admin. If the administrator creates user accounts and finds a password disclosure. According to the study, incorrect login trials with a certain password are recognized for every user and result in Honey Pot accounts. We have created and saved the password with the fake password set on the proposed system. We analyses the approach of honeywords and make a few comments about system security. In the case of unauthorized users trying to enter the system, the alarm is triggered and administrator will receive notification since unauthorized users receive decoy documents.

II. LITERATURE SURVEY

In this paper [1], they check the honeyword system and present some remarks to highlight possible weak points. Also, they suggest an alternative approach that selects the honeywords from existing user passwords in the system in order to provide realistic honeywords – a perfectly flat honeyword generation method – and also to reduce storage cost of the honeyword scheme.

In this paper [2], they create the honeyword, i.e. a false word, using a perfectly flat honeyword generation method. Hence that time they catch the unauthorized user and also the attacker not getting the original data. Now a day, rapid expansion of the accessing data from the network or from the other system the security becomes biggest issue. Storage of information also becomes the insecure because of attacker. And encryption of the data is also having some deficiencies. So the security of data is the latest issue. As the solution of above problem is the honey word generation is the best option for providing security to the data. In existing systems they only concentrated on the security of the password file but the different issues come. For solving this here we create the honeyword, i.e. a false word, using a perfectly flat

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 8, August 2021 ||

| DOI:10.15680/IJIRSET.2021.1008100 |

honeyword generation method. Hence that time we catch the unauthorized user and also the attacker not getting the original data.

In this system [3] the tools create fake accounts and how they manage to circumvent existing security measures. It also helps to get an insight into what websites do in order to handle fake accounts, both during the account sign-up process, as well as and after the fake accounts have been created.

Using deception techniques [4] (as in honeypots), they propose the user-verifiable authentication scheme (Uvauth) that tolerates, instead of detecting or counteracting, guessing attacks. Uvauth provides access to all authentication attempts; the correct password enables access to a legitimate session with valid user data, and all incorrect passwords lead to fake sessions.

Over the last year [5], there have been many high-profile login leaks, including LinkedIn, Yahoo, and eHarmony. Although you never want vulnerabilities that give hackers access to your password hashes, you also want to make sure that if the hashes are broken, hackers can't easily generate passwords from them. Large businesses are using poor hashing mechanisms that make it easy to break user passwords, as these leaks have shown. In this article, I'll go through the fundamentals of password hashing, look at password cracking software and hardware, and talk about how to use hashes safely.

In this paper [6], proposed comparing password distributions with a uniform distribution which would provide equivalent security against different forms of guessing attack, author estimate that passwords provide fewer than 10 bits of security against an online, trawling attack, and only about 20 bits of security against an optimal offline dictionary attack.

Author [7] create partial guessing metrics that include a new type of guesswork that is parameterized by the attacker's desired success rate. Our new metric is simple to calculate and directly applicable to security engineering. We estimate that passwords provide less than 10 bits of protection against an online trawling attack and just about 20 bits of security against an optimal offline dictionary attack by comparing password distributions to a uniform distribution that would provide equal security against various types of guessing attacks. Surprisingly little variation in guessing difficulty exists; every recognisable group of users created a password distribution that was similarly poor. Security motivations like registering a credit card have no more influence than demographic factors like age and nationality. Also constructive attempts to encourage users to use better passwords by providing graphical input are ineffective.

Authors create [8] partial guessing metrics that include a new type of guesswork that is parameterized by the attacker's desired success rate. Our new metric is simple to calculate and directly applicable to security engineering. We estimate that passwords provide less than 10 bits of protection against an online trawling attack and just about 20 bits of security against an optimal offline dictionary attack by comparing password distributions to a uniform distribution that would provide equal security against various types of guessing attacks. Surprisingly little variation in guessing difficulty exists; every recognizable group of users created a password distribution that was similarly poor. Security motivations like registering a credit card have no more influence than demographic factors like age and nationality. Also constructive attempts to encourage users to use better passwords by providing graphical input are ineffective.

In this paper [9] we studied one of the Honeyword generation method i.e. chaffing-with-tweaking provide some possible improvements which are easy to implement and introduce an enhanced model as a solution to an open problem also overcomes almost all the drawbacks of previously proposed honeyword generation approaches.

Author develop[10] an efficient distributed method for calculating how effectively several heuristic password-guessing algorithms guess passwords. Leveraging this method, we investigate (a) the resistance of passwords created under different conditions to guessing; (b) the performance of guessing algorithms under different training sets; (c) the relationship between passwords explicitly created under a given composition policy and other passwords that happen to meet the same requirements; and (d) the relationship between guessability, as measured with password-cracking algorithms, and entropy estimates. Our findings advance understanding of both password-composition policies and metrics for quantifying password security.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 8, August 2021 ||

| DOI:10.15680/IJIRSET.2021.1008100 |

III. System Overview

We know for any system main focus is protecting user password. Password plays important role. Leaked passwords make the users target of many possible cyber-attacks. In order to resolve these problems of defence, two aspects should beconsidered: First, passwords need to be covered with sufficient precautions and preservation using salting or some other complicated mechanisms for their hash values. Therefore, for an opponent to obtain plaintext keys, it must be difficult to invert hashes. Secondly, a safe device can identify the occurrence or non-accessibility of a login file leakage. We examine the approach of honeywords and comment on device security. In addition, we note that the main element of this approach is that honeywords are so created that they cannot be distinguished from proper passwords. Therefore, we suggest a new solution that uses other user's passwords in the scheme for honeyword sets. In addition, this approach decreases the cost of storage relative to the honeyword. We do use honeywords to detect password cracking in our proposed model. However, we prefer that you use current passwords to create honeywords rather than create the honeywords and add it to a password register. If an intruder types in a faulty password, the user will get notice and the administrator will then decoy file if the password enters incorrectly 3 times. Honey checker is used to store right indexes for each account and we believe that it authentically interacts with the main server through a secure channel. If any password mismatch then alarm will generate to admin that our password file is hack.

A. Architecture

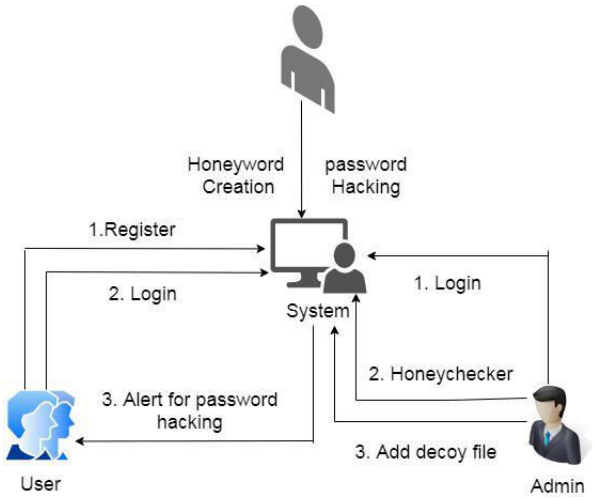

Fig. 1. Proposed System Architecture

USER:

Also user is going to register into system. Then while registration for give password by user system will gen-erate Honey Words and their Hash Values and Store into the table. Along with Hash Values the original password hash is also store at specific random position. An also user get one generated key for his uploaded file encryption and decryption.Here user is going to Login into the System. If password matches with the hash password then user can Login. Log creation is done for each user action to the system and which is store into the database.

ADMIN:

Here admin can Login into the system. Once login He can handle all administrative functions. Also admin add the decoy file for the uploaded file if unauthorized user tries password combination three times then he can get access to files but those file are Decoy files.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 8, August 2021 ||

| DOI:10.15680/IJIRSET.2021.1008100 |

HACKER:

Here hacker is going to login the system. Here if hacker tries to break the system and if he enters any honey word then the alert is given to the real user. And if suppose he try combination of password and it goes more than three attempt and also entered password does not match with the honey words then he is his get access the file but all files are decoy files.

B. Algorithm

Algorithm for Honey Word Generator:

1. Take input as a Position(pos) and Password(pass).

2. Reverse the Password.

3. Apply for loop from 1 to 20.

4. if(i == position)

realPassword[i] = pass; hashPassword[i] = generatorHash(pass); 5. else

realPassword[i] = replace(password1); hashPassword[i] = generatorHash(pass);

Fig. 1. passResult.put(”real”, realedPassword); passResult.put(”hash”, hashedPassword);
passResult is HashMap.
Fig. 2. return passResult;

C. . Mathematical Model

Input

Input given to the system is: - password.

Output

Check entered password is matched if not matched then send notification to user that any one try to login your system and if password is entered three times wrong then generate alert to admin that our password file is hacked. Send notification to user.

Process

Attack:

pz : is a assigned by system as a honeyword. K: honeywords are assigned to each account. pr :is the probability that pz. Ia assigned one of the honeyword for uy. Password Guessing: pr(Success): is probability that the adversary makes a correct guess for randomly picked username. T : no. of honeypots in system .

k : No. Sweetword.

Storage cost:

N stands for no. of users in system.

h:denotes length of password hash in bytes.
k: denotes no. of Sweetword assigned to each account.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 8, August 2021 ||

| DOI:10.15680/IJIRSET.2021.1008100 |

IV. CONCLUSION

In this system we analyzed the possibility of using Honeyword as an ideal mechanism to protect passwords if the problems it addresses could be resolved. We have indicated the strength of the honeyword system will be determined by the honeyword creation mechanism we have adapted. Another point we want to emphasize is that the system has to guarantee the low probability of DoS since it may present a serious threat. The proposed model has been compared to other models in terms of resistance, flatness and usability in terms of safety and usability compared to other models.

REFERENCES

[1] Imran Erguler," Achieving Flatness: Selecting the Honeywords from Existing User Passwords", DOI 10.1109/TDSC.2015.2406707, IEEE Transactions on Dependable and Secure Computing.
[2] Ms. Manisha B. Kale, Prof. D. V. Jadhav, " Security Analysis of Honey words Generation Scheme to Evade Unauthorized Access" , Department of Computer Engineering, Zeal College of Engineering and Research, Pune, India1, Tech. Rep. Issue 7, July 2016.
[3] A. Pathak, "An Analysis of Various Tools, Methods and Systems to Generate Fake Accounts for Social Media," Ph.D. dissertation, Northeastern University Boston, 2014.
[4] L. Zhao and M. Mannan, "Explicit Authentication Response Considered Harmful," in Proceedings of the 2013 Workshop on New Security Paradigms Workshop–NSPW '13. New York, NY, USA: ACM, 2013, pp. 77–86. [Online]. Available: http://doi.acm.org/10.1145/2535813.2535822
[5] K. Brown, "The Dangers of Weak Hashes," SANS Institute InfoSec Reading Room, Tech. Rep., 2013.
[6] A. Juels and R. L. Rivest, "Honeywords: Making Password cracking Detectable," in Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, ser. CCS'13. New York, NY, USA: ACM, 2013, pp. 145–160. [Online].Available: http://doi.acm.org/10.1145/2508859.2516671
[7] J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords," in Security and Privacy (SP), 2012 IEEE Symposium on. IEEE, 2012, pp. 538–552.
[8] D. Malone and K. Maher, "Investigating the Distribution of Password Choices," in Proceedings of the 21st International Conference on World Wide Web, ser. WWW '12. New York, NY, USA: ACM, 2012, pp. 301–310. Q111111111[Online]. Available: http://doi.acm.org/10.1145/2187836.2187878
[9] Z. A. Genc, S. Kardas, and K. M. Sabir, "Examination of a New Defense Mechanism: Honeywords," Cryptology ePrint Archive, Report 2013/696, 2013
[10] P. G. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, and J. Lopez, "Guess again (and gain and again): Measuring Password Strength by Simulating Password-cracking Algorithms," in Security and Privacy (SP), 2012 IEEE Symposium on. IEEE, 2012, pp. 523–537.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.569

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details