Generating Honey words using Toughnuts

Harikrishna Reghu[1], Gibi Thomas[2], Pranav P[3], Kavya Prasad[4], Akhija Lakshmi R[5]
[1,2,3,4] *B.Tech Student, Mount Zion College of Engineering, Kadammanitta, Kerala, India*
[5]*Assistant Professor, Department of Computer Science and Engineering, Mount Zion College of Engineering, Kadammanitta, Kerala, India*

***Abstract*- As society relies on digital world, the threat continues to quickly increase. once a year new mechanism against cyber security threats is introduced. At same time the cybercriminals in addition produce new techniques to beat these efforts. One in every of the important security issue is with revealing of positive identification file. To tune up this issue the construct of honey words i.e. false password is introduced. we tend to tend to point out an easy methodology for raising the protection of hashed passwords: the upkeep of extra "honey words" (false passwords) associated with each user's account. AN opponent WHO steals a file of hashed passwords and inverts the hash function cannot tell if he has found the password or a honey word. The tried use of a honey word for login triggers AN alarm. AN auxiliary server (the "honeychecker") can distinguish the user positive identification from honey words for the login routine, and can go off AN alarm if a honey word is submitted.**

Index Terms- Honey words, Authentication, Password

I. INTRODUCTION

Information security has become a most outstanding demand during this era that is finished mistreatment some authentication technique. Many different ways for authentication exists (e.g. Patterns, Passwords etc.). Now-a-days most usually used technique for authentication is passwords. Security of countersign is a crucial issue. A countersign could be a secret word, that a user should input during a login, solely at the moment it's potential to induce access. In password primarily based systems, developers should pay attention that passwords should not be keep in databases in plaintext or with unseasoned hash values. In past, several hacking makes an attempt had created possible for attackers to achieve unauthorized access to the sensitive information similarly as user passwords keep in information. The mechanism of Password protection helps us to protect info from unauthorized users. an opponent who has taken a file of hashed passwords will typically use brute-force search to search out a password p whose hash worth H(p) equals the hash worth keep for a given user's arcanum, so permitting the opponent to impersonate the user.

A recent report by Mandiant1 illustrates the importance of cracking hashed passwords within the current threat surroundings. password cracking was instrumental, as an example, during a recent cyberespionage campaign against the new Times. The past year has additionally seen varied status thefts of files containing consumers' passwords; the hashed passwords of Evernote's fifty million users were exposed as were those of users at Yahoo, LinkedIn, and eHarmony, among others. The LinkedIn passwords were mistreatment the SHA-1 algorithmic program while not a salt and similarly the passwords within the eHarmony system were additionally keep mistreatment unseasoned MD5 hashes. Indeed, once a countersign file is stolen, by mistreatment the password cracking techniques just like the algorithmic program of Weir et al. it's simple to capture most of the plaintext passwords.

Harley and Florencio projected a replacement approach to sight the malicious behavior on each incorrect or unauthorized login. for each single user false login makes an attempt with few passwords can generate honeypot accounts (fake accounts) so malign behavior is caught. Recently, Juels and Rivest have presented the honey word mechanism to sight an opponent who attempts to login with cracked passwords. The construct is that for every username they build a collection of sweetwords during which one word is the real password and also the others area unit honey words (false passwords). once an opponent tries to induce access mistreatment any of the honey word an alarm is triggered that notifies the administrator concerning the countersign file breach.

This approach isn't very deep, however it ought to be quite effective, as it puts the opponent in danger of being detected with each tried login employing a password obtained by brute-force resolution a hashed password. Consequently, honey words will offer a really helpful layer of defense. In any case, our hope is that this paper can facilitate to encourage the utilization of honey words.

II. AIM

Objective for this project is listed below:

- Monitoring data access patterns where system will generate honey words to keep user data secure.
- Decoy data will be stored in the database, alongside the users real data also serve as sensors to detect illegitimate access or exposure is suspected.
- To validate the alerts issued by the anomaly detector that monitors user access behavior.
- Launch a disinformation attack by returning large amounts of decoy information to the attacker.

III. SCOPE

Honey words are used in authentication system The main aim of project is to validating whether data access is authorized or not when abnormal information access is detected.

1. Confusing the attacker with fake information.
2. This protects against the misuse of the user's real data.
3. Here, we propose a different approach for securing the cloud using decoy information technology, that we have come to call fog computing.
4. We use this technology to launch disinformation attacks against malicious insiders, preventing them from distinguishing the real sensitive customer data from fake worthless data.

IV. RELATED WORKS

A. Juels and Rivest propose a basic technique for enhancing the security of hashed passwords.

The support of extra "honey words" connected with every client's record. A foe who takes a file of hashed passwords and transforms the hash capacity can't tell on the off chance that he found secret key or a honey word. The endeavoured utilization of a honey word for login sets off an alert. A helper server can recognize the client secret key from honey words for the login schedule, and will set off an alert if a honey word is submitted.

B. Bojinov, Bursztein, Boyen and Boneh explain methodology about Kamouflage

Loss-safe Password Management is a framework to secure the secret word database on a cell phone from assaults that are frequently disregarded by conveyed watchword administrators. The framework influences our insight into client watchword determination conduct to generously build the normal online work required to abuse a stolen secret key database.

C. Imran Erguler have investigated the security of the honey word framework

In this appreciation, we have brought up that the quality of the honey word framework straight forwardly relies on upon the era calculation, i.e. flatness of the generator calculation decides the shot of recognizing the right secret word out of particular sweetwords. They likewise give system to resistance against. They clarify diverse honey word era technique told about shortcoming and focal points of honey word era strategies.

V. SYSTEM CONFIGURATION

A. System Overview

The basic system requirements would be enough for the smooth functionality of the program since there is no use of hardware components. The basic operating system that the device should be having is Windows 7. The operating system helps in ensuring the smooth running of the program with the essential Integrated Development Environment Tool (IDE).

B. NetBeans 7.2.1

NetBeans is an Integrated Development Environment for Java. It allows the development of applications using a modular approach. NetBeans platform is a framework for simplifying the development of Java based desktop applications. NetBeans is the tool that is essentially used for the development of the

program by bifurcating it into modules such as admin module and user module.

C. MySQL

MySQL is a relational database management system which is open source. MySQL is used by many database driven web applications like Wordpress. It can be built and installed manually from the source code. However it is more commonly installed under a binary package until special customizations seem necessary. MySQL is used here for the storage of data with respect to the user module which can be monitored by the admin module.

VI. PROPOSED SYSTEM

Here we propose the development and use of generating honey words with the help of Chaffing with Toughnuts technique. In this technique four types of input are collected from the user:

1. A letter
2. A number
3. A word
4. A Mobile Number

All these details are acquired from the user which are called honey words. These honey words are provided into a hashing algorithm with the help of which these honey words are converted to a hash code. Each digit from each one of the honey words in collected and then the next digit from each one of the honey words is the order which is maintained for providing the honey word as input into the algorithm. The hashing algorithm used here will be Secured Hashing Algorithm(SHA-1). The user can thus login using the hash code once it is generated which makes it hard for the intruder to find out in comparison to the conventional passwords as these hash codes are longer in size and have no similarity with the honey words nor the conventional passwords as they are hexadecimal digits which are acquired as output from the algorithm and given to the user.

A. HoneyChecker

Honeychecker is the auxiliary server that is employed inorder to keep records with the existing user password and each time the login attempt is made. The Honeychecker thus has to maintain a close eye on the database where the userid and password are stored once the user is registered. Further which each time there is a login attempt either by user or intruder, the honeychecker checks with the database stored in MySQL and allows entry only if the passwords match. In any other case they would deny the entry.

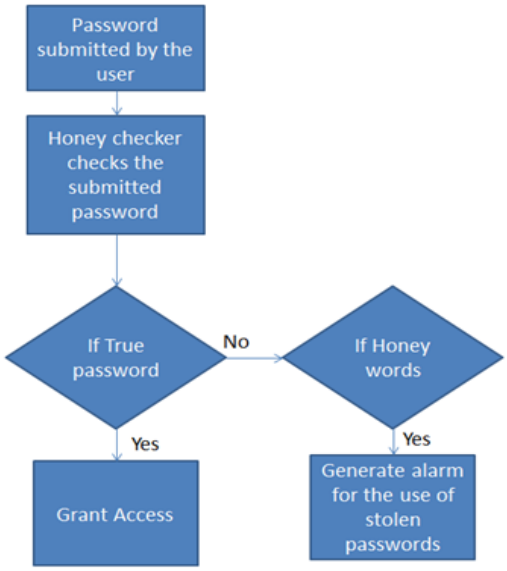

Fig.1. Block Diagram of HoneyChecker

B. MODULES

The paper mainly discusses about two modules throughout.

1. *USER MODULE*

The user module is where the user enters their details and interacts with the program. A new user, at first, needs to register by entering their personal details which includes name, mobile number, date of birth, mail id which acts as the user id. Upon entering these details the user is provided with an otp to the mobile number. The user can login at first with the user id which is the mail id and the otp as password. Further logging in, the user needs to complete a set of three security questions that will enable the user to get another otp just in case the user makes a misentry in the password. Then the user comes across honey words where the user needs to enter a set of four honey words which will be provided as input to the hashing algorithm. After the generation of the hash code the user needs to provide the hash code as password for further logins.

2. *ADMIN MODULE*

The admin module mainly monitors on the actions of the user. There is not much functionality provided to

the admin. The admin can also login through the login interface provided further which the admin can monitor the three sets of databases that are with respect to the user. These databases include the user details, honey word database and the security questions database. The admin can also view the otp that is assigned to each user.

VII. CONCLUSION

Honey word generation techniques has already been proposed with respect to security, usability, flatness, DOS resistance and storage. The use of decoy data mechanism will secure the confidential data of the authorized users from the hacker. In honey word based authentication approach, it is sure that the attacker will be detected. The main aim of project is to validate whether data access is authorized or not when abnormal information access is detected. Confusing the attacker with decoy data protects from the misuse of the user's real data. The admin keeps the data of the tracked IP's with them and use them to block access on their network. Use of honey words is very useful and works for every user account. With developing technology and further more developing tools that act as a threat to information security it is necessary that a system like honey word generation be used. Coming up with unique passwords seems almost impossible while posing an issue of remembering the password. In such a scenario it is better to use honey word generation technique and use hash codes as password which gives a better level of security.

ACKNOWLEDGEMENT

The authors would like to thank everyone for having given us this opportunity to conduct this study. We would also like to thank Mount Zion College of Engineering and APJ Abdul Kalam Technological University for giving us this platform.

REFERENCES

[1] D. Mirante and C. Justin, ―Understanding password database compromises,‖ Dept. of Computer Science Polytechnic Inst. of NYU, New York, NY, USA: Tech. Rep. TR-CSE-2013-02, 2013.

[2] A. Juels and R. L. Rivest, ―Honey words: Making password cracking detectable,‖ in Proc. ACM SIGSAC Conf. Computer. Commun. Security, 2013, pp. 145–160.

[3] K. Brown, ―The dangers of weak hashes,‖ SANS Institute InfoSec Reading Room, Maryland US, pp. 1– 22, Nov. 2013, http://www.sans.org/reading-room/ whitepapers/authentication/dangers-weak-hashes-34412.

[4] A. Vance, ―If your password is 123456, just make it hack me,‖ New York Times, Jan, 2010.

[5] C. Herley and D. Florencio, ―Protecting financial institutions from brute-force attacks,‖ in Proc. 23rd Int. Inform. Security Conf., 2008, pp. 681–685.

[6] H. Bojinov, E. Bursztein, X. Boyen, and D. Boneh, ―Kamouflage: Loss-resistant password management,‖ in Proc. 15th Eur. Conf. Res. Comput. Security, 2010, pp. 286–302.

[7] J. A and L. R. R, "Honey words: Making Password cracking Detectable," in ACM SIGSAC conference on Computer & communications security, November 2013.

[8] Imran Erguler, "Achieving Flatness: Selecting the Honey words from Existing User Passwords," IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 2, pp. 284 - 295, February 2015.

[9] M. Dell'Amico, P. Michiardi and Y. Roudier, "Password Strength: An Empirical Analysis," INFOCOM'10: Proceedings of the 29th Conference on Information Communications, vol. 10, pp. 983-991, 2010.

[10] Mirante, Dennis and Justin Cappos, "Understanding Password Database Compromises," Dept. of Computer Science and Engineering Polytechnic Inst. of NYU, 2013.

[11] R. Morris and K. Thompson, "Password Security: A Case History,"Communications of the ACM, vol. 22, no. 11, pp. 594-597, 1979.

[12] P.G. Kelley, S. Komanduri, M.L. Mazurek, R. Shay, T. Vidas, LBauer, N. Christin, L.F. Cranor, and J. Lopez. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In IEEE Symposium on Security and Privacy (SP), pages 523–537, 2012.