ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

Honey Words "Detecting Password Cracking With Hacker Tracking"

Vinay Maslekar[1], Akshay Takale[1], Snehal Aher[1], Ashish Dhamal[1], Dr. Prashant Kumbharkar[2]

UG Student , Dept. of CSE, Dr.D.Y.Patil School of Engineering , Lohngoan Pune, Maharashtra, India[1]

Professor, Dept. of CSE, Dr.D.Y.Patil School of Engineering , Lohngoan Pune, Maharashtra, India.[2]

ABSTRACT: The new developments in the field of information technology offered the people enjoyment, comforts and convenience, but there are many security related problems. One of them is password file. Password files have got a lot of security problem that has affected millions of users as well as many companies. Password file is generally stored in encrypted format, if a password file is hacked or theft by using the password cracking techniques and decryption technique it is easy to capture or find most of the plaintext and encrypt passwords. For troubleshoot this here we produce the honey word password, i.e. a False password using a perfectly flat honey word generation method, and try to attract illegal or unauthorized user. Hence that time we find the unauthorized user. Here we also protect the original data from unauthorized user. As mentioned above, in this system we have used Honey words also called as Sweet Password Security Strategy.

KEYWORDS: Authentication, Decoy, Honey words, Login, Password, Security

I. INTRODUCTION

By and large in numerous organizations and programming businesses store their information in databases like ORACLE or Mysql or might be other. In this way, the section purpose of a framework which is required client name and secret key are put away in scrambled shape in database. Once a secret key document is stolen, by utilizing the watchword splitting strategy it is anything but difficult to catch the greater part of the plaintext passwords. So to avoid it, there are two issues that ought to be considered to defeat these security issues: First passwords must be ensured and secure by utilizing the fitting calculation. Furthermore, the second point is that a safe framework ought to distinguish the section of unapproved client in the framework. In the proposed framework we concentrate on the nectar words i.e. fake passwords and records. The manager deliberately makes client accounts and distinguishes a watchword exposure, if any of the nectar pot passwords get utilized it is effectively to identify the administrator. As per the study, for every client mistaken login endeavors with a few passwords prompt Honey pot accounts, i.e. pernicious conduct is perceived. In proposed framework, we make the secret word in plain content, and put away it with the fake watchword set. We investigate the nectar word approach and give a few comments about the security of the framework. At the point when unapproved client endeavors to enter the framework and get to the database, the alert is activated and gets notice to the head, since that time unapproved client get bait reports. i.e. fake database. For the most part genuine passwords are anything but difficult to identify and consequently hack the framework. So here the fundamental inspiration is to stay away from this sort of hacking by the making of nectar words. The human personality is unequipped for precisely putting away a lot of information. Indeed we can now and then not by any means recall that one secret word effortlessly. This is the reason a nectar word based security framework is expected to spare essential records from going into wrong hands that can control critical information for a wrong utilize and mischief somebody by and by or hurt the entire business or organization. Utilizing this procedure the primary client simply needs to recollect that one unique secret key that he sets for the record. Whatever remains of it is dealt with by the working of the nectar word security set up.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

II. RELATED WORK

Paper Name	Author Name	Proposed System	For this paper we referred
1.Achieving Flatness: Selecting the Honeywords from Existing User Passwords	Imran Erguler	In this paper, theycheck thehoneyword system and present some remarks to highlight possibleweak 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 honeywordscheme.	1. In this paper, we have referred the solution to the detectionof password file disclosure events, and also referred how to reduce storage cost of the honeywordscheme.
2.Security Analysis of Honey words Generation Scheme to Evade Unauthorized Access	Ms. Manisha B. Kale and Prof. D. V. Jadhav	In this paper,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.	1. In this paper, we have referred how to create honeywords i.e. false word, using a perfectly flat honeyword generation method.
3.An analysis of various tools, methods and systems to generate fake accounts for social media	AvanishPathak	In this system 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.	1. In this system we have referred how to manage security of user accounts and how to handle fake accounts during sign-up process and after the accounts have been created.
4.Explicit Authentication Response Considered Harmful	Lianying Zhao and Mohammad Mannan	Using deception techniques (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.	1. In this paper we have referred password authentication of user instead of detecting or guessing attacks.
5. The Dangers of Weak Hashes	Kelly Brown	In this paper, discussed the basics of password hashing, look at password cracking software and hardware, and discussed best practices for using hashes securely.	1. In this system we have referred if we want to prevent our data or our password then we have to store passwords as hashes using strong encryption algorithms.

III. PROPOSED SYSTEM APPROACH

In the proposed system, we would like to refine our model by involving hybrid generation algorithms to also make the total hash inversion process harder for an adversary in getting the passwords in plaintext form from a leaked password

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

hash file. Hence, by developing such methods both of two security objectives – increasing the total effort in recovering plaintext passwords from the hashed lists and detecting the password disclosure – can be provided at the same time.

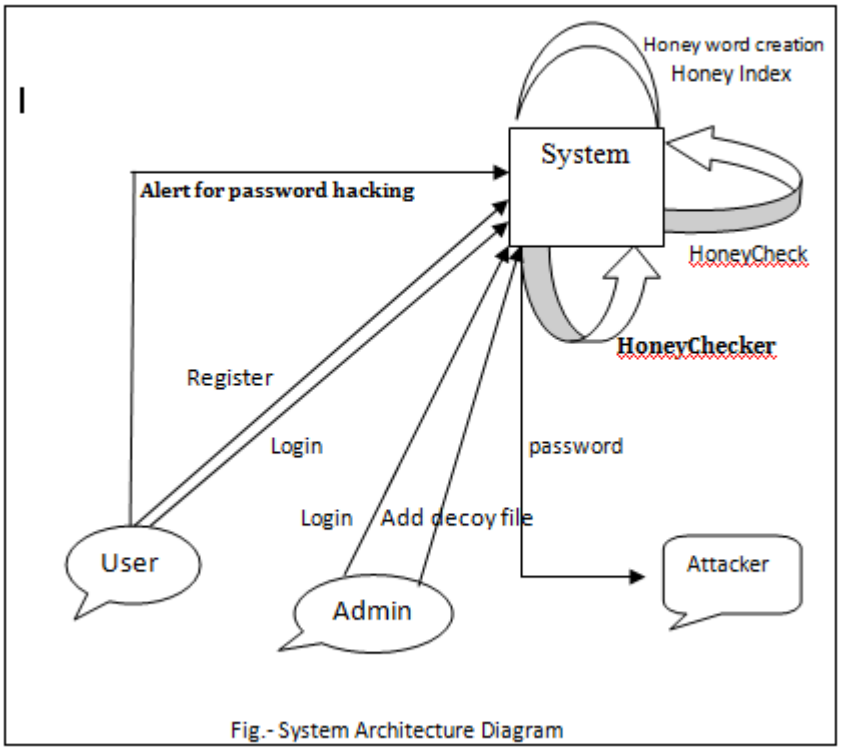


Fig.- System Architecture Diagram

IV. ADVANTAGES

1. System protects the original data from unauthorized user.
2. System protects against the misuse of the users real data.

V. WORKING

Dos Attack:

I. p_z:is a assigned by system as a honeyword.
II. K :honeywords are assigned to each account.
III. p_r:is the probability that p_z. Ia assigned one of the honeyword for u_y.

Password Guessing:

I. p_r(success): is probability that the adversary makes a correct guess for randomly picked username.
II. T : no. of honeypots in system .
III. k : No. Sweetword.

Storage Cost:

I. N stands for no. of users in system.
II. h:denotes length of password hash in bytes.
III. k: denotes no. of sweetword assigned to each account.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

Mathematical Equation:

$$P_r\left(p_{z\epsilon}\, W_y\right) = 1 - \left(\frac{N-m}{N}\right)^k \quad \text{----------(1)}$$

$$P_r\ (\text{success}) = \frac{N-T}{N}.\frac{1}{k} \quad \text{----------(2)}$$

For our approach we assume that each index requires 4bytes and the storage cost becomes:

$$4kN+hN+4N \quad \text{----------(3)}$$

To measure the gain in storage compared to original method, we give the ratio as

$$\frac{4kN + hN + 4N}{khN} = \frac{4k + h + 4}{kh}$$

So problem is NP Complete

Hashing:

Hash algorithms are one way functions. They turn any amount of data into a fixed-length "fingerprint" that cannot be reversed. They also have the property that if the input changes by even a tiny bit, the resulting hash is completely different (see the example above). This is great for protecting passwords, because we want to store passwords in a form that protects them even if the password file itself is compromised, but at the same time, we need to be able to verify that a user's password is correct.

The general workflow for account registration and authentication in a hash-based account system is as follows:

1. The user creates an account.
2. Their password is hashed and stored in the database. At no point is the plain-text (unencrypted) password ever written to the hard drive.
3. When the user attempts to login, the hash of the password they entered is checked against the hash of their real password (retrieved from the database).
4. If the hashes match, the user is granted access. If not, the user is told they entered invalid login credentials.
5. Steps 3 and 4 repeat every time someone tries to login to their account.

In step 4, never tell the user if it was the username or password they got wrong. Always display a generic message like "Invalid username or password." This prevents attackers from enumerating valid usernames without knowing their passwords.

It should be noted that the hash functions used to protect passwords are not the same as the hash functions you may have seen in a data structures course. The hash functions used to implement data structures such as hash tables are designed to be fast, not secure. Only **cryptographic hash functions** may be used to implement password hashing. Hash functions like SHA256, SHA512, RipeMD.

It is easy to think that all you have to do is run the password through a cryptographic hash function and your users' passwords will be secure. This is far from the truth. There are many ways to recover passwords from plain hashes very quickly.

Hashing is the transformation of a string of characters into a usually shorter fixed-length value or key that represents the original string. Hashing is used to index and retrieve items in a database because it is faster to find the item using the shorter hashed key than to find it using the original value. It is also used in many encryption algorithms.

Hashing password:

A hash function can be as simple as "add 13 to the input" or complex like a Cryptographic Hash such as MD5 or SHA1. There are many things that constitute a good hash function like:

I. Low Cost: Easy to compute
II. Deterministic: if I hash the input a multiple times, I am going to get the same output each time
III. Uniformity: The input will be evenly distributed among the possible outputs. This falls in line with something called the Pigeonhole Principle. Since there are a limited number of outputs we want f() to place those outputs

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

evenly instead of in the same bucket. When two inputs compute to the same output this is known as a collision. It's a good thing for a hash function to produce fewer collisions.

Hashing applied to Passwords:

The hashing of passwords is the same process as described above, however it comes with some special considerations. Many of the properties that make up a good hash function are not beneficial when it comes to passwords.

Take for example determinism, because hashes produce a deterministic result when two people use the same password the hash is going to look the same in the password store. This is a bad thing! However this is mitigated by something called a salt.

Parameter	Existing System	Proposed System
Duplicate User Role	Absent	Present
Notification	Using email	Using email as well as SMS
Speed of performance evaluation	Slower	Faster
Knowledge Sharing Point	Absent	Present
Speed of Administrative Actions	Slower	Fatser
Data Access	Slower	Faster

VI. IMPLEMENTATION AND ANALYSIS

The proposed system is implemented on three tier architecture in which the client interface is simply a web browser, XAMP 1.7.1 is configured as a web server, Java script is used as scripting language, MySQL Database connectivity. After successful completion of all the mentioned in proposed system, we are able to achieve the following aspects related to access control in any organization:

 Services offered enhances overall performance: Notification service, GUI, and Knowledge Sharing Point enhance the overall performance of the system and provides satisfaction to the users. This trade-off between precision and recall can be observed using the precision-recall curve, and an appropriate balance between the two obtained.

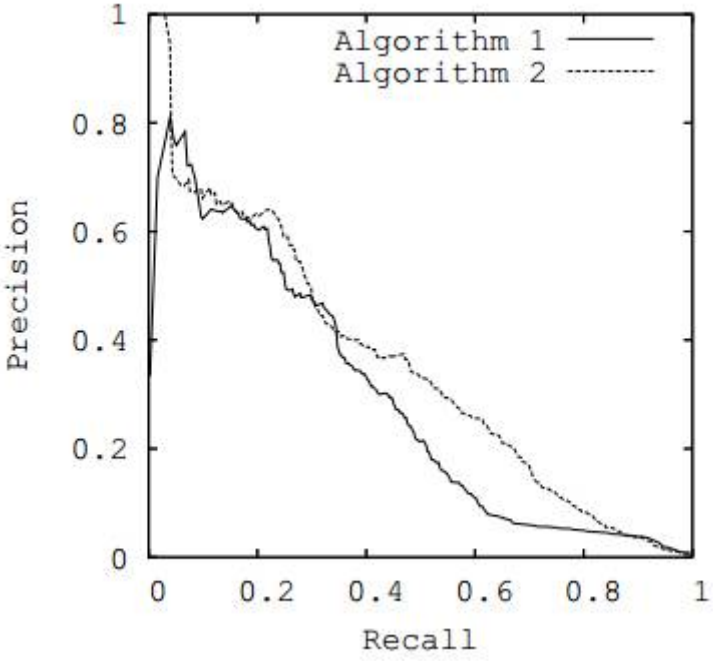
ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017



The precision-recall curves for two algorithms are shown. Depending on the requirement (high precision at the cost of recall, or high recall with lower precision), an appropriate algorithm can be chosen and get the Honey word result.

VII. CONCLUSION

We present a standard approach to securing personal and business data in the system. We propose monitoring data access patterns by profiling user behavior to determine if and when a malicious insider illegally accesses someone's documents in a system service. Decoy documents stored in the system alongside the user's real data also serve assessors to detect illegitimate access. Once unauthorized data access or exposure is suspected, and later verified, with challenge questions for instance, we inundate the malicious insider with fake information in order to dilute or divert the user's real data. Such preventive attacks that rely on disinformation technology could provide unprecedented levels of security in the system and in social networks model.

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 ofWeak Hashes," SANS Institute InfoSecReading 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.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

[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. [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.