Honeywords: A New Approach For Enhancing Security

Manisha Jagannath Bhole

Lecturer,Computer Engineering,V.P.M'Polytechnic Thane ,Maharashtra,India

---

Abstract: ***From the Previous Research it has been found that many password hashes were not safe by hackers. So the method of Honeywords (Decoy passwords) which is used to detect attacks against hashed password databases introduced. For an attacker it becomes easier to steal hash passwords and enter into the account through authenticate user by cracking the hash password. An adversary can recover a user's password using brute-force attack on password hash. A secure server called "Honeychecker" which can distinguish a user's real password among honeywords of each user and immediately sets off an alarm whenever a honeyword is used. In this research paper will examine 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.***
.

Key Words: *Authentication, honeywords, login, password, password cracking*

1. INTRODUCTION:

In the authentication process it becomes difficult to handle security of passwords that's why password became the most important asset to login. But users choose weak passwords (for easy to remember) that can be predicted by the attacker using brute force, dictionary, rainbow table attacks etc. So it becomes much easier to crack a password hash. An adversary can recover a user's password using brute-force attack on password hash. Once the password has been recovered no server can Detect any illegitimate user authentication. So Honeywords plays an important role to defense against stolen password files. Specifically, they are
bogus passwords placed in the password file of an authentication server to deceive attackers. Honeywords resemble ordinary, user-selected passwords. An auxiliary service called a honeychecker checks whether a password submitted by a user on login is her true password or a honeyword. The password system itself

stores a given user's password randomly along with honeywords. The past year has also seen numerous high profile thefts of files containing consumers' passwords; the hashed passwords of Evernote's 50 million users were exposed[19] as were those of users at Yahoo, LinkedIn, and e-Harmony, among others [18].One approach to improving the situation is to make password hashing more complex and time-consuming.

1.1 Contribution: This work analyzes the honeyword system according to both functionality and the security perspective. Then suggests improvements for number of honeywords per user and managing old passwords. Finally introduce an enhanced model of honeywords which may

2. LITERATURE SURVEY:

Users reuse the passwords for login high important account and the reason behind that was it easy to remember also passwords were extremely weak: being too short, containing lowercase letters only, digits only or a combination of the two, or being easily found in dictionaries or lists of names.[2]. Typical computer user suffers from password overload. Users still need education and assistance when choosing passwords for important accounts [3]. Algorithm Metropolis-Hastings that can guide users to distribute their passwords more uniformly without having to know a list of common passwords in advance[10]. LinkedIn, Yahoo, and eHarmony these sites have been suffered from several high publicity password leaks. LinkedIn passwords were using the SHA-1 algorithm without a salt and passwords on eHarmony system were also stored using unsalted MD5 hashesh. SHA-2 algorithm is cryptographically strong[16]. new mechanism Honeywords to detect an adversary who attempts to login with cracked passwords[17]. Erguler suggested a new approach that

selects the honeywords from the existing user password that provide Realistic honeyword.

2.1 Type of Attacks

There are numerous attacks to obtain a user's password. The six of these techniques are depicted in figure:

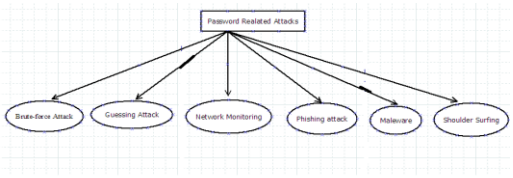

Fig-1: Type of Attacks

Password attacks can be classified as follows:

Brute-force attack: An adversary can steal the password hash file and crack the hashes using brute force computation. He may also use a precomputed dictionary of password hashes [6]

Guessing attack: Many users choose weak passwords such that an adversary can find out the passwords of some users of a system by trying common passwords while attempting to login to that system [4], [5]. Spafford suggest good password choice should avoid common words and names [13].

Network monitoring: If the communication between the user and the system is unsecured, i.e. unencrypted, an adversary may monitor the network traffic and obtain the passwords or interrupt the traffic while a user creating her password and change it to another one [12]. This attack is also called man-in-the-middle-attack [1].

Phishing attack: A user can submit her login information to a web page prepared by an adversary Which seems very likely to the original system's login screen? This technique is relatively new, the First attempt was reported in the mid-1990s [15].

Malwares: A Trojan program can capture the key strokes and send this information to the adversary [7]. There are some advanced malwares that can steal the login information from messenger like Software's some of which does not keep the login information encrypted [8]. Sun et al. proposed pass Which uses a user's cellphone and short message service (SMS) to prevent password stealing [14]?

Visible passwords: A password that is written to a stickie can be seen by an adversary. He can also Watch a user while she enters her password (shoulder surfing). Kumar et al. propose Eye Password, gaze-based password entry, to overcome direct observation [11].The authors in [9] focus on brute-force attack scenario where an adversary has stolen a file of user names and associated password hashes from the server.

2.2 Problem Statement

Most users use same password on different systems. An old password of a user on some system may be the current password of that user on another system. Thus taking advanced security measurements may not guarantee the safety. An adversary may attack to a weaker system that the targeted user have an account on it and obtain her old passwords and submit them on a more secure system. This scenario constitutes a security risk. Hence proposed the honeyword generation method Chaffing-With-Tweaking which removes Brute-force attack to some exetent .

3. METHODOLOGY:

Our proposed approach is identified as **Chaffing-With-Tweaking**. In this method, the user password seeds the generator algorithm which tweaks selected character positions of the real password to produce the honeywords. For instance, each character of a user password in predetermined positions is replaced by a randomly chosen character of the same type: digits are replaced by digits, letters by letters, and special characters by special characters .The honeywords system is only designed to withstand off-line attacks. In this scenario, we assume, as the authors mentioned in [9], that the adversary has only stolen the password hashes but did not compromise the system on a persistent basis, i.e, the adversary did not hack the system or did not gain the admin rights. However, the authors in [9] mentioned about a problem which we believe is still open: How can a honeyword system be best designed to withstand active attacks.

Proposed Modules in this paper are:

3.1 Design

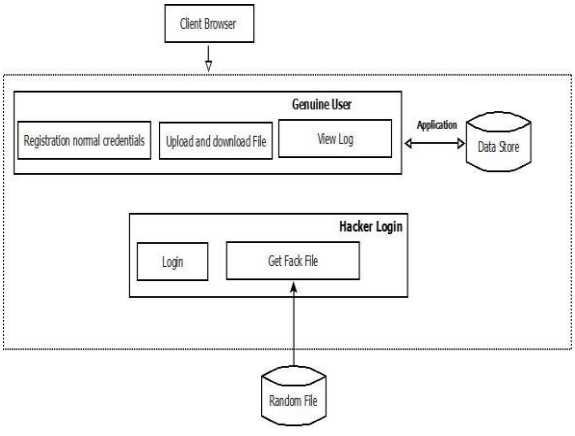

Fig-2: System Architecture

3.2 Modules:

Registration: Here user is going to register into system. Then while registration for give password by user system will generate HoneyWords 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.

Login: Here user is going to Login into the System. If password matches with the hash password then user can Login.

Hacker: Here hacker is going to login the system. Here if hacker tries to break the system and if he enters any honeyword then the alert is given to the Actual 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 honeywords then he is his get access the file but all files are decoy files.

File Upload and View: Authenticated user to the system can upload file into the System. And the uploaded file is encrypted by the encryption algorithm by the user encryption key. To view fie or download file user has to enter the decryption.

Admin Login: Here admin can Login into the system. Once login He can handle all administrative functions.

Decoy File Upload: Here admin add the decoy file for the uploaded file if unauthorised user tries password combination three times then he can get access to files but those file are Decoy files.

Log Creation: Log creation is done for each user action to the system and which is store into the database.

Valid User Behaviour Tracking: After valid user login, the system will track the valid user operations and track IP Address, mac address and data size of resources downloaded by each user per session.

User Behaviour Analysis : The parameters tracked above will be analyzed using similarity vector analysis to identify behaviour of each user. If invalid detected, the user will be delivered decoy data for all downloads.

According to above module floe of method is:

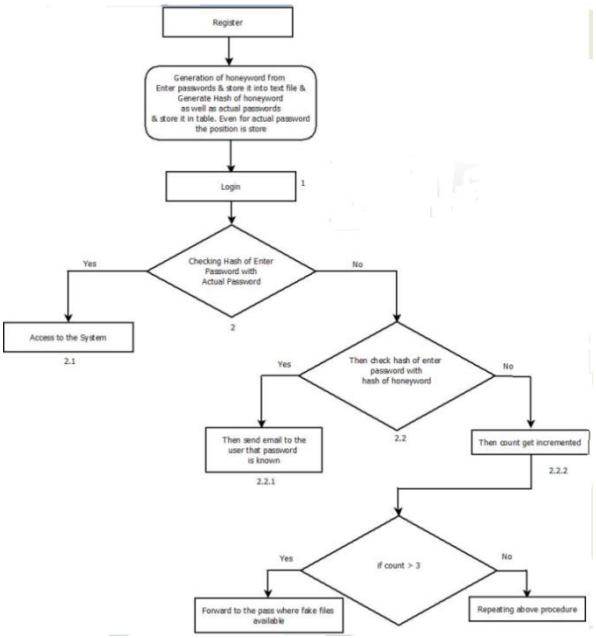

Fig-3: Flow of System

Chaffing- With-Tweaking algorithm: Using this honeyword generation method honeywords are randomly generated.

Take input as a position (pos) and password (pass).

```
Apply for loop from 1 to 10.
if (if==pos)
realPassword[i] =pass;
hashPassword[i] =generateHash(pass);
else
realPassword[i] =replace(pass1);
hashPassword[i] =generateHash(pass);
PassResult.put("Real",realPassword);
PassResult.put("Hash",HashedPassword);
return PassResult;
```

Using the previously proposed honeyword generation algorithms system maintains k-1 extra passwords along with the original password of user, in the password file F. On the other hand index of the original password of the user is maintained in “honeychecker" server. Using this honeyword generation method almost removes the Brute-force attack.

4 FUTURE WORKS:

In the future, 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 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.

5. CONCLUSION:

Honeyword based techniques are getting popular as it provides several advantages over additional password based schemes. However, the storage cost is one of the major overhead of honeyword based schemes. In this paper we have proposed a novel honeyword generation approach which reduces the storage overhead and also it addresses majority of the drawbacks of existing honeyword generation techniques.

6. REFERENCES

1. National information assurance (ia) glossary, 2010.
2. D. Florencio and C. Herley, "A Large-scale Study of Web Password Habits," in Proceedings of the 16th international conference on World Wide Web. ACM Press, 2007, pp. 657–666.
3. G. Notoatmodjo and C. Thomborson, "Passwords and Perceptions,"in Proceedings of the Seventh Australasian Conference on Information Security–AISC 2009. Australian Computer Society, Inc., 2009, pp. 71–78.
4. J. Bonneau. Guessing human-chosen secrets. Technical Report UCAM-CL-TR-819, University of Cambridge, Computer Laboratory, May 2012.
5. J. Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *IEEE Symp. Security and Privacy*, 2012.
6. A. Conklin, G. Dietrich, and D. Walz. Password-based authentication: A system perspective. In *Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 7 - Volume 7*, HICSS '04, pages 70170.2–, Washington, DC, USA, 2004. IEEE Computer Society.
7. D. Elser and M. Pekrul. Inside the password-stealing business: the who and how of identity theft, 2009.
8. J. Erasmus. Malware attacks: Anatomy of a malware attack. *Netw. Secur.*, 2009(1):4–7, Jan. 2009.
9. A. Juels and R. L. Rivest. Honeywords: Making password cracking detectable. Unpublished draft.
10. 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.
11. M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd. Reducing shoulder-surfing by using gaze-based password entry. In *Proceedings of the 3rd symposium on Usable privacy and security*, SOUPS '07, pages 13–19, New York, NY, USA, 2007.
12. P. G. Neumann. Risks of passwords. *Commun. ACM*, 37(4):126–, Apr. 1994.
13. E. H. Spafford. Opus: preventing weak password choices. *Comput. Secur.*, 11(3):273–278, May 1992.
14. H.-M. Sun, Y.-H. Chen, and Y.-H. Lin. opass: A user authentication protocol resistant to password stealing and password reuse attacks. *Information Forensics and Security, IEEE Transactions on*, 7(2):651–663, 2012.
15. A. van der Merwe, M. Loock, and M. Dabrowski. Characteristics and responsibilities involved in a phishing attack. In *Proceedings of the 4th international symposium on Information and communication technologies*, WISICT '05, pages 249–254. Trinity College Dublin, 2005.
16. K. Brown, "The Dangers ofWeak Hashes," SANS Institute InfoSec Reading Room, Tech. Rep., 2013.
17. A. Juels and R. L. Rivest, "Honeywords: Making Passwordcracking 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
18. C. Gaylord. LinkedIn, Last.fm, now Yahoo? don't ignore news of a password breach. Christian Science Monitor, 13 July 2012.
19. D. Gross. 50 million compromised in Evernote hack. CNN, 4 March 2013.