USING PRESENT USER PASSWORDS TO OBTAIN HONEYWORDS AND ACHIEVE FLATNESS

[1] *CHINTAN KOTADIYA,* [2] *SWASTICA SWAPN,* [3] *SHUBHAM DIXIT,*
[4] *PROF A.V SAGARE*

Sinhgad Institute Of Technology, Lonavala
Department Of Computer Engineering, Sinhgad Institute Of Technology, Lonavala

aposs8@outlook.com

ABSTRACT : *Disclosure of password files is a severe issue in information security now a days.We suggest a simple method of improving the security of hashed passwords: the maintainance of additional "honeyword"(false password) associated with each user's account .An adversary who steals a file of hashed passwords and inverts the hash function can not tell if he has found the password or a honeyword. Solution includes an auxiliary secure server called "honeychecker" which can distinguish user's real password among their honeywords and immediately sets off an alarm whenever a honeyword is used.We are implementing a web application which provides a secure login using honeywords to the organization server which we are working under the guidance of .For each user name a honeyword is constructed. Hence,when an adversary tries to enter with a honeyword, an alarm is triggered to notify the system about the password leakage. We are also working on alarm preaching i.e. notifying the user via email if somebody intrudes the user's account.*

Keywords : Passwords, Honey words, Authentication, Hashed Passwords, Security, Password Hacking, chaffing

INTRODUCTION

In this 21st centuary internet and modern technology is the base of every work being done in both domestic as well as industries. With this easy access to the thing called internet, the security is a crucial matter of concern as well. The amount of hackers are increasing day by day and our data is not completely safe. From online data to email ID passwords, everything is in danger. Email is the most used form on professional conversation which includes a lot of important data transfer as well as detailed discussion about the privacy of companies. Hacking of these email IDs results in loss of private data and it can leak into the hands of dangerous people who can use it for their personal benefits and can harm the company and its people. This is why honeywords are used now-a-days to protect our emails and serious conversations. Honeywords are generated from the real password and incase any hacker tries to hack into the account by guessing the password the main user is sent alerts in form of a mail or some message so he knows that somebody is trying to log into his or her account. The hacker is given access after three trails, he is shown decoy files and the real remain safe with the user.

MOTIVATION

Generally real passwords are very easy to detect and thus hack the system. So here the main motivation is to avoid this kind of hacking by the creation of honeywords. The human mind is incapable of accurately storing a large amount of data. In fact we can sometimes not even remember one password easily. This is why a honeyword based security system is needed to save crucial files from going into wrong hands who can manipulate important data for a wrong use and harm someone personally or harm the wholeindystry or company. Using this process the main user just needs to remember one original password that he sets for the account. The rest of it is taken care of by the working of the honeyword security set up.

LITERATURE SURVEY

Author/ Date	Theoretical/ Conceptual Framework	Research Question(s)/ Hypotheses	Methodology	Analysis & Results	Conclusions
Imran Erguler, 2015	Honeywords are selected properly, a cyber-attacker who steals a file of hashed	1.Passwords must be protected by taking appropriate	1. Chaffing-by-tweaking 2. Chaffing-with-a-password-	1. Denial-of-service Attack- we point out that if a strict policy is	We have analyzed the security of the honeyword system and addressed a number of flaws

	passwords cannot be sure if it is the real password or a honeyword for any account. Moreover, entering with a honeyword to login will trigger an alarm notifying the administrator about a password file breach.	precautions and storing with their hash values computed through salting or some other complex mechanisms. 2. A secure system should detect whether a password file disclosure incident happened or not to take appropriate actions.	model 3. Chaffing with "Tough Nuts" 4. Hybrid Method	executed in a honeyword detection, system may be vulnerable to DoS attacks affecting the whole system. 2. Brute-force Attack- we describe the following attack to demonstrate an adversary can capture an amount of accounts in case of a light policy	that need to be handled before successful realization of the scheme. In this respect, we have pointed out that the strength of the honeyword system directly depends on the generation algorithm, i.e. flatness of the generator algorithm determines the chance of distinguishing the correct password out of respective sweetwords.
Lianying Zhao and Mohammad Mannan	Automated online password guessing attacks are facilitated by the fact that most user authentication techniques provide a yes/no answer as the result of an authentication attempt. In addition, we suggest using adapted distorted images and pre-registered images/text as a complement to convey an authentication response, especially for accounts that do not host much personal data.	Pinkas and Sander first proposed the use of Reverse Turing Tests (RTTs, e.g., captchas) to restrict large-scale online password dictionary attacks. The protocol challenges users with RTTs for a small fraction (e.g., 5%) of all possible userid-password pairs to reduce the server-load (of generating RTTs) and usability impact (of answering RTTs), while keeping the cost of launching a large-scale guessing attack significantly high.	1.In user-level authentication, we introduce the idea of programmably leaving the result of authentication on the server (verifier). Such hiding of authentication results may enable effective protection against online guessing attacks. 2. We propose the use of adapted distorted image as a computer-cipher/human-decipher channel to communicate short messages in human-machine interaction. 3. Our proposal requires no changes on the client side software or	In designing Uvauth, we explicitly consider such threats and provide limited protection (possibly significantly more than existing technologies). Implementing Uvauth fake sessions would require server-side support, but no changes are needed on the client-side software or existing password input UI (including browser mechanisms such as "keep me logged in" and cookies)	It can effectively deceive an attacker assuming fake sessions can be efficiently generated.

			existing password input UI, and can be used with any authentication scheme vulnerable to online guessing attacks.		
Ari Juels, Ronald L. Rivest	We propose a simple method for improving the security of hashed passwords: the maintenance of additional "honeywords"(false passwords) associated with each user's account.	How can a honeyword system best be designed to withstand active attacks, e.g., malicious messages issued by a compromised computer system or code modification of the computer system (or the honeychecker)? • Can a honeyword system be designed to protect at all against persistent attacks, in which an adversary observes passwords submitted to the computer system? • How well can targeted attacks help identify users' passwords for particular honeyword-generation methods?	1. "Random pick" honeyword generation 2. Typo-safety 3. Managing old passwords 4. Storage optimization 5. Hybrid generation methods	The use of an honeychecker thus forces an adversary to either risk logging in with a large chance of causing the detection of the compromise of the password-hash file F, or else to attempt compromising the honeychecker as well. Since the honeychecker's interface is extremely simple, one can more readily secure the honeychecker. The use of honeywords may be very helpful in the current environment, and is easy to implement. The fact that it works for every user account is its big advantage over the related technique of honeypot accounts.	It inherit many of the well known drawbacks of passwords and something-you-know authentication more generally. Eventually, passwords should be supplemented with stronger and more convenient authentication methods, or give way to better authentication methods completely, as recently predicted by the media. every breach of a password server has the potential to improve future attacks
Gilbert Notoatmodjo and Clark Thomborson 2009	Participants thus demonstrated awareness of the basic tenets of password safety, but they did not behave safely in all respects.	Morris and Thompson (1979) studied a corpus of 3,289 passwords from many users over a	1. Participants were asked to write all their passwords in a piece of paper. 2. We explained our	1. Password Properties- test indicated that there was no evidence that there is a significant correlation	By using passwords which they perceived to be more 'secure' on accounts that they considered important, our

	Almost half of our participants reused at least one of the passwords in their high-importance accounts. Our findings add to the body of evidence that a typical computer user suffers from 'password overload'.	long period of time and discovered that 86% of these passwords were extremely weak. Riddle, Miron et al. (1989) analyzed 6226 user generated passwords from IBM CMS environment used by students and staff at Syracuse University in 1987, finding that many passwords were extremely short and consisted of English words or persons' names. Adams and Sasse (1999) conducted a study of password related user behaviors, including password construction, frequency of use, password recall and work practices. They concluded that their participants lacked security motivation and understanding of password policies, and tended to circumvent password restrictions	hypothesis regarding how people organize their passwords by mentally grouping them. 3. Participants were asked to complete a table, by assigning numbers and codes to their passwords according to the way they group their passwords based on their perceived similarities. We also instructed the participants to describe the similarities that they use as a basis for grouping their passwords together. The worksheet containing this table was used only during the next step. 4. Participants were instructed to describe each of their passwords by completing the following columns: 1.Length The total number of characters in each password). 2.Perceived security level Measured on a five point Likert scale, from one (least secure) to five (most	between length and perceived security level of passwords 2. Growth of Accounts and Passwords- longer exposure to computers or the internet would translate into more accounts. 3. Password Reuse Statistics- It is evident in the scatter plot below that our participants reused more as they accumulate more accounts.	participants demonstrated their awareness of the importance of using strong passwords to protect their valuable information.

		for the sake of convenience. Dhamija and Perrig (2000) conducted an interview-based study involving 30 participants.	secure). 3.Difficulty of recall Measured on a 5 point Likert scale, from one (least difficult) to five (most difficult).		

PROPOSED SYSTEM

Here we are generating honeywords.Honeywords are generated from the real password and incase any hacker tries to hack into the account by guessing the password the main user is sent alerts in form of a mail or some message so he knows that somebody is trying to log into his or her account. The hacker is given access after three trails, he is shown decoy files and the real remain safe with the user. Following are Modules used in the proposed system.

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.

ALGORITHM

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);
6. passResult.put("real", realedPassword);
passResult.put("hash", hashedPassword);
passResult is HashMap.
7. return passResult;

Honey Word Checker:

```
if (honeyPassList[i].equals(passwordHash) && i != Integer.parseInt(pos)) {
}
```

CONCLUSION

In this study, we have analyzed the security of the honeyword system and addressed a number of flaws that need to be handled before successful realization of the scheme. In this respect, we have pointed out that the strength of the honeyword system directly depends on the generation algorithm. The use of this honeyword based security system is the best way to sure all sorts of accounts and conversations be it personal or professional. The user does not have to worry about anything but remember his/her original password. The system takes care of the rest to save and secure the data inside of it. This system is not costly as well and easy to handle and work on.

REFERENCES

1) 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

2) 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

3) 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

4) G. Notoatmodjo and C. Thomborson, "Passwords and Percep-tions," in Proceedings of the Seventh Australasian Conference on Information Security–AISC 2009. Australian Computer Society, Inc., 2009, pp. 71–78.

5) D. Florencio and C. Herley, "A Large-scale Study of Web Pass-word Habits," in Proceedings of the 16th international conference on World Wide Web. ACM Press, 2007, pp. 657–666.