Achieving Flatness: Selecting the Honeywords from Existing User Passwords

Ritu Ranjan Jha[1] Pranita Dhane[2] Prof. Yuvaraj N N[3]
[1,2]BE Student [3]Assistant Professor
[1,2,3]Department of Computer Engineering
[1,2,3]DYPSOEA Ambi, India

Abstract— It may be represented the honeyword mechanism to detect an adversary who tries to login with cracked passwords. New password is the mix of existing user passwords called honey words. Fake passwords are few things though the honey words basically, for every username some sweet words is constructed in ways that merely one element will be the correct password as well as the others are honey words (decoy passwords). Hence, when an adversary tries to enter into the device with a honey word, a security is triggered to notify the administrator of a password leakage. Honey words to detect attacks against hash password database. Per user account the legitimate password saved in way of honey words. If attacker Attack on password i.e. honey words it cannot be sure it really is real password or honey word. On this study, we to check in more detail with careful attention the honey word system and provide some comment to target supply weak spots. Also concentrate on pragmatic password, reduce storage price of password, and alternate any to choice the newest password from existing user passwords.

Key words: Authentication, Honeypot, Honey words, Password Cracking

I. Introduction

On this there are two issues that should be thought to overcome these security problems. First passwords has to be paid by taking appropriate precautions and storing using hash values computed through salting as well as other complex mechanisms. Hence, for an adversary it should be though to invert hashes to get plaintext passwords. The other point is the fact that a safe and secure system should detect whether your password strength file disclosure incident happened or you are not to adopt appropriate actions. Within this study, we focus on the later issue and handle fake passwords or accounts being a simple and easy cost effective treatment for detect compromise of passwords. Every time a user sends a login request, the login server determines the transaction of her one of the users, and the order with the submitted password among her sweet words. The login server sends a message from the form to a secure server which is called honey checker, for that user and her sweet word. The honey checker will determine whether the submitted word is often a password or perhaps a honey word. If a honey word is submitted, this will raise an alarm or take an action that's previously chosen. The honey checker cannot know everything about the user's password or honey words. It retains a single database made up of merely the order in the true password one of the user's sweet words.

II. Literature Survey

A. *Paper name: Guess again: Measuring password strength by simulating password-cracking algorithm*

Authors: P.G. Kelley, S. Komanduri , M.L Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L.F. Cranor and J. Lopez

Description: here authors found several notable results about the comparative strength of different composition policies. Although NIST considers basic16 and comprehensive8 equivalents, we found that basic16 is superior against large numbers of guesses. Combined with a prior results that basic16 is also easier for users, this suggests basic16 is the better policy choice. We also found that the effectiveness of a dictionary check depends heavily on the choice of dictionary; in particular, a large blacklist created using state-of-the -art password guessing techniques is much more effective than a standard dictionary at preventing users from choosing easily guessed passwords.

B. *Paper Name: a large-scale study of web password habits*

Authors: D. Florencio and C. Herley

Description: Here authors report the results of a large scale study of password use and password re-use habits. The study involved half a million users over a three month period. A client component on users' machines recorded a variety of password strength, usage and frequency metrics. This allows us to measure or estimate such quantities as the average number of passwords and average number of accounts each user has, how many passwords she types per day, how often passwords are shared among sites, and how often they are forgotten. Also authors get extremely detailed data on password strength, the types and lengths of passwords chosen, and how they vary by site. The data is the first large scale study of its kind, and yields numerous other insights into the role the passwords play in users' online experience.

C. *Paper Name: Examination of a new defense mechanism: Honey words*

Authors: Z. A. Genc, S. Kardas, and M. S. Kiraz

Description: Here authors decoy passwords i.e. honey words to identify attacks against hash password database. For each and every user account the legitimate password kept in kind of honey words. If attackers Attack on password i.e. honeywords it wouldn't make sure it is real password or honeyword. It is less difficult to crack your password hash together with the advancements within the graphical processing unit (GPU) technology. Entering which has a honey word to login will trigger an alarm notifying the administrator of a password file breach.

D. Paper Name: Password Cracking Using Probabilistic Context-Free Grammars

Authors: M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek

Description: Choosing the foremost effective word-mangling rules to use once playing a dictionary-based parole cracking attack may be a troublesome task. During this paper authors have a tendency to discuss a replacement technique that generates parole structures in highest chance order. Here authors first mechanically produce a probabilistic context-free descriptive linguistics based mostly upon a coaching set of antecedently disclosed passwords. This descriptive linguistics then permits United States to get word-mangling rules, and from them, parole guesses to be utilized in parole cracking. Also authors conjointly show that this approach appears to supply a more practical thanks to crack paroles as compared to ancient ways by testing our tools and techniques on real password sets.

III. Existing System

We separate the honeyword approach and give some notice about the security of the system. We point out that the key item for this method is the generation algorithm of the honeywords such that they shall be indistinguishable from the correct passwords. Therefore, we propose a new method that created the Honeywords using the existing user passwords combination in hash format.

A. Disadvantages of Existing System

- Not secure.
- Performance is low.
- Difficult to locate malicious activities.

IV. Proposed System

Within this study, we concentrate on the security issue and cope with fake passwords or accounts as being a simple and cost-effective means to fix detect compromise of passwords. Honeypot is probably the methods to identify occurrence of the password database breach. On this approach, the administrator purposely creates deceit user accounts to lure adversaries and detects a password disclosure, if any one of the honeypot passwords get used. In this paper we now have proposed a novel honeyword generation approach which cuts down on the storage overhead and also it addresses majority of the drawbacks of existing honeyword generation techniques. Proposed model is determined by utilization of honey words to identify password-cracking. We propose to work with indexes that map to valid passwords from the system. The contribution of our own approach is twofold. First, using this method requires less storage when compared to the original study. In your approach passwords of other users are employed as the fake passwords, so guess ones password is fake and that is correct gets to be more complicated for an adversary.

A. Advantages of Proposed System

- It is more secure.
- It detects the all malicious activities of users.
- It’s a trustful network.

V. Algorithm

A. Chaffing with Toughnut

In this method, the system intentionally injects some special honeywords, named as tough nuts, such that inverting hash values of those words is computationally infeasible, e.g. fixed length random bit strings should be set as hash value of a honeyword. Moreover, it is noted that number and positions of tough nuts are selected randomly. By means of this, it is expected that the adversary cannot seize whole sweet word set and some sweet words will be blank for her, thereby deterring the adversary to realize her attack.

B. Chaffing with Tweaking

In this method, user password seeds the generator algorithm which tweaks selected character positions of the real password to produce the honeywords. For instance, each character of 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. Number of positions to be tweak, denoted as t should depend on system policy etc. As an example t = 3 and tweaking last t characters may be a method for generator algorithm Gen (k, t). Another approach named in the study as” chaffing-by-tweaking-digits” is executed by tweaking the last t positions that contain digits. For example, by using last technique for the password 42hungry and t = 2, the honeywords 12hungry and 58hungry may be generated.

C. Tail

Tail is combining the strength of different honeyword generation methods, e.g. chaffing-with-a-password-model and chaffing-by-tweaking-digits. By using this technique, random password model will yield seeds for tweaking-digits to generate honeywords. For example let the correct password be apple1903. Then the honeywords angel2562 and happy9137 should be produced as seeds to chaffing-by-tweaking-digits.

VI. System Architecture

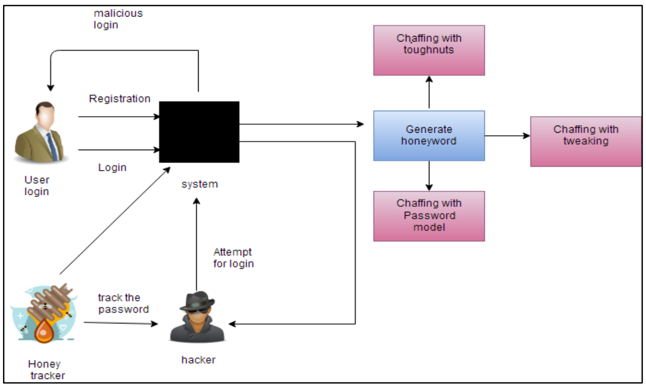


Fig. 1: Proposed System Architecture

In this system, If user entered right username and password is the honey word which is generated at the time of registration then the system will allow user next two times to enter his correct password. The honey encryption methods used by using some passwords + keys. We have generated the many to many relationships. And Compare to each key with seed space. Then XOR operation performed. Even if after giving three chances user enters the honey word then system will lock the account. And he has waits for activation from admin. If user entered right username but if password is wrong also password is not a honey word then system will block that particular user and request to admin for activate the account.

VII. Modules

1) User
2) Admin
3) Hacker
4) Honey Tracker

A. User

1) Registration:

– User will register to the system, at the time of registration user will enter the 3 Honey words.
– Also system will generate no. of Honey words with the help of user password by three methods:
  - Chaffing with Toughnut
  - Chaffing with Tweaking
  - Tail

2) Login:

– If user entered right username and password is the honey word which is generated at the time of registration then the system will allow user next two times to enter his correct password.
– Even if after giving three chances user enters the honey word then system will lock the account. And he has waits for activation form admin.
– If user entered right username but if password is wrong also password is not a honey word then system will block that particular user and request to admin for activate the account.

B. Admin

– Admin will activate the blocked user account.
– Admin will protect the passwords by using Honey Encryption method.
– The honey encryption methods used by using some passwords+keys.We have generated the many to many relationships. And Compare to each key with seed space. Then XOR operation performed.

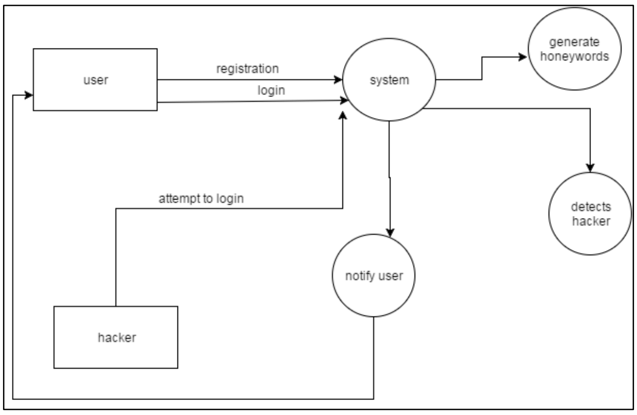
C. Hacker

– Hacker will login into the system.
– Then hacker will get wrong passwords for requested user.

D. Honey Tracker

It will track the user’s record i.e. number of wrong passwords and number of honey words for particular user login.

VIII. Functional Model



IX. System Requirements

Hardware Requirements:

System	:	Pentium IV 2.4 GHz.
Hard Disk	:	40 GB.
Floppy Drive	:	1.44 Mb.
Monitor	:	15 VGA Colour.
Mouse	:	Logitech.
Ram	:	512 Mb.

Software Requirements:

Operating system	:	Windows XP/7.
Coding Language	:	JAVA/J2EE, Hibernate.
IDE	:	Java eclipse.
Web server	:	Apache Tomcat 7.
Front End	:	JSP, CSS etc.
Back End	:	MySQL as database server.

X. Future Scope

1) 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.
2) 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.

XI. Conclusion

Finally proposed the security in the honeyword system and introduce numerous defect that need to be fitted with before successful realization with the scheme. This is because, we now have pointed out that the forte with the honeyword system directly is determined by the generation algorithm finally we've got presented a whole new approach to help make the generation algorithm as close regarding man's instinct by generating honeywords with randomly picking passwords owed with users from the system. We present an ordinary method of securing business and personal data from the system. We propose monitoring data access patterns by profiling user behavior to ascertain when and if a malicious insider illegally accesses someone’s documents inside a system service. Decoy documents kept in the device alongside the user’s real data also work as sensors to detect illegitimate access. Once unauthorized data access or exposure is suspected, and then verified, with challenge questions for example, we inundate the malicious insider with fake information to be able to dilute or divert the user real data. Such preventive attacks that depend on disinformation technology could provide unprecedented numbers of peace of mind in the machine plus internet sites model.

In the future, we'd like to refine our model by involving hybrid generation algorithms also to result in the total hash inversion process more difficult for an adversary in enabling the passwords in plaintext form a leaked password hash file. Hence, by developing such methods both two security objectives helping the total effort in recovering plaintext passwords through the hashed lists and detecting the password disclosure can be provided as well.

REFERENCES

[1] Kelley, Patrick Gage, et al. "Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms." Security and Privacy (SP), 2012 IEEE Symposium on. IEEE, 2012.

[2] Florencio, Dinei, and Cormac Herley. "A large- scale study of web password habits." Proceedings of the 16th international conference on World Wide Web. ACM, 2007.

[3] Genc, Ziya Alper, and Süleyman Kardaş. "Examination of a new defense mechanism: Honeywords." Proceedings of the 11th WISTP International Conference on Information Security Theory and Practice. Springer, 2017.

[4] Weir, Matt, et al. "Password cracking using probabilistic context-free grammars." Security and Privacy, 2009 30th IEEE Symposium on. IEEE, 2009.

[5] National information assurance (ia) glossary, 2010.

[6] Password cracking. Web Site, 2013. www.golubev.com/hashgpu.htm.

[7] H. Bojinov, E. Bursztein, X. Boyen, and D. Boneh. Kamouflage: Loss-resistant password management. In ESORICS, pages 286– 302, 2010.

[8] J. Bonneau. Guessing human-chosen secrets. Technical Report UCAM-CL-TR-819, University of Cambridge, Computer Laboratory, May 2012.

[9] M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek, “Password cracking using probabilistic context-free grammars,” in Proc. 30thIEEE Symp. Security Privacy, 2009, pp. 391–405.

[10] F. Cohen, “The use of deception techniques: Honeypots and decoys,” Handbook Inform. Security, vol. 3, pp. 646–655, 2006.

[11] M. H. Almeshekah, E. H. Spafford, and M. J. Atallah, “Improving security using deception,” Center for Education and Research Information Assurance and Security, Purdue Univ., West Lafayette, IN, USA: Tech. Rep. CERIAS Tech. Rep. 2013-13, 2013.

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

[13] 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.

[14] A. Juels and R. L. Rivest, “Honeywords: Making password cracking detectable,” in Proc. ACM SIGSAC Conf. Comput.Commun. Security, 2013, pp. 145–160.

[15] M. Burnett. The pathetic reality of adobe password hints. [Online].Available: https://xato.net/windows-security/adobe-passwordhints, 2013.