Review on Honeyword Generation for User Authentication

[1]Manisha J.Bhole, [2]Prof. Dr. Girish K. Patnaik,
[1]P.G.Student, [2]H.O.D.,
[1]Computer Department,
[1]SSBT'S COET, Bambhori, Jalgaon, India

Abstract : With an emphasis on Digital India and Government's encouragement for cashless transactions, it has become essential for organizations to maintain the secrecy of login credentials of their employees and clients. At the same time, it is also necessary to avoid any suspicious activities/attempts to steal such data by the unauthenticated user. There are many methods to achieve secure logins like OTP and Token generators. But these methods require additional devices to be carried by the authenticated user. Loss or change of the additional devices can obstruct the authenticated user from logging in. Organizations can increase the authenticate user's comfort while maintaining the secrecy by storing a bunch of decoy passwords or "honeywords" corresponding to the correct password in the hashed password database. Foe the purpose of 'password cracking' an unauthenticated user hacking the hashed password database would not be able to identify the decoy password and an attempted use of honeyword can set off an alarm. In the proposed work the Honeyword generation method i.e. chaffing-with- tweaking provide an enhanced password security as a solution to an open problem that also overcomes password-cracking problem which the drawbacks of previously proposed honeyword generation approaches.

Index Terms **- authentication. Honeywords, passwords, password authentication**.

I. INTRODUCTION

The term "honeywords" is a play on "honeypot," which in the information security really refers to creating fake servers and then learning how attackers attempt to exploit them in effect, using them to help detect more widespread intrusions inside a network. "Honeywords are a simple but clever idea". Seed password files with dummy entries that will trigger an alarm when used. That way a site can know when a hacker is trying to decrypt the password file. While there are many attacks against password protection systems, password cracking refers to the process of extracting passwords from data. The most straightforward attack is called the brute-force attack. Simply trying every possible combination of characters until you find a matching hash value. So may become very time consuming particularly with long passwords. The difficulty also increases when more characters are allowed in passwords. The numbers grow quite large as the length and complexity of the password increases and would seem to make password cracking impossible. While it's true that the length of time to brute-force passwords increases with complexity, there are several other techniques that crackers can use to expose these passwords. The RockYou attack, in particular, revealed millions of commonly used passwords and has become part of the standard dictionary used to crack passwords. Password leaks are becoming a common occurrence on the internet with several large-scale leaks .happing every year. These leaks have revealed the poor practice many companies employ when storing their passwords. The widely available lists of common passwords, an expanding knowledge base on how user select passwords and advances in password cracking technologies have made basic hashes more vulnerable than ever.

Section 1.1 Describes an overview of Honeywords. Motivation is described in Section 1.2. Section 1.3 Describes contribution. Organization of the report is described in Section 1.4. Finally, a summary of the chapter is given in the last section.

1.1 Overview of Honeywords

Basically, the simple but clever idea behind the study is insertion of false pass- words – called as honey words – associated with each user's account. When an adversary gets the password list, she recovers many password candidates for each account and she cannot be sure about which word is genuine. Hence, cracked password files can be detected by system administrator if a login attempt is done with a honeyword by the adversary. Some honeyword generation methods were explained as follows:

1 **Chaffing-by-tweaking** 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, letters by letters, replace digits and special characters by special characters. A number of positions to be tweaked denoted, as t should depend on system policy. As an example t = 3 and tweaking last t characters may be a method for the generator algorithm Gen(k, t).

2 **Chaffing-by-tweaking-digits** It is executed by tweaking the last t positions that contain digits. For example, by using the last technique for the password 42hungry and t = 2, the honeywords 12hungry and 58hungry may be generated.

3 **Chaffing-with-a-password-model** In this approach, the generator algorithm takes the password from the user and relying on a probabilistic model of, [8] real passwords it produces the honeywords, [9]. The authors give the model of as an example of this method named the modeling syntax. In this model, the password is split into character sets. For instance, mice3blind is decomposed as 4-letters + 1-digit + 5-letters L4 + D1 + L5 and replaced with the same composition like gold5rings.

4 **Chaffing with "Tough Nuts"** 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 the hash value of a honeyword. An illustrative example of a tough nut is given, in [9] as '9,50PEe[KV.0?RIOtcL-:IJ"b+Wol¡*]!NWT/pb'. It is stated that the number and positions of tough nuts are selected randomly. By means of this, it is expected that the adversary cannot seize whole sweetword set and some sweet words will be blank for her, thereby deterring the adversary to realize her attack. In [9] is discussed that in such a situation the adversary may pause before attempting a login with cracked passwords.

5 **Hybrid Method** 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.

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

1.2 MOTIVATION

Real passwords are often weak and easily guessed; either by sharing passwords, using names of loved ones, dictionary words, and brute force attacks. The first step is to study and identify websites that are being actively targeted by account creation tools. Specifically, chose to target: Twitter, Facebook, Pinterest, LinkedIn, Wordpress, eBay, and Hotmail. All of these sites were well represented in the forums (i.e. many people were selling accounts from, and tools targeting these sites) and they are extremely popular with web users in general. To provide protection against online attacks generating honeywords against the password is may be the solution for these problems.

1.3 CONTRIBUTION

The contribution in the proposed system is to make a difficult situation for an unauthenticated user by making the protocol, which increases the chance of login in dummy environment.

1.4 OBJECTIVES

Increasing the probability of unauthenticated user guessing a Honeyword by at least 5%. Unauthenticated user access to dummy environment and thereby protecting the user account.

1.5 PROPOSED SOLUTION

Creating Honeywords by randomly replacing the characters also creating Dummy environment where files uploaded by unathenticated user will be deleted after the end of section and user with Wrong password is given error for wrong password. User with Honeyword is given access to Dummy environment. User with correct password is given access to his account.

1.6 ORGANIZATION OF REPORT

The organization of the report is given as:-Chapter 1, titled Introduction,consists of sections like information about domain, motivation.Chapter 2,titled Literature Survey, consists the section background and related work.Chapter 3,titled Proposed solution, describes the proposed honeyword generation method. It also presents the flowcharts and algorithms used in the proposed methods.chapter 4,titled Conclusion and Future Work, concludes and provides directions for further work.

II. LITERATURE SURVEY

On any computer system that controls resources for more than one user, the ability to authenticate different users is imperative. A password has long been the most common way users prove their identity to a computer. The attractiveness of password-based authentication lies not in its security, but in its simplicity, practicality, ease of use, and low cost. Businesses should seed their password databases with fake passwords and then monitor all login attempts for use of those credentials to detect if hackers have stolen stored user information. That's the thinking behind the "honeywords" concept first proposed in "Honeywords:

Making Password-Cracking Detectable," a paper written by Ari Juels, chief scientist at security firm RSA, and MIT professor Ronald L. Rivest, who co-invented the RSA algorithm. The term "honeywords" is a play on "honeypot", which in the information security really refers to creating fake servers and then learning how attackers attempt to exploit them in effect, using them to help

detect more widespread intrusions inside a network. “Honeywords are a simple but clever idea,” said Bruce Schneier. Seed password files with dummy entries that will trigger an alarm when used.

2.1 Background

This guideline is used for all journals. These are the manuscript preparation guidelines used as a standard template for all journal submissions. Author must follow these instructions while preparing/modifying these guidelines. This A are the manuscript preparation guidelines used as a standard template for all journal submissions. Author must follow these instructions while preparing/modifying these guidelines. This guideline is used for all journals. This guideline is used for all journals. These are the manuscript preparation guidelines used as a standard template for all journal submissions. Author must follow these instructions while preparing/modifying these guidelines. This guideline is used for all journals. This guideline is used for all journals. These are the manuscript preparation guidelines used as a standard template for all journal submissions. Author must follow these instructions while preparing/modifying these guidelines. This guideline is used for all journals.

In password-based authentication, the identity of an individual is verified solely by his/her ability to present a previously agreed word. This results in a significant vulnerability. Security is compromised if an adversary learns a single word. The adversary who knows a user’s password will be able to impersonate this user and to access the resources to which this user is entitled. An adversary may mount several types of attacks on password authentication systems.

Identification of three main categories, based on the target of the attacks,

1. Attacks on the system end: This type of attack is targeted at the passwords stored in the system. An example of this type of attack is password-guessing attack

2. Attacks on the communication channel: These attacks target any communication channel through which passwords are transmitted. Definition of communication channel includes all devices, media and protocols which connect the user to the system which stores the password (or its hash). Examples are replay, eavesdropping, and man-in-the-middle attacks.

3. Attacks on the user end: These are directly targeted at the user. Examples are social engineering, shoulder surfing, dumpster diving, and phishing. For cracking purpose the attacker makes a guess as to the value of the original password. The attacker then hashes that guess using the appropriate password- hashing algorithm and compares the two hashes. If the two hashes match, the attacker has discovered the original password, or in the case of a poor password hashing algorithm, they at least have a password grant access. The two most commonly used methods to make these guesses are brute-force and dictionary attacks. With brute-force, the attacker attempts to try all possible password combinations. While this attack is guaranteed to recover the password if the attacker manages to brute-force the entire password space, it often is not feasible due to time and equipment constraints. If no salting is used, brute-force attacks can be dramatically improved through the use of pre- computation and powerful time-memory trade-off techniques ,in [15] [17].

2.2 Related Work

Figure 2.1 shows the structure of literature survey. Password security form most users have a low awareness of their vulnerability and of the scope of damage that can be inflicted if their passwords are compromised. Password security can be achieved through measuring password strength, web password habit and honeyword generation technique.

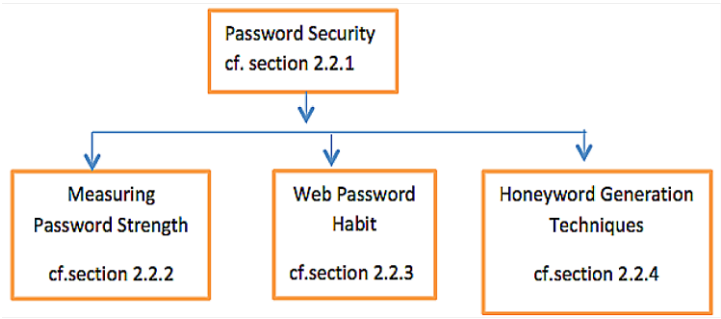

FIGURE 2.1: Structure of Literature Survey

2.2.1 Password Security

Gilbert Notoatmodjo and Clark Thomborson, in [7], presented user’s perspective to their accounts and passwords. Authors described three main categories of attacks are, namely, attacks on the system end, attacks on the communication channel and attacks on the user end. By using passwords which they perceived to be more ‘secure’ on accounts that they considered

important,participants demonstrated their awareness of the importance of using strong passwords to protect their valuable information.

Hristo Bojinov et al., in [1], presents kamouflage-based password manager a new technique to prevent theft-resistant. The study states to use salts and slow hash functions to slow down a dictionary attack on the master password but unfortunately these methods do not prevent dictionary attacks. Authors states the main difficulties to overcome to make kamouflage work are, human-memorable passwords, related passwords, relation to master password and site restrictions. The authors have done with a survey that shows how users choose passwords. Authors have also described threat model, decoy set generation and fingerprinting. They ended with the conclusion stating kamouflage and fingerprinting technique provides security at high level.

Brown and Kelly, in [12], proposed that that the damaged caused by password leaks can be reduced by implementing few good practices.In June 2012 many companies like LinkedIn, Yahoo etc... Were affected by password leaks, which were publicly spread. Secure system should not have any loop holes that will allow intruders to get access to password files and should make sure that if the password hashes are been hacked it should not be easy to generate passwords from the hashes. Due to weak hashing mechanism these companies were highly affected is proved. In this paper the author has discussed about basics of password hashing and best practices that should be followed while password storage.

Kelley et al., in [9], proposed that the effects of password-composition policies on the guessability of pass- words. Seven dfferent password-composition policies are used online to apply on a dataset of 1200 plaintext passwords. Described development approaches for calculating time consumed to guess each password they col- lected,also implemented guess-number calculator to evaluate the electiveness of password-guessing attacks. Results also reveal important information about conducting guess-resistance analysis. Effective attacks on passwords created under complex or rare-in-practice composition policies require access to abundant, closely matched training data. Shannon entropy provides only a rough correlation with guess resistance and is unable to correctly predict quantitative differences in guessability among password sets.

2.2.2 Measuring Password Strength

Kelley et al., in [5], proposed that Seven different password-composition policies are used online to apply on a dataset of 1200 plaintext passwords. They have developed approaches for calculating time consumed to guess each password they collected. They have implemented guess-number calculator to evaluate the effectiveness of password-guessing attacks. Results also reveal important information about conducting guess- resistance analysis. Effective attacks on passwords created under complex or rare-in-practice composition policies require access to abundant, closely matched training data. Shannon entropy provides only a rough correlation with guess resistance and is unable to correctly predict quantitative differences in guess ability among password sets. Password strength is a numerically expressed measure of how uncrack able a password is by considering the length and complexity of the password. Since password strength is measured by length and complexity, would it be safe to simply follow the generally recommended guidelines—use more than 8 characters and mix numbers, symbols, upper and lowercase letters. Altogether, there are 96 possible characters when choosing from A to Z in both upper and lowercase, 0-9, and all available keyboard symbols. A password with 8 characters could be any one of these 96, taken to the eighth power for the varying patterns. That means over 7,200 trillion password options! Not even a computer could easily handle cracking a password with that type of complexity.

Joseph Bonneau in [3], proposed that the evaluation of large password data sets by collecting a massive password data set legitimately and analyzing it in a mathematically rigorous manner. In previous paper, Shannon entropy and guessing entropy not worked with any realistically sized sample, therefore, they developed partial guessing metrics including a new variant of guesswork parameterized by an attacker's desired success rate. In their study most troublesome is how little password distributions seem to vary, with all populations of users.

2.2.3 Web Password Habit

Dinei Florencio and Cormac Harley, in [8], proposed that study of password used and password reused habits. They measured average number of passwords and average number of accounts each user has, as well as measured number of times user enters password per day. They calculated this data and estimated password strength, password vary by site and number of times user

forgotten password. In their findings, it showed users choose weak password; they measured exactly how weak. They measured number of distinct passwords used by a client vs. age of client in days also, number of sites per password vs. age of client in days. They also analyzed password strength. We are able to estimate the number of accounts that users maintain the number of passwords they type per day, and the percent of phishing victims in the overall population. 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. Honeypot is one of the methods to identify occurrence of password database breach. In 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, [17], [18]. Idea has been modified by Harley and Florencio in, [19] ,to protect online banking accounts from password brute-force attacks. According to the study, for each user incorrect login attempts with some passwords lead to honeypot accounts, i.e. malicious behavior is recognized. For instance, there are 108 possibilities for a 8- digit password and let system links 10000 wrong password to honeypot accounts, so the adversary performing the brute-force attack 10000 times more likely to hit a honeypot account than the genuine account.

2.2.4 Honeyword Generation Technique

Joseph Bonneau 2012 et al., in [3], proposed that the evaluation of large password data sets by collecting a massive password data set legitimately and analyzing it in a mathematically rigorous manner. In previous paper, Shannon entropy and guessing entropy not worked with any realistically sized sample, therefore, they developed partial guessing metrics including a new variant of guesswork parameterized by an attacker's desired success rate. In their study most troublesome is how little password distributions seem to vary, with all populations of users.

Dhinaharan Nagamalai et al., in [13], proposed that the characteristics of spam and technology used by spammers. They observed that spammers use software tools to send spam with attachment. To track and represent the characteristics of spam and spammers they setup a spam trap in their mail server. The paper is discussed in two types i.e. first type spam with attachment and second type is spam without attachment. They concluded, for spam without attachment, senders use non-sophisticated methods but for spam with attachment, senders use sophisticated software to spam end users.

Ziya Alper Genc et al., in [11], proposed that hash passwords are used to improve security. For user authentication false passwords are added in hashed password file i.e. honeywords. They analyzed the honeyword system according to both functionality and the security perspective. They also elaborated how the system will respond to six passwords related attacks. Improvements for honeywords is described briefly i.e. number of honeywords, typo-safe honeyword generation and old passwords problem. Assumptions are illustrated to an active attack against honeyword system. They concluded that honeyword system is the powerful defense mechanism where an adversary steals the file of password hashes and inverts most or many of the hashes.

Lianying Zhao and Mohammad Mannan , in [4], proposed that technology called Uvauth to hide authentication results from attackers to mitigate the risk of online password guessing. They propose the use of adapted distorted image as a computer-cipher/human-decipher channel to communicate short messages in human-machine interaction. The authors have discussed Uvauth and CAPTCHA for selfevidence of authentication that may make the scheme feasible. They have also elaborated possible attacks from attacker's perspective and some of them are limitations to current design. Limitations are they have not evaluated the server side load for generating and running a large number of fake sessions. They also have not tested how effectively users can detect implicit results from an authentication attempt, or whether messages via adapted distorted images can be used in practice. It can effectively deceive an attacker assuming fake sessions can be generated.

Ari Juels and Ronald L. Rivest, in [2], proposed that honeywords technology to improve security level for authenticating fake users. The authors have also described briefly attacks on different scenarios, but have focused on stolen files of password hashes scenario. They have described various types of attacks on hon- eyword system that shows how it will manage and overcome it. The attacks are, namely, general password guessing, targeted password guessing, attacking the honeychecker, likelihood attack, DOS attack and multiple systems. The study shows to limit the impact of a DOS attack against chaffing-by-tweaking; one possible approach is to select a relatively small set of honeywords randomly from a larger class of possible sweet- words. Methods used . "Random pick" honeyword generation2. Typo-safety3. Managing old passwords4. Storage optimizations. It inherits 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.

Imran Erguler 2015 et al., in [1], proposed new honeyword generation algorithm in which honeywords are generated from the existing user passwords is proposed. It provides realistic honeywords. It shows better results with respect to flatness, DOS resistance and storage. New honeyword generation method 1. Chaffing- by-tweaking 2. Cha ng-with-a-password-model3. Chaffing with ”Tough Nuts”4. Hybrid Method reduces storage cost of the honeyword scheme. It 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, i.e. flatness of the generator algorithm determines the chance of distinguishing the correct password out of respective sweetword.

Analyzed honeyword approaches and security of the system. Furthermore, point out that the key item for method is the generation algorithm of honeywords such that they were indistinguishable from the correct passwords. Therefore, propose a new approach that uses passwords of other users in the system for honeyword sets, i.e. realistic honeywords are provided.

2.3 Summary

In this chapter, background and related work about honeyword generation techniques are described. In the next chapter, Proposed Solution is presented.

III. PROPOSED PTOTOCOL

The research focuses on honeywords generation, i.e. the user passwords stored with sweet words in a hashed file and storing in random position as an encrypted file. A user gets a key when the account is created. The users can use key to encrypt/decrypt the files to view the files through their accounts. Authentication is done through the login to an account. Create honeywords of a saved password in the database. Check whether the user is genuine or not, if yes, File upload/download rights, Can use key to encrypt/decrypt rights. If no, the hacker/spammer will be redirected to decoy data environment Fake user should not be login if password is incorrect.

3.1. Security Analysis of Honeyword

To protects against the misuse of the user’s real data. Conducted in a local file setting and password combination that provide evidence that this approach may provide unprecedented levels of user data security. Scrutinizing the honeyword system and highlights possible weak points. Also suggests the selection of the honeywords from existing user passwords in the system in order to provide realistic honeywords – a perfectly flat honeyword generation method where system stores all the passwords using honeywords.

Figure 3.1 shows System architecture for authentication of user .

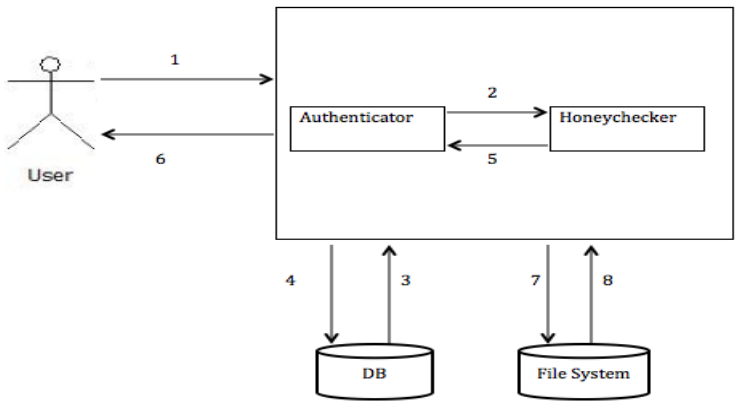

Figure 3.1: System Architecture for Authentication

3.2 Proposed Protocol for Login

Proposed protocol gives the detailed description about login conditions of authenticate user, unauthenticated user and fake user. From Figure 3.1 Users can be categorized according to their authentication as below,

• Case 1:

I. User submitting username and password to the system.

II. Authenticator authenticate user by submitting username and password to the honeychecker.

III. Honeychecker access database to validate user. Honeychecker checks user's password with the hash value of password.

IV. Honeychecker gets information from database.

V. If matches, honeychecker gives response to authenticator, user is valid.

VI. Accordingly authenticator gives response to user to access account.

VII. Honeychecker access database to views user's actions.

VIII. User may access file system to download and upload files.

• Case 2:

I. User submitting username and password to the system.

II. Authenticator authenticate user by submitting username and password to the honeychecker.

III. Honeychecker access database to validate user. Honeychecker checks user's password with the hash value of password.

IV. Honeychecker gets information from database.

V. If matches with one of the honeyword stored in the database, honeychecker gives response to authenticator, user is unauthenticated.

VI. Accordingly authenticator gives response to user to access account (but in dummy environment).

VII. Honeychecker access databse to views user's actions.

VIII. User may access file system to download and upload files in first login.After next login files must be removed from temp folder.

• Case 3:

I. User submitting username and password to the system.

II. Authenticator authenticate user by submitting username and password to the honeychecker.

III. Honeychecker access database to validate user.Honeychecker checks user's password with the hash value of password.

IV. Honeychecker gets information from database.

V. Ifnotmatcheswithhashofpasswordorwithoneofthehoneywordstoredinthedatabase, honeychecker

VI. gives response to authenticator, user is fake user.

VII. Accordingly authenticator gives response to user, username and password is incorrect.

3.3 Summary

In this chapter, working and design of proposed approach and proposed system are described. In the next chapter, Results and Discussion are presented.

REFERENCES

[1] Ali, A. 2001.Macroeconomic variables as common pervasive risk factors and the empirical content of the Arbitrage [1] H. Bojinov, E. Bursztein, X. Boyen, and D. Boneh, Kamouflage: Loss-resistant Password Management,in Computer Security–ESORICS 2010. Springer, 2010, pp. 286–302.

[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] 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..

[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] 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 again and again): Measuring Password Strength by Simulating Password- cracking Algorithms,in Security and Privacy (SP), 2012 IEEE Symposium on. IEEE, 2012, pp. 523–537.

[6] J. Bonneau and S. Preibusch, The Password Thicket: Technical and Market Failures in Human Authen- tication on the Web, in WEIS, 2010.

[7] G. Notoatmodjo and C. Thomborson, Passwords and Perceptions,in Proceedings of the Seventh Aus- tralasian Conference on Information Security–AISC 2009. Australian Computer Society, Inc., 2009, pp. 71–78.

[8] 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.

[9] Kelley, Saranga Komanduri, Michelle L. Mazurek, Richard Shay, Timothy Vidas Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor and Julio Lopez, Guess again and again and again: Measuring password strength by simulating passwordcracking algorithms ,in IEEE Symposium on Security and Privacy, pp. 523-537, May- 2012.

[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. [Online]. Available: http://doi.acm.org/10.1145/2187836.2187878

[11] Z. A. Genc, S. Kardas, and K. M. Sabir, Examination of a New Defense Mechanism: Honey- words,Cryptology ePrint Archive, Report 2013/696, 2013.

[12] Brown and Kelly, The dangers of weak hashes,SANS Institute InfoSec Reading Room, November 2013.

[13] D.Nagamalai,B.C.Dhinakaran,J.K.Lee,AnIn-depthAnalysisofSpamandSpammers,arXivpreprint arXiv:1012.1665, 2010.

[14] Imran ErgulerAchieving Flatness: Selecting the Honeywords from Existing User Passwords,IEEE Trans- actions on Dependable and Secure Computing, vol. 13, no. 2, pp. 284 - 295, February 2015.

[15] M.Hellman.Acryptanalytictime-memorytrade-o↵ .IEEETransactionsonInformationTheory,Volume 26, Issue 4, pages 401-406, 1980.

[16] P. Oechslin. Making a Faster Cryptanalytic Time-MemoryTrade-O↵ . Proceedings of Advances in Cryp- tology (CRYPTO 2003), Lecture Notes in Computer Science, Volume 2729, pages 617-630, 2003. Springer.

[17] Cohen, F.: The use of deception techniques: Honeypots and decoys. Handbook of Information Security 3 (2006) 646–655.

[18] Almeshekah, M.H., Spa↵ ord, E.H., Atallah, M.J.: Improving security using decep- tion. Technical Report CERIAS Tech Report 2013-13, Center for Education and Research Information Assurance and Security, Purdue University (2013)

[19] Herley, C., Florencio, D.: Protecting financial institutions from brute-force attacks. In: SEC'08. (2008) 681–685