A Literature Survey on Making Password Cracking Detectable using Honeywords

Prof. M. P. Deshmukh[1] Shekhar A. Pansare[2] Tejas K. Dhage[3] Akshay K. Pandit[4] Kunal A. Kasar[5]
[1,2,3,4,5]Department of Information Technology
[1,2,3,4,5]Rajarshi Shahu College of Engineering, Tathawade, Pune, India

Abstract— Honey word mechanism is used to detect an adversary who attempts to login with cracked passwords. 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. An adversary who steals a file of hashed passwords and inverts the hash function cannot tell if he has found the password or a honeyword. The attempted use of a honeyword for login sets off an alarm. An auxiliary server (the "honeychecker") can distinguish the user password from honeywords for the login routine, and will set off an alarm if a honeyword is submitted. This paper does a survey on how the passwords are stored on the server. This paper also discusses the different password composition policies implemented by different systems. Finally, this paper discusses about various approaches to overcome the challenges in the current authentication systems.

Key words: Honeyword, Honeypot, Sweetword, Sugarword, Password, Authentication

I. INTRODUCTION

In this there are two issues that should be considered to overcome these security problems: First passwords must be protected by taking appropriate precautions and storing with their hash values computed through salting or some other complex mechanisms. Hence, for an adversary it must be hard to invert hashes to acquire plaintext passwords. The second point is that a secure system should detect whether a password file disclosure incident happened or not to take appropriate actions. In this study, we focus on the latter issue and deal with fake passwords or accounts as a simple and cost effective solution to detect compromise of passwords. When a user sends a login request, the login server will determine the order of her among the users, and the order of the submitted password among her sweet words. The login server sends a message of the form to a secure server which is called "honeychecker", for the user and her sweet word. The honey checker will determine whether the submitted word is a password or a honey word. If a honey word is submitted, then it will raise an alarm or take an action that is previously chosen. The honey checker cannot know anything about the user's password or honey words. It maintains a single database that contains only the order of the true password among the user's sweet words.

II. LITERATURE REVIEW

In Password Cracking Using Probabilistic Context-Free Grammars [1], the paper says that choosing the most effective word-mangling rules to use when performing a dictionary-based password cracking attack can be a difficult task. In this paper we discuss a new method that generates password structures in highest probability order. We first automatically create a probabilistic context-free grammar based upon a training set of previously disclosed passwords. This grammar then allows us to generate word-mangling rules, and from them, password guesses to be used in password cracking. We will also show that this approach seems to provide a more effective way to crack passwords as compared to traditional methods by testing our tools and techniques on real password sets.

In Examination of a new defense mechanism: Honey words [2], the paper states that the decoy passwords i.e., honey words to detect attacks against hash password database. For each user account the legitimate password stored in form of honey words. If attacker Attack on password i.e., honey words it cannot be sure it is real password or honey word. It is much easier to crack a password hash with the advancements in the graphical processing unit(GPU) technology. Entering with a honey word to login will trigger an alarm notifying the administrator about a password file breach.

In Guess again: Measuring password strength by simulating password-cracking algorithms [3], We found several notable results about the comparative strength of different composition policies. Although NIST considers basic16 and comprehensive8 equivalent, we found that basic16 is superior against large numbers of guesses. Combined with a prior result that basic16 is also easier for users [46], 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.

In A large-scale study of web password habits [4], we 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. We 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.

In Improving Security Using Deception [5], as the convergence between our physical and digital worlds continues at a rapid pace, much of our information is becoming available online. In this paper we develop a novel taxonomy of methods and techniques that can be used to protect digital information. We discuss how information has been protected and show how we can structure our methods to achieve better results. We explore complex relationships

among protection techniques ranging from denial and isolation, to degradation and obfuscation, through negative information and deception, ending with adversary attribution and counter-operations. We present analysis of these relationships and discuss how they can be applied at different scales within organizations. We also identify some of the areas that are worth further investigation. We map these protection techniques against the cyber kill-chain model and discuss some findings.

Moreover, we identify the use of deceptive information as a useful protection method that can significantly enhance the security of systems. We posit how the well-known Kerckhoffs's principle has been misinterpreted to drive the security community away from deception-based mechanisms. We examine advantages these techniques can have when protecting our information in addition to traditional methods of hiding and hardening. We show that by intelligently introducing deceptive information in information systems, we not only lead attackers astray, but also give organizations the ability to detect leakage; create doubt and uncertainty in any leaked data; add risk at the adversaries' side to using the leaked information; and significantly enhance our abilities to attribute adversaries. We discuss how to overcome some of the challenges that hinder the adoption of deception-based techniques and present some recent work, our own contribution, and some promising directions for future research.

In Honeywords: MakingPassword-Cracking Detectable [6] if passwords are the only authentication mechanism in place, the adversary can then log in to the accounts of those users in a reliable and undetected manner. One approach to improve this situation is to make password hashing algorithm more complex and time-consuming. This approach not only help, but also slows down the authentication process for authrorized users, and doesn't make password cracking easier to detect. Honeywords overcomes this problems by having multiple possible passwords for each account, only one of which is genuine. The attempted use of a honeyword to log in sets on an alarm, as an adversarial attack has been reliably detected. This approach is not much deep, but it is quite eff ective, as it puts the attacker (hacker) at risk of being detected with every attempte to login using a password obtained by brute-force solving a hashed password. Consequently, honeywords also provides a very useful layer of defense.

The system can also incorporate an auxiliary secure server called the "honeychecker"to help with the use of honey words. The honeychecker is thus a separate independent computer system where such secret information can be stored. We assume that the computer system can communicate with the honeychecker when a login attempt is made on the computer system, or when a user changes his/her password. We also assume that the honeychecker is also capable of raising an alarm when an irregularity or unauthorized access is detected. The alarm signal may be sent to an administrator (user) or other party diff erent than the computer system itself. When it detects that something is going wrong with the login attempt, it could signal or communicate to the computer system that login should be denied. It sends a "silent alarm" to an administrator or user. Honeychecker can also be called as "login monitor" rather than a "honeychecker."

We have study carefully the security of the honeyword system and introduce a number of defect that need to be fitted with before successful realization of the scheme. In this respect, we have pointed out that the strong point of the honeyword system directly depends on the generation algorithm finally, we have presented a new approach to make the generation algorithm as close as to human nature by generating honeywords with randomly picking passwords that belong to other users in the system. 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 assensors 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. 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 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.

III. Challenges and Corresponding Preventing Approaches in Existing Systems

A. *CHALLENGES in existing system*

Passwords are a notoriously weak authentication mechanism. Users frequently choose poor passwords. An adversary who has stolen a file of hashed passwords can often use brute-force search to find a password p whose hash value H(p) equals the hash value stored for a given user's password, thus allowing the adversary to impersonate the user.

B. *APPROACHES to prevent this security issues*

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.

IV. Conclusion

We have study carefully the security of the honeyword system and introduce a number of defect that need to be fitted with before successful realization of the scheme. In this respect, we have pointed out that the strong point of the honeyword system directly depends on the generation algorithm Finally, we have presented a new approach to make the generation algorithm as close as to human nature by

generating honeywords with randomly picking passwords that belong to other users in the system.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 assensors 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. 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 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.

V. 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.

REFERENCE

[1] M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek, "Password cracking using probabilistic context-free grammars," in Proc. 30th IEEE Symp. Security Privacy, 2009, pp. 391–405
[2] Z. A. Genc, S. Kardas, and M. S. Kiraz, "Examination of a new defense mechanism: Honeywords," IACR Cryptology ePrint
[3] 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 Proc. IEEE Symp. Security Privacy, 2012, pp. 523–537.
[4] D. Florencio and C. Herley, "A large-scale study of web password habits," in Proc. 16th Int. Conf. World Wide Web, 2007, pp. 657–666.
[5] 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.
[6] Juels and R. L. Rivest, "Honeywords: Making passwordcracking detectable," in Proc. ACM SIGSAC Conf. Comput. Commun. Security, 2013, pp. 145–160.
[7] G. Notoatmodjo and C. Thomborson, "Passwords and perceptions," in Proc. 7th Australasian Conf. Inform. Security, 2009, pp. 71–78.

