International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER
SSRN

IJIEMR Transactions, online available on 6th Mar 2018. Link
:http://www.ijiemr.org/downloads.php?vol=Volume-08&issue=ISSUE-03

Title: FALSE DETECTION USING RELIABLE HONEY WORDS

Volume 08, Issue 03, Pages: 1–6.

Paper Authors

MR.K.SRINIVASA RAO, B.NARASIMHA RAO

Vignan's Lara Institute of Technology & Science

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

FALSE DETECTION USING RELIABLE HONEY WORDS

MR.K.SRINIVASA RAO[1], B.NARASIMHA RAO[2]

Assistant Professor[1], Department of M.C.A ,Vignan's Lara Institute of Technology & Science

M.C.A Student[2], Department of M.C.A Vignan's Lara Institute of Technology & Science

Abstract:

Digital certificates, based on X.509 PKI standard, are located at the core of many security mechanisms implemented in services and applications. However, the usage of certificates has revealed flaws in the certificate validation process (e.g., possibility of unavailable or non-updated data). This fact implies security risks that are not assessed. In order to address these issues that such flaws entail, we propose a novel probabilistic approach for quantitative risk assessment in X.509 PKI, together with trust management when there is uncertainty. We have evaluated our risk assessment approach and demonstrated its usage, considering as a use case the secure installation of mobile applications. The results show that our approach provides more granularity, appropriate values according to the impact, and relevant information in the risk calculation than other approaches.

Introduction

Password based authentication is the most widely accepted and cost effective authentication technique. In general practice, passwords are never stored in clear text to ensure confidentiality. Instead they are hashed and then stored along with other user related information. The process of performing a one-way transformation on the password and to obtain another string called the 'hashed' password is known as 'password hashing'. User selected passwords are mostly predictable, since humans have a tendency to choose non-random and easy to remember passwords [1]. 'Dictionary attack' [2] is the most widely used attack technique for retrieving a password from its hash value. In 'dictionary attack', the attacker creates a dictionary of commonly used passwords and computes their corresponding password hashes using the password hashing algorithm. Dictionaries with commonly used passwords can be efficiently created using inexpensive and massively parallelizable hardware such as Graphics Processing Units (GPUs). Any attacker with access to this precomputed dictionary, only needs to get access to the server database. He can then easily compare the entries and learn client passwords. A salt for password hashing refers to an additional public random input to the password hashing algorithm. It is stored in the database along with the password hash. Salts help randomize the otherwise deterministic password hashing algorithm. As a result the same password can be mapped to different password hashes. Use of salt prevents specialized attacks like the rainbow table

attack [3], when considering a large collection of hashes. For simplicity of presentation, we ignore the usage of salt in our constructions. However, our proposed schemes can be naturally extended to include the usage of salts and it is strongly recommended to use them. There are several ways to prevent an attacker from performing a dictionary attack by increasing the complexity of this attack manifolds. Making the password hashing algorithm more resource consuming is one way to prevent the adversary from pre-computing the dictionary. This was the main objective behind the Password Hashing Competition (PHC) [4] that ran from 2013-2015. To further improve the security, use of cryptographic module for password hashing is explained in [5]. Another approach is to introduce confusion by adding a list of fake passwords along with the correct password. This would discourage the adversary to mount dictionary attack even after compromising the database. This approach, proposed by Juels et.al. [6], of using fake passwords can help in detecting password database breaches. Specifically, any login attempt with one of the fake passwords detects the breach. The idea was influenced from some other existing techniques mentioned below. The honey pot technique [7], introduced in early 90's, is a system or component which influences the adversary to attack the wrong targets, namely honey pot accounts. Honey pot accounts are fake accounts created by the system administrator to detect password database breaches. Honey token is a honey pot that contains fake entries like social security or credit card numbers [8] to identify malicious activity. Kamouflage [9] is a theft-resistant password manager that creates multiple decoy password lists along with the correct password list. Frequent cases of password database breaches like that of LinkedIn in 2012 [10], Adobe in 2013 [11], eBay in 2014 [12], Ashley Madison in 2015 [13] etc., are indicative of security issues in the current password based authentication systems which can fail to ensure user privacy. In the case of LinkedIn, breach of 6.5 million passwords was reported in 2012. However, in May 2016, additional 100 million passwords were found, that were reportedly leaked in the same breach in 2012 [14]. In response to this, LinkedIn invalidated all the passwords that were not changed since 2012 [10]. No efficient solution to detect such database breaches had been reported in the literature prior to [6]. Therefore, the Honey words technique [6] is a significant contribution towards detecting breaches of the password database. In this technique, the server generates multiple fake passwords called honey words for each user, and stores them along with the actual password chosen by the user. Even if an attacker gets access to the password database, she would not be able to distinguish the actual password from honey words. Therefore with a very high probability, she is expected to enter a honey word to carry out the attack. If a honey word is entered instead of the password, the system raises an alarm, thus detecting the compromise of password database. The

efficiency of this system basically depends on the ability of the honey word generation scheme to generate honey words that are indistinguishable from the real password. The authors in [6], provide some heuristic honey word generation techniques, along with detailed analysis of the system implementing the honey words technique. Continuing along the same line of research, we provide an experimental method for quantifying the flatness of honey word generation schemes. We also implement a distance-measure between password and honey word using 'Levenshtein distance' [15] to avoid false detection when a legitimate user makes a typing error and enters a honey word.

Existing system:

There are several ways to prevent an attacker from performing a dictionary attack by increasing the complexity of this attack manifolds. Making the password hashing algorithm more resource consuming is one way to prevent the adversary from pre-computing the dictionary. This was the main objective behind the Password Hashing Competition (PHC). To further improve the security, use of cryptographic module for password hashing. Another approach is to introduce confusion by adding a list of fake passwords along with the correct password. This would discourage the adversary to mount dictionary attack even after compromising the database. This approach, of using fake passwords can help in detecting password database breaches. Specifically, any login attempt with one of the fake passwords detects the breach. The idea was influenced from some other existing techniques mentioned below. The honeypot technique, introduced in early 90's, is a system or component which influences the adversary to attack the wrong targets, namely honey pot accounts.

Drawbacks:

Honey pot accounts are fake accounts created by the system administrator to detect password database breaches. Honey token is a honey pot that contains fake entries like social security or credit card numbers to identify malicious activity. Is a theft-resistant password manager that creates multiple decoy password lists along with the correct password list. Frequent cases of password database breaches(like that of LinkedIn in 2012 , Adobe in 2013 , eBay in 2014 , Ashley Madison in 2015 etc.,) are indicative of security issues in the current password based authentication systems which can fail to ensure user privacy. No efficient solution to detect such database breaches had been reported

Proposed system

The Honey words technique is a significant contribution towards detecting breaches of the password database. In this technique, the server generates multiple fake passwords called honey words for each user, and stores them along with the actual password chosen by the user. Even if an attacker gets access to the password database, she would not be able to distinguish the actual password from honey words. Therefore with a very high probability, she is expected to enter a honey word to carry out the attack. If a honey word is entered instead of the password, the

system raises an alarm, thus detecting the compromise of password database. The efficiency of this system basically depends on the ability of the honey word generation scheme to generate honey words that are indistinguishable from the real password. The authors provide some heuristic honey word generation techniques, along with detailed analysis of the system implementing the honey words technique. Continuing along the same line of research, we provide an experimental method for quantifying the flatness of honey word generation schemes. We also implement a distance-measure between password and honey word using 'Levenshtein distance' to avoid false detection when a legitimate user makes a typing error and enters a honey word.

Advantages:

a new attack model called 'Multiple System Intersection attack considering Input'. We show that the 'Paired Distance Protocol' defined is not secure against this attack model.

Propose efficient and practical honey word generation techniques that can generate honey words indistinguishable from real passwords.

Modules:

Password Policy:-

Our system imposes the following less restricted and practical to implement conditions on password selection.

1) Username or its sub-string should not appear in the password.

2) The password should contain at least 8 characters including alphabets, special symbols and digits.

Typo-Safety:

Very high probability by maintaining a minimum distance between the password and each generated honey word. We suggest using 'Levenshtein distance' to compute the distance between password and honey words. 'Lavenshtein distance' is calculated by counting the number of deletions, insertions, or substitutions required to transform one string into another. It can be used to calculate distances between variable length strings. In this way, all types of human typing errors can be taken into account. Legacy-UI password changes:

Evolving password model:-

Define the key terms for the better understanding of the scheme. These terms are defined with respect to the available disclosed password databases.

Token: We consider token as a sequence of characters that can be treated as a single logical entity. In our context, for a given password, tokens are the alphabet-strings (A), digit-strings (D) or special-character-strings(S).

Pattern: The different combinations of tokens form patterns for a password, e.g., ADS1, AS2D, S1AS1D etc. Note: To create honey words indistinguishable from user password we do not preserve length of alphabets and digits, however we preserve the length of special-characters. Therefore the length of the special-character is mentioned as a subscript of S in the representation of pattern.

Frequency: It is the number of occurrences of the tokens or the password pattern in the available password database.

User-profile model: It generates honey words by combining some details from the user profile and checks the threshold of minimum distance with the password.

i) Create separate list of tokens named, token digits, token alphabet, token special Char from the information provided in user profile.

ii) To create k honey words, take k different combinations of elements from each token lists, satisfying the password policy of the service.

iii) Compare the tokens of the password with the tokens of the honey word. Reject the honey word if more than one token matches with password.

Conclusion:

A new honey word generation tech- niques which overcome several limitations of the existing honey word generation techniques. Our proposed methods produce honey words that are indistinguishable from the password and hence achieve 'approximate flatness'. To pre- vent false detection, in cases where legitimate user unin- tentionally enters a honey word, we implement Levenshtein distance to maintain minimum required distance (3 for our experiment) between password and honey words. We propose a new attack model and show that the 'Paired Distance Protocol' defined in [16] is completely broken in our attack model. The detailed analysis of existing honey- word techniques and their comparison with our proposed techniques is also provided.

References

1. Sharing in MULTICS. In Proceedings of the Fourth Symposium on Operating System Principles, SOSP 1973, Thomas J. Watson, Research Center, Yorktown Heights, New York, USA, October 15-17, 1973.

2. Robert Morris and Ken Thompson. Password Security: A Case History, 1979. http://cs-www.cs.yale.edu/homes/arvind/cs422/doc/unix-sec.pdf.

3. Philippe Oechslin. Making a faster cryptanalytic time-memory trade-off. In Dan Boneh, editor, Advances in Cryptology – CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings, volume 2729 of Lecture Notes in Computer Science, pages 617–630. Springer, 2003.

4. Password Hashing Competition (PHC), 2014. https://password-hashing.net/index.html.

5. Donghoon Chang, Arpan Jati, Sweta Mishra, and Somitra Kumar Sanadhya. Rig: A simple, secure and flexible design for password hashing. In Dongdai Lin, Moti Yung, and Jianying Zhou, editors, Information Security and Cryptology - 10th International Conference, Inscrypt 2014, Beijing, China, December 13-15, 2014, Revised Selected Papers, volume 8957 of Lecture Notes in Computer Science, pages 361–381. Springer, 2014.

6. Ari Juels and Ronald L. Rivest. Honeywords: making passwordcracking detectable. In 2013 ACM SIGSAC Conference on Computer and

Communications Security, CCS'13, Berlin, Germany, November 4- 8, 2013, 2013.
7. Fred Cohen. The Use of Deception Techniques: Honeypots and Decoys. http://all.net/journal/deception/Deception Techniques .pdf.
8. Lance Spitzner. Honeytokens: The Other Honeypot, 2003. http://www.symantec.com/connect/articles/ honeytokens-other-honeypot.
9. Hristo Bojinov, Elie Bursztein, Xavier Boyen, and Dan Boneh. Kamouflage: Loss-resistant password management. In Computer Security - ESORICS 2010, 15th European Symposium on Research in Computer Security, Athens, Greece, September 20-22, 2010. Proceedings, pages 286–302, 2010.
10. Wikipedia contributors. 2012 LinkedIn hack. Wikipedia, The Free Encyclopedia, Date retrieved: 29 May 2016. Available at: https://en.wikipedia.org/w/index.php?title=2012 LinkedIn hack&oldid=722095159.
11. Bruce Schneier. Cryptographic Blunders Revealed by Adobe's Password Leak. Schneier on Security, 2013. Available at: https://www.schneier.com/blog/archives/2013/11/ cryptographic b.html.
12. Swati Khandelwal. Hacking any eBay Account in just 1 minute, 2014. Available at: http://thehackernews.com/2014/09/ hacking-ebay-accounts.html.
13. Wikipedia contributors. Ashley Madison data breach. Wikipedia, The Free Encyclopedia, Date retrieved: 29 May 2016. Available at: https://en.wikipedia.org/w/index.php?title= Ashley Madison data breach&oldid=721001290.
14. Troy Hunt. Observations and thoughts on the LinkedIn data breach, 2015. Available at: https://www.troyhunt.com/ observations-and-thoughts-on-the-linkedin-data-breach/.
15. Michael Gilleland. Levenshtein Distance, in Three Flavors. Available at: http://people.cs.pitt.edu/_kirk/cs1501/assignments/editdistance/Levenshtein%20Distance.htm.

15. Bibliography

(1) Java Complete Reference by Herbert Shield
(2) Database Programming with JDBC and Java by George Reese
(3) Java and XML By Brett McLaughlin
(4) Wikipedia, URL: http://www.wikipedia.org.
(5) Answers.com, Online Dictionary, Encyclopedia and much more, URL: http://www.answers.com
(6) Google, URL: http://www.google.co.in
(7)Project Management URL: http://www.startwright.com/project.html