International Journal of Scientific Research and Management (IJSRM)
||Volume||5||Issue||06||Pages||5516-5624||2017||
Website: www.ijsrm.in ISSN (e): 2321-3418
Index Copernicus value (2015): 57.47 DOI: 10.18535/ijsrm/v5i6.36

An Efficient Privacy Preserving Technique Using Decoy Passwords

***Gannamaneni Siri Venkata Bhanu*[1], *Vankamamidi S Naresh*[2]**
[1,2]Dept of CSE, Sri Vasavi Engineering College, Tadepalligudem, India
[1] *siri.gannamaneni@gmail.com*, [2] *vsnaresh111@gmail.com*

Abstract

As per the rapid growth and essentiality in the field of information security, the data authentication and data access control are the major concerns. Now-a-days many users prefer to share their data through websites are bound to have many security related issues such as leakage of password file. To provide security against cyber threats in the above dimensions We proposed a decoy based passwords system. Honeywords are the Decoy (fake) passwords. If an attacker steals the password file, it will include original password and honeywords and password cracking becomes tougher. Further we improved the flatness of honeyword generation system using masked password generation algorithm based on randomized technique. So, for an adversary it will become harder to distinguish between original password and a honeyword. The administrator gets notification if any illegal login done apart from the genuine user. In this system the main advantage is that sensitive data can't be viewed by the unauthorized user. For this Identity Based Encryption (IBE) has been implemented. By this, only the user with genuine credentials allowed to view the original data.

Key Words: Authentication, honeywords, encryption, decoy passwords, password cracking, masked password generation.

I. Introduction

Information security has become a most prominent requirement in this era which is secured using some authentication method. It is the practice of preventing unauthorized access, use, disclosure, modification, inspection, recording or destruction of information. Many different methods for authentication exists (e.g. PINs, Patterns, Passwords etc.). Now-a-days the most popular authentication technology is the password based authentication. Security of password is an important aspect. A password is a secret word, which a user must input during a login, only after that the user is possible to get access. The selection of user password and storing them with a proper technique is the major issue. People generally pick the words that are easy to remember as their passwords. The password selection must be easy to remember and hard to guess. Attacks such as brute force attacks, Dos attacks are major issues. Here a legitimate users access rights to a computer and network resources

DOI: 10.18535/ijsrm/v5i6.36

are compromised by identifying the user id/password combination of the legitimate user. Password guessing attacks can be classified into two types:

- **Brute Force Attack:** A Brute Force attack is a type of password guessing attack and it consists of trying every possible code, combination, or password until you find the correct one. This type of attack may take long time to complete. A complex password can make the time for identifying the password by brute force long.
- **Dictionary Attack:** A dictionary attack is another type of password guessing attack which uses a dictionary of common words to identify the user's password.

Leakage of password files is a severe security problem that has affected millions of users and companies like Yahoo, RockYou, LinkedIn, eHarmony and Adobe [1], [2], since leaked passwords make the users target of many possible cyber-attacks. The entry point of a system which is required user name and password are stored in encrypt form in database. Once a password file is stolen, by using the password cracking technique it is easy to capture most of the plaintext passwords.

These recent events have demonstrated that the weak password storage methods are currently in place on many web sites. For example, the LinkedIn passwords and eHarmony system were also stored using unsalted MD5 hashes [3]. Indeed, once a password file is stolen, by using the password cracking techniques like the algorithm of Weir et al. [4] it is easy to capture most of the plaintext passwords. Password cracking is a process of recovering passwords that have been stored in or transmitted by a computer system. The best method of preventing a password from being cracked is to ensure that attackers cannot get access even to the hashed password. This makes it harder for a malicious user to obtain the hashed passwords in the first instance.

In this respect, 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.

Data access control and data authentication is an efficient way to ensure the data security in the internet. Online services are providing an effective solution for sharing information through website. This proposal initiates the study of two specific security threats on online security based password authentication in distributed systems. Honey words-based password authentication is one of the most popular security mechanisms to keep the passwords safe and secure. Recently some authors were proposed honeywords, which is also known as decoy passwords to detect attacks against hashed password databases. The individual user's password is stored with several honeywords in order to sense imitation. If honeywords are selected properly, a cyber-attacker who steals a file of hashed passwords cannot be sure if it is the real password or a honeyword for any account. In addition, entering with a honeyword to login will notify the administrator about a password file breach. And the proposed system also consists of randomization technique, which frequently changes the honeywords according to the popularity. In this system the main advantage is that sensitive data can't be viewed by the unauthorized user. For this Identity Based Encryption (IBE) has been implemented. By this, only the user with genuine credentials allowed to view the original data.

II. Related Work

A study was undertaken by Dennis Mirante, Justin Cappos [1] to research information posted on the web concerning recent, high profile website intrusions, wherein user login credentials and other data were compromised. It includes the attack mechanism utilized, the format in which the login data was stored, and the location of any password dumps pilfered from the site. News stories from trade related journals, press releases from the Victim Company, hacker sites, and blogs from individuals and companies engaged in security analysis were, in particular, searched in order to find related information. A total of thirty four breaches were researched According to many posts dealing with password security, good storage practice would dictate the use of bcrypt or PBKDF2 hash algorithms, a salt, and a large number of rounds.

The Psychology of Password Selection [2] In December of 2009 a social gaming site RockYou.com was hacked and 32 million passwords were exposed (Signler, 2009). An analysis of the passwords (Imperva, 2010) revealed several trends how users select passwords. People tend to use short passwords; 30% of the passwords were six characters or less and over 50% where eight characters or less. People tend to use a limited set of characters for passwords; 40% of people choose passwords consisting only of lower case letters, 16% of people used only numeric characters in their passwords, and less than 4% of people used special characters. People use common words for their passwords; 50% of people choose slang, dictionary words, or trivial passwords consisting of adjacent letters, numbers, or simple patterns for their passwords such as a word followed by a one or more numbers. The most common password was "123456". Of the 32 million accounts leaked, there were only 14.5 million unique passwords meaning there was a lot of duplication of passwords. The "123456" password for example, was used by over 290,000 different accounts. The RockYou password leak was a critical turning in password cracking.

In June of 2012 a hacker posted more than 8 million passwords to the internet belonging to LinkedIn and eHarmony [3]. Within hours, over two million of the passwords were cracked and posted on-line. Within a week, 99% of the passwords had been cracked. The eHarmony

DOI: 10.18535/ijsrm/v5i6.36

passwords were also stored using poor cryptographic practices as unsalted MD5 hashes. A month later, 450,000 were leaked from Yahoo (Gross, 2012). In this case, passwords were stored in clear text. MD5 is no longer considered to be a good hash algorithm because weaknesses have been found in the algorithm. Once you have salted your passwords and used a slow hashing algorithm, the next step is to introduce encryption.

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 [5].

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. One could imagine other uses of an auxiliary server to support of password-based authentication. The individual user’s password is stored with several honeywords in order to sense imitation. In addition, entering with a honeyword to login will trigger an alarm notifying the administrator about a password file breach [9].

ID-based encryption, or identity-based encryption (IBE) [12], is an important primitive of ID-based cryptography. It is a type of public-key encryption in which the public key of a user is some unique information about the identity of the use. This means that a sender who has access to the public parameters of the system can encrypt a message using receiver's name or email address as a key. The receiver obtains its decryption key from a central authority, which needs to be trusted as it generates secret keys for every user. A trusted third party, Private Key Generator (PKG), generates the corresponding private keys. To operate, the PKG first publishes a master public key, and retains the corresponding master private key (referred to as master key). To obtain a corresponding private key, the party authorized to use the identity ID contacts the PKG, which uses the master private key to generate the private key for identity ID. As a result, parties may encrypt messages (or verify signatures) with no prior distribution of keys between individual participants. This is extremely useful in cases where pre-distribution of authenticated keys is inconvenient or infeasible due to technical restraints. However, to decrypt messages, the authorized user must obtain the appropriate private key from the PKG. A PKG must be highly trusted, as it is capable of generating any user's private key. IBE can be used to build security systems that are more dynamic, lightweight and scalable [13].

A. Honeyword Generation Methods:

1. *Chaffing-by-tweaking:*

Applying chaffing by tweaking scheme, the using client password generates Honeywords. In this method, 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. After selection of that location we shuffle character from password. There is some limit while generation of honeyword because if it's doesn't there is chance that honeywords allocate lot of memory while generating honeywords. For example, by using this technique for the password 42hungry, the honeywords 12hungry and 58hungry may be generated.

2. *Chaffing with tough nuts:*

It means extra string added into the plain text. In this honeyword generation methodology our system inserts some tough word into the password so it's hard to crack password from hash files. So whenever password inserted by user there is some special string and character so and salty with original password so at that time it's hard to get original password. Using *Chaffing with tough nuts* method there is chance that attacker ignore the tough nuts.

3. *Chaffing-with-a-password-model:*

This model comprises of the password, splitted into character sets. On the off chance that the username and the password is co-related, the password can be easily recognized from the honeywords. E.g., the password NRGP143 with a username NRGP can be successfully recognized from the comparing honeywords.

III. Proposed Work

In this paper, we suggest an alternative approach that selects the honeywords from existing user passwords in order to provide realistic honeyword – a perfectly flat honeyword generation method.

Most users use same password on different systems. An old password of a user on some system may be the current password of that user on another system. Thus taking advanced security measurements may not guarantee the safety.

In the proposed system, we store all the passwords using honeywords. So the security increased in this mechanism. Admin has rights to add the decoy file for the uploaded file if illegal user tries password grouping then he can get access to files but those file are Decoy files (fake file).The main aim of project is to validate whether data access is legal or not when unusual information access is detected. Use of honeywords is very beneficial and works for each user accounts.

This application is extended in such a way if any other user other than owner attempts to login with any one of honey word other than original password, we can assume this login may be fake login, the proposed system recognizes this kind of user as fake user and after login, entire information is not accessed by that user, sensitive information may be hidden from that user, by this fake user can see only non-sensitive information. In this system IBE based encryption is used such that the fake user doesn't get any sensitive information related to that genuine account. Here if intruder attempts to break the system and if he/she enters any honeyword then the alert is given to the administrator.

A. Masked Password Generation Algorithm

DOI: 10.18535/ijsrm/v5i6.36

1. User login to system by entering username and password.

2. If login success user can perform all the appropriate operations

3. Else

4. Generate the sequences.

5. alpha=”0-9&&A-Z&&a-z&&symbols”

6. Select the password from user.

7. for (i=1; i<=n; i++)

8. The characters in the password string are replaced with any variables in alpha in a random manner.

9. Finally the resultant string gets shuffled and n honeywords are generated.

10. Again login done with the honeywords and it repeats the step 4.

11. end if

B. Identity Based Encryption

1. User login with username and password
2. If login success user page will be displayed
3. Until the user gets authorized by the admin, the sensitive data is hided from user.
4. Setup()
    - i. Run by the Private Key Generator (PKG) one time for creating the whole IBE environment.
    - ii. The master key is kept secret and used to derive user private keys, while the system parameters are made public.
    - iii. Output: Public system parameters P , a master-key Km which is known only to the PKG.
5. Key Extract()
    - i. The process which the PKG generates the private key for user.
    - ii. Input: system parameters P, master-key Km and any arbitrary ID (i.e., the public key)
    - iii. Output: private key d
6. Encryption ( )
    - i. Input: A message M, the identity IDs of the sender, the private key ds of the sender and an identity set {ID1, ID2, . . . , IDt} .
    - ii. Output: cipher text C
    - iii. C = Encrypt (params, IDs, ID1, ID2…, IDt ,M, ds).
7. Decryption ( )
    - i. Input: cipher text C and the secret key di of user IDi.
    - ii. Output: plaintext message M
    - iii. M = Decrypt (params, C, di).

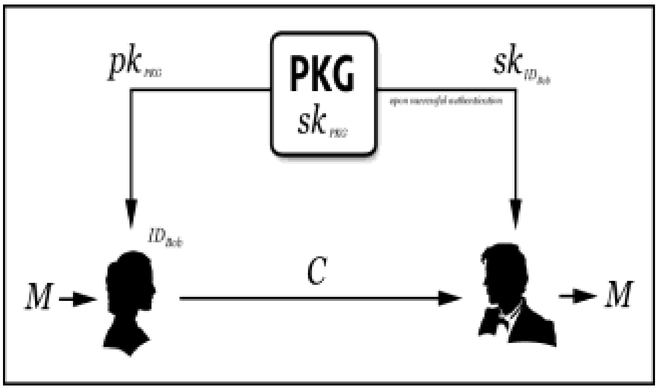

Figure 1: Identity Based Encryption

C. Proposed Architecture

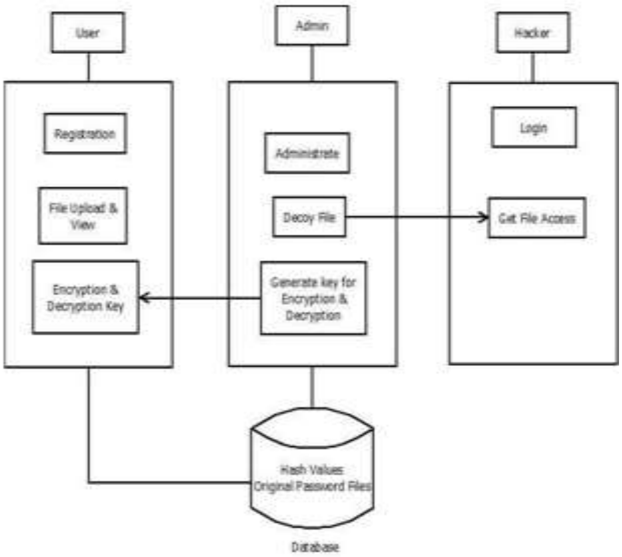

Figure 2: Proposed Architecture

IV. Results and Analysis

Table 1: Comparision of Honeywords generation methods

Honeyword generation Method	Dos Resistance	Flatness	Storage Cost
Tweaking	Weak	Weak	hN^*
Take-a-tail	Weak	Weak	hN^*
Our model (Masked Password Generation)	Strong	Strong$^+$	khN^*

The comparison results are summarized in Table. Note that the same expressions of [9] are used for these table entries. By weak DoS resistance we mean an adversary who knows the password can hit the one of corresponding honeywords with a non-negligible chance; while by strong we mean that this chance is ignorably small. The + is used for condition that its strength depends on how the real password list is used. The * indicates optimization technique is considered in storage cost calculation.

V. Conclusion and Future Enhancement

We proposed a novel approach to secure personal and business data in the system. Further we studied monitoring data access patterns by profiling user behaviour to determine if and when a malicious insider illegitimately accesses someone’s documents in a system service. The honeyword system is powerful defense mechanism in this scenario. Namely, even if the adversary has broken all the hashes in the password file, he cannot login to the system without a high risk of being detected. Success of the method in flatness depends on how honeywords are generated. The honeywords generated using the masked password generation algorithm can fulfil its claims provided that the generator algorithm is flat. The Identity Based Encryption provides privacy for the data by hiding the original details. In the future, we would like to enhance our work by using hybrid honeywords generation algorithms.

DOI: 10.18535/ijsrm/v5i6.36

References

[1] D. Mirante and C. Justin, “Understanding password database compromises,” Dept. of Comput. Sci. Eng. Polytechnic Inst. of NYU, New York, NY, USA: Tech. Rep. TR-CSE-2013-02, 2013.

[2] A. Vance, “If your password is 123456, just make it hackme,” New York Times, Jan. 2010.

[3] K. Brown, “The dangers of weak hashes,” SANS Institute InfoSec Reading Room, Maryland US, pp. 1–22, Nov. 2013, [Online]. Available: http://www.sans.org/reading-room/ whitepapers/authentication/dangers-weak-hashes-34412.

[4] 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.

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

[6] 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.

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

[8] 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.

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

[10] Erguler, Imran. "Achieving Flatness: Selecting the Honeywords from Existing User Passwords."IEEE Transactions on Dependable and Secure Computing (2015): 1-14.

[11] Z. A. Genc, S. Kardas, and M. S. Kiraz, “Examination of a new defense mechanism: Honeywords,” IACR Cryptology ePrint Archive, Report 2013/696, 2013.

[12] https://en.wikipedia.org/wiki/ID-based_encryption

[13] https://www.voltage.com/technology/data-encryption/identity-based-encryption

[14] L. Martin, Introduction to Identity Based Encryption, Artech House Publishers; 1 edition

About the Authors:

Gannamaneni Siri Venkata Bhanu is pursuing M.Tech in Sri Vasavi Engineering College in the branch of Computer Science and Engineering.

Dr. Vankamamidi Srinivasa Naresh is currently working as Associate professor in Sri Vasavi Engineering College. He obtained an M.Sc. in Mathematics from Andhra University, an M.Phil. in Mathematics from Madurai Kamaraj University and an M.Tech & Ph.D in Computer Science and Engineering from J.N.T.UK- Kakinada. He is also a recipient of U.G.C.-C.S.I.R. JUNIOR RESEARCH FELLOSHIP and cleared NET for

LECTURERSHIP He completed UGC Minor Research Project with a financial assistance of Rs. 3,70,000.