Enhancing the security using Honeywords

Bhagyashri D.Suryawanshi
Computer engineering
SSBT COET
Jalgaon, India

Paresh B.Tayade
Computer engineering
SSBT COET
Jalgaon, India

Akshay V. Patil
Computer engineering
SSBT COET
Jalgaon, India

Jeevan B. Patil
Computer engineering
SSBT COET
Jalgaon, India

Deepak V. Rajput
Computer engineering
SSBT COET
Jalgaon, India

***Abstract*—Disclosure of password files is a severe security concerns making millions of vulnerable to cyber-attack. A cyber security is nothing but cracking the user's password and stolen files. A brute force attack on the stolen hashfiles can be easily recover the user's password. Honeyword (Decoy Password) method is used to detect attack against hashed password database. A secure server can distinguish on user's real password among Honeyword and send a mail whenever a honeyword is used. Thus the Honeyword Technique is used to detect the unauthorized attempt and protect from stolen.**

Keywords— honeyword, hashfiles, vulnerable

I. INTRODUCTION

In the authentication process it becomes difficult to handle security of passwords that's why password became the most important asset to login. But users choose weak passwords (for easy to remember) that can be predicted by the attacker using brute force, dictionary, rainbow table attacks. Hence it becomes much easier to crack a password. An attacker can recover a user's password using brute-force attack on password hash. Once the password has been recovered no server can detect any illegitimate user authentication. So Honeywords plays an important role to defense against stolen password files.

II. LITERATURE SURVEY

Actually passwords are weak authentiocaiton mechanism.But users prefer weak password that can be easily cracked by an attacker using brute force attack, dictionary attack. A password p having hash value H (p) equals the stored hash value is selected by an adversary who has stolen a file of hashed passwords and it can often use brute-force search to impersonate the user [2].An effective approach is needed in the scenario where an adversary has stolen the password hash files. Honeypot is one of the methods to identify occurrence of a password database breach. In the approach,user account are purposely creates by administrator to lure attacker and detect a password disclosure if any one of the honeypot account get used. The idea has been modified by Herley and Florencio 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. Use of decoys for building theft-resistant was introduced by Bojinov et al. in called as Kamouage.

Juels and Rivest have proposed the honeyword mechanism to detect an attacker who try to login with cracked passwords. Basically for each username a set of sweetwords is constructed.In those sweetwords only one is user real password and remaining all are honeywords[1]. So when attacker tries to login user account with honeyword an alarm is generated to notify adminstrator about a password leakage Following are some honeyword generation methods tweaking, take-a-tail and hybrid method. First in tweaking method it alters the character at specified position. Second in take-a-tail method it appends some digits at the end. and in hybrid method it is combination of above two methods. [1] The proposed system uses case alteration method in which only the case of specific characters is altered i.e.from uppercase to lowercase and vice-a-versa.

III. METHODOLOGY

A. Existing System

Two types of approaches are used to generate Honeywords.

1. Chaffing by Tweaking.

2. Take-a-Tail

In the first approach, it Tweak selected character positions of the password to obtain the honeywords. And For each selected position the character of the real password is replaced by a randomly-chosen character of the same type [1][2].

Alternatives approaches are also available i.e

1. Chaffing-by-tail-tweaking: tweak last t positions of Password.

2. Chaffing-by-tweaking-digits: tweak last t positions containing digits.

Example where t = 5

Password77 = PassHord12 PassCord45 PassVord67

In the second approach,it Request password from user and then modify it with a randomized tail.

Submitted password: Password

Append '77' to make new password

Actual Password: Password77

Generated honeywords :

Password12

Password34

Password56

Password67

Password78

Password90

In the next approach, above explain methods are combined known as 'Hybrid generation method'. In this approach,Combining several methods can result in better honeywords. It Require the user to use digits at the end of the password. [1]

1. Chffing-with-a-password to generate new random words.

2. Chffing-by-tweaking-digits on all words

Example: Actual Password: Alice77

Generated honeywords:

Alice85 Bob49

Alice65 Bob14

B. *Proposed System*

Our proposed approach is identified as 'Case alteration Method'. It takes full advantage of the case sensitive nature of password authentication systems. It uses the actual password as a seed to generate decoy passwords or honeywords. The actual password may contain lowercase characters, uppercase characters or combination of both. A set of honeywords is generated such that the position of the characters remains same for each honeyword but some lowercase characters are converted into respective uppercase characters and some uppercase characters are converted into respective lowercase characters. The approach does not alter the semantic significance of the generated passwords if any. The case sensitive nature of authentication system can distinguish between an actual password and honeyword.

For example, Let an actual password be Password. It contains 1 uppercase letter and 7 lowercase letters. The proposed case alteration system generate different number of honeywords by altering cases of some characters of the word Password. The generated honeywords may contain more than one uppercase characters and/or more than one lowercase characters. Some examples of the generated honeywords using the proposed case alteration system are as follows: password,PASSWORD,PassWORD,paSSWORD,PaSsWoRd, pAsSwOrD,PASSword,etc. for 8 digit password 256 honeywords can be generated by using this simple approach. For a password of length alphabetic characters, 2 to raise n honeywords can be generated by using the simple approach. It uses permutation of 2 cases for every alphabetic character namely uppercase and lowercase. The approach maintains semantic significance of the honeywords and hence makes it difficult for attacker to guess the actual password as all the honeywords appear semantically similar. Further embellishments like substitution of special symbol for some characters and vice- a versa can be added for generating more honeywords.

E.g. 1l,2z,3E,4A, 5S,6G, 7T,8g, 9q,0O. These takes care of digits contained in passwords.

While generating honeywords it makes sure that the honeywords look similar in appearance without altering semantic significance if any. These takes care of worst case scenario for the proposed approach. i.e. A password containing all digits like phone numbers, etc.

The following diagram shows the system architecture .The flow of the system is describes in this Fig. 1.

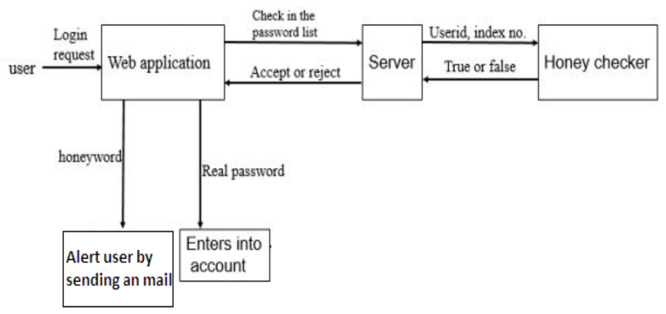

Fig. 1. System Architecture.

Honeyword Generation Algorithm

```
Honeyword(pass)
l=length(pass)
char =char .replace(oldchar,newchar) //for special symbol and digits char[]=char;
for j<- 1 to l;
for n<- 1 to l;
if j mod n equals to even number
convert charAt[j] into uppercase
else
convert charAt[j] into lower case
end if
end for
end for
```

In our approach, the honeychecker is an auxillary server which stores the index of correct password. The Main Role of honeychecker is when the user will submit the password ,the main server will send the index of the submitted password the honeychecker. After that honeychecker checks the index of the submitted password with index of the real password,. Based on that it will take corrective action.

Honeychecker Algorithm

```
Honeychecker(User name,index)
i1=index of original password
i2=index of Entered password
if i2 equals to i1 are equal then Access
else if i2 not equals to i1 then Send mail
else Access Denied
end of else
end of else
end of if
```

IV. Result And Analysis

It describes the result obtained for both the server and the clients of the proposed system. These results are examined and analyzed to infer that they are according to the expected results or not. At the client side, 1st the user will do the registration. After registration server will generate the honeywords from their registered password. then he/she will do the login.In the login process if the user will enter the real password then the user can get the access to the system.if the user will enter the honeyword then it will send an email along with the IP.

The computer system has n number of users say user1, user2, . . . , user n; here ui is the username for the ith user. for user ui pi is the passsword. This is the correct, legitimate, password; it is what user ui uses to log in to the system. The system uses a cryptographic hash function H for computing the hash value and stores hashes of passwords rather than raw passwords. That is, the system maintains a file F listing username / password-hash pairs of the form (ui,H(pi)) for i = 1, 2, . . . ,n. The idea is that the system may include an auxiliary secure server called the honeychecker to assist with the use of honeywords. The honeychecker is thus a separate computer system where a secret information can be stored and when a login attempt is made on the computer system, or when a user changes her password can communicate with the computer system. In case of an irregularity the honeychecker is capable of sending an email. To generate the honeywords we follow a simple procedure. For each user ui, a list Wi of distinct words (called potential passwords or more brief, sweetwords) is represented: Wi = (wi,1,wi,2, . . . ,wi,k) .For simplicity we assume that k is a fixed system –wide parameter, although k could be chosen in a per-user manner. The list Wi of sweetwords thus contains one sugarword (the password) and (k 1) honeywords (the char).They also allow a sweetword called a tough nut that is, a very strong password whose hash the adversary is unable to invert. A honeyword, or the password itself, may be a tough nut.

TABLE I. Result and Analysis

Hybrid Methods	DOS Resistance	Flatness
Tweaking	Weak	Weak
Take-a-Tail	Weak	Weak
Hybrid	Strong	Strong
Case alteration Method	strong	Strong

for example: suppose,

real password=Password

Honeyword=PaSSword,

PAssword,

pAssWoRD,

PaSSwoRd

Therefore, our approach is simpler and more practical for implementation

TABLE II. Result and Analysis

Input	Output	Access to the system
Password	Accepted	Access to the system
PaSSwoRd	Honeyword	Send a mail
PasSWORD	Rejected	Access denied

V. Discussion

A. Advantages

- This is little bit simple algorithm compare to previous one.
- No additional database is required for chaffing.
- Overhead of additional stuffing is absent as in the take a tail method.
- Generated honeywords looks similar in appearance making it difficult to guess.
- In worst case scenario also 50% security is assured. Suppose, length of the password is 1 Still 2 honeywords will be generated.

B. Diadvantges

- Overhead of implementation of additional server called honeychecker is required.
- The honeyword are generated in the form of 2n i.e as the length of n increases the size of the database also increases.
- Thus, complexity increases.

Conclusion

Honeywords provide many Advantages. Exposed password files provide attackers with insight into how users include their passwords. The models can then refine by attacker of user password selection and design faster password cracking algorithms. Thus every break of a password server has the capacity to improve future attacks. Some honeyword generation techniqes, specifically chaffing-by-Tweakin ones, obscure actual user password choices, and thus complicated models are built for hash crackers. It may even be useful to muddy attacker's knowledge of user's composition choices intentionally by drawing some honeywords from slightly perturbed probability distributions. Instead of their benefits over common methods for password management, honeywords are not a wholly satisfactory approach to user authentication. They inherit many of the well-known drawbacks of passwords and something-you-know authentication more generally.passwords should be provided with stronger and more convenient authentication methods or give way to better authentication methods completely, as recently predicted by the media. for user convenience,further attempts can be provided in case the user mistakenly enters a wrong password instead of changing password immediately. this has to be implemented while maintaining the balance between user convenience and security measures.

References

[1] A. Juels and R. L. Rivest. Honeywords: Making password cracking Detectable Unpublish draft.

[2] Achieving Flatness:selecting The Honeywords From Existing User Passwords vol.13,No.2,March/April,2016.

[3] M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek, Password cracking using probability context free grammar,in Proc.30th IEEE symp.Security Privacy,2009

[4] Honeywords: A New Approach For Enhancing Security, Manisha Jagannath Bhole. IRJET,volume 2,Nov 2015.

[5] J. Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million password.In IEEE symp.Security Privacy ,2012

[6] D. Elser and M. Pekrul. Inside the password-stealing business: the who and how of identity theft, 2009.

[7] L. Spitzner. Honeytokens: The other honeypot. Symantec SecurityFocus, July 2003.L. Spitzner. Honeytokens: The other honeypot. Symantec SecurityFocus, July 2003.

[8] D. Mirante and C. Justin, Understanding password database compromises, Dept. of Comput. Sci. Eng. Polytechnic Inst. of NYU, New York.