International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue V, May 2019- Available at www.ijraset.com

Achieving Flatness Using Honeywords Generation Algorithm

Sonali Bamane[1], Monika Shinde[2], Mrs.Amrapali Mhaisgawali[3], Mangal Bargaje[4], Dr.Sanjay Pawar[5]
[1, 2, 3, 4]*Computer Science and Technology, Usha Mittal Institute of Technology, SNDT Women's University, Santacruz(W), Mumbai-400049.*

Abstract: ***Honey words (decoy passwords) have been proposed to detect attacks against hashed password databases. For each user account, the original password is stored with many honey words in order to thwart any adversary. The honey words are selected deliberately, such that a cyber-attacker who steals a file of hashed passwords cannot be sure, if it is the real password or a honey word for any account. Moreover, entering with a honey word to login will trigger an alarm notifying the administrator about a password file breach. At the expense of increasing the storage requirement by 24 times, the authors introduce a simple and effective solution to the detection of password file disclosure events. In this study, we scrutinize the honey word system and highlight possible weak points. Also, we suggest an alternative approach that selects the honey words from existing user information, a generic password list, dictionary attack, and by shuffling the characters. Four sets of honey words are added to the system that resembles the real passwords, thereby achieving an extremely flat honey words generation method. To measure the human behaviors in relation to trying to crack the password, a tested engaged with by 820 people was created to determine the appropriate words for the traditional and proposed methods. The results show that under the new method it is harder to obtain any indication of the real password (high flatness) when compared with traditional approaches and the probability of choosing the real password is 1/k, where k = number of honey words plus the real password.***
Keywords: Authentication, honeypots, honeyword, honeycheker, security.

I. INTRODUCTION

Leakage of the password file is the main security issue that has been affected to many user's accounts. Hence, we introduce a new technique for improving safety of the password by inserting some honeywords i.e. false passwords with original passwords to every user's accounts. If the password file is get cracked by the attacker then also attacker will not be able to identify that either he is generating real password or fake passwords i.e. honeywords. If the attacker uses honeywords to login units off an alarm. In this improved security system honey checker is used to differentiate real passwords and fake passwords. Whenever unauthorized user submits hone words to the login system them alarm will get triggered and send notification via mail. There are two major problems that should be consider to solve security problems: first is Password must be secured by taking right precautions and storing with their hash values through the complex mechanism. Second is that, secure system must be able detect whether the password file leakage incident happened or not for taking any appropriate action. considering increase in the use of internet and simultaneously increase in cyber-attacks, there is no doubt that, strong security is vital for protecting organization systems and information. For some time, network security has been strongly protected by using some traditional network security devices such as routers, firewalls etc. Honeypots also playing an important role in security problem solutions honeypots are the systems which is used to analyzed and record the action of attackers. it is also used to take away attackers from his goal. Only honey pots cannot solve system security problems, they are only tools that supports to the traditional network devices.So proposed system is the solution for the mentioned security problems. Here we generate a set of honeywords that contained only one correct password and others are fake passwords. Hence when the attacker tries to enter into the system with fake password, and alarm will get triggered and send notification to the administrator about the password file breach.

II. RELATED WORK

"Achieving flatness: Selecting the honeywords from existing user passwords." which includes the recent knowledge as well as theoretical and methodological contributions to a paper.[8] it is mainly proposed for reducing security issues. Honeywords generation algorithm which is used in this system to overcome security problems shows positive results towards the flatness of the system. once honeywords are selected properly, a attacker who bargains a file of hashed passwords cannot sure if it is original password or fake passwords.[8] If the attacker enters wrong passwords i.e. honeywords then alarm will get triggered.

It is easy to crack password with the help of graphical processing unit (GPU) technology. By the brute-force attack attacker can crack user password. Once the password is get stolen by the attacker system cannot detect to any unauthorised user. Hence by considering security issues Rivest and Jules published paper for solving passwords security problems. the evaluated a concept of a fake passwords that is honeywords for the authorized user authentication. If the password file is get stolen the attackers then it includes honeywords i.e. fake passwords along with the original password. One secure server is added to the system to detect original and fake passwords. When the attackers uses honeywords to login then alarm will triggered automatically. So in this paper we analysed some possible improvement in security which is easy to implement as well as we introduce a new technique model which is the solution to an password security issues [1].

Password leakage is the main issue related to the security hence to overcome this problem we implement new system which can minimise damage caused by the leakage password. A system is said to be secured if it does not have passivity or weakness that may cause any unauthorised access of password. If the password hashes get hacked, then also it should not easy to create password from the hashed password. It was found that many accounts get affected because of password cracking in 2012.[2] In this paper author has discussed about password hacking and improved tricks that are used while password storage.[2]

David Malone, Kevin Maher NUI Maynooth ,"Investigating the Distribution of Password Choices" In this paper we will look at the distribution with which passwords are selected. Zipf?s Law is commonly observed in lists of selected words. Using password lists from four di□erent online sources, we will scrutinize if Zipf?s law is a good candidate for describing the frequency with which passwords are selected. We look at a number of standard statistics, used to estimate the security of password distributions, and see if modelling the data using Zapf's Law produces good evaluation of these statistics. We then look at the analogy of the password distributions from each of our sources, using guess as a metric.[3] This shows that these distributions provide adequate tools for cracking passwords. Finally, we will show how to frame the distribution of passwords in use, by asking users to choose a di□erent password.[3]

Mohammed Alma shekh, Eugene H. Spa□ord, Mikhail J. Atallah,"Improving Security Using Deception" As the merge between our physical and digital worlds continues at a rapid step, much of our information is getting available online. In this paper, they developed a novel taxonomy of methods and techniques that can be used to protect digital information. They discuss how information has been secured and show how we can structure our methods to achieve better output. They explore complex relationships among security techniques ranging from denial and isolation, to degradation and confusion, through negative information and deception, ending with attacker attribution and counter-operations. They present analysis of these relationships and discuss how they can be applied at di□erent place within organizations. They also found some of the areas that are worth further investigation. We map these security techniques against the cyber kill-chain model and discuss some findings. They identify the use of ambiguous information as a useful security method that can significantly improve the security of systems. [4]

Ari Juels, Ronald L. Rivest ,"Honeywords: Making Password-Cracking Detectable" They suggested a simple method for enhancing the security of hashed passwords: the maintenance of other honeywords" (decoy passwords) related with each user's account. An attacker who steals a file of hashed passwords and inverts the hash cannot tell if he has found the password. The attempted use of a honeyword for login sets an alarm. An auxiliary server (the honeychecker") can di□erentiate the user password from honeywords for the login, and will set o□ an alarm if a honeyword is submitted.[5]

Matt Weir, Sudhir Aggarwal, Breno de Medeiros, Bill Glodek,"Password Cracking Using Probabilistic Context-Free Grammars " Selecting the most e□ective word-distorting rules to use when performing a dictionarybased password cracking attack can be a tough task. They discuss a new method that generates password structures in highest probability order. They first automatically create a probabilistic context free grammar depend on a training set of previously disclosed passwords. This grammar then allows us to create word-mangling rules, and from them, password guesses to be used in password cracking. They will also show that this approach seems to provide a more e□ective way to crack passwords as compared to traditional techniques by testing our tools and techniques on real password sets.[6]

Dinei Florencio and Cormac Herley,"A Large-Scale Study of Web Password Habits"This paper describes the 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.[7]

Dhinaharan Nagamalai, Beatrice Cynthia Dhinakaran and Jae Kwang Lee,"An In-Depth Analysis of Spam and Spammers "This paper describes 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.[8]

Patrick Gage Kelley, Saranga Komanduri, Michelle L. Mazurek, Richard Shay, Timothy Vidas Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor and Julio Lopez ,".Guess again (again and again): Measuring password strength by simulating password-cracking algorithms"This paper describes the e□ects of password-composition policies on the guess ability of passwords. Seven di□erent 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 e□ectiveness of password-guessing attacks. Results also reveal important information about conducting guess-resistance analysis. E□ective 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 di□erences in guessability among password sets.

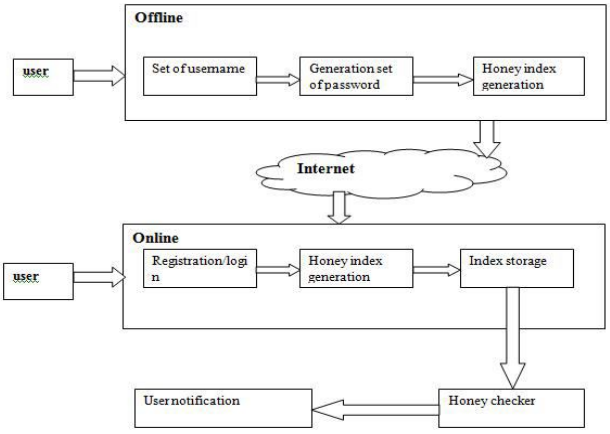
III. SYSTEM ARCHITECTURE

A. *Problem Statement*

Suggest an alternative approach that selects the honeywords from existing user passwords in order to provide realistic honeyword a perfectly flat honeyword generation method. Leakage of password files is a severe security problem that has affected millions of users Accounts and companies like Yahoo, LinkedIn and Adobe, since disclosed passwords make the users target of many possible cyber-attacks. These recent activities have demonstrated that the weak password storage methods are currently in place on many web-servers. For example, the LinkedIn passwords encryption were using the SHA-1 algorithm without a salt and similarly the passwords in the eHarmony system were also stored using unsalted MD5 hashes.[1] Indeed, once a password file is stolen, by using the password cracking techniques like the algorithm of Weir et al. it is easy to capture most of the plaintext passwords. In this respect, there are two problems that should be considered to solve these security problems: First, passwords must be protected by taking right precautions and storing with their hash values computed through salting or other complex mechanisms. Hence, for an attacker it must be hard to invert hashes to get plaintext passwords. The second point is that a secure system should detect whether a password file leak incident happened or not to take appropriate actions. In this study, we focus on the latter issues and deal with fake passwords or accounts as a simple and cost effective solution to detect compromise of passwords. Honeypot is one of the best solution to identify occurrence of a password database breach.

B. *Existing system*

Now days the password hacking is major issue and it leads so many security threads. The user_id and password is used to validate the authentication of user in many applications. If the password is hacked, then attacker can take important data frothuser account. This hack activity happen by taking the password files forms the database.



. Fig 1:-Project Overview

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue V, May 2019- Available at www.ijraset.com

C. Proposed System

Attacks against password hashes are offline attacks. This means the attackers have already stolen the password file and are using their own computer to attempt various password combinations to find a hash in the stolen password hash file. Since this is an offline attack, controls such as account lockout or capuches provide no value. Those controls are valid only for online attacks against the login page of a running web server.

Suggest an alternative approach that selects the honey words from existing user passwords in order to provide realistic honey word a perfectly flat honey word generation method. 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.

IV. METHODOLOGY

By using various algorithms such as MD5 and Hybrid method this dissertation ideas implemented. All the approaches used for problem solving are based on Honey word generation Method to detects the unauthorized user which will also provide privation from DOS attack and Brute-force attack. Various algorithms are used for the purpose of detecting unauthorized user and safe our password.

Hence, all these things helps to solve all the problems regarding a perfectly at Honey word generation method.

1) Algorithm: MD5 (for hash value)

MD5 Algorithm Description

The following steps are performed to compute the message digest of the message.

a) Step 1. Append Padding Bits

The input message is "padded" (extended) so that its length (in bits) equals to 448 mod 512. Padding is always performed, even if the length of the message is already 448 mod 512. Padding is performed as follows: a single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits of the padded message becomes congruent to 448 mod 512. At least one bit and at most 512 bits are appended.

b) Step 2. Append Length

A 64-bit representation of the length of the message is appended to the result of step1. If the length of the message is greater than 2^64, only the low-order 64 bits will be used.

The resulting message (after padding with bits and with has a length that is an exact multiple of 512 bits. The input message will have a length that is an exact multiple of 16 (32-bit) words.

c) Step 3. Initialize MD Buffer

A four-word buffer (A, B, C, D) is used to compute the message digest. Each of A, B, C, D is a 32-bit register. These registers are initialized to the following values in hexadecimal, low-order bytes first):

word A: 01 23 45 67
word B: 89 ab cd ef
word C: fe dc ba 98
word D: 76 54 32 10

d) Step 4. Process Message in 16-word Block

Four functions will be defined such that each function takes an input of three 32-bit words and produces a 32-bit word output.

F (X, Y, Z) = XY or not (X) Z
G (X, Y, Z) = XZ or Y not (Z)
H (X, Y, Z) = X xor Y xor Z
I (X, Y, Z) = Y xor (X or not (Z))

e) Step 5. Output

i) MD5 is simple to implement, and provides a "fingerprint" or message digest of a message of arbitrary length.
ii) It performs very fast on 32-bit machine.
iii) MD5 is being used heavily from large corporations, such as IBM, Cisco Systems, to individual programmers.

2) *Algorithm:* honey checker and honey word generation

a) Inputs

i) T fake user accounts (honey pots)

ii) index value between [1;N]

iii) index list ,which is not previously assign to user

b) Procedure

i) Step 1: Honey pots creation: fake user account

1. For each account honey index set is created like Xi =>(xi;1; xi;2; : : : ; xi;k); one of the elements in Xi is the correct index (sugar index) as Ci
2. Create two password _le _le f1 and _le f2. F1 Store username and honey index set <hui,xi> Where hui is honey pot account and F2 keeps the index number and the corresponding hash of the password <Ci;H(pi)>

ii) Step 2: Generation of honey-index set

In Step 1 we insert honey index set in _le F1 but dont know how to create that We use
honey index generator algorithm Generator(k; SI) Ci;Xi Generate Xi.

1. select xi randomly selecting k-1 numbers from SI and also randomly picking a number ci SI .
2. Ui; ci pair is delivered to the honey checker and F1, F2 _les are updated.

iii) Step 3: Honey checker

Set: Ci, Ui Sets real password index Ci for the user Ui Check: Ui, j

Checks whether ci for Ui is equal to given j. Returns the output and if equality does not hold, notices system a honey word situation.

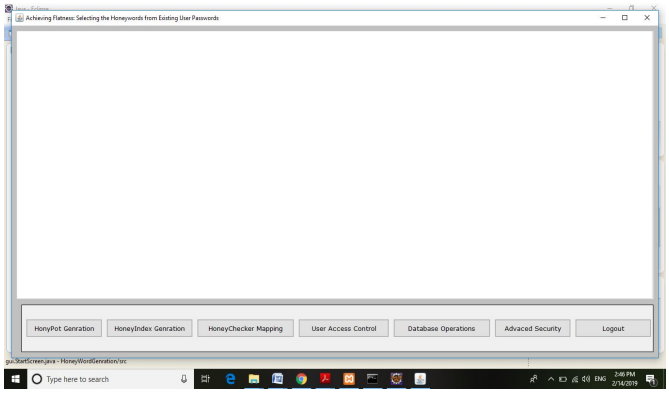
iv) Step 4: Encryption

1. We have a user message (password) space M which contains all possible messages. We outline these messages to a seed space S by using a DTE (distribution transforming encoder).
2. The seed space is the space of all n-bit binary strings for some predetermined n.Each message in m 2 M is mapped to a seed range in S.
3. The size of the seed range of m is directly related to how most likely m is in the message space M. We require some knowledge about the message space M in order for the DTE to map messages to seed ranges, specifically the DTE requires the CDF(cumulative distribution function) of M and some information on the ordering of messages.
4. Additionally, the seed space S must be large so that even the message with smallest probability in the message space S is assigned at least one seed. With this information, we can find the cumulative probability range corresponding the message m and relate it to the same percentile seed range in Space S.

V. RESULT

Main Module:-

All operation in this page which is done by Admin as well as Registered User. It includes Honey pot generation, Honey word generation, Honey index generation, Database operation, Honey Checker Mapping.



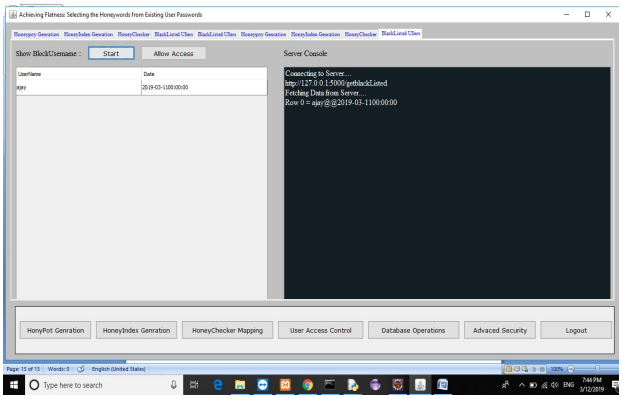


Fig: Main Module Fig :- Honey Checker mapping

Honey Checker Mapping module:-It shows block user list.

VI. CONCLUSION

The utilization of honeywords might be extremely useful in the present environment, and is anything but difficult to actualize. The way that it works for each client record is its huge favorable position over the related method of honeypot records. One could envision different employments of an assistant server to support of watchword based validation. In any case, the design proposed here is perfect and basic, returns to mongrel lease rehearse if assistant server documents are bargained, and is even vigorous against helper server disappointment (on the off chance that one permits logins with honeywords). Honeywords likewise give another advantage. Distributed pass-word records (e.g., one stolen from LinkedIn [10]) give assailants understanding into how clients form their passwords. Assailants can then refine their models of client watchword determination and plan quicker secret word splitting calculations [9In existing frameworks, we store every one of the passwords encoding with help of some encryption component. The methods for decoding the standard calculation are notable and programmers effectively deal with to get the secret key. In this way every break of a secret key server can possibly enhance future assaults. Some honeyword era systems, especially cha□ng ones, darken real client secret word decisions, and in this manner convolute model working for would-be hash wafers. It might even be helpful to sloppy aggressors' information of clients' structure decisions deliberately by drawing some honeywords from somewhat bothered likelihood dispersions.

VII. ACKNOWLEDGMENT

We are extremely grateful to Prof. Amrapali Mhaisgawali and Head of Department Dr. Sanjay Pawar sir for their guidance and support. We would like to thank all those who have contributed to the completion of the project and help us with valuable suggestion for improvement.

REFERENCES

[1] Defense Information Systems Agency (DISA) for the Department of Defense (DoD). Application security and development: Security technical implementation guide (STIG), version 3 release 4, 28 October 2011.
[2] A. Forget, S. Chiasson, P. C. van Oorschot, and R. Biddle. Improving text passwords through persuasion. In SOUPS, pages 1–12, 2008
[3] William Cheswick. Rethinking passwords. Comm. ACM, 56(2):40–44, Feb. 2013.
[4] J. Bonneau. Guessing human-chosen secrets. PhD thesis, University of Cambridge, May 2012.
[5] B. M. Bowen, S. Hershkop, A. D. Keromytis, and S. J. Stolfo. Baiting inside attackers using decoy documents. In SecureComm, pages 51–70, 2009.
[6] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In WWW, pages 551–560, 2009.
[7] C. Herley and P. Van Oorschot. A research agenda acknowledging the persistence of passwords. IEEE Security & Privacy, 10(1):28–36, 2012.
[8] Erguler, Imran. "Achieving flatness: Selecting the honeywords from existing user passwords." IEEE Transactions on Dependable and Secure Computing 13.2 (2016): 284-295.
[9] 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 IEEE Symposium on Security and Privacy (SP), pages 523–537, 2012.
[10] I. Paul. Update: LinkedIn confirms account passwords hacked. PC World, 6 June 2012.