ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

EFFICIENT APPROACH FOR HIGH LEVEL SECURITY USING HONEYWORD

Prof.Vina M.Lomte, Kiran R.Pawar, Rushikesh M.Shivsharan, Kiran V.Swami,Akshay A.Vishwasrao
Department of Computer Engineering, RMDSSOE, Pune, India

ABSTRACT

Now a days, password files has a lot of security problem that has affected millions of users and many companies. password file is generally stored in encrypt format, if a password file is stolen, by using the password cracking techniques and decryption technique it is easy to capture most of the plaintext and encrypt passwords.For troubleshoot this here we create the honeyword password,i.e.a False password,using a perfectly flat honeyword generation method,and try to attract unauthorized user.Hence that time we detect the unauthorized user.Here we also protectthe original data from unauthorized user.

KEYWORDS: Honeywords, Honeypot, Login, OTP, Authentication, Password cracking,Passwords,Decoy documents.

INTRODUCTION

Generally in many companies and software industries store their data in database. 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.So for avoiding it, there are two issues that should be considered to overcome these security problems: First, passwords must be protected and secure by using the appropriate algorithm.And the second point is that a secure system should detect the enrty of unauthorized user in the System.In the proposed system we focus on the honeywords i.e.fake passwords and accounts.The administrator purposely creates user accounts and detects a password disclosure, if any one of the honeypot passwords get used it is easily to detect the admin. According to the study, for each user incorrect login attempts with some passwords lead to Honeypot accounts, i.e. malicious behavior is recognized.

In proposed system , We create the password in plane text,and stored it with the fake password set. we analyze the honeyword approach and give some remarks about the security of the system. When unauthorized user attempts to enter the system and get access the database, the alarm is triggered and get notification to the administrator ,since that time unauthorized user get decoy documents. i.e. Fake database.

LITERATURE SURVEY

Imran Eregular said in the that how the honeyword is created,the password are stored in honeyword form.The password file i.e.false password file is visible to the hacker,and this is the Merits of that systems But in this system some drawback has occure after the use of this system ,like less authentication process,is used as in this system,so all this conclude we create our proposed System, is used present novel approach for securing personal and business data.[1]

Honeyword i.e. false password forces to attacker to brute force the hashes one at a time by a **D.Mirante and C.justin,** instead of attaking them as a group.
High profile website instusion is occurred where as user login credentials and other data were compromisaed.Thus a study was undertaken to research information posted on the web concerning recent,is done.[2]

Purpose And Scope:

- The main aim of project is to validating whether data access is authorized when abnormal information access is detected.
- Confusing the attacker with bogus information.
- This protects against the misuse of the user's real data.

- We propose a completely different approach to securing the cloud using decoy information technology, that we have come to call fog computing.
- We use this technology to launch disinformation attacks against malicious insiders, preventing them from distinguishing the real sensitive customer data from fake worthless data.

PROJECT OBJECTIVE

The proposal is for *"Making Data Inconspicuous In system* based applications for the purpose to avoid the attack of Insider on confidential data.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.

SYSTEM ARCHITECTURE

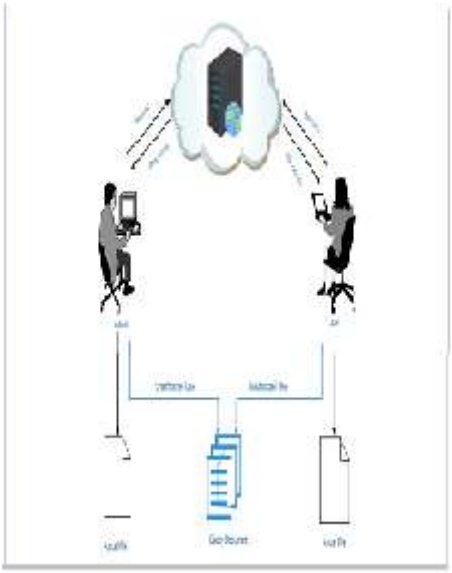


Figure 1: System Architecture

SYSTEM ALGORITHM

Step 1-Start
Step 2-Enter the user name.
Step 3-if(username!= true)go to step 8
Step 4-Enter the password.
Step 5-if(password!= true)go to step 8
Step 2-Enter the answer of question.
Step 3-if(answer!= true)go to step 8
Step 6-Enter the OTP.
Step 7-if(OTP!= true)go to step 8
Step 8-Create the honeyword i.e. false password using the SHA-1 Algorithm.
hash to result so far:
h0 = h0 + a
h1 = h1 + b
h2 = h2 + c
h3 = h3 + d
h4 = h4 + e
Produce the final hash value (big-endian) as a 160 bit number:
hh = (h0 leftshift 128) or (h1 leftshift 96) or (h2 leftshift 64) or (h3 leftshift 32) or h4
Step 9-Enter the system, using false password,but unauthorized person don't know,the password is false.
Step 10-System show only fake database to the unauthorized user.
Step 11-When unauthorized user download it Admin/User get triggering notification.
Step 12-Unauthorized user get fake data.
Step 13-Stop.

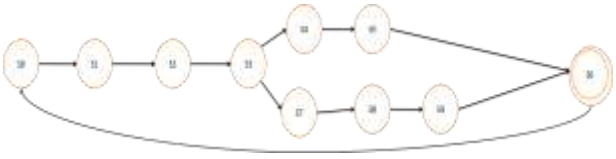
MATHEMATICAL MODEL

Let us consider that we have database 'D' and 'n' number of attribute such as user name, user id etc.
$D = \{A|A \varepsilon$ Information of user$\}$
Here D is the set of all A such that A is information of user which is to be store on server
Consider following function STORE (D, SERVER): Here admin enter the user information into database at server.
Let us consider that the receiver provide us with value "X" for every input it obtain from the every time login account of the particular user .so we can further assume to have a set 's to have value 'n' number of detect value at particular instance. Let us denote the current situation in the following manner
$S = \{X| \forall X \varepsilon D \exists$ ID for attacker$\}$
Here S is the set all X such that for all X there exits Id for user.
Now, for some X value that match with some value inside the database when admin check user account update.
1. GET(D,X,SERVER): Admin get all information about the user account from server.
2. PUT(X,ATK,SERVER): Here admin will upload attacker's information on server.
3. PUTP(X,REPORT,SERVER) : Here admin upload daily report on server.

Initialize variables:
h0 = 0x67452301
h1 = 0xEFCDAB89
h2 = 0x98BADCFE
h3 = 0x10325476
h4 = 0xC3D2E1F0

Initialize hash value for this chunk:
a = h0
b = h1
c = h2
d = h3
e = h4

Main loop
forifrom 0 to 79
if $0 \leq i \leq 19$ then
f = (b and c) or ((not b) and d)
k = 0x5A827999
else if $20 \leq i \leq 39$
f = b xor c xor d
k = 0x6ED9EBA1
else if $40 \leq i \leq 59$
f = (b and c) or (b and d) or (c and d)
k = 0x8F1BBCDC
else if $60 \leq i \leq 79$
f = b xor c xor d
k = 0xCA62C1D6
temp = (a leftrotate 5) + f + e + k + w[i]
e = d
d = c
c = b leftrotate 30
b = a
a = temp
Add this chunk's hash to result so far:
h0 = h0 + a
h1 = h1 + b
h2 = h2 + c
h3 = h3 + d
h4 = h4 + e
Produce the final hash value (big-endian) as a 160 bit number:
hh = (h0 **leftshift** 128) **or** (h1 **leftshift** 96) **or** (h2 **leftshift** 64) **or** (h3 **leftshift** 32) **or** h4
State Diagram:.



S0: User login password
S1: User security question
S2: OTP generation
S3: Enter Hash key and convert to SHA1 value
S4: Set Authenticated
S5: Download original File
S6: If find documents else go to S0.
S7: Set Decoy
S8: Convert to honeywords
S9: Download Decoy File

CONCLUSION AND FUTURE SCOPE

We present a novel 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 illegitimately accesses someone's documents in a system service. Decoy documents stored in the system alongside the user's real data also serve as sensors 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 bogus information in order to dilute 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 from 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.

REFERENCES

1. Imran Erguler," Achieving Flatness: Selecting the Honeywords from Existing User Passwords", DOI 10.1109/TDSC.2015.2406707, IEEE Transactions on Dependable and Secure Computing.
2. D. Mirante and C. Justin, "Understanding Password Database Compromises," Dept. of Computer Science and Engineering Polytechnic Inst. of NYU, Tech. Rep. TR-CSE-2013-02, 2013.
3. A. Vance, "If Your Password is 123456, Just Make It Hackme," The New York Times, vol. 20, 2010.
4. K. Brown, "The Dangers of Weak Hashes," SANS Institute InfoSec Reading Room, Tech. Rep., 2013.
5. M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek, "Password Cracking Using

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

Probabilistic Context-Free Grammars," in Security and Privacy, 30th IEEE Symposium on. IEEE, 2009, pp. 391–405.