INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 2 ISSUE 4 (2016) PAGES 1126 - 1135
RECEIVED : 2/08/2016. PUBLISHED : 14/08/2016
August 14, 2016

AN EFFICIENT WAY TO PROTECT PASSWORD DATABASE USING PAIRED DISTANCE PROTOCOL WITH DB2

[*1]Ms. BRINDHA J., *[2] Ms. ANITA MADONA M.

[*1] M.Phil Research Scholar, Department of Computer Science, Auxilium College(Autonomous), Vellore, TamilNadu, India.

[*2]Assisant Professor, Department of Computer Science, Auxilium College(Autonomous), Vellore, TamilNadu, India.

ABSTRACT

Password based authentication techniques are widely used to protect user information in the years. All passwords database are very important because intruders can break the password or information of users stored in online and use it for their benefit. Sometimes user's information is shared in social networks, commercial and company secrets. In this situation commercial users purchase products in online shopping by giving their shipping address and credit card information. If honeywords are selected appropriately, a cyber-attacker who steals a file of hashed passwords cannot be sure if it is the genuine password or a honeyword for any account. But when implementing honeywords DB manager increase storage requirements 20 times more. This research mainly focuses on how to handle honeywords efficiently and protect the password database. This work proposes a new algorithm with Paired Distance Protocol that is used to maintain the password database. In PDP user need to provide the user name and password then the system will generate that password file which can be spitted and stored in three different database then the PDP measures the distance between the consecutive elements of password with respect to the elements stored in three database and DB2 is the new tool that provides IBM.DB2 is designed to store, analyze and retrieve the data efficiently and it's more secure compared to all other SQL server.

Key Words: *Honeywords, Pairing Distance Protocol (PDP), DB2 Tool (DataBase2).*

I. INTRODUCTION

1.1 Password Database

A large number of research contributions have been made towards increasing the security and usability of password-based authentication. Many of these attempts require account providers to change how they handle authentication by augmenting or outright replacing passwords; e.g., one-time passwords, dual-factor, single-sign on, biometrics, graphical passwords, etc. Recently, researchers have argued that despite the wide-held sentiment from the security and usability communities that passwords needs to be replaced, because the incumbency, familiarity, and low cost of traditional passwords continues to hamper widespread adoption of an alternative, as well as a lack of consensus on what exactly the alternative should provide.

Password managers are designed to relieve password fatigue, reduce cognitive load on users, and reduce log-in time. They can also indirectly facilitate better password quality and a reduction in password reuse. A naive password manager simply stores the passwords, while security-conscious managers lock the stored passwords under a master password.

1.2 Password Database Protection

Password based authentication techniques are widely used to protect the user information in the years. It's very compatible to manage network users to secure their information unless the

attackers crack their credential. However password based authentication techniques are challenging factors in modern days. The no of password cracking techniques used by the attackers to crack the password database and steal the user passwords. In current revelation of password files is very stern security problem that affects lakhs of network users and some social network companies like Yahoo, LinkedIn. These incidents are clear examples why it is important to protect password database. Some of recently developed attack models are Dictionary attack, Brute force attack, Rainbow Table, Phishing, Key Logger attack and Inversion attack. In Dictionary attack the attackers break the password shield server by analytically entering every word in dictionary as a password. Dictionary attacks work because lots of computer clients and businesses persist on using normal terms as passwords. Dictionary attacks are hardly ever successful alongside systems that utilize multiple-word phrases, and failed against systems that employ unsystematic combinations of uppercase and lowercase letters varied up with ciphers.

1.3 Motivation and Scope of This Research Work

The conventianl Password Scheme is an old and most widely used password scheme. In this scheme the user enters or logs into the system through his username and password. The system first authenticates the user from the user database and on the basis of authentication of the user and then grants the access to the system. The advantage of conventional password scheme is that it provides the security of data by allowing only authenticated users to access the system. However, such a scheme is vulnerable to attacks like Shoulder Surfing Key loggers, Brute force attack, Phishing Attacks and Login Spoofing.

II. RELATED WORK

2.1 ACHIEVING FLATNESS: SELECTING THE HONEYWORDS FROM EXISTING USER PASSWORDS

I.Erguler et al [8] The new honeyword techniques called honey indexes. The honeyindexes are randomly created newly assigned user password, a random index number is given to this account and the correct password is kept with the correct index in a list. On the other hand, in another list is stored with an integer set which is consisted of the honeyindexes and the correct index. So, when an adversary analyzes the two lists, she recognizes that each username is paired with numbers as sweet indexes and each of which points to real passwords in the system. The tentative password index hampers an opponent to make an accurate conjecture and cannot be with no trouble sure about which index is the correct one. The disadvantages in these techniques not clearly in Simulation system, Expert systems, Social Networks.DTK is not handled other than brute force attack.Even it’s not clear in password strength rules.

.2 PROTECTING FINANCIAL INSTITUTIONS FROM BRUTE-FORCE ATTACKS

Cormac Herley et al [2]The protecting online banking accounts holder passwords from brute force attack. This method is proposed honeypot method to create large number of passwords pairs. This method makes confuse the attacker which is the original password file. This honeypot method is to copy traits from the pond of genuine accounts, but enter illusory traits for name, address, beneficiaries *etc*. This guarantees that the honeypot hold appropriate account history and transaction details. The Bank allows the real users access to honeypot account when more times of honeypot

credential is obtainable. This Honeypot Technique is used to Hashing algorithm to store password database.

An attacker who enters a honeypot account will have access to the full range of services, with the exception of course that the bank will not actually remit any money to anyone. It will however pretend that it has done so. Only attackers will enter a honeypot, and the goal of the account is to create the illusion of reality. Thus the bank will do everything possible to perpetuate the illusion except part with money.

2.3 THE USE OF DECEPTION TECHNIQUES: HONEYPOTS AND DECOYS

F. Cohen et al [1]The dictionary attacks make use of the plan that users normally prefer weak or easy to keep in mind passwords. The dictionary attack analytically tries every word in a dictionary or passwords from a common password list. Nevertheless users hardly ever use words on their own owing to strict password policies that vigor the user to contain digits or symbols in their password. Users see this is a annoyance and characteristically append a digit (usually a „1") or a symbol (usually a „!") to the end of their chosen word. Superior dictionary attacks effort to imitate these general patterns and apply a list of word smashing rules to every word that is tested. This approach was proposed probabilistic context-free grammars G=(N,∑ ,V,P), Where G is the grammar N- non terminals, ∑ set of terminals, V start variable, P set of production based on aka rules. This algorithm was simply replacing the non terminal values. It is not promising to recourses through each probable arrangement as this would lead to less probable supposition being tested before more credible guesses be in the right places to dissimilar base structures.

III. PROPOSED METHOD

3.1 PAIRED DISTANCE PROTOCOL

Inversion Attack Model

An Inversion attack model, Attackers or Password database Hackers can successfully invert the Hash Password from the compromised password data table. Some techniques are recently developed to crack the password data file Like Probabilistic Context-Free Grammars.

In that approach they have two Phases to crack the Password.

1) Training: This phase generates the context-free grammar from a training set of disclosed real user passwords. The observed base structures and their frequencies are derived from the training set of passwords. Probability information for digits, special characters and capitalization are also obtained. This information is used to generate the probabilistic context- free grammar. The probability of any string derived from the start symbol is then the product of the probabilities of the productions used in its derivation.

2) Generating Guesses: The guess generation phase generates the possible password guesses in decreasing probability order using the context-free grammar obtained from the previous step. Multiple dictionaries can be used with probabilities associated to each.

An Inversion attacks seek to exploit correlations between the target sensitive attributes, known non-sensitive attributes and the model output.

3.2 Password Hashing

Hash algorithms are one way functions. They turn any amount of data into a fixed-length "fingerprint" that cannot be reversed. They also have the property that if the input changes by even a tiny bit, the resulting hash is completely different (see the example above). This is great for protecting

passwords, because want to store passwords in a form that protects them even if the password file itself is compromised, but at the same time, users need to be able to verify that a user's password is correct.

The general workflow for account registration and authentication in a hash-based account system is as follows:

1. The user creates an account.
2. Their password is hashed and stored in the database. At no point is the plain-text (unencrypted) password ever written to the hard drive.
3. When the user attempts to login, the hash of the password they entered is checked against the hash of their real password (retrieved from the database).
4. If the hashes match, the user is granted access. If not, the user is told they entered invalid login credentials.
5. Steps 3 and 4 repeat every time someone tries to login to their account.

In step 4, never tell the user if it was the username or password they got wrong. Always display a generic message like "Invalid username or password." This prevents attackers from enumerating valid usernames without knowing their passwords. It should be noted that the hash functions used to protect passwords are not the same as the hash functions you may have seen in a data structures course. The hash functions used to implement data structures such as hash tables are designed to be fast, not secure. Only cryptographic hash functions may be used to implement password hashing. Hash functions like SHA256, SHA512, MD5, and AES are cryptographic hash functions.

3.3Advantages of Paired Distance Protocol

Storage Cost

A typical password file system requires N plus storage for usernames, where N stands for the number of users in the system and h denotes length of password hash in bytes. Storage cost much lesser than other existing methods.

Inversion Resistance

This proposed approach does not compromise Inversion attack model. Even if compromise any singe peace password can compromise it will not affected to master password.

Usability

Approach with the simple model in terms of practicality and ease of use. By considering the simple model whose password list is constructed with composition of numerous real passwords and randomly generated passwords, one can argue about how the real password source is provided. If the same resource of real passwords is used indifferent sites, similar inherited weaknesses related to Pair distance protocol may be observed.

IV. SYSTEM IMPLEMENTATION

This proposed approach is identified as Paired Distance Protocol or PDP. Using the proposed approach user needs to provide two information i.e., User name and Password when they started used

the applications. Here the proposed approach considers applying some online application for example online banking.

The user provide basic information about them like user name, password, Ac no and mobile no etc., Here the user needs to give strong password

User registration:

User name

Account No

Mail id

Password

Confirm Password.

4.1 PDP Algorithm:

Storing Password:

Step 1 P = Password characters

Step 2 PL = Find the Length of Password.

PL= {p1, p2, p3} – Splitting the Password Characters.

P1 [p1 (0-2 character)]

P2 [p1 (3-6 character)]

P3 [p1 (4< PL - character)].

H(P1)--P1-Hash the Value Using SHA- 1 Functions.

H(P2)--P1-Hash the Value Using MD- 5 Functions.

H(3)--P1-Hash the Value Using AES Functions.

Store Table 1- User information (User information 1) + H(P1).-->D=

Store Table 1- User information (User information 2) + H (P2).--> DB

Store Table 1- User information (User information 3) +(H) P3.--> DB

Login

User Name and password provide by the user.

PL=Password Length

Find password the password length (PL)

Split password characters P={p1,p2, lp3}

If (check User name)==DB.

ThenFind P Value from DB

P1= select corresponding User name and P1 (H) from DB.

INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 2 ISSUE 4 (2016) PAGES 1126 - 1135
RECEIVED : 2/08/2016. PUBLISHED : 14/08/2016
August 14, 2016

P2= select corresponding User id and P2 (H) from DB.

P3= select corresponding User Ac/No and P3 (H) from DB.

If(P== (p1+p2+p3))

Then allow accessing the next page.

Else

Not allow to access the next page.

Else

Ask to enter proper user information.

Here the proposed algorithm implemented in IBM DB2 data server. That DB2 is more secure and working efficiently compare to all other data server.

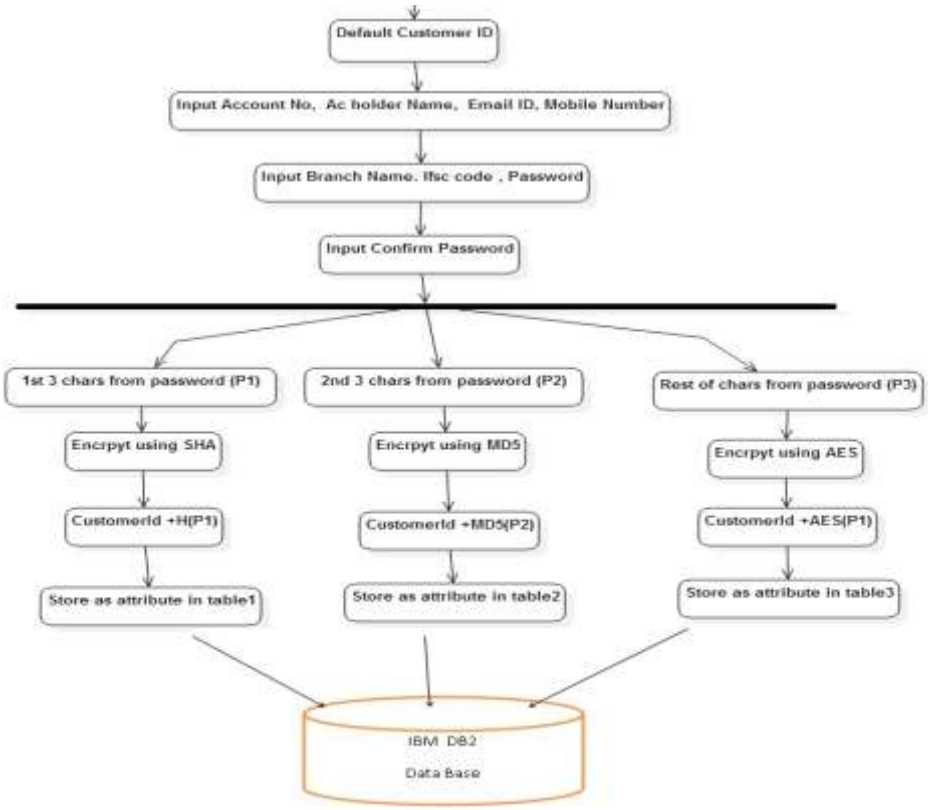

Fig 4.1 Registration page

V. IMPLEMENTATION

5.1 IBM DB2 Tool

INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 2 ISSUE 4 (2016) PAGES 1126 - 1135
RECEIVED : 2/08/2016. PUBLISHED : 14/08/2016
August 14, 2016

DB2 is a database product from IBM. It is a Relational Database Management System (RDBMS). DB2 is designed to store, analyze and retrieve the data efficiently. DB2 product is extended with the support of Object-Oriented features and non-relational structures with XML.

Initially, IBM had developed DB2 product for their specific platform. Since year 1990, it decided to develop a Universal Database (UDB) DB2 Server, which can run on any authoritative operating systems such as Linux, UNIX, and Windows.

5.2 Password Stored in DB2

The user passwords are split and stored within the different format as well as different tables. That may confuse the attackers and also its working under command prompt it gives more secure, reliable and less storage.

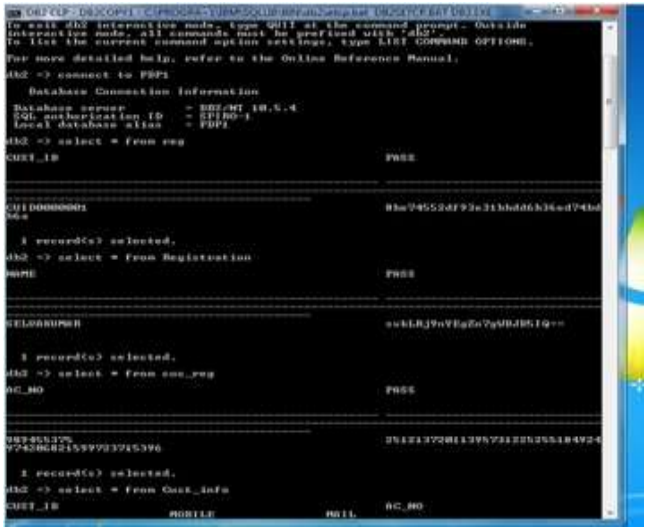

Fig 5.1 Password are Stored in DB2

In this research has conducted some set of passwords with number of different format this is very flexible for online users that more secure compare to all other previous algorithm. This algorithm is very simple there is no complicated input given by the users but it's powerful for password attackers.

There are three well defined security parameters for evaluating the robustness of any password generation algorithms 1) Multiple System Vulnerability 2) Flatness 3) Inversion attack resiliency. The user may use the same password for multiple pages which is used same encryption algorithm. Now if two such different systems are compromised then adversary can get the original password of user by performing an intersection operation. This is because, for a given password, a honeyword generation algorithm produces different honeywords at each run with very high probability. Thus, for a given password, a honeyword generation algorithm produces different honeywords for each system

Now if PDP is adopted by Z different systems which secretly share a password, then for a given passwords all the system generated distance chains will be same. Thus, even if two different accounts of a user are compromised then also MSV will not occur.

INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 2 ISSUE 4 (2016) PAGES 1126 - 1135
RECEIVED : 2/08/2016. PUBLISHED : 14/08/2016
August 14, 2016

If system maintains k sweet words against a user ui then attacker may get confused among k possible options once Wi is compromised. Now sometimes it may happen that, adversary can easily identify the password chosen by the user from the list Wi (e.g. if there exists a correlation between username and password). A honeyword generation algorithm is said to be perfectly-flat if adversary has no advantage while identifying the user's original password from the list of Wi. If the honeyword generation algorithm is perfectly flat then probability of selecting the original password of user from list Wi is 1=k. If the probability of selecting user password from the list Wi is slightly greater than 1=k, then the honeyword generation algorithm is called approximately flat. A good honeyword generation algorithm is required to be perfectly-flat. In PDP creation of passwords very different and PDP perfectly flat and randomness is high.

Using the previously proposed honeyword generation algorithms system maintains k-1 extra passwords along with the original password of user, in the password file F. On the other hand index of the original password of the user is maintained in "honeychecker"server. If assume that for storing a single password, system requires Θ memory space then for storing password information of n users, would require nΘk space. Whereas the required space in "honeychecker" is nΘ. Using PDP, for each user, system stores three tables for information (password and difference field). Thus, the password storing cost for n users in password file F becomes 2nΘ. Though, PDP maintains password of size but the storage cost does not depend on number of users. So storage cost of password is negligible. As memory cost of storing an index value in the "honeychecker" is very similar with storing a digit/alphabet so, required space in "honeychecker" is same as nΘ. Thus, PDP saves a memory overhead nΘ (k-2) .As any standard honeyword system maintains the value of k as 20 for moderate detection rate thus PDP saves a memory overhead 18nΘwhich is a huge benefit.

In PDP approach, if a system senses that the any one of password has been compromised (after generates negative feedback for E(> 1) users accounts) then it will broadcast a security issue to all other systems generating blocking the user , by using the same password or wrong password that account will blocked until user given original password.

VI. EVALUATION RESULT:

From the Existing Honeyword based technique takes input password from the user and create 20 more duplicate password files and DCT, that stored in the DB. The storage cost calculate by bytes. In honeyword based technique are occupies 450 bytes but the proposed PDP occupies only lesser than 300 bytes. It reduced storage cost compare to all other honeyword techniques.

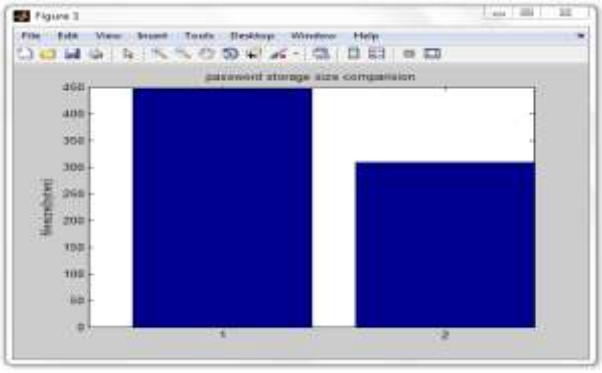

Fig 5.1 Password Storage Size Comparison between Honeyword Technique and PDP

From the Existing Honeyword based technique takes input from the user it's storage time calculate by seconds. In honeyword based techniques are occupies 3.5 seconds but the proposed PDP occupies only lesser than 1 seconds. The results shows in figure 5.2

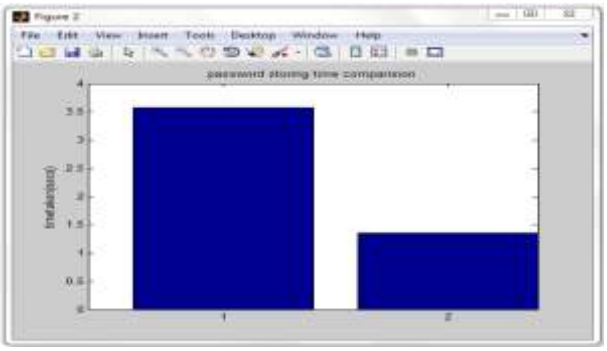

Fig 5.2 Password Storing Time Comparison between Honeyword technique and PDP

CONCLUSION

Authentication on the web is dominated by passwords, mandating that users choose strong passwords to achieve security. As the number of accounts each user is meant to support continues to grow, it is intractable for users to maintain secure account management practices without aid. Honeyword based techniques are powerful as it presents conventional password based schemes. In literature review discuss with various security vulnerabilities that the database suffers from and the need for security to alleviate these vulnerabilities. Here presented Paired Distance Protocol (PDP) based password database security techniques that can be employed to augment and enhance the security of the database against some known attacks and security threats. However, the storage cost is one of the major overhead of honeyword based schemes. This proposed a novel PDP approach which reduces the storage overhead and also it addresses majority of the drawbacks of existing honeyword generation techniques. The only shortfall of PDP is, have create additional data table for password creation but there is no overhead for users.

6.2 Future Enhancement

In future work extended to analyze the possibility of developing a honeyword generation technique without using additional table for store a data. In future work addressing secure backup of managed credentials would increase robustness of the supervisor, helping avoid a time intensive password reset process in the event of device malfunction or loss.

REFERENCES:

1. F. Cohen, *"The Use of Deception Techniques: Honeypots and Decoys,"* Handbook of Information Security, vol. 3, pp. 646–655, 2006.

2. Cormac Herley and Dinei Florencio*," Protecting Financial Institutions from Brute-Force Attacks"*,Proceedings of the IFIP TC 11 2rd International Information Security conference: IFIP 20th World Computer Congress, IFIP SEC'08, September 7-10, 2008.

3. M. Weir, S. Aggarwal, B. De Medeiros, and B. Glodek*," Password cracking using probabilistic context-free grammars. In Security and Privacy"*, 30th IEEE Symposium on, pages 391–405. IEEE"-2009.

4. H. BOJINOV, E. BURSZTEIN, X. BOYEN, AND D. BONEH, *"KAMOUFLAGE: LOSS-RESISTANT PASSWORD MANAGEMENT,"* IN COMPUTER SECURITY–ESORICS 2010. SPRINGER, 2010, PP. 286–302.

5. 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.

6. A. JUELS AND R. L. RIVEST, *"HONEYWORDS: MAKING PASSWORD CRACKING DETECTABLE,"* IN PROCEEDINGS OF THE 2013 ACM SIGSAC CONFERENCE ON COMPUTER & COMMUNICATIONS SECURITY, SER. CCS "13. NEW YORK, NY, USA: ACM, 2013, PP. 145–160.

7. John the ripper password cracker [online document] [cited 2008 oct 07] available http://www.openwall.com.

8. I. Erguler. *"Achieving flatness: Selecting the honeywords from existing user passwords"*.