SKIRMISH AGAINST PASSWORD DENOUNCE USING GRAPH BASED MAZE GENERATION ALGORITHM

[1]R.Shamili , [2]J.Jeyaram
[1]Research Scholar, Department of Computer Science, Vivekanandha Institute of Engineering & Technology for Women, Elayampalayam
[2]Assistant Professors, Department of Computer Science & Engineering, Vivekanandha Institute of Engineering & Technology for Women, Elayampalayam
[1]rstnisha1111@gmail.com

Abstract: Reveal of Password files are not kidding security issues that have influenced innumerable. The client name and watchword acknowledge a vital part in a security framework. So shield that from outsider Authentication. This recommendation starts the examination of online security based secret enter endorsement in passed on framework. Nectar words based watchword endorsement is one of the outstanding security instruments to keep the honest to goodness riddle enter in secure way. The essential puzzle key which typifies the nectar words, an automated attacker who takes a watchword can't guarantee the way and co-ordinate purpose of the relating watchword. So the insistence gives a Graph Theory based Maze Generation Algorithm (MGA).

Keywords: Honey word, Hybrid legacy UI, Tweaking digits, Graph Theory based Maze Algorithm.

1. INTRODUCTION

For the most part the product organizations put away their subtle elements or data in database with the assistance of User name and Password and they are put away in scramble shape in the database. Utilizing the secret word splitting strategy, once the watchword record is stolen it ought to be anything but difficult to recover the plaintext secret key. To conquer these security issues there are two approaches to characterize: to start with, utilizing some Salting instrument to ensure the authentic passwords in secure way. What's more, second, recognize the passage of unapproved client in the specific record.In the Existing System focused to create realistic honey words to detect password cracking. However instead of generating the honey words and storing them in the password files, they use the existing passwords to simulate honey words. Generating honey indexes for each and every account of the system using Honey word generation Algorithm Gen(). Therefore the authors introduce a definition as the flatness of algorithm such that it measures the chance of getting the correct password from the honey word.

In this review, we consolidate the few strategies and give some notice about the security of the framework. We call attention to that the key things for this technique is plot the nectar words in graphical frame with the assistance of Maze Generation Algorithm (MGA). In this manner we skirmish against un-authentication administrations utilizing MGA.

2. RELATED WORK

Imran Ergular [1] et al., suggest to use the existing user password to simulate the honey word and storing them in a password file. The password guessing attack perform the attacker cannot exactly determine which password belong to which users.

Jules & Rivest [2] et al., suggest the method of giving multiple passwords for each account whether one password is correct and other used as honey word. If the hashed password file is stolen by the cyber attacker and easily convert into hashed functions for getting correct password in this file honey word and it also stored with the password. Example, if the password is lucky den honey words like lucky953, lucky413 etc… Here 953 and 413 are honey words.

Data and password authentication is a major aim of all applications. Several companies were affected by security violations like adobe, yahoo, Rock you, eHarmony [3]. 50 million hashed user passwords were stolen from evernote in 2013. The leaked passwords create much more problems for the respective companies. Current system was protecting the real

passwords using fake passwords methods [4]. Secure the original passwords files using Secure Hash Algorithm [SHA1] without any salting mechanism [5]. This will increase the password stealing threats.

3. PROBLEM STATEMENT

Propose an Alternative approach, utilizing MGA increment the aggregate exertion in getting passwords from the chart and distinguishing the passwords revelation can be given at same time.

4. PROPOSED SYSTEM

In the proposed framework, we utilize Maze Generation strategy for expanding the security components of the nectar word era. We consolidate both legacy UI methods. Legacy UI (User Interface) in which secret key change the UI is unaltered, the client picks the genuine watchword. The created nectar words are put away in the chart position. It makes the aggregate hash reversal prepare harder for an adversary in getting the secret key in plaintext shape from a spilled watchword hash record. Henceforth by building up the strategy increment the aggregate exertion in recovering plaintext passwords from the hashed list and distinguishing the password database breach.

a. Maze Generation Algorithm [MGA]:

In this method we using Graph based theory with the help of Maze Generation Algorithm. A maze can be generated by starting with a predetermined arrangement of cells (most commonly a rectangular grid but other arrangements are possible) with wall sites between them. This predetermined arrangement can be considered as a connected graph with the edges representing possible wall sites and the nodes representing cells. The purpose of the maze generation algorithm can then be considered to be making a subgraph in which it is challenging to find a route between two particular nodes.

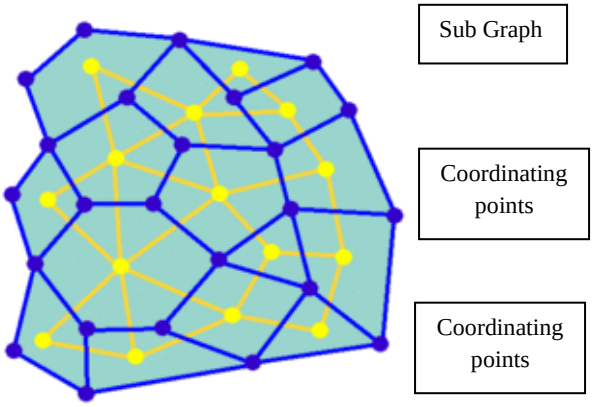

Figure 1: Graph model for Plot the values.

4.2 Chaffing by Tweaking Digits:

Tweaking the last L position that contains Digits. For example, by using the last technique for the password 38 orange here t= 2 and password = Hungry. Therefore the honey words 13 orange and 42 orange may be generated. The data digit will be replaced with the randomly selected digits.

Here,

U_i = User name.

P_i = Password of the U_i.

W_i = list of potential password

K = Number of Elements in W_i.

t = Number of elements in P_i

Gen (k) = Procedure used to generate W_i of length k of honey words.

For Instance,

Generator alg. = Gen (k, t)

User Password=Agnes 32

Fix: t=4; k=9 (k denotes the potential combination of the user password). Therefore Gen (k) is

Table 1: Combination of Passwords

Agnes 15	Agnes21	Agnes 23
Agnes 22	Agnes 11	Agnes 32
Agnes 24	Agnes 14	Agnes 28

4.3 Chaffing with a Password Model

In this approach, the generator takes the password from the user and depends on the probabilistic model of real password it produce the Honey word [2]. As an example for this Method named as the Modeling syntax.

For Instance,

Gold9 kings is fissured as

4 letters + 1 digit + 5 letters

L_4 + D_1 + L_5 are the substitutes with same composition like bond5queen.

Real password = gold 9 kings

Here is a list of honey words Generated by One simple model.

Bond5queen pink2color

Rose5queen boat5water

Rose9queen blue9queen

Agni7water pink3queen

Blue8rocks very9rocks

- **Modeling syntax:** Bojinov et al., [5] propose an interest approach chaffing with a password model in which the honey word are generated using the same syntax as password. In this strategy, the password is replaced into sequence of "tokens" each character representing by a Distinct syntactic elements.

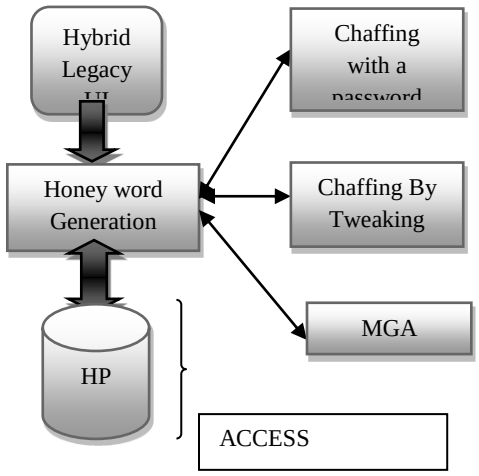

Figure 2: MGA FRAME WORK

The proposed model shown in the FIG.2 includes the scheme which is named as MGA (Maze Generation Algorithm). The combination of Hybrid Legacy UI (User Interface) for generating the honey words (use salting mechanism) to prevail against cyber attacker in the legitimate system.

The proposed system performs Chaffing with-a-secret word show and Chaffing by-tweaking digits. By utilizing these two strategies, gives nectar words and the relating nectar words are put away in the diagram focuses, the nectar words are splitted and put away in the inside and outer (sub chart focuses) planning focuses in the particular chart position.

5. MODULE INCISIONS

5.1 Initialization

Step 1: Take user accounts T (honey pots) are created with their passwords

Step 2: Store the corresponding Index value between (1, N) not used previous value of the Index.

Step 3: Then the random numbers are selected from the index list as k-1.

Step 4: Create the index number for the corresponding username.

Example 1:

The honeypot username/password pair is generated like <pinky, pinky1993> by the system. Then an index number is selected randomly, for instance 2008, and assigned as the correct index of this account.

Index No	Hash of Password
.	.
.	.
2008	H(pinky1993)
.	.
.	.

Then, k-1 numbers are randomly chosen and combined with correct index 2008 in a random manner to produce the index group, For an example, if k=4, such a group (56,45789, **2008,** 34576, 8204) may be generated.

Username	Index Set
.	.
.	.
Pinky1993	**2008**
.	.
.	.

5.2 Registration

After the initialization process, system is ready for user registration. In this phase, Legacy UI is preferred. In which the username and password are required from the user as (u_i,p_i) is register in to the system.

Step 5: Receive the username and password generate the honey index and Index number for the legitimate username and password from the authorized user.

5.2 Honey Checker & MGA

The optimal strategy for an adversary when tough nuts are experienced. We believe that "tough nuts" method is a double-edged-sword.

Take password from user & relying on a probabilistic model of real password (using chaffing-with-a-password-model) &(chaffing-with-a-digits) model.

Step 6: Then store the correct index in the graphical path.

Step 7: Store the index value and password in the inner and outer coordinating points in the graph.

Step 8: Then produce the path to increase the effort of getting the original password and the corresponding document.

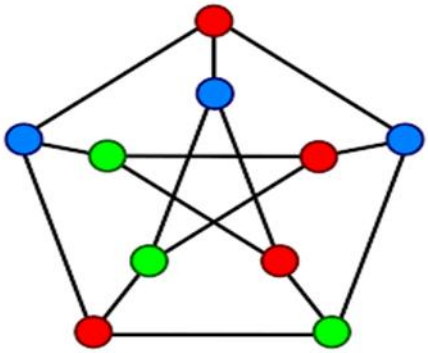

Figure 3: Plot the values in the Graph Model

Store the value of the Index Number

Store the user name u_i

Store the password p_i

5.3 Password Cracking

After this entire steps the attacker not getting that hecatch. He thinks that he got the original password & original password file/doc. But when authentication fail

that time administrator get the attacker attempt and get all the information about that attacker like the physical address, IP address etc., And then take appropriate action.

6. CONCLUSION & FUTURE ENHANCEMENT

In the end, passwords ought to be supplemented with more grounded and more advantageous confirmation techniques. We presented a basic and intense new line of resistance in the security of hashed passwords. In this strategy will diminish the estimation of the stolen watchword hash documents and furthermore makes the secret key breaking noticeable. This paper is to give higher security to creating the nectar words and put away in a safe chart demonstrate. In future we give nectar pictures to build the security level into abnormal state.

REFERENCES

[1] Imran Erguler, "Achieving Flatness: Selecting the Honey words from Existing User Passwords," in proceedings of IEEE Transaction on Dependable and Secure Computing (Volume: 13, Issue: 2, March-Agnes 2016)

[2] A. Juels and R. L. Rivest, "Honey words: Making Password cracking Detectable," in Proceedings of the 2013 ACM SIGSAC Conference Computer & Communications Security, ser. CCS'13. New York, NY, USA: ACM, 2013, pp. 145–160. [Online] availablehttp://doi.acm.org/10.1145/2508859.2516671

[3] 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.

[4] A. Vance, "If Your Password is 123456, Just Make It Hackme,"The New York Times, vol. 20, 2010.

[5] H. Bojinov, E. Bursztein, X. Boyen, and D. Boneh, "Kamouflage: Loss resistant Password Management," in Computer Security ESORICS 2010. Springer, 2010, pp. 286–302.

[6] J. Brainard, A. Juels, B. Kaliski, and M. Szydlo. A New two-server approach for authentication with short Secrets. In USENIX Security, pages 201–214, 2003.

[7] J. Camenisch, A. Lysyanskaya, and G. Neven. Practical yet universally composable two-server Password-authenticated secret sharing. In ACM CCS, Pages 525–536, 2012.

[8] William Cheswick. Rethinking passwords. Comm. ACM, 56(2):40–44, Feb. 2013.

[9] F. Cohen. The use of deception techniques: Honeypots and decoys. In H. Bidgoli, editor, Handbook of Information Security, volume 3, pages 646–655. Wiley and Sons, 2006.

[10] EMC Corp. RSA Distributed Credential Protection.http://www.emc.com/security/rsa-distributedcredential-protection.htm, 2013.

[11] A. Czeskis, M. Dietz, T. Kohno, D. Wallach, and D. Balfanz. Strengthening user authentication through opportunistic cryptographic identity assertions. InACM CCS, pages 404–414, 2012.

[12] 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.

[13] A. Forget, S. Chiasson, P. C. van Oorschot, and R. Biddle. Improving text passwords through persuasion. In SOUPS, pages 1–12, 2008.

[14] C. Gaylord. LinkedIn, Last.fm, now Yahoo? Don'tignore news of a password breach. Christian ScienceMonitor, 13 July 2012.

[15] D. Gross. 50 million compromised in Ever note hack. CNN, 4 March 2013.

[16] C. Herley and P. Van Oorschot. A research agenda acknowledging the persistence of passwords. IEEE Security & Privacy, 10(1):28–36, 2012.

[17] S. Houshmand and S. Aggarwal. Building better Passwords using probabilistic techniques. In ACSAC, pages 109–118, 2012.

[18] 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.

[19] O. Kharif. Innovator: Ramesh Kesanupalli's biometric passwords stored on devices. Bloomberg BusinessWeek, 28 March 2013.

[20] Microsoft TechNet Library. Password must meet complexity requirements. Referenced March 2012 at http://bit.ly/YAsGiZ.

[21] R. Morris and K. Thompson. Password security: a case history. Commun. ACM, 22(11):594–597, November 1979.

[22] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In IEEE Symposium on Security andPrivacy (SP), pages 173–187, 2009.

[23] U.S. House of Representatives. H.R. 624: The Cyber Intelligence Sharing and Protection Act of 2013. 113thCong., 2013.

[24] B.-A. Parnell. LinkedIn admits site hack, adds pinch of salt to passwords. The Register, 7 June 2012.

[25] I. Paul. Update: LinkedIn confirms account passwords hacked. PC World, 6 June 2012.

[26] D. Perito, C. Castelluccia, M. A. Kaafar, and P. Manils. How unique and traceable are usernames? In Privacy Enhancing Technologies, pages 1–17, 2011.