Anonymous System: Automatic Secret Attack in Utility and Detectability Application

[1]M.Bharathi, [2]P.Rayavel

[1]Assistant Professor, [2]Assistant Professor
Computer Science and Engineering
Sri Sai Ram Institute of Technology, Chennai, India

Abstract : **In some case, system may be vulnerable because of DDOS attacks it affecting the whole system. The passwords are easily hacked by the hacker using online guessing attacks. So there is no big security implementation was introduced in the existing system. We propose a simple security system for improving the authentication .Our implementation is to avoid DDOS attack by inducing hackers in the form of honey words. The user has to register with the server and it generates Random set of Passwords to the user called Honey words. User's Original password is hashed and stored along with the Honey words. Attacker will fetch any one of the password so that intermediate server will filter the wrong password based queries so that DDOS can be avoided. Honey words based on the user information provided and the original password is converted into another format and stored along with the Honey words. We deploy Intermediate server, Shopping server and Cloud server for purchasing and maintaining user account details respectively. Attacker who knows the E mail account of original user can easily change the password of the cloud server. Attackers are induced to do attack in this Project, so they can found out easily. The attacker logins into the purchase portal, where tracking is done and he is allowed to do purchase. Server identifies the attacker and blocks him doing further transaction from his original account and also sends the attacker details to the original user. The main purpose behind the introduction of the honey words is to overcome password-crack detection problem, we believe that security policies should not be loosened to mitigate the DoS attacks.**

KEYWORDS – Authentication, honeywords, decoy, login, passwords, security

I. INTRODUCTION

Generally in many companies and software industries store their data in databases like Mysql or may be other. So, the user name and password gets stored in an encrypted form in the database which is the entry point of a system. The password cracking technique is used to capture most of the plain text passwords that are stolen from the password file. Two issues are used to overcome these security problems: First passwords must be protected and secured by using the appropriate algorithm. And the **second** point is that an unauthorized user's entry has to be detected. In the proposed system we focus on the honey words i.e. fake passwords and accounts. The administrator purposely creates user accounts and detects a password disclosure, if any one of the honey pot 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 Honey pot accounts, i.e. malicious behaviour is recognized. In proposed system, we create the password in plain text, and stored it with the fake password set. The security of the system is analyzed using the honey word approach. The mail gets triggered and notification is sent to the administrator, when unauthorized user attempts to enter the system and access the database. By the time an unauthorized user gets decoy documents. i.e. fake database. In this article, we shall present a secure and an efficient password based authentication scheme that follows honey words generation. In contrast, The proposed scheme can resist DOS attack, DDOS attack, Brute force attack and further provides security analysis. By complexity analysis, the storage and computation cost is shown to be very efficient. In section 2, we discuss the honey word generation method and in section 3, we generate the honey checker model. In section 4, we describe the attacks induced in the existing model. In section 5, we described the techniques used in the proposed model and in section 6 feasibility study has been described and in section 7 comparison of honey words has been made. In section 8 we demonstrate the related works and future enhancement of the proposed system and finally in section 9, we conclude this paper.

II. HONEYWORD GENERATION METHOD

In this section, we summarize the honeyword password model generation pro- posed by the Juels and Rivest. Then, we discuss some points that can cause some security problems.

Basically, the honeywords is the insertion of false passwords associated with each user's account. When an adversary gets the password list, they recovers many password candidates for each account and they cannot be sure about which word is genuine. Hence, cracked password files can be detected by system administrator if a login attempt is done with a honeyword by the adversary.The honeyword mechanism works simply as follows: For each user ui, the

The honeyword mechanism works simply as follows: For each user ui, the honeyword generation algorithm Gen(k) generates the honeyword Wi.This procedure takes input k as the number of sweetwords and outputs both the password list Wi = (wi,1,wi,2,...,wi,k) and ci is generated as the correct password (sugarword) stored in a server called honeychecker. The username and hashes of the sweetwords as < ui,(vi,1,vi,2,...,vi,k) > tuple is kept in database of the main server. By storing the password

hashes in one server and ci in the honeychecker – makes the system as a whole harder to compromise i.e. provides a basic form of distributed security [9]. Notice that in a traditional password technique < ui,H(pi) > pair is stored for each account, while < ui,Vi > tuple is kept in database, where Vi = (vi,1,vi,2,...,vi,k). The login procedure of the scheme is summarized below: – User ui enters a password g to login to the system. – Server firstly checks whether or not H(g) is in list Vi. If not, then login is denied. Otherwise system checks to verify if it is a honeyword or the correct password. – Let v(i,j) = H(g). Then the value is delivered to honeychecker in a hashed secure manner. Honeychecker checks whether the equality holds and returns a TRUE value, otherwise it responses hacker's information to the original user. Before discussing honeyword generation methods, we want to talk about honeyword generator algorithm Gen(). Note that strength and effectiveness of the method indeed is directly related to how the Gen() is constructed. Thus the chance of detecting sweetword can be measured using the Gen(). In other words, if a honeyword generation method is €-flat, then they have at least a 1-€ chance of picking a honeyword. For example the attacker has a chance of at most 25% of picking the correct password pi from Wi for € = 1/4. In short, if the algorithm is not flat enough, real password stands out from the remaining fake passwords and an adversary can easily reveal the original one[3].
The expressions and definitions depicted to simplify the description of honeyword scheme.

Expressions:

H()	Cryptographic hash function used to compute hash of the passwords
ui	Username for the ith user.
pi	Password of ith user
Wi	List of potential passwords for
k	Number of elements in Wi
ci	Index of correct password in list Wi
Gen(k)	Procedure used to generate Wi of Length k of sweetword
Sweetword:	Each element of Wi
Sugarword:	Correct password in Wi honeyword: Fake passwords in Wi

III.HONEYCHECKER

We assume that the system may incorporate auxiliary secure server called the“honeychecker” to assist with the use of honeywords. Since the system has the file F, the password and the other hashed parameters can be stolen. Thus, there is no place for storing secret information. Thus the secret information can be stored using honeychecker. The system communicates with the honeychecker whenever the user makes an attempt to login or changes the password.This communication is made over an encrypted form. The honeychecker should have extensive instrumentation to detect anomalies of various sorts. This honeychecker also detects the hacker's information. The hacker's details are sent to the original user and he can be easily tracked. Depending on, the honeychecker replies to the computer system when a login is attempted. When the system detects something wrong with the login attempt it sends the signal to the user. On the other hand the login attempts proceed. We could also call the honeychecker a“login monitor”. The proposed honeychecker also maintains a single database value c(i) for each user ui.

IV. ATTACKS IDENTIFIED IN EXISTING MODEL

Denial-of-service (DoS)

In computing, the multiple users target the resources of a system using one or more servers is denial-of-service attack (DoS attack). A DoS attack is analogous to a group of people crowding the entry door and not allowing the authorized parties enter into the shop and also disrupts the normal operations.
Criminal perpetrators of DoS attacks often target sites or services hosted on high-profile web servers such as banks or credit cardpayment gateways. Revenge, blackmail and activism can motivate these attacks. A denial-of-service (DoS) attack is discussed for the following scenario: Adversary knows the used Gen() procedure and can produce all possible honey- words for a given a password. For example if chaffing-by-tweaking-digits is em- ployed in the system and with a small t adversary may generate whole possible honeywords from a known password. Consider the case, let password of a user be test42, then for t = 2 she can generate 100 possible honeywords and k of these honeywords are stored in the system password list[8] . Let Pr(g = wi|pi) denote the probability of correctly guessing a valid honeyword of Wi, where correct pass- word pi is available to the adversary. Hence if this probability is a non negligible value, the adversary may attempt to login with the guessed honeyword to trigger an alarm condition. In fact, this may be serious, if a strong policy is set by the administrator e.g. a global password reset in response to a single honeyword hit. In the above example for k = 20 and t = 2, Pr(g = wi|pi) = (k−1)/99 = 0.19. In order to mitigate this risk,

the authors suggest to choose a relatively small set of honeywords randomly from a larger class of possible sweetwords. For the previous example, success probability of the attacker is about 19% for k = 20, while this chance is decreased to 2% by only changing t = 3. Nevertheless, we want to consider the case that an adversary knows m username– password pairs. Perhaps, she previously created these accounts in the system to make a DoS attack. Also suppose that there exists a limit for unsuccessful login attempts as n and success probability of guessing a valid honeyword for a known password is Pr(g = wi|pi) = 1 α. Then it is more likely that the adversary can succeed in DoS attack, if she makes about α trials. Notice that the adversary can make at most m·n attempts. For the above example Pr(g = wi|pi) = 0.02, so it is highly possible to raise an alarm condition if an adversary make about 50 trials. That is to say if the false attempt limit n is (say) five, 10 known account/passwords pairs will be enough to realize the mentioned.

Algorithm

```
DosAttack(pi,T(pi),n)
for j ← 1 to |T(pi)| do
if (mod(j,n)_ = 0
then Login(pi)−comment: To reset unsuccessful login attempts
else  Login(Guessj)  −comment: Make jth guess; Guessj ∈ T(pi)
```

Likelihood Attack

If the adversary has stolen F and wishes to maximize his chance of picking pi from Wi, he can proceed with a “likelihood attack” as follows. We assume here that we are dealing with an approach based on generating honeywords using a probabilistic model. Let G(x) denote the probability that the hon- eyword generator generates the honeyword x. Similarly, let U(x) denote the probability that the user picks x to be her password. (This may not be mathemat- ically well-defined; it can be interpreted as a Bayesian prior for the adversary on such probabilities, and may or may not be user-specific.) Let Wi = {wi,1,...,wi,k}. The likelihood that c(i) = j, given Wi, is equal to U(wi,j)Y j06=j G(wi,j0) = C R(wij) where C =Y j0 G(wi,j0) and where R(x) = U(x)/G(x) is the relative likelihood. Note that it is desirable that for all eligible x, G(x) > 0 (that is, the honeyword generator is capable of generating may be recognizable as one the honeyword generator could not possibly have produced. The adversary wants to maximize his likelihood of picking the password, so he will pick the one maximiz- ing R(wij). This is the password that is maximally more likely to be picked by the user than to be generated by the honeyword generator[8] .

Brute-force Attack

An attempt to decrypt an encrypted password by an attacker is so called brute force attack. The consecutive guesses of the desired data is done using the automation software. These types of attacks are used by the attackers to encrypt data or to tests the network security and is also known as brute force cracking attack.In previous attack, we point out that if a strict policy is executed in a honeyword detection, system may be vulnerable to DoS attacks affecting the whole system. On the other hand, a soft policy weakens the influence of honeywords. The following attack is used for capturing the large number of accounts. We suppose an adversary has obtained a password file F and cracked numer- ous user passwords. Then, she tries to login with any accounts in the list instead of compromising a specific account. Furthermore, we assume that the adversary has no advantage in guessing correct password by analyzing corresponding hon- eywords, i.e. Pr(g = pi) = 1/k. Last, if one of the user’s honeywords is entered, system takes the appropriate action according to one of the example policies as follows:

– Login proceeds as usual

– User’s account is shutdown until the user establishes a new password.

The common point of the above policies is that even a honeyword entrance is detected, system gives a local or no response. As a result of this, an adversary can carry out a brute-force search until a successful login is obtained. For example, even a user’s account is locked due to a honeyword attempt, she continues to search with another user’s account, i.e. single guess for each user. She likely makes a correct guess after k trials, since Pr(g = pi) = 1/k. As an illustrative example for k = 20, it is highly possible that the adversary finds a correct password after 20 attempts. It is equivalent to say that if there exists N users in the system, the adversary may recover genuine passwords of N/k users by using brute-force search.

Attacking the Honeychecker

The hackers tries to attack the honeychecker and also communicates with the system. The updated commands has to be sent to the honeychecker and it has to be authenticated so that it gets correctly updated to the database. The requests (“Check” commands) sent to the honey- checker also need to be authenticated, so that the adversary can’t query the honeychecker and cause an alarm to be raised. The replies from the honeychecker should be authen- ticated, so that the computer system doesn’t improperly allow the adversary to login. By disabling communications between the computer system and the honeychecker, the adversary can cause a fail over[8] . The computer system then either has to disallow login or take the risk of temporarily allowing login based on a honeyword and buffering mes- sages for later processing by the honeychecker. While our intention is that the honeychecker should be hardened and of minimalist design, the deployment of the computer system and the honeychecker as two distinct systems itself brings the usual benefits of separation of duties in enhancing security. The two systems may be placed in different administrative domains, run different operating systems, and so forth. The proposed system is employed to avoid the following above attacks.

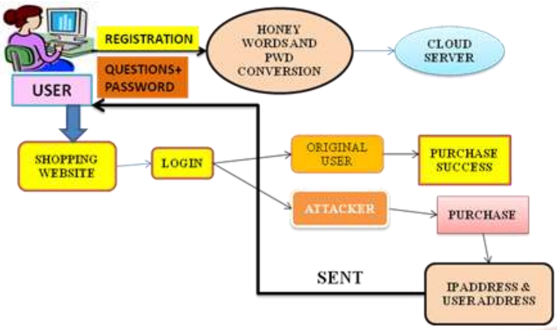
V. TECHNIQUES UESD

A modular design reduces complexity, facilities change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different part of system. The interfaces are simplified using effective modularity of software. The separately named modules are finally integrated to produce the final outcome and is to be indicated in the software architecture.

The five important criteria that enable us to evaluate a design method with respect to its ability to define an effective modular design are: Modular decomposability, Modular Com ps ability, Modular Understandability, Modular continuity, Modular Protection.
The modules are planned in aid to complete the project with respect to the proposed system.

- USER REGISTRATION
- SERVER VERIFICATION
- RANDOM PICK HONEY WORDS GENERATION
- INTERMEDIATE SERVER DEPLOYMENT & SHOPPING SERVER DEPLOYMENT
- PASSWORD HACKING PROCESS
- IDENTIFICATION OF ATTACKERS & AVOIDANCE OF DDOS ATTACK

ARCHITECTURAL DIAGRAM



DESCRIPTION OF PROPOSED MODEL

USER REGISTRATION

In this module the User application is created by allowing him to access the data from the Server. The User can access the network only when he creates an account.Once the User creates an account, they are allowed to login into their account to access the application.The server responds based on the requests of user's.

All the User details will be stored in the Database of the Server. In this module bank user details are registered with the fields like username, password, and personal details with some set of questions and answers. These details are saved into the server. After proper registration only the user cal allowed to login into the server.

SERVER VERIFICATION

In this Server module server will deployed to access the database and web based application. Server will verify the users and generates honey word for save the users password. In case illegal actions happened means server will generates alert and intimate it to user.The user's information in their database is monitored and verified by the server and also it stores and updates the entire information of user. The Server also establishes the connection to communicate with the Users. The Server will authenticate each user before they access the Application. So that the Server will prevent the Unauthorized User from accessing the Application.

RANDOM PICK HONEYWORDS GENERATION

“Randompick” honeywordgeneration We now present a modified-UI procedure that is perfectly flat. At a high level, a good way of generating a password and honeywords is by producing the list Wi of k distinct honeywords in some manner (which may involve interaction with the user) and then pick an element of this list uniformly at random to be the new password; the other elements become honeywords. As an example of user in- volvement, we might just ask the user for k potential pass- words. The value c(i) is set equal to the index of (randomly chosen) password pi in this list. The random pick method is perfectly flat, no matter how the list Wi of sweetwords was generated, since the given procedure is equivalent to choosing c(i) uniformly at random from{1,2,...,k} independentof the actual sweetwords; there is thus no information in Wi that can aid in determining c(i). It is probably a bad idea, however, to ask the user for k sweetwords. Not only is this burdensome on the user, but the user may remember and mistakenly enter a sweetword supplied by her and used by the system as a honeyword. Instead, the random pick method is probably better applied to a set of k sweetwords output by an algorithmic pass- word generator.
The millions of users has been affected due to the security problems generated in Password files.The encryped Password file can be hacked using the cracking techniques and decryption technique and it is easy to capture most of the plaintext and encrypt passwords.So in this module we deployed honey word creations. That is the user’s password and registered questions are combined and then it will generate a key as unknown Name.

TABLE
Password File F1 for the Proposed Model[2]

Username	Honeyindex Set
agent-lisa	(93; 16; 626; . . . ; 94; 931)
alexius	(15; 476; 51; 443; . . . ; 88; 429)
baba	(133; 62107; . . . ; 91; 233)
....	

....
....
zack_tayland (1; 009; 23; 471; . . . ; 47; 623)
zoom (63; 51234; . . . ; 72; 382)

INTERMEDIATE SERVER & SHOPPING SERVER DEPLOYMENT

The requests to a resource manager program are handled by an intermediate server on behalf of a user program. The user program can be referred to as a client of the intermediate server.Here we will generate the Intermediate server to make communication between user and Server. All requests comes from the users are first sent to the intermediate server to verifies the password and user details.Shopping server is to collect the details from customer and sent to the details to the intermediate server for verification.

PASSWORD HACKING PROCESS

The process of getting passwords from data that has been stored in or transmitted by a computer system is so called as hacking. The two approaches are of guessing the password and changing the password if its been forgotten. Password Hacking is blocked in this Module. Because we modifies the users original passwords into unknown Name and saved into server.

IDENTIFICATION OF ATTACKERS & REMOVE DDOS ATTACKS

A distributed denial of service (DDos) attack is an attempt when the multiple systems target the single system.The important resources and information from banks and news websites are targeted by DDOS attack.If there is anybody trying with wrong password or any illegal action means server will block that action and intimate to the Specified Users. If the same request comes from same user or from different user's means server will blocks that actions also. This is done in DDOS attack.

VI.FEASIBILITY STUDY OF THIS MODEL

The project's feasibility study is estimated and is planned accordingly.The system analysis is done to analyse the feasibility study of the proposed system. This is to ensure that the proposed system is not a burden to the company. The major requirements of the system is essential for the feasibility analysis.

Three keypoints involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- OPERATIONAL FEASIBILITY
- **6.1 ECONOMICAL FEASIBILITY**
- The economic feasibilty of the project is carried out for an organization.The research and development of the company fund can be limited.The expenditures must be justified. Thus the limited fund due to the freely available technologies has been employed in the developed system.Only the customized products had to be purchased

TECHNICAL FEASIBILITY

The technical requirements of the system can be checked using the feasibility study techniques.The available technical resources on any system developed must not have a high demand provided. This will lead to high demands on the available technical resources and also on the client. The modest requirement has to be provided in the developed system, as implementing system requires only the minimal or null changes.

OPERATIONAL FEASIBILITY

- The aspect of study is to check the level of acceptance of the system by the user. The user has to be trained well to use the system effectively.The user must feel it as an important one.The different methods are employed to the user to make him familiar about the system.He is also able to make some constructive criticism with respect to the raise in level of confidence which is welcomed, as he is the final user of the system.

VII. COMPARISON OF HONEYWORDS BETWEEN THEIR METHODS

METHODS	DOS RESISTENCE	FLATNESS	STORAGE COST
Tweaking	Weak	Weak	Hn
Password model	Strong	Strong	Khn
Our model	Strong	Strong	4kN þ hN þ 4N

Table - Comparison of honeyword-generation methods[2],[8] . All methods can achieve excellent (1/k)-flatness under some conditions. By "weak" DoS (denial of service) resistance, the user submits a honeyword given knowledge of the password; by "strong" DoS resistance we mean that such attack is improbable, honeywords are distributed like user passwords in the view of the adversary. These means "tough nuts" are not useful on their own. The storage costs assume generation of k−1 honeywords. Nevertheless, how the source of the real passwords is attained for this model should be answered before judging its applicability.

Finally, we have presented a new approach to
make the generation algorithm as close as to human nature by generating honeywords with randomly picking passwords that belong to other users in the system.The proposed model is compared with other methods with respect to DoS resistance, flatness, storage cost and usability properties. The comparisons have indicated that our scheme has advantages over the chaffing-with-a-password model in terms of storage, flatness and usability.

VIII.RELATED WORKS

Cloud computing

Is a internet based computing and fast-growing technology that provides sharing of resources and it has been established itself in the next generation.Cloud services have become a powerful architecture to perform complex large-scale computing tasks. The process of storing the large datas has been let to the use of cloud computing. The cloud is used for deploying the large number of scientific applications for extensive experiments. In addition, cloud service providers helps the users to integrate the parallel data processing for accessing the cloud resources.

The number of computational resources (e.g., networks, server, storage, application, and services) can be accessed using the cloud computing.The service provider interaction helps in releasing these resources.Cloud computing has a number of favorable aspects to address the rapid growth of economies and technological barriers. The computations, infrastructures, and storage cloud services are combined together to provide a highly attractive environment. Cloud service models typically consist of PaaS, SaaS, and IaaS.

Because of the limited processing capability, storage capacity, and battery lifetime of each device has taken cloud computing to high level. This condition has led to the emergence of a mobile cloud computing paradigm. The users can outsource tasks to external service providers by using mobile cloud. Mobile cloud applications, are being currently used. Juniper research predicts that cloud-based mobile applications will increase to approximately 9.5$ billion by 2014 which improves the performance.

Cloud Setup

Cloud Service Provider will contain the large amount of data in their Data Storage. Also the user's information can be authenticated and can be stored using the Cloud Service provider. The Cloud Service Provider stores the user's information. Also the Cloud Server will redirect the User requested job to the Resource Assigning Module to process the User requested Job. The Resource Assigning Module is used for processing the resources of all the users.The Cloud Server will establish connection between the client and the other modules of the cloud network.

Private Cloud

Private clouds are provided only for one user and cannot be shared. The resource can be provided in-house or externally. Private cloud provides the same resource as public clouds. only the limited number of people are provided for using the services of private cloud and it is also permitted only behind the firewall.The direct control is also provided over the data.

8.2.2 Public Cloud

Public cloud provides data transfers, storage, and processing. It is also provided free or on a pay free model. The organiztion manages the public cloud it provides. Visualization makes access to other customers' data extremely difficult.

Hybrid Cloud

The hybrid cloud architecture merges private and public cloud deployments.The hybrid cloud provides services to the multiple resources and provides rent for a short period. This requires a great deal of operational ability in the organization to seamlessly scale between the private and public cloud. On the long-term the additional expense of the hybrid approach often is not justifiable since cloud providers offer major support.

MYSQL IS A DATABASE MANAGEMENT SYSTEM

The structured collection of data is provided in the database. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. The database management systems are good at handling large amount of data and also it plays a central role in computing process of different applications.

MySQL databases are relational.What is MySQL?

A relational database stores data in separate tables rather than putting all the data in one big storeroom.The rules are set up for governing the relationships between tables. By setting these rules the well designed database has been developed. The SQL part of “MySQL” stands for “Structured Query Language”.The standardized language to access the databases is SQL. SQL is defined by the ANSI/ISO SQL Standard.

IX.CONCLUSION

In this study, we have analyzed the security of the honeyword system and addressed a number of flaws that need to be handled before successful realization of the scheme. In this respect, we have pointed out that the strength of the honeyword system directly depends on the generation algorithm, i.e., flatness of the generator algorithm determines the chance of distinguishing the correct password out of respective sweet words. Another point that we would like to stress is that defined reaction policies in case of a honeyword entrance can be exploited by an adversary to realize a DoS attack. This will be a serious threat if the chance of an adversary in hitting a honeyword given the respective password is not negligible. To combat such a problem, also known as DoS resistance, low probability of such an event must be guaranteed. Unpredictable usage of honeywords can be able to minimize the risks. Hence, we have noted that the security policy should strike a balance between DoS vulnerability and effectiveness of honeywords. Furthermore, we have demonstrated the weak and strong points of each method introduced in the original study. It has been shown that DoS resistance of the chaffing-by-tweaking method is weak and also its flatness can be questioned by regarding Remark. Although some weaknesses of the chaffing-by-tweaking techniques are accepted by their creators, we believe that it should not be considered as alternative method due to its predictable nature and a potential DoS weakness.

X. References

[1] A Review on Honey Words “Detecting Password Cracking with Hacker Tracking” https://www.ijircce.com/upload/2016/october/132_Hunney%20paper.pdf by Dr.Prashant Kumbharkar, Snehal Aher, Ashish Dhamal, Vinay Maslekar, Akshay Takale Vol. 4, Issue 10, October 2016

[2 Examination of a New Defense Mechanism:Honeywords” http://www.ijettjournal.org/archive/ijett-v27p238 Rohit Guj, Rahul Dhumal, Shrinath Shelke,Pravin Hinge,Prof.Prashant Suryavanshi International Journal of Engineering Trends and Technology (IJETT) – Volume 27 Number 4 - September 2015

[3] “Some Remarks on Honeyword Based password cracking detection”
https://www.iacr.org/cryptodb/data/paper.php?pubkey=25626
Author Imran Erguler. Publication, IACR Cryptology ePrint archive 2014.

[4] Security Analysis of Honey words Generation Scheme to Evade Unauthorized Access” http://www.ijarcce.com/volume-5-issue-7.html by Ms. Manisha
B.Kale and Prof.D. V. Jadhav in University Boston, 2014.

[5 The Dangers of Weak HashesKelly Brown” https://www.sans.org/reading-room/whitepapers/authentication/dangers-weak-hashes-34412 by SANS Institute Reading Room in November 15, 2013.

[6] Kamouage: Loss-Resistant Password
Management” by Hristo Bojinov, Elie Bursztein, Xavier Boyen, and Dan Boneh
(IJSR) 2014

[7] Improving Security using Deception”
By Mohammed H. Almeshekah, Eugene H.Spafford and Mikhail J. Atallah in November 11, 2013.

[8] Honeywords: Making Password-Cracking Detectable “https://dspace.mit.edu/openaccess-disseminate/1721.1/90627 by Ari Juels Ronald L. Rivest MIT ,May 2, 2013

[9] “Improvingsecurity using deception,” M. H. Almeshekah, E. H. Spafford, and M. J. Atallah, Center for Education and ResearchInformation Assurance and Security, Purdue Univ., WestLafayette, IN, USA: Tech. Rep. CERIAS Tech. Rep. 2013-13, 2013.

[10] “Secure Hash Standard”
National Institute of Standards and Technology. (2012, March). Retrieved August 7, 2013, from Information Technology Laboratory:
http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf
NVIDIA

[11] “Kamouflage:Loss-resistant password management,” H. Bojinov, E. Bursztein, X. Boyen, and D.oneh, in Proc. 15th Eur. Conf.Res. Comput. Security, 2010, pp. 286–302

[12] “Password cracking using probabilistic context-free grammars,” by M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek in Proc. 30thIEEE Symp. Security Privacy, 2009, pp. 391–405.

[13“Protecting financial institutions frombrute-force attacks,” C. Herley and D. Florencio, in Proc. 23rd Int. Inform. Security Conf., 2008,pp. 681–685.

[14]. Password Authentication Schemes: Current Status and Key Issues” by Chwei-Shyong Tsai1, Cheng-Chi Lee2, and Min-Shiang Hwang1 In International Journal of Network Security, Vol.3, No.2, PP.101–115, Sept. 2006 (http://ijns.nchu.edu.tw/)

[15] “The use of deception techniques: Honeypots and decoys,” F. Cohen, Handbook Inform. Security, vol. 3, pp. 646–655, 2006.

[16] "A formal framework and evaluation method for network denial of service" C. Meadows, in Proc. 12th IEEE computer security foundations workshop, 1999, pp. 4-13.

[17] “Honeyfiles: deceptive files for intrusion detection,” J. Yuill, M. Zappe, D. Denning, and F. Feer, in
Information Assurance Workshop, 2004. Proceedings from the Fifth Annual IEEE SMC, pp. 116–122, IEEE.

[18] Bonneau, J., Preibusch, S.: “The password thicket: Technical and market failures in human authentication on the web” In: WEIS. (2010) 19.

[19]Notoatmodjo, G., Thomborson, C.: “Passwords and perceptions” In: Proceedings of the Seventh Australasian Conference on Information Security–AISC 2009, Aus- tralian Computer Society, Inc. (2009) 71–78 20.

[20]Florencio, D., Herley, C “A large-scale study of web password habits” In: Proceed- ings of the 16th international conference on World Wide Web, ACM Press (2007) 657–666

[21]Percival, C. (2009). “Stronger Key Derivation Via Sequential Memory-Hard Functions” BSDCan.
Ottawa, Canada.

[22]Percival, C., & Josefsson, S. (2013, September 24). “The scrypt Password-Based Key Derivation
Function “ Retrieved August 13, 2013, from Internet Engineering Task Force:
http://tools.ietf.org/html/draft-josefsson-scrypt-kdf-01 Provos, N., & Mazières, D. (1999). A Future-Adaptable Password Scheme. USENIX ATC '99.

[23]Montgomery, CA: Usenix Association.
RainbowCrack Project. (2013, August 10).List of Rainbow Tables. Retrieved August 10, 2013, from RainbowCrack Project: http://project-rainbowcrack.com/table.htm
RSA Laboratories. (2000, September).

[24] PKCS #5: Password-Based Cryptography Specification. Retrieved August 13, 2013, from Internet Engineering Task Force:
https://tools.ietf.org/html/rfc2898

[25]New York: John Wiley & Sons, Inc.
Signler, M. (2009, December 14). One Of The 32 Million With A RockYou Account? You May Want To Change All Your Passwords. Like Now. Retrieved August 3, 2013, from
TechCrunch:http://techcrunch.com/2009/12/14/rockyou-hacked/

[26]Steube, J. (2012). Exploiting a SHA1 Weakness in Password Cracking. Password^12. Oslo, Norway