A Project Report

On

"Improving Security and Achieving Flatness to the User-Password by Using Honeyword"

Submitted to

Savitribai Phule Pune University

In Partial Fulfillment of the Requirement for the Award of

Bachelor of Engineering In
Information Technology

By

Lalit Suthar	**Exam No : B121048560**
Vimal Singh	**Exam No : B121048564**
Mahesh Umale	**Exam No : B121048562**
Hemalata Khandagle	**Exam No : B121048531**

Under The Guidance of
Mr. Anand Bone

Sinhgad Institutes

Department of Information Technology

SKN Sinhgad Institute of Technology and Science, Lonavala.

Gate No.309/310, Off Pune-Mumbai Express Way,Kusgaon (BK), Lonavala, Pune - 410401

2016-2017

Affiliated to

Savitribai Phule Pune University

SKN Sinhgad Institute of Technology and Science, Lonavala.

Department of Information Technology

Gate No.309/310, Off Pune-Mumbai Express Way,Kusgaon (BK), Lonavala, Pune - 410401

Sinhgad Institutes

CERTIFICATE

This is certify that the project entitled

"Improving Security and Achieving Flatness to the User-Password by Using Honeyword"

submitted by

Lalit Suthar	**Exam No : B121048560**
Vimal Singh	**Exam No : B121048564**
Mahesh Umale	**Exam No : B121048562**
Hemalata Khandagle	**Exam No : B121048531**

is a bonafide work carried out by them under the supervision of **Mr. Anand Bone** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University for the award of the Degree of Bachelor of Engineering (Information Technology). This project report has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

Date: / /

(Mr. Anand Bone)
Project Guide

(Prof. Ganesh Kadam)
HOD, IT

External Examiner

(Dr. M. S. Rohokale)
Principal

Acknowledgment

We are profoundly grateful to **Mr. Anand Bone** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

We would like to express deepest appreciation towards **Dr. M. S. Rohokale**, Principal, SKN Sinhgad Institute of Science and Technology, Lonavala.

Prof. Ganesh Kadam, Head of Department of Information Technology, whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Information Technology Department who helped me directly or indirectly during this course of work.

Lalit Hajarimal Suthar (B120148560)
Vimal Chetan Singh (B120148564)
Mahesh Arun Umale (B120148562)
Hemalata Manohar Khandagle (B121048531)

Abstract

The new developments in the field of information technology offered the users enjoyment, comforts and convenience,but there are many security thread related issues. One of them is Password file. Password files have found a lot of security problem that has affected billions of users as well as many companies/Industries. Password file is generally stored in encrypted format like Hash file, if a password file is stolen or theft by using the password cracking techniques and decryption technique it is easy to get most of the plaintext and encrypt passwords. For which Juels and Rivest proposed Honeywords (decoy passwords) to detect attacks against Hash password databases. The creation of honeyword password, i.e. a decoy password using a perfectly flat honeyword generation method, and try to attract adversary / illegal or unauthorized user. Hence that time server detect the unauthorized user and Alert the User and Administrator. Here the given provision also protect the original data from unauthorized user.

Keywords: *Authentication, Confidentiality, Impersonation, Infraction, Honeywords, Login, Sweetword, Sugarword, Cyberpunk, Honeypot.*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

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. 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. Here focus is 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. In this method, the administrator purposely creates fake user accounts to lure adversaries and detects a password leakage, if any one of the honeypot passwords get used. This idea has been modified by Herley and Florencio to protect online banking accounts from password brute force attacks. According to the study, for each user incorrect login try with some passwords lead to honeypot accounts, i.e., malicious behavior is recognized. For instance, there are 108 possibilities for a eight-digit password and let system links 10,000 wrong password to honeypot accounts, so the adversary performing the bruteforce attack 10,000 times more likely

to access a honeypot account than the genuine account. Use of decoys for building theft-resistant was introduced by Bojinov et al.in called as Kamouflage. In this model, the fake password sets are stored with the real user password set to mask the real passwords, thereby forcing an adversary to carry out a considerable amount of work before getting the Right information. Recently, Juels and Rives have presented the honeyword mechanism to detect an adversary who attempts to login with cracked passwords. Basically, for each username a set of sweet-words is generated such that only one element is the correct password and the others are honeywords (decoy passwords). Hence, when an adversary tries to enter into the system with a honeyword, an alarm is triggered to notify the administrator about a password leakage.

Here analysis on honeyword approach and putting some remarks about the security of the system. Furthermore, Figuring out that the key term for this method is the generation algorithm of the honeywords such that they shall be identical from the correct passwords.

1.1 Project Idea

In our society, the threat of cyber-attacks is increasingly high and harmful. With the rise of usage in computers, criminal activity has also shifted from physical intrusion into cyber attacks. So to Overcome the problem of Confidentiality, Data integrity, Data loss and Privacy works on security of User Password from adversary.

Honeywords provide the ability to identify security breaches in a system. A security breach will be any action the owner of the system deems unauthorized. In the existing system they have implemented a general model algorithm which detects multiple passwords brute force attack. In proposed system implementation of an advanced Honeywords model algorithm is done. In proposed system it using advanced approach for file system. Here Implementation of Real Time Honeywords detection System which will actively detect brute force attacks and Make adversary fool by decoy login.

In order to sense impersonation the legitimate password is stored with several Honeywords for each user Account. If proper Honeywords are chosen, a cyber-attacker who steals a file of hashed passwords would be confused, whether it is a real password or Honeywords for user Account. for each sweetword many to many mapping is done with key bits. EX-ORing of Key and key bits will redirect again to mapped honeywords. Login with Honeywords will trigger an Alarm, which will notify the administrator about a password file Infraction. To identify or detect

occurrence of password database infraction honeypot concept is used. So improvisation and implementation of an easy and efficient solution which will protect and detect exposure events of passwords file.

1.2 Motivation

Providing number, test, special character validation passwords are the more generally used authentication method in computer systems. Backward references shown that passwords are often simple for attackers to disclose. A general threat model is an attacker who take without permission a list of hashed passwords, empower him to end favour to become fissure them offline at his leisure.

Although it is generally believed that password composition policies make passwords difficult to think, and hence more free from, research has struggled to quantify the level of resistance to guessing provided by different password-composition policies or the individual requirements they comprise.

1.3 Objectives

- Create honeyword from existing user password.
- Make Decoy Password indistinguishable from the correct passwords.
- Create Decoy Login.
- Detect attacks against hashed password databases.
- Reduces the storage cost.
- Maintain Data Confidentiality.
- Triggers alerts to make the administrator and User aware of Attacks.
- Detect whether a password file disclosure incident happened or not to take appropriate actions.
- Take actions to stop brute force attacks.
- Create Many to Many mapping Between sweetwords and Seed bits to avoid prediction of sugarword by adversary.
- Protect Computers and Networks from malicious attacks.

1.4 Scope

- **Accuracy**: All the functionally bonded logical dependencies must be integrated.
- **Efficiency**: The whole system should work under all circumstances and on a long run it should work efficiently irrespective of their proprietary format.
- **Any Prerequisite for the use**: The existing systems are not altered, and integration is done.
- Make the cyberpunk to get confuse by including the fake information of the account. Here the password misuse will be protected from the cyberpunk for the further access of the user account.
- Validate whether data access is authorized when abnormal information access will detect.
- Use of this technology to launch disinformation attacks against malicious insiders, preventing them from distinguishing the real sensitive customer data from fake worthless data.

Chapter 2

Literature Survey

2.1 Recent work

2.1.1 The Use of Deception Techniques : Honeypots and Decoys.

Author : *Cohen*
Publication : *Handbook of Information Security, vol. 3, pp. 646–655, 2006*
Techniques Used : Deception techniques have the demonstrated ability to increase attacker workload and reduce attacker effectiveness.
Advantages : The most critical work that must be done in order to make progress is the systematic study of the effectiveness of deception techniques against combined systems with people and computers.
Disadvantages : Modern defensive computer deceptions are in their infancy, but they are moderately effective, even in this simplistic state.
Remark : This article has summarized a great deal of information on the history of honeypots and decoys for use in defense of computer systems.

2.1.2 Protecting financial institutions from brute-force attacks.

Authors : *Herley and D Florencio*
Publication : *Microsoft Research One Microsoft Way Redmond(2008), WA*
Techniques Used : Show that is simple to ensure that a brute-force attacker will encounter hundreds or even thousands of honeypot accounts for every real break-in.
Advantages : Activity in the honeypots provides the data by which the bank learns the attackers attempts to tell real from honeypot accounts, and his cash out strategy.
Disadvantages : Examine the problem of protecting online banking accounts from password brute-forcing attacks.
Remark : In a brute-force attack repeated credential pairs are tried in an attempt

to gain access to an account. The simplest is directed against a single account: the attacker tries all possible passwords for one user ID until one succeeds.

2.1.3 Kamouflage: Loss-resistant Password Management.

Authors : *H.Bojinov, E.Bursztein, X.Boyen, and D.Boneh*
Publication : *Computer Security - ESORICS 2010. Springer, 2010, pp. 286-302.*
Techniques Used : Introduce Kamouflage : new architecture for building the ft-resistant password managers. An attacker who steals a laptop or cell phone with a Kamouflage-based password manager is forced to carry out a considerable amount of online work before obtaining any user credentials.
Advantages : Replacement for the built-in Firefox password manager, and provide performance measurements and the results from experiments with large real-world password sets to evaluate the feasibility and effectiveness of our approach.
Disadvantages : Presented a system to secure the password database on a mobile device from attacks that are often ignored by deployed password managers.
Remark : Kamouflage is well suited to become a standard architecture for password managers.

2.1.4 Password Cracking Using Probabilistic Context-Free Grammars .

Authors : *M.Weir, S.Aggarwal, B.de Medeiros, and B.Glodek*
Publication : *Password Cracking Using Probabilistic Context-Free Grammars," in Security and Privacy,30th IEEE Symposium on. IEEE,2009, pp.391-405.*
Techniques Used : This grammar allows us to generate word-mangling rules, and from them, password guesses to be used in password cracking.
Advantages : This approach seems to provide a more effective way to crack passwords as compared to traditional methods.
Disadvantages : Approach was able to crack 28%
Remark : Honeyword approach was able to crack 28% to 129% more passwords than John the Ripper, a publicly available standard password cracking program.

2.1.5 Improving Security using Deception.

Authors : *M.H.Almeshekah , E.H.Spafford, and M.J.Atallah*
Publication : *Center for Education and Research Information Assurance and Security, Purdue University, Tech. Rep. CERIAS Tech Report 2013-13, 2013.*
Techniques Used : Exploring complex relationships among protection techniques ranging from denial and isolation, to degradation and obfuscation, through negative information and deception, ending with adversary attribution and counter-operations.
Advantages : Outlined a new classification scheme for deception techniques in cyber security.
Disadvantages : Have shown how some of these techniques have been known and used for many years, but that the field is under-developed.
Remark : Explained how systems can be augmented to use deception and false information to protect them and their data, to degrade attacks, to expose attackers, to enhance attribution, and possibly to be used to damage or degrade attacker capabilities.

2.1.6 If Your Password is 123456, Just Make It Hack me.

Author : *Ashlee Vance.*
Publication : *The Dangers of Weak Hashes, SANS Institute InfoSec Reading Room, Tech. Rep., 2013.*
Techniques Used : Basics of password hashing, look at password cracking software and hardware, and discuss best practices for using hashes securely .
Advantages : Hashes are compromised it is not easy for hackers to generate passwords from the hashes.
Disadvantages : Password leaks are becoming a common occurrence on the Internet with several large scale leaks happening every year.
Remark : Don't try to create your own hashing algorithm Don't use outdated algorithms (such as MD5 or SHA1) Use SHA2 or similar strength algorithm.

2.1.7 Understanding password Database Compromises.

Authors : *D.Mirante and C.Justin*
Publication : *Department of Computer Science and Engineering Polytechnic Institute of NYU(2013).*
Techniques Used : It forces the attacker to brute force the hashes one at a time, instead of attacking them as a group.
Advantages : Offering the benefit of flexibility, with the ability to provide resources almost instantaneously as necessary to avoid site shutdown.
Disadvantages : High profile website intrusions, wherein user login credentials and other data were compromised.
Remark : A study was undertaken to research information posted on the web concerning recent, high profile website intrusions.

2.2 Existing System

Discloser of password files is a severe problem that has affected millions of users and companies like Yahoo, RockYou, LinkedIn, eHarmony and Adobe since leaked passwords make the users target of many possible cyber-attacks. These recent events have demonstrated that the weak password storage methods are currently in place on many web sites.

2.3 Limitation of Existing System

- Discloser of password files is a severe security problem that has affected millions of users and companies.
- Disclosure of Key leads to cyber-attacks.
- Leaked passwords make the users target of many possible cyber-attacks.
- Less secure

Chapter 3

Problem Statement

3.1 Problem Statement

To design and develop an alternative approach that selects the honeywords from existing user passwords in order to provide realistic honeyword – a perfectly flat honeyword generation method.

Username is useful to find the particular user and the password for the authorization of the user. The username-password checking is more important in the security system, so to protect password from third party here implementation for each user account, the valid password is converted new password using honeywords and hash password.

New password is the combination of existing user passwords called honeywords. Fake password is nothing but the hone words, If honeywords are choice properly, a cyber-attacker who to take a file of hashed passwords cannot be sure if it is the real password or a honeyword for any account. Moreover, entering with honeywords to login will trigger an alarm inform the administrator about a password file an infraction, so introduction of an easy and capable, solution to the detection of password file exposure events. examination in detail with careful attention the honeywords system and present some comment to focus be used weak points. Also focus on pragmatic password, reduce storage cost of password, and alternately to choice the new password from existing user passwords.

Chapter 4

Proposed System

4.1 Proposed System

Here emphasis is given on the security issue and deal with fake passwords or accounts as a simple and cost effective solution to detect compromise of passwords. Honeypot is one of the methods to identify occurrence of a password database breach. In this approach, the administrator purposely creates deceit user accounts to lure adversaries and detects a password disclosure, if any one of the honeypot passwords get used. Proposal of a novel honeyword generation approach which reduces the storage overhead and also it addresses majority of the drawbacks of existing honeyword generation techniques. Proposed model is based on use of honey words to detect password-cracking. Proposal of using indexes that map to valid passwords in the system. The contribution of our approach is twofold. First, this method requires less storage compared to the original study. Within our approach passwords of other users are used as the fake passwords, so guess of which password is fake and which is correct becomes more complicated for an adversary.

Chapter 5

Software Requirements Specification

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

5.1 Hardware Requirements

- System : Pentium IV 2.4 GHz.(min)
- Hard Disk : 10 GB.(min)
- Ram : 512 Mb.(min)

5.2 Software Requirements

- Operating system : Windows XP/7.(min)
- Coding Language : JAVA/J2EE
- IDE : Java eclipse (Kepler).
- Web server : Apache Tomcat 7.(min)
- Front End : JSP, Javascript, CSS, HTML, etc.
- Back End : MySQL as database server.

5.3 Functional Requirements

- Alert admin and user if any attempt is made to access the account with wrong password.
- Trace the IP address of attacker and mail to Admin and User.
- Block the IP address of attacker to avoid further more attacks.

- Admin should able to get list of Blocked account and will be able to recover the account on user request.
- Provide at most 3 Attempt to login with Honeyword and 1 for wrong password.
- Provide user the recovery my account option with request to admin by mail or support.
- Provide User friendly Interface and Interactive as per standards.

5.4 Non-functional Requirements

5.4.1 Performance Requirements

- The Application database should accommodate large no. of user account details while registering without any fault.
- Responses to Login shall not take longer than 5 Second to appear on the User Screen.
- Alert of attempt made by attacker shall not take more than 10 seconds to mail to admin and user.
- Quick Access should be able to support 300 user and 500 request / minute concurrently.

5.4.2 Availability Requirements

- Application should be available 24 hours in order to provide access to user without any server down / fail.
- Database backup and recovery plan should be proper in order to avoid any unexpected downtime of Application.

5.4.3 Security Requirements

- Administrator will have full access to Application to resolve any issues.
- Normal user can just read information but they cannot edit or modify anything except their personal information.

5.4.4 Usability Requirements

- Ease of Use requirement address the factor that constitute the capacity of software to be understand, learned and used by its intended user.

Chapter 6

System Design

6.1 System Architecture

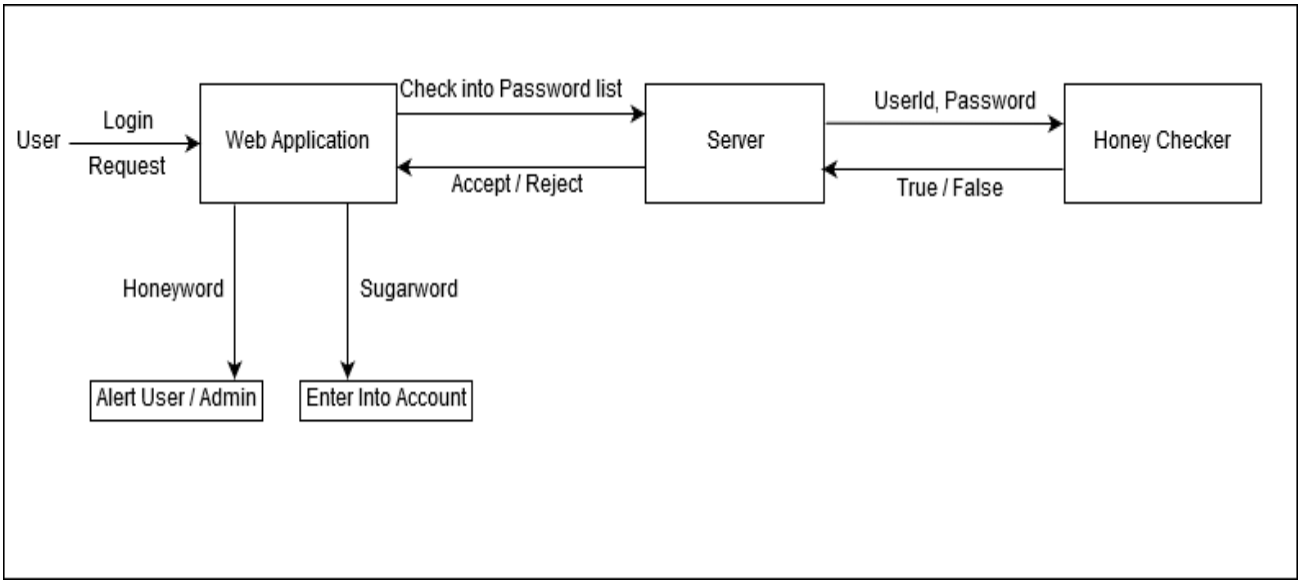

Figure 6.1: Honeyword system architecture

System Architecture represents interaction between various modules of Honeyword system. user interacts with web application through which it accept authentication credential and send to server which also verify credential at Honey-checker. server deals with honeychecker as well web application for passing the credential data. after retrieving the verification of credential web application either alert user/admin or Enter into account. verification also include identification the honeyword or sugarword. application provide only 3 login attempt with honeyword and 1 login attempt with sugarword.

6.2 Use-Case Diagram

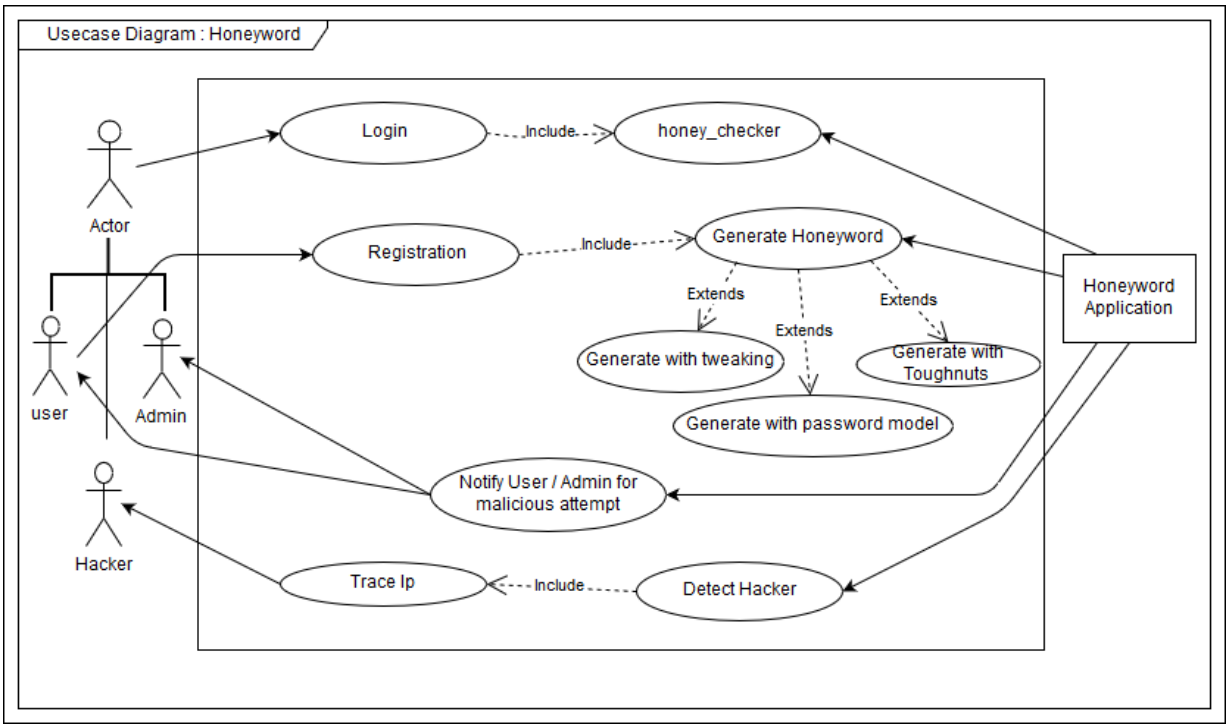

Figure 6.2: Use-Case Diagram

Use-case Diagram Shows various usecases between actors such as user, admin and with different modules of honeyword application. Here login activity include honeychecker module. Registration activity include Honeyword generation activity like Tweaking, Tough-nuts and Password model. Alert/notification activity includes Tracing IP address of malicious user. Rectangular Block shows the Boundaries of Use-cases of application. An actor represents a role that an outsider takes on when interacting with the business system. For instance, an actor can be a customer, a business partner, a supplier, or another system. Here actor generalization is used to show actor can be user, admin, hacker, etc.

Actor : User, admin, hacker, system, Honeyword application.

UseCases : login, registration, traceIP, notification, Alert, Honeyword Generation, Honey Checker.

6.3 Sequence Diagram

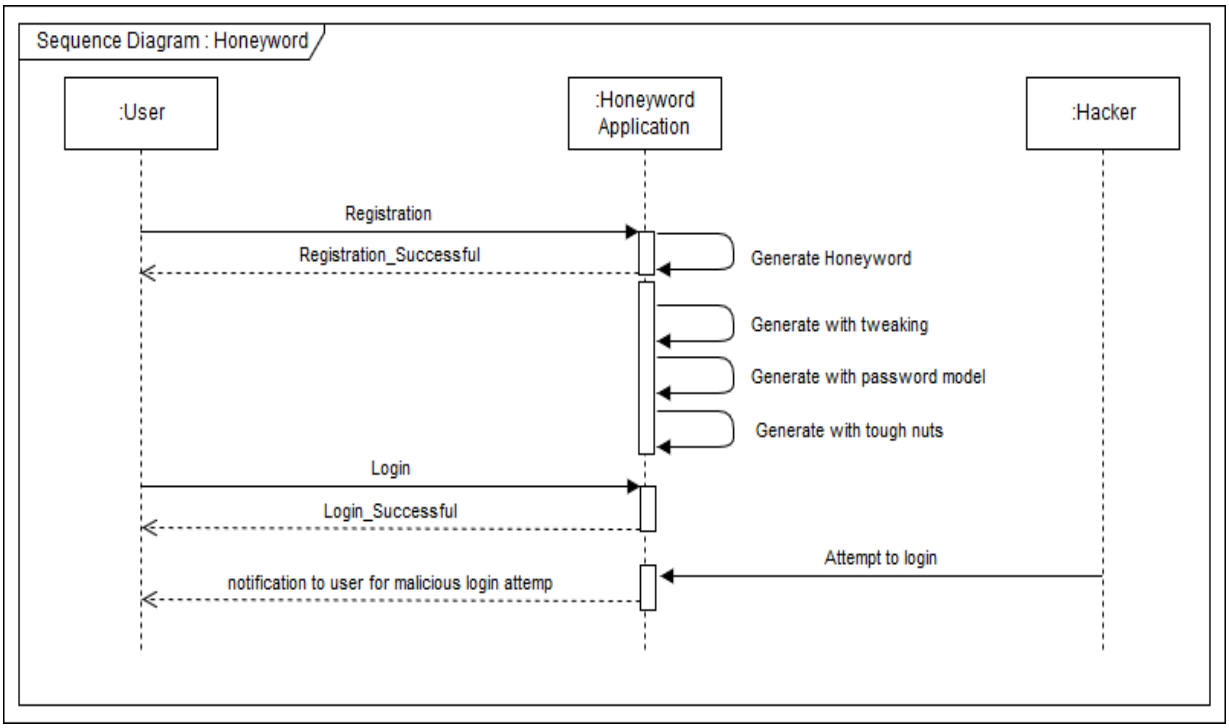

Figure 6.3: Sequence Diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. An important characteristic of a sequence diagram is that time passes from top to bottom. Registration activity which is must for new user then Sequence of activity leads to login activity. Dark arrow headed line between objects shows activity execution and Dotted arrow headed line shows acknowledge. Vertical dotted line show lifeline of objects involved in application. vertical rectangular shape entity shows time period of activity. Types of Messages in Sequence Diagrams : Synchronous Message, Asynchronous Message, Reply or Return Message, Self Message, Create Message, Delete Message, Found Message.

6.4 Class Diagram

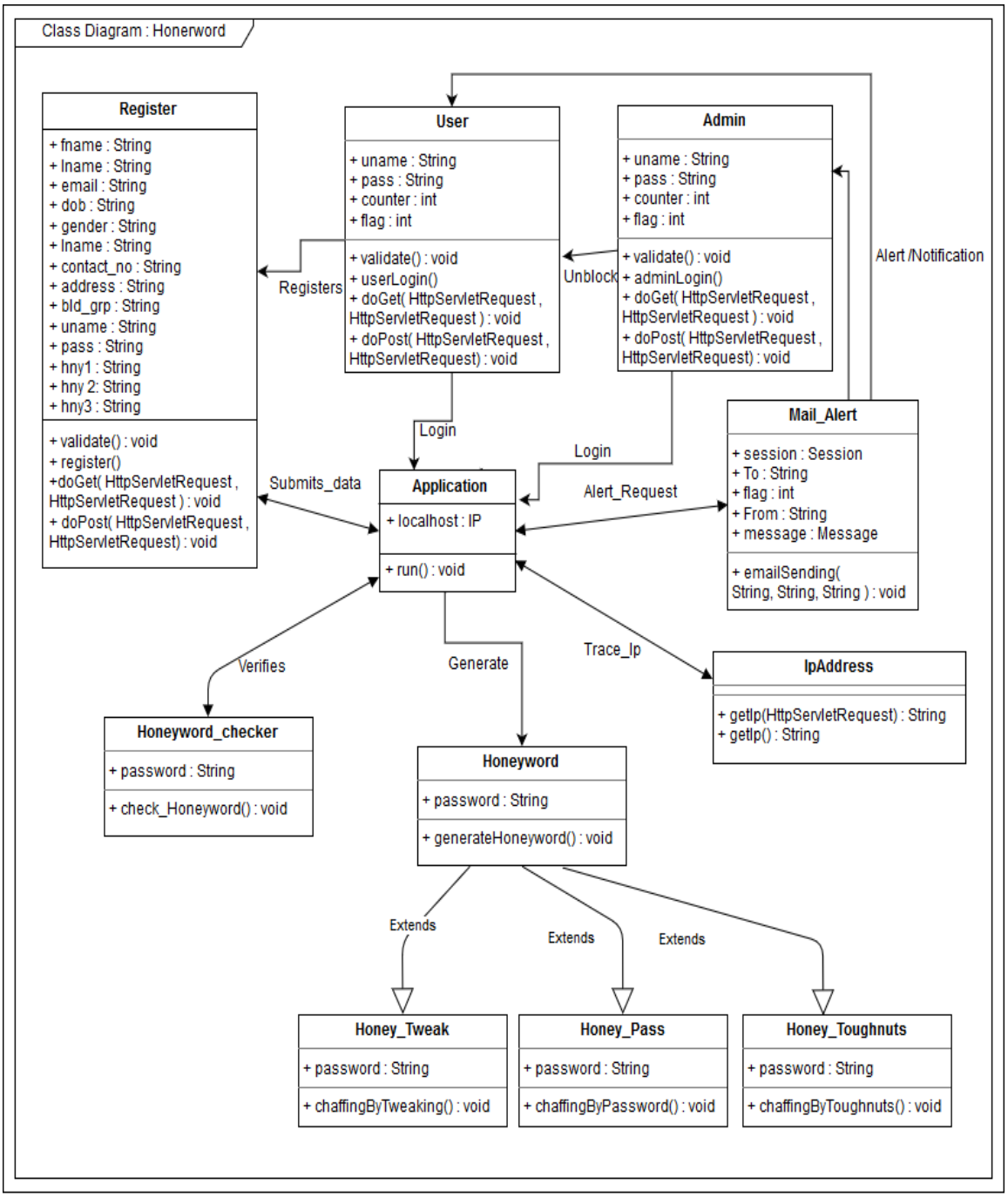

Figure 6.4: Class Diagram

Class Diagram is structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. this also represents various kind of relationship like Dependency, Association, Aggregation, Composition, Generalization and Multiplicity. register class include various attribute like name, add, contact, etc. and methods like validate, register,etc. To specify the visibility of a class member no-

tation are used : + for Public, - for Private, # for Protected and / for Derived.

6.5 Activity Diagram

Activity diagram, which is related to program flow plans, are used to illustrate activities. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

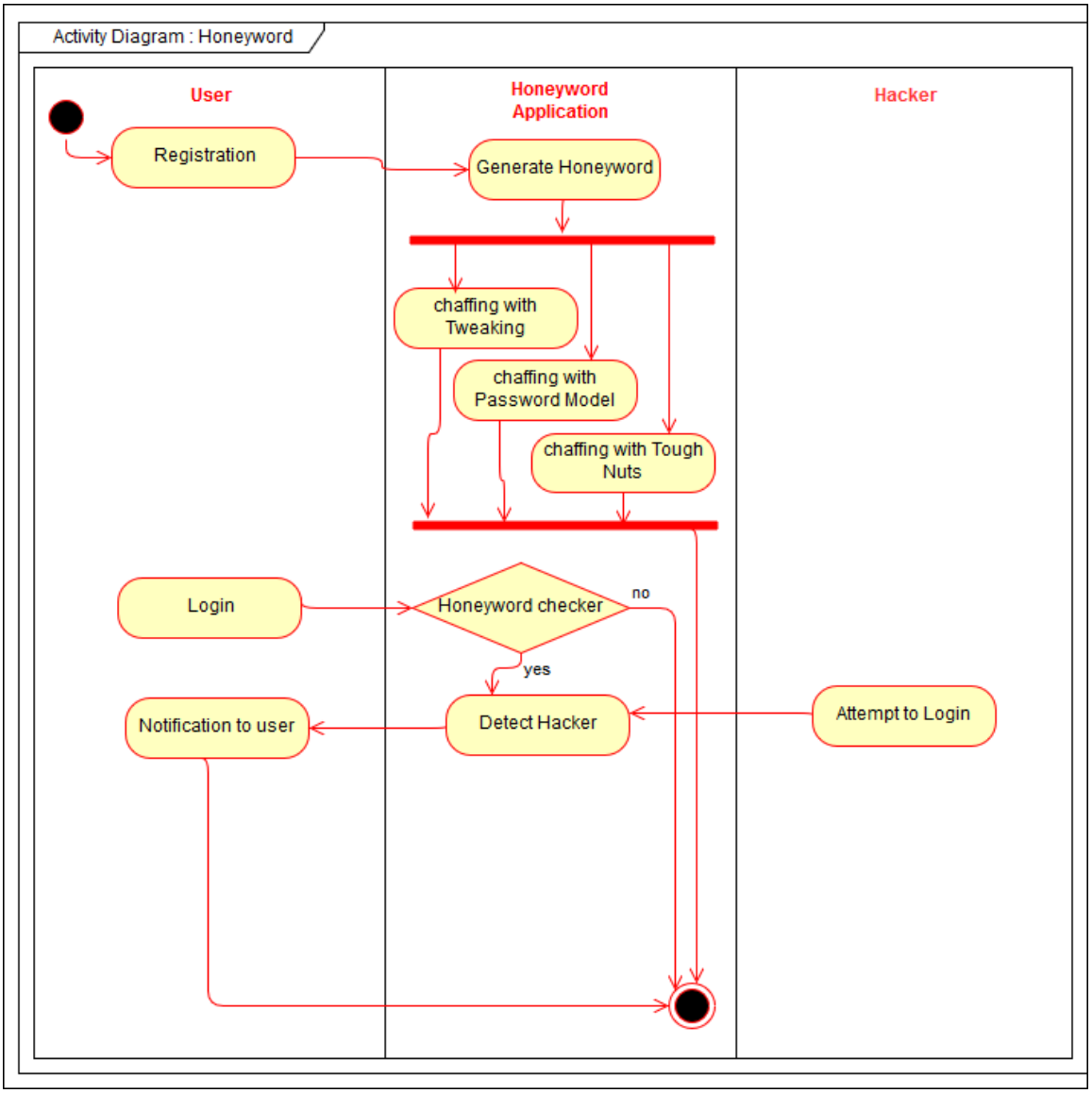

Figure 6.5: Activity Diagram

Rounded rectangles represent *actions*.
Diamonds represent *decisions*.
Bars represent the *start (split) or end (join)* of concurrent activities.
a **Black circle** represents the *start (initial node)* of the workflow.
an **Encircled Black circle** represents the *end (final node)*.

Chapter 7

Implementation

7.1 System Working

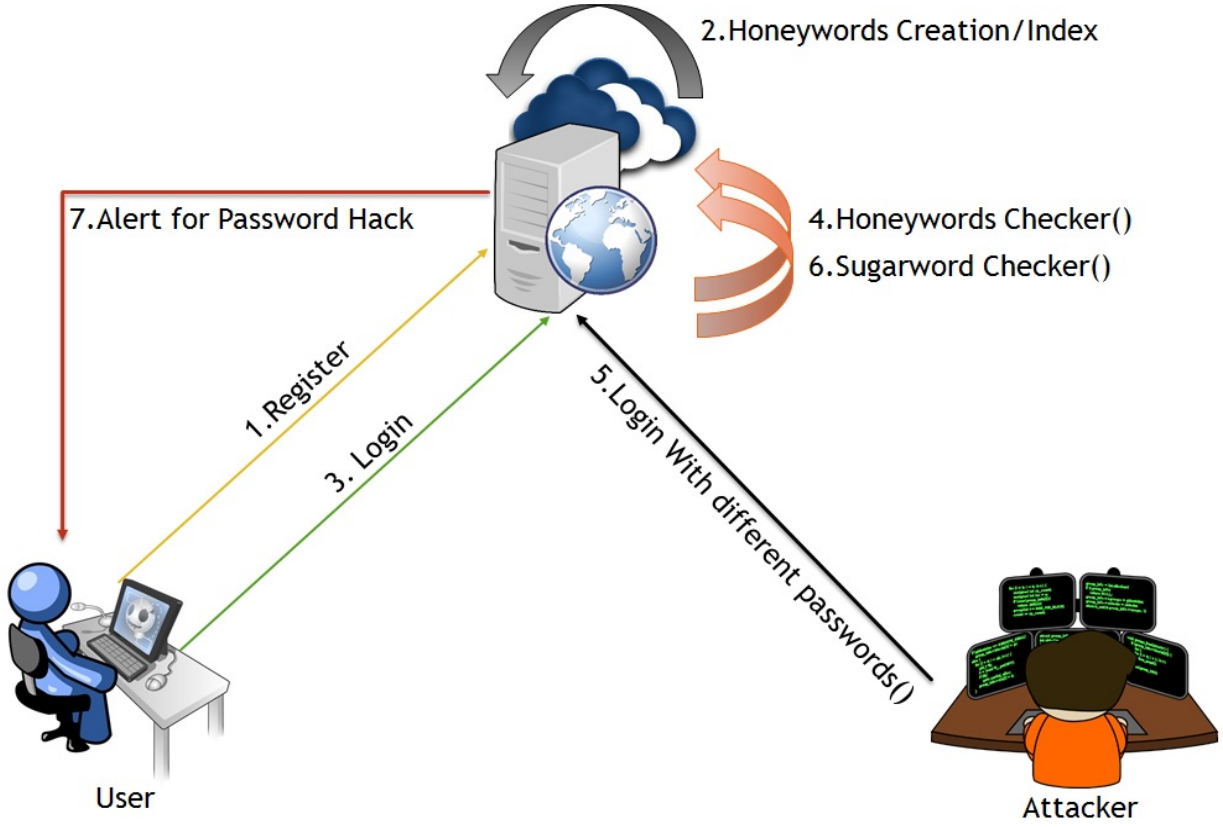

Figure 7.1: System Working

7.2 Algorithm

Inputs:

1). T fake user accounts (honey pots)

2). Index value between [1; N]

3). Index list, which is not previously assign to user.

Procedure:

Step 1: Honeypots creation: fake user account

- For each account honey index set is created like Xi $\rightarrow$(xi;1; xi;2; : : : ; xi;k); one of the elements in Xi is the correct index (sugar index) as Ci
- create two password file file f1 and file f2. F1 Store username and honey-index set $\langle$ hui,xi$\rangle$ Where hui is honey pot account and F2 keeps the index number and the corresponding hash of the password $\langle$ Ci;H(pi)$\rangle$

Step 2: Generation of honey-index set In Step 1.

- Inserted honey index set in file F1 but don't know how to create that. By using honey index generator algorithm Generator(k; SI) Ci:Xi Generate Xi.
- select xi randomly selecting k-1 numbers from SI and also randomly picking a number ci SI .
- Ui; ci pair is delivered to the honey checker and F1, F2 files are updated.

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, notifies system a honey word situation.

Step 4: Encryption

- Let a user message (password) space M which contains all possible messages. By outlining these messages to a seed space S by using a DTE (distribution-transforming encoder).
- 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.
- The size of the seed range of m is directly related to how most likely m is in the message space M. There is Need to 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.
- 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. This information is useful to find the cumulative probability range corresponding the message m and relate it to the same percentile seed range in Space S.

7.3 Mathematical model

Let,

S⇒{I,O}

Where, I→{Input}

I⇒{H, U, A, AD, P, E, K, HE}

Where,

H→{Honeywords}

U→{Users}

A→{Attacker}

AD→{Admin}

P→{Password}

E→{Email}

K→{Key}

HE→{Honey encryption}

O⇒{Output}

Considering that database 'D' and 'n' number of attribute such as user name, user id etc. D→ {A | A ∈ 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 enters 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 further assume to have a set 'S' to have value 'n' number of detect value at particular instance. Let's denote the current situation in the following manner 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.

- GET(D,X,SERVER): Admin get all information about the user account from server.
- PUT(X,ATK,SERVER): Here admin will upload attacker's information on server.
- PUTP(X,REPORT,SERVER) : Here admin upload daily report on server.

7.4 Problem Class

7.4.1 What is P ?

- P is set of all decision problems which can be solved in polynomial time by a deterministic.
- Since it can be solved in polynomial time, it can be verified in polynomial time.
- Therefore P is a subset of NP.

Whenever a user types password in any organization's Login box, the hacker intercepts the password. The threat of such hackers is pervasive. Username is useful to find the particular user and the password for the authorization of the user. Once a password file is stolen, by using the password cracking technique it is easy to capture most of the plaintext passwords.

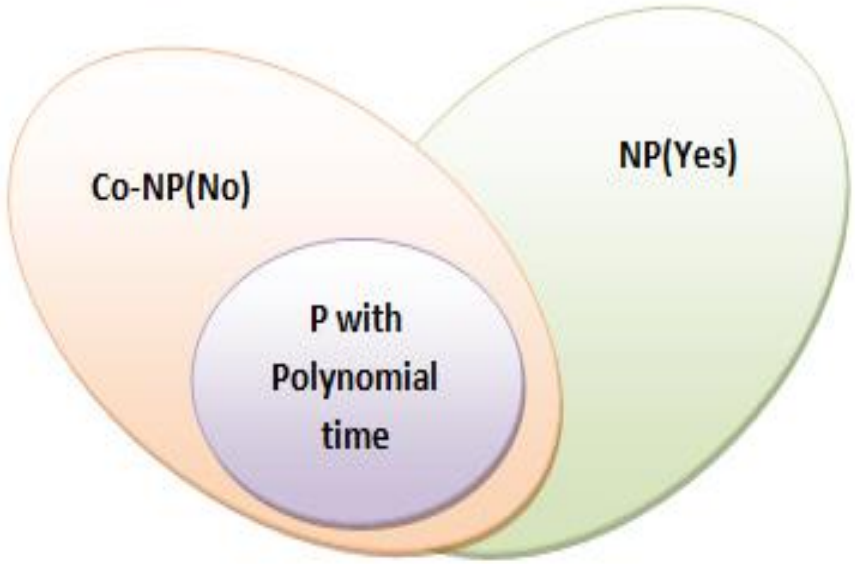

Figure 7.2: Polynomial Time

7.4.2 What is NP ?

"NP" means "it can be solve in polynomial time if able to break the normal rules of step-by-step computing".

What is NP Hard ?

A problem is NP-hard if an algorithm for solving it can be translated into one for solving any NP-problem (non-deterministic polynomial time) problem. NP-hard therefore means "at least as hard as any NP-problem," although it might, in fact, be harder.

Here focus is on the security issue and deal with fake passwords or accounts as a simple and cost effective solution to detect compromise of passwords. Honeypot is one of the methods to identify occurrence of a password database breach. In this approach, the administrator purposely creates deceit user accounts to lure adversaries and detects a password disclosure, if any one of the honeypot passwords get used.

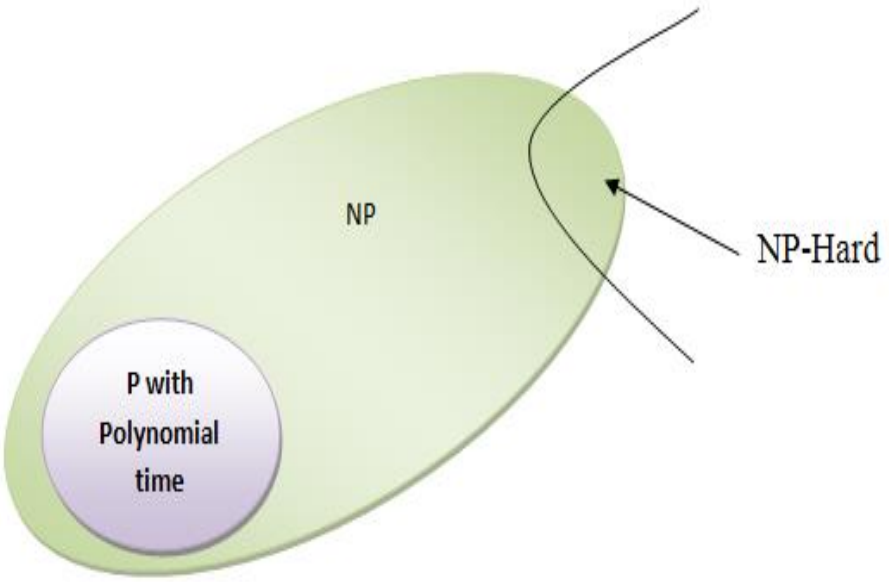

Figure 7.3: NP-Hard Problem

What is NP-Complete ?

- Since this amazing "N" computer can also do anything a normal computer can, we know that "P" problems are also in "NP".
- So, the easy problems are in "P" (and "NP"), but the really hard ones are *only* in "NP", and they are called "NP-complete".
- It is like saying there are things that People can do ("P"), there are things that Super People can do ("SP"), and there are things only Super People can do ("SP-complete").

After study carefully the security of the honeyword system and introducing a number of defect that need to be fitted with before successful realization of the scheme. In this respect, Pointing out that the strong point of the honeyword system directly depends on the generation algorithm Finally, Here 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. Here presenting a standard approach to securing personal and business data in the system.

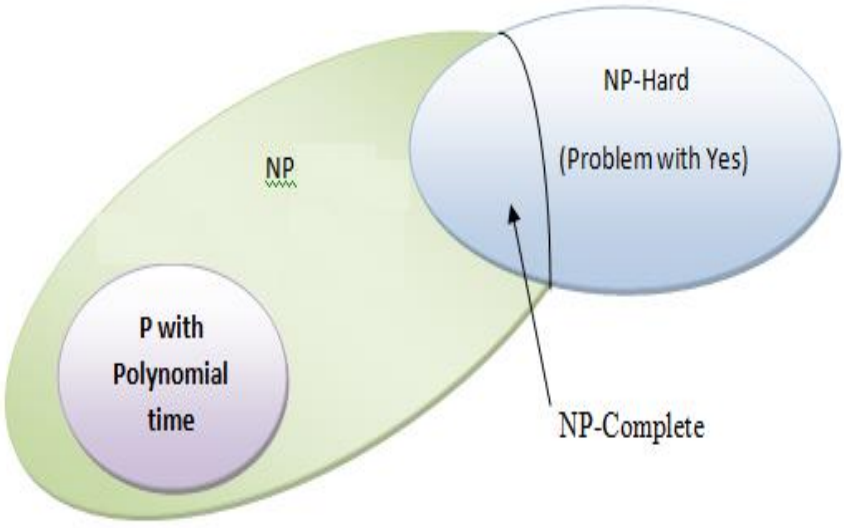

Figure 7.4: NP-Complete Problem

7.5 Benefits of system

- Enhancing Security in user authentication through honeyword.
- For each user set of honeyword is generated
- Achieving Flatness
- Providing privacy and security to user account.
- Proposed system identifies password file disclosure before getting harm to the system.
- Hybrid method which combines strength of others method is used for generation of honeywords.

7.6 Advantages

- Honeywords provide high security
- Honeywords confuses hackers
- Easy to use

Chapter 8

GUI

8.1 Welcome Screen

Figure 8.1: welcome Screen

8.2 User Registration

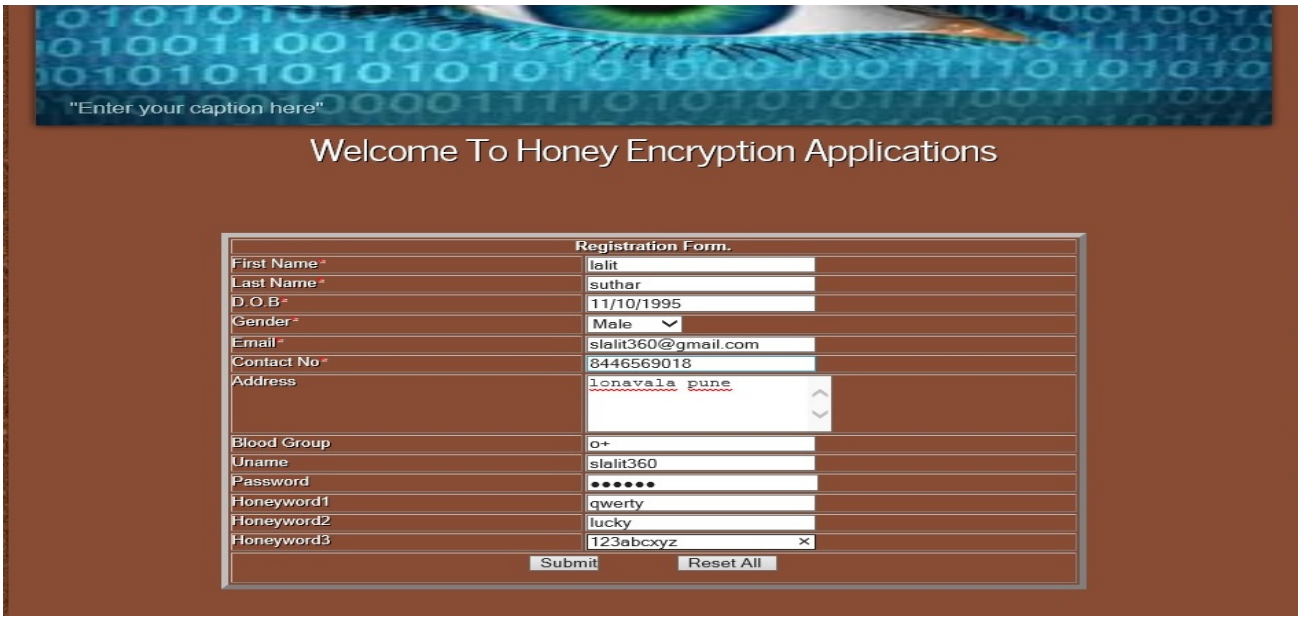

Figure 8.2: User Registration

8.3 User Registered successfully

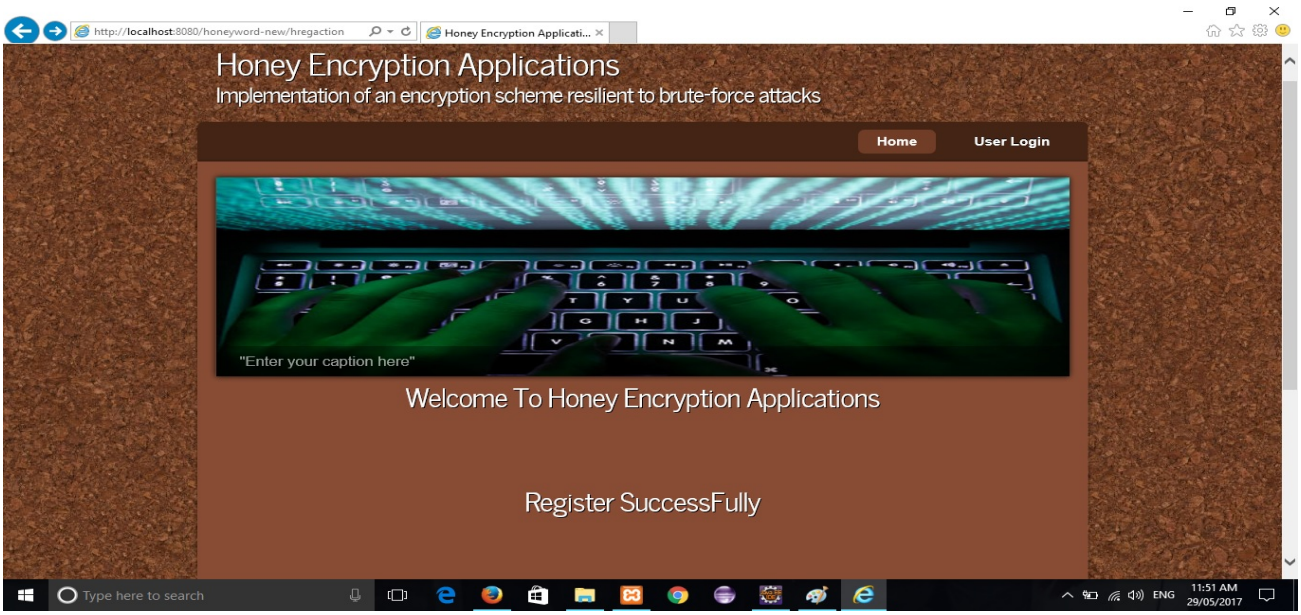

Figure 8.3: User Registered successfully

8.4 User Login

Figure 8.4: User Login

8.5 User Login successfully

Figure 8.5: User Login successfully

8.6 User Login (Tried with Honeyword)

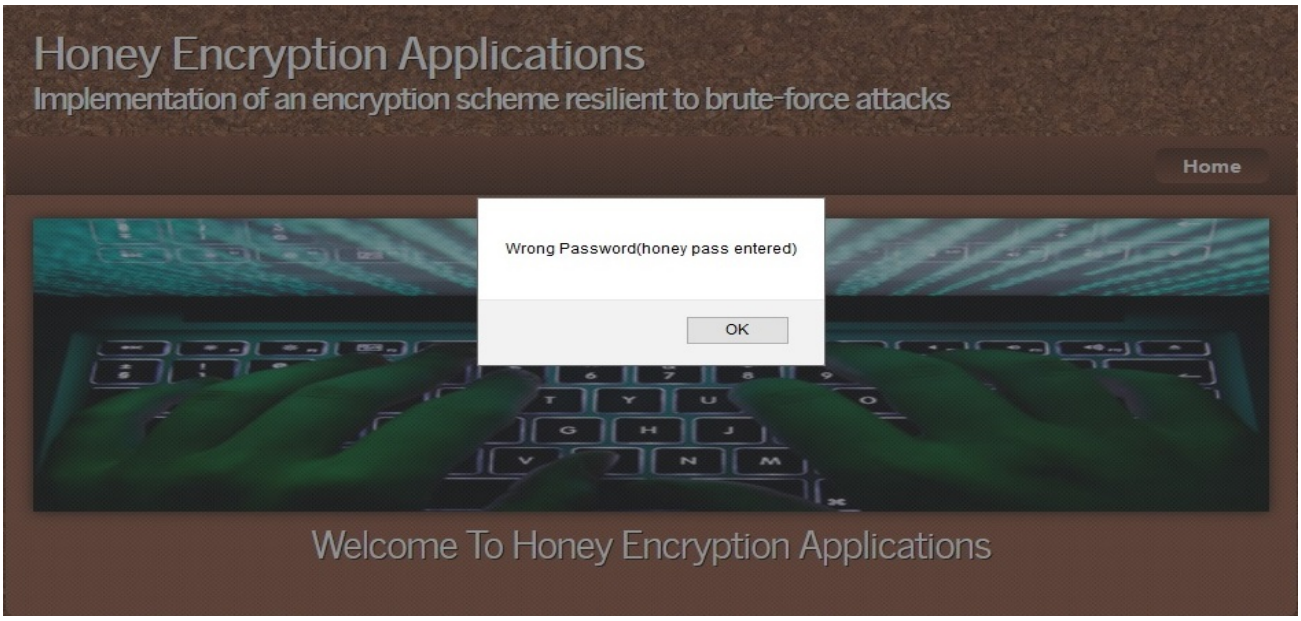

Figure 8.6: User Login (Tried with Honeyword)

8.7 User Login (Entered Wrong Password)

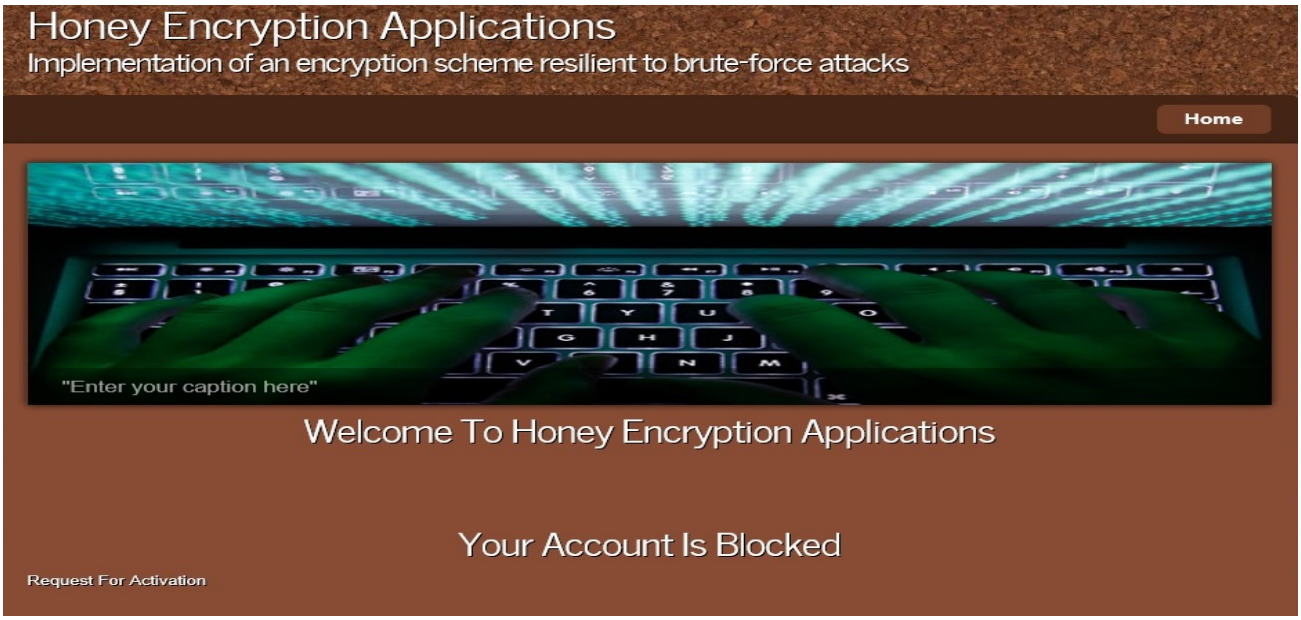

Figure 8.7: User Login (Entered Wrong Password)

8.8 User Login (Account Recovery Request)

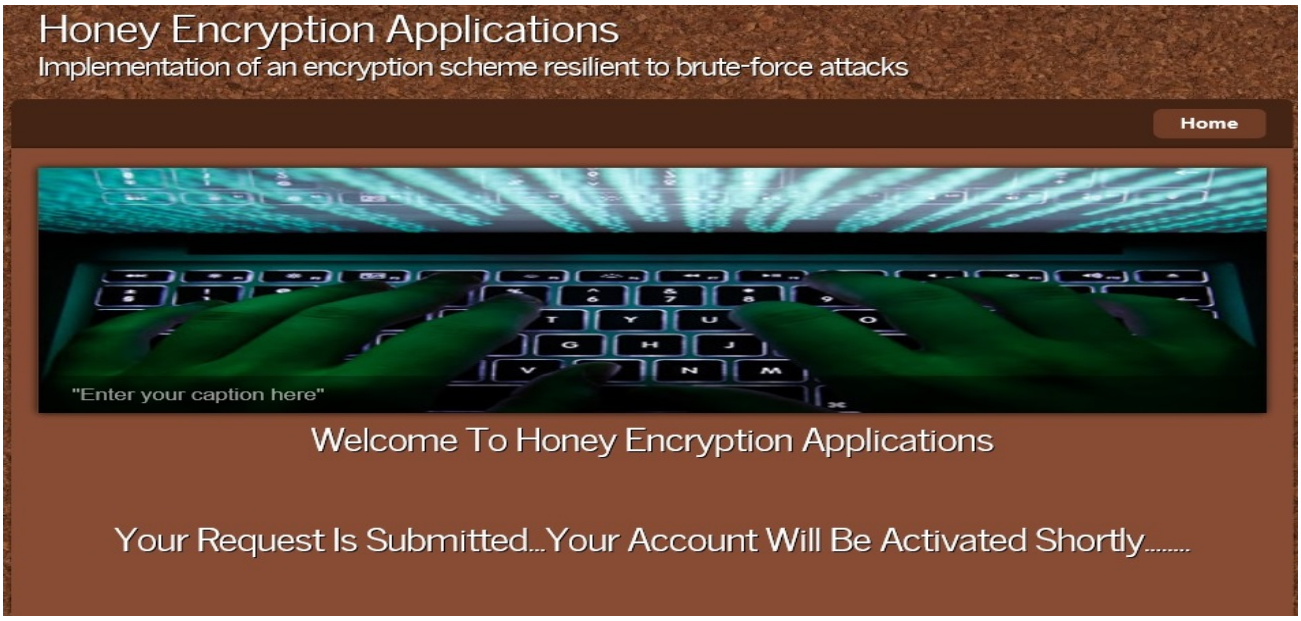

Figure 8.8: User Login (Account Recovery Request)

8.9 Admin Login

Figure 8.9: Admin Login

8.10 Admin Dashboard

Figure 8.10: Admin Dashboard

8.11 Admin Dashboard (Check Honeyword)

Figure 8.11: Admin Dashboard (Check Honeyword)

8.12 Admin Dashboard (Activation Request)

Figure 8.12: Admin Dashboard (Activation Request)

8.13 Hacker Login

Figure 8.13: Hacker Login

8.14 Hacker (Attempt to hack)

Figure 8.14: Hacker (Attempt to hack)

8.15 Honey-Tracker Login

Figure 8.15: Honey-Tracker Login

8.16 Honey-Tracker (login Reports)

Figure 8.16: Honey-Tracker (login Reports)

8.17 Email

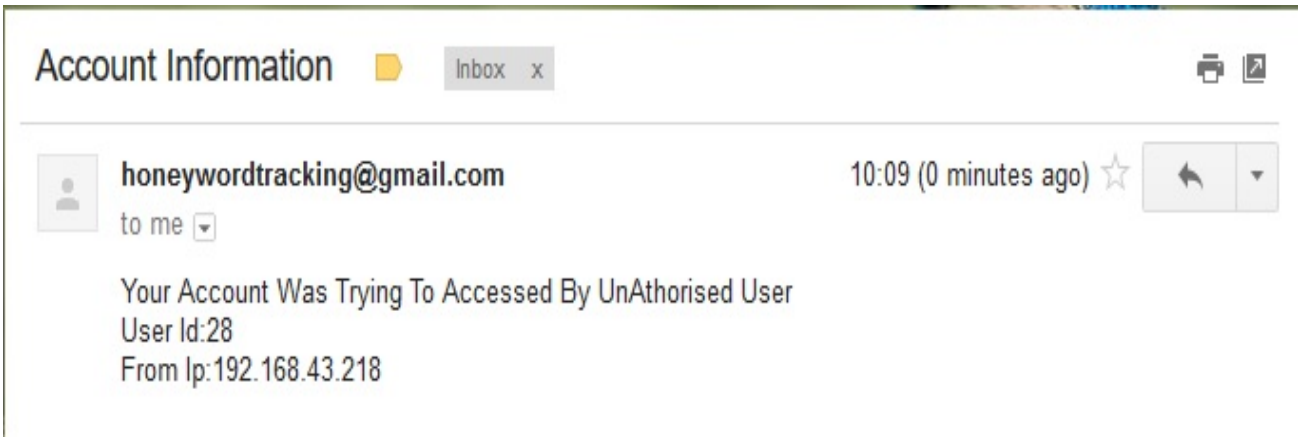

Figure 8.17: Alert Email to User and Admin

8.18 Database Schema/Tables

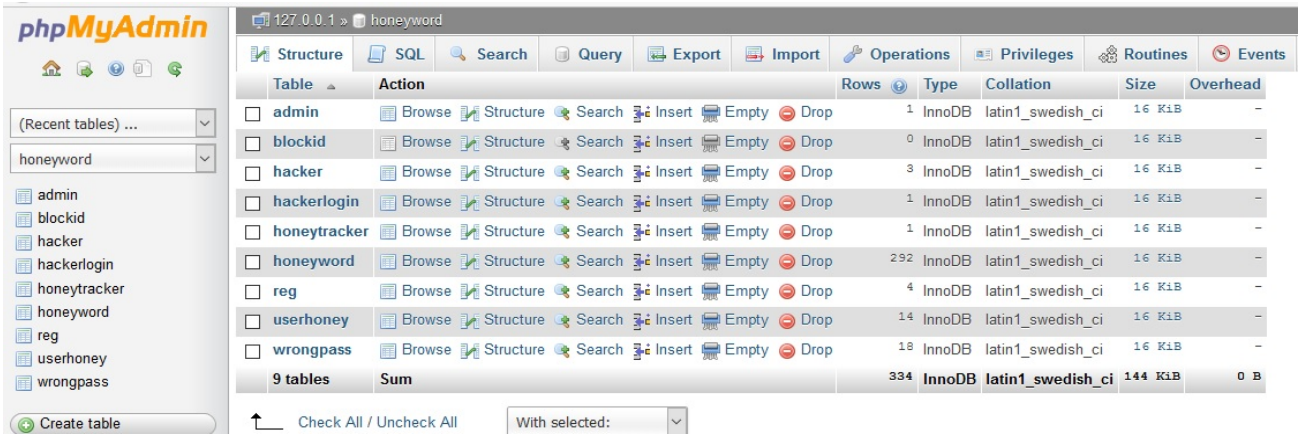

Figure 8.18: Database Schema/Tables

Chapter 9

System Testing

9.1 Purpose

Testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs. Software testing can also be stated as the process of validating and verifying that a software program or application or product :

1. Meets the business and technical requirements that guided its design and development.
2. Works as expected.
3. Can be implemented with the same characteristics.

9.2 Scope of Testing

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the software development methodology adopted. Different software development models will focus the test effort at different points in the development process. Newer development models, such as Agile, often employ test driven development and place an increased portion of the testing in the hands of the developer, before it reaches a formal team of testers. In a more traditional model, most of the test execution occurs after the requirements have been defined and the coding process has been completed.

Software testing methods are traditionally divided into white-box and black-box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

9.2.1 White-box testing

In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases.

9.2.2 Black-box testing

Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation. The testers are only aware of what the software is supposed to do, not how it does it.

9.2.3 Grey-box testing

Grey-box testing involves having knowledge of internal data structures and algorithms for purposes of designing tests, while executing those tests at the user, or black-box level. The tester is not required to have full access to the software's source code.

9.3 Test Plan

To test the application following the proper sequencing of testing like unit, integration, validation, GUI, Low level and High level test cases, major scenarios likewise. Performing with the GUI testing first and then integration testing. After integration testing performs the high level test cases and major scenarios which can affect the working on the application. Performing the testing on the data transmitted using the various inputs and outputs and validate the results. It also intends to cover any deviations that the project might take from the initially agreed Test Strategy in terms of scope, testing methodology, tools, etc.. This test plan covers details of testing activities for this project and scope.

9.4 Test Cases and Test Results

Test Case Title	**Honeyword generation**
Objective	System will generate the honeywords using some methods such as chaffing by tweaking, chaffing with password model, chaffing with tough nuts.
Expected Result	Successful.

Table 9.1: Test Case - Honeyword generation

Test Case Title	**Email Alert**
Objective	System will send Email alert for malicious login attempt.
Expected Result	User will be notified by mail Received .

Table 9.2: Test Case - Email Alert

Test Case Title	**User Blocked request**
Objective	System will provide interface for activation request for blocked user.
Expected Result	Request Received By Admin.

Table 9.3: Test Case - User Blocked request

Test Case Title	**Notification**
Objective	System will notify user for malicious login attempt.
Expected Result	Successful.

Table 9.4: Test Case - Notification

Test Case Title	**User Account Activation**
Objective	Admin will Reactivate user account after Verifying user identity.
Expected Result	User will able to login again with Right Credential.

Table 9.5: Test Case - User Account Activation

Test Case Title	**Honeyword Checker**
Objective	System will show generated honeyword of specific user to admin.
Expected Result	Successfully Honeyword Found.

Table 9.6: Test Case - Honeyword Checker

Test Case Title	**Hacker Activity**
Objective	System must show false password of users when tried by hacker with Bits keys.
Expected Result	Fake password Received by Hacker.

Table 9.7: Test Case - Hacker Activity

Test Case Title	**Honeyword Tracking**
Objective	System will display logs of user's honeywords and sugarword.
Expected Result	True logs and activity performed by user received.

Table 9.8: Test Case - Honeyword Tracking

Note: Testing was performed manually

9.5 GUI Testing

Graphical User Interface (GUI) testing is the one of the mechanism in which user interface developed System Under some graphical rules. GUI testing includes checking various controls- menus, buttons, icons, dialog boxes and windows etc. Proposed system is tested for user inputs against different modules, validations are done. GUI is tested for appearance of different controls, visibility graphs is tested. GUI testing involves following actions:

1. Check all elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.
2. Overall functionality related with performance of users graphical interface are checked.
3. Check Error Messages are displayed correctly.

4. Check the font, layout details, style and display warning messages if it is false.
5. Check the positioning of GUI elements.

Test Case Title	**User Registration**
Objective	System will Provide registration form for individual user.
Expected Result	if already registered it will prompt registration failed(User already Exist) else Registered Successful.

Table 9.9: Test Case - User Registration

Test Case Title	**Login Screen - Sign up**
Objective	Click on sign up button then check all required/ mandatory fields with leaving all fields blank.
Expected Result	All mandatory fields should display with symbol "*". Instruction line "* field(s) are mandatory" should be displayed .

Table 9.10: Test Case - Login Screen- Sign up

Test Case Title	**Create a Password - Text Box** **Confirm Password - Text Box**
Objective	Check the validation message for Password and Confirm Password field.
Expected Result	Correct validation message should be displayed accordingly or "Password and confirm password should be same" in place of "Password mismatch".

Table 9.11: Test Case - Text Box

Chapter 10

Conclusion

Carefully study of security factors of the honeywords system that introduce a number of defect that need to be fitted with before successful realization of the scheme. In this respect, Pointing out that the strong point of the honeywords system directly depends on the generation algorithm finally, Here presenting a new approach to make the generation algorithm as close as to human nature by generating honeywords with randomly selecting passwords that belong to other users in the system. Here presenting a standard approach to securing personal and business data in the system. Decoy documents stored in the system along with 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, Here barraging of malicious insider with fake information in order to dilute or divert the user' s real data. Such preventive attacks that rely on dis-information technology could provide unusual levels of security in the system and in social networks model. In the future, Refinement of model by involving hybrid generation algorithms to also make the total hash inversion process difficult for an adversary in getting the passwords in plaintext form a leaked password hash file. Hence, by developing such methods both of two security goal – 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.* 13.2 (2016): 284-295.

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

[3] C. Herley and D. Florencio. "Protecting financial institutions from brute-force attacks." *Microsoft Research One Microsoft Way Redmond*, WA, 2008.

[4] H. Bojinov, E. Bursztein, X. Boyen, and D. Boneh . "Kamouflage: Loss-resistant Password Management." *Computer Security - ESORICS 2008.* Springer, 2010, pp. 286-302.

[5] 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. IEEE, 2009, pp.391-405.

[6] M. H. Almeshekah , E. H. Spafford, and M. J. Atallah. "Improving Security using Deception, 2013." *Center for Education and Research Information Assurance and Security, Purdue University, Tech.* Rep. CERIAS Tech Report 2013-13, 2013.

[7] The Dangers of Weak Hashes. "If Your Password is 123456, Just Make It Hack me." *SANS Institute InfoSec Reading Room, Tech.* Rep., 2013.

[8] D. Mirante and C. Justin. "Understanding password Database Compromises." *Department of Computer Science and Engineering Polytechnic Institute of NYU*, 2013.

[9] Genc, Ziya Alper, Süleyman Kardas, and Mehmet Sabir Kiraz. "Examination of a New Defense Mechanism: Honeywords." *IACR Cryptology ePrint Archive 2013* (2013): 696.

[10] Malone, David, and Kevin Maher. "Investigating the distribution of password choices." *Proceedings of the 21st international conference on World Wide Web.* ACM, 2012.

[11] Almeshekah, Mohammed H, Eugene H. Spafford, and Mikhail J. Atallah. "Improving security using deception." *Center for Education and Research Information Assurance and Security*, Purdue University, Tech. Rep. CERIAS Tech Report 13 (2013).

[12] Juels, Ari, and Ronald L. Rivest. "Honeywords: Making password-cracking detectable." *Proceedings of the 2013 ACM SIGSAC conference on Computer and communications security.* ACM, 2013.

[13] Weir Matt et al. "Password cracking using probabilistic context-free grammars." *Security and Privacy, 2009.* 30th IEEE Symposium on. IEEE, 2009.

Appendix A

Published Papers

Vol-2 Issue-6 2016 *IJARIIE-ISSN(O)-2395-4396*

IMPROVING SECURITY AND ACHIEVING FLATNESS TO THE USER-PASSWORD BY USING HONEYWORDS

Lalit Hajarimal Suthar[1], Vimal Chetan Singh[2], Mahesh Arun Umale[3], Hemalata Manohar Khandagle[4]

[1] Student, Department of Information Technology,
SKN Sinhgad Institute of Science & Technology, Lonavala, Maharashtra, India
[2] Student, Department of Information Technology,
SKN Sinhgad Institute of Science & Technology, Lonavala, Maharashtra, India
[3] Student, Department of Information Technology,
SKN Sinhgad Institute of Science & Technology, Lonavala, Maharashtra, India
[4] Student, Department of Information Technology,
SKN Sinhgad Institute of Science & Technology, Lonavala, Maharashtra, India

ABSTRACT

Authentication and Authorization of user is important to get access into any secured system for which username and password is necessary. Disclosure of password files is a serious security Issue, for which Juels and Rivest proposed honeywords (decoy passwords) to detect attacks against Hash password databases. To protect hash password databases from third party, we implement Honeywords which converts a valid password into a new hash passwords. New passwords is the combination of existing password called honeywords, which are nothing but the Fake passwords. If proper honeywords are chosen, a cyber-attacker who steals a file of hashed passwords would be confused, whether it is a real password or honeyword for user Account. Login with honeywords will trigger an Alarm, which will notify the administrator about a password file Infraction. To identify or detect occurrence of password database infraction honeypot concept is used. So we implement an easy and efficient solution which will protect and detect exposure events of passwords file.

Keyword: - *Authentication, Authorization, Infraction, honeywords, login, honeypot.*

1. INTRODUCTION

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. 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. In this method, the administrator purposely creates fake user accounts to lure adversaries and detects a password leakage, if any one of the honeypot passwords get used. This idea has been modified by Herley and Florencio to protect online banking accounts from password bruteforce attacks. According to the study, for each user incorrect login try with some passwords lead to honeypot accounts, i.e., malicious behavior is recognized. For instance, there are 108 possibilities for an eight-digit password and let system links 10,000 wrong password to honeypot accounts, so the adversary performing the bruteforce attack 10,000 times more likely to access a honeypot account than the genuine account. Use of decoys for building theft-resistant was introduced by Bojinov et al. in called as Kamouflage. In this model, the fake password sets are stored with the real user password set to mask the real passwords, thereby forcing an adversary to carry out a considerable amount of work before getting the Right information. Recently, Juels and Rivest have presented the honeyword mechanism to detect an adversary who attempts to login with cracked passwords. Basically, for each username a set of sweet-words is generated such that only one element is the correct password and the others are honeywords (decoy passwords). Hence, when an adversary tries to enter into the system with a honeyword, an alarm is triggered to notify the administrator about a password leakage.

In this study, we analyze the honeyword approach and give some remarks about the security of the system. Furthermore, we point out that the key term for this method is the generation algorithm of the honeywords such that they shall be identical from the correct passwords.

2. TECHNICAL DESCRIPTION

2.1 Literature Survey

2.1.1 Examination of a New Defence Mechanism: Honeywords

Authors: Ziya Alper Genc, Suleyman Kardas, Mehmet Sabir Kiraz
Description: It has become much easier to crack a password hash with the improvement in the graphical processing unit (GPU) technology. An attacker can recover a user's password using brute-force attack on password hash. Once the password has been recovered no server can detect any invalid user authentication. In this context, Juels and Rivest published a paper for improving the security of hashed passwords. They propose a method for user authentication, in which some fake passwords, i.e., "honeywords" are added into a password file, in order to detect enactment their solution includes a secure server called "honeychecker" which can differentiate a user's real password among her honeywords and quickly sets off an alarm whenever a honeyword is used. In this paper, we analyse the security of the proposal, provide some possible improvements which are easy to implement and introduce an improved model as a solution to an open problem.

2.1.2 Investigating the Distribution of Password Choices

Authors: David Malone, Kevin Maher NUI Maynooth
Description: 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 different 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 Zipf's Law produces good evaluation of these statistics. We then look at the analogy of the password distributions from each of our sources, using geuss as a metric. 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 different password.

2.1.3 Improving Security Using Deception

Authors: Mohammed Alme shekah, Eugene H. Spafford, Mikhail J. Atallah
Description: 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 different 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. They posit how the well-known Kerckhoffs's principle has been misread to drive the security community away from deception based mechanisms. They examine advantages of these techniques can have when securing our information in addition to traditional methods of hiding. They show that by intelligently introducing ambiguous information in information systems, we not only lead attackers awry, but also give organizations the ability to detect leakage of data; create doubt and uncertainty in any leaked data; add risk at the adversaries' side to using the leaked information; and significantly enhance our abilities to attribute attackers. They discuss how to overcome some of the challenges that abstract the adoption of deception-based techniques

2.1.4 Honeywords: Making Password-Cracking Detectable

Authors: Ari Juels, Ronald L. Rivest
Description: 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 differentiate the user password from honeywords for the login, and will set off an alarm if a honeyword is submitted.

2.1.5 Password Cracking Using Probabilistic Context-Free Grammars

Authors: Matt Weir, Sudhir Aggarwal, Breno de Medeiros, Bill Glodek.
Description: Selecting the most effective word-distorting rules to use when performing a dictionary-based 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 effective way to crack passwords as compared to traditional techniques by testing our tools and techniques on real password sets.

2.2 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.

2.3 Proposed System

In this study, we focus on the security issue and deal with decoy passwords or accounts as a simple and cost effective solution to detect compromise of passwords. Honeypot is one of the methods to identify occurrence of a password database breach. In this approach, the administrator purposely creates fake user accounts to lure adversaries and

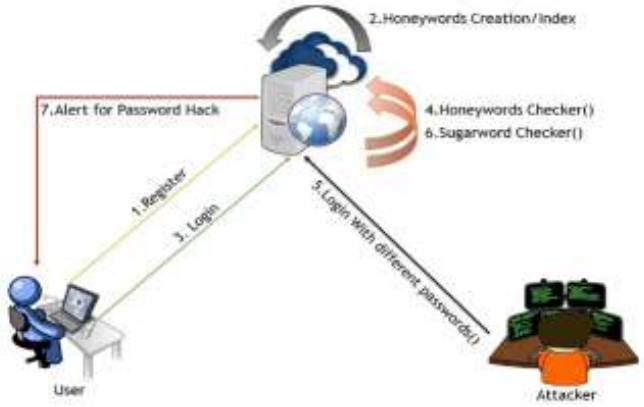

Fig -1: Working of System

Detects a password disclosure, if any one of the honeypot passwords get used. In this paper we have proposed a novel honeyword generation approach which decreases the storage overhead and also it addresses majority of the drawbacks of existing honeyword generation techniques. Proposed model is based on use of honey words to detect password-cracking. We propose to use indexes that map to valid passwords in the system. The contribution of our approach is two-fold. First, this method requires less storage compared to the original study. Within our approach passwords of other users are used as the decoy passwords, so prediction of which password is fake and which is correct becomes more complicated for an adversary.

2.4 Implementation

2.4.1 Introduction

In this project, we separate the honeyword approach and give some notice about the security of the system. We point out that the key for this method is the generation algorithm of the honeywords such that they shall be in-distinguishable from the correct passwords. Therefore, we propose a new method that created the Honeywords using the existing user passwords combination in hash format.

2.4.2 Implementation Details

Algorithm:
Inputs:

1. T fake user accounts (honey pots)
2. index value between [1;N]
3. index list ,which is not previously assign to user

Procedure:
Step 1: Honey pots creation: fake user account

a. 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
b. Create two password file file f1 and file 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)>

Step 2: Generation of honey-index set

In Step 1 we insert honey index set in file F1 but don't know how to create that We use honey index generator algorithm Generator(k; SI) →Ci;Xi

Generate Xi.

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

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, notifies system a honey word situation.

Step 4: Encryption

- 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).
- 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.
- 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.
- 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.

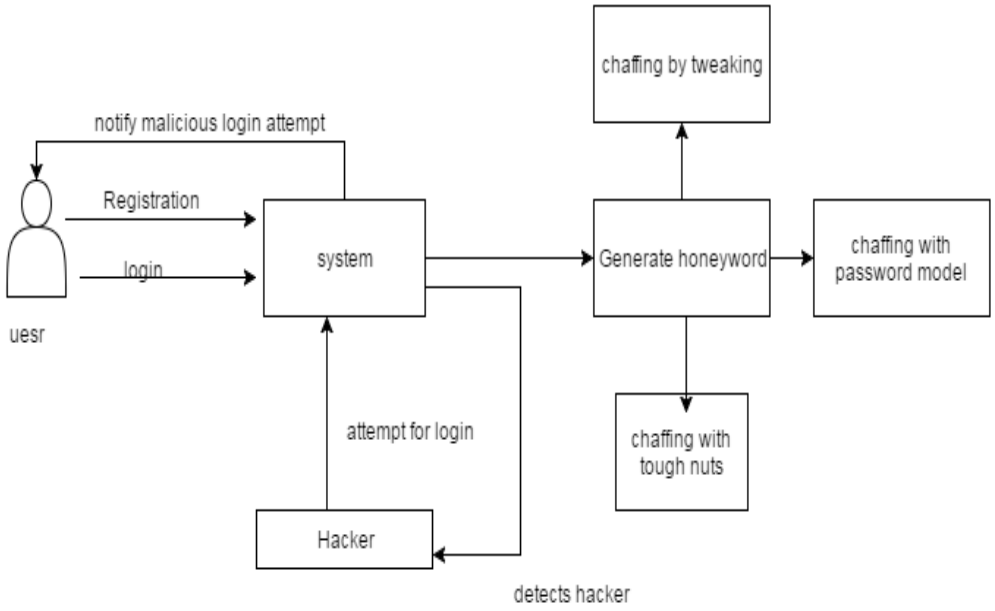

Fig -2: System Architecture

3. Honeyword Generation Methods

3.1 *Chaffing-by-Tweaking*

In this method, the generator algorithm which twist selected character positions of the real Password to produce the honeywords. Each character of a user password in pre-determined positions is replaced by a randomly selected character of the same type: digits by digits, letters are replaced by letters, and special characters by special characters. Number of positions to be twisted, denoted as t should depend on sys policy. As
An example t ¼ 3 and tweaking last t characters may be a method for the generator algorithm Gen(k,t). Another Approach named in the study as "chaffing-by-tweaking-digits" is executed by tweaking the last t positions that contain digits. For example, by using the last technique for the password 42hungry and t ¼ 2, the honeywords 12hungry and 58hungry may be generated.

3.2 *Chaffing-with-a-Password-Model*

In this approach, the generator algorithm takes the password from the user and relying on a probabilistic model of correct passwords it generates the honeywords. In this model, the password is spitted into character sets. For instance, mice3blind is decomposed as four-letters + one-digit + five-letters) L4 þ D1 þ L5 and replaced with the same composition like gold5rings.

3.3 *Chaffing with "Tough Nuts"*

In this method, the system intentionally injects some special honeywords, named as tough nuts, such that inverting hash values of those words is computationally inefficient, e.g. fixed length random bit strings should be set as the hash value of a honeyword. An example for a tough nut is given in as '9,50PEe [KV.0? RIOtcL-:IJ"b+Wol !*]!NWT/pb'. It is stated that the number and positions of tough nuts are selected randomly. By means of this, it is expected that the attacker cannot seize whole sweet-word set and some sweet-words will be blank for her, thereby deterring the adversary to realize her attack. It is discussed that in such a situation the adversary may pause before attempting login with cracked passwords.

3.4 Comparison of Honeyword Generation Models

Method	**DoS Resistance**	**Flatness**	**Storage Cost**
Tweaking	*Weak*	*Weak*	hN*
Password-model	*Strong*	*Strong+*	khN
Our model	*Strong*	*Strong+*	4kN+hN+4N

Table -1: comparison of Honeyword generation Method

4. CONCLUSIONS

We have study carefully the security factors of the honeyword system and introduce a number of defect that need to be fitted with before successful realization of the scheme. In this respect, we have pointed out that the strong point of the honeyword system directly depends on the generation algorithm finally, we have presented a new approach to make the generation algorithm as close as to human nature by generating honeywords with randomly selecting passwords that belong to other users in the system. We present a standard approach to securing personal and business data in the system. We propose monitoring data access patterns by marking user behaviour to determine if and when a malicious insider illegally accesses someone's documents in a system service. Decoy documents stored in the system along with 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 barrage the malicious insider with fake information in order to dilute or divert the user' s real data. Such preventive attacks that rely on dis-information technology could provide unusual 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 difficult for an adversary in getting the passwords in plaintext form a leaked password hash file. Hence, by developing such methods both of two security goal – increasing the total effort in recovering plaintext passwords from the hashed lists and detecting the password disclosure – can be provided at the same time.

5. ACKNOWLEDGEMENT

The authors would like to thanks Prof. Anand Bone who, guided us through the paper work and such a great Motivation. The authors would also like to thanks Prof. Ganesh Kadam who provided all related help while working on this task.

6. REFERENCES

[1]. D. Mirante & C. Justin, "Understanding passwords databases compromises," Dept. of Computer. Sci. Eng. Polytechnic Inst. of NYU, New York, NY, USA: Tech. Rep. TR-CSE-2013-02, 2013.
[2]. D. Mirante & C. Justin, "Understanding passwords databases compromises," Dept. of Computer. Sci. Eng. Polytechnic Inst. of NYU, New York, NY, USA: Tech. Rep. TR-CSE-2013-02, 2013.
[3]. K. Brown, "The dangers of weak hashes," SANS Institute InfoSec Reading Room, Maryland US, pp. 1–22, Nov. 2013,[Online]. Available: http://www.sans.org/reading-room/ whitepapers/authentication/dangers-weak-hashes-34412
[4]. M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek, "Password cracking using probabilistic context-free grammars," in 30th IEEE Symp. Security Privacy, 2009, pp. 391–405.
[5]. F. Cohen, "The use of deception techniques: Honeypots and decoys," Handbook Inform. Security, vol. 3, pp. 646–655, 2006.
[6]. M. H. Almeshekah, E. H. Spafford, and M. J. Atallah, "Improving security using deception," Center for Education and Research Information Assurance and Security, Purdue Univ., West Lafayette, IN, USA: Tech. Rep. CERIAS Tech. Rep. 2013-13, 2013.

[7]. C. Herley and D. Florencio, "Protecting financial institutions from brute-force attacks," in Proc. 23rd Int. Inform. Security Conf., 2008, pp. 681–685.
[8]. A. Juels and R. L. Rivest, "Honeywords: Making password cracking detectable," in Proc. ACM SIGSAC Conf. Comput. Commun. Security, 2013, pp. 145–160.
[9]. J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords," in Proc. IEEE Symp. Security Privacy, 2012, pp. 538–552.
[10]. D. Malone and K. Maher Investigating the distribution of password choices. in Proc. 21st Int. Conf. World Wide Web, 2012,pp. 301–310.
[11]. M. Burnett. The pathetic reality of adobe password hints. [Online]. Available: https://xato.net/windows-security/adobe-passwordhints, 2013.
[12]. H. Bojinov, E. Bursztein, X. Boyen, and D. Boneh, "Kamouflage: Loss-resistant password management," in Proc. 15th Eur. Conf. Res. Comput. Security, 2010, pp. 286–302.

BIOGRAPHIES

	Lalit Hajarimal Suthar slalit360@gmail.com Bachelor of Engineering In Information Technology SKN Sinhgad Institute of Science & Technology, Lonavala, 410401
	Vimal Chetan Singh *vimalsinghkashyapp@gmail.com* Bachelor of Engineering In Information Technology SKN Sinhgad Institute of Science & Technology, Lonavala, 410401
	Mahesh Arun Umale *mahesharunumale@gmail.com* Bachelor of Engineering In Information Technology SKN Sinhgad Institute of Science & Technology, Lonavala, 410401
	Hemalata Manohar Khandagle *hemlata7509@email.com* Bachelor of Engineering In Information Technology SKN Sinhgad Institute of Science & Technology, Lonavala, 410401

Improving Security and Achieving Flatness to the User-Password by using Honeywords

Lalit Hajarimal Suthar[1], Vimal Chetan Singh[2], Mahesh Arun Umale[3], Hemalata Manohar Khandagle[4]
Department of Information Technology,
SKNSITS, Lonavala

Abstract --- The new developments in the field of information technology offered the users enjoyment, comforts and convenience, but there are many security thread related issues. One of them is Password file. Password files have found a lot of security problem that has affected billions of users as well as many companies/Industries. Password file is generally stored in encrypted format like Hash file, if a password file is stolen or theft by using the password cracking techniques and decryption technique it is easy to get most of the plaintext and encrypt passwords. For which Juels and Rivest proposed Honeywords (decoy passwords) to detect attacks against Hash password databases. For this here we create the honeyword password, i.e. a decoy password using a perfectly flat honeyword generation method, and try to attract adversary/illegal or unauthorized user. Hence that time server detects the unauthorized user and Alert the User and Administrator. Here we also protect the original data from unauthorized user.

Keywords **--- Authentication*, Confidentiality, Impersonation, Infraction, Honeywords, Login, Sweetword, Sugarword, Cyberpunk, Honeypot.***

1. INTRODUCTION

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 encryptions were using the SHA-1 algorithm without a salt and similarly the passwords in the eHarmony system were also stored using unsalted MD5 hashes. 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 solutions to identify occurrence of a password database breach. In this method, the administrator purposely creates fake user accounts to lure adversaries and detects a password leakage, if any one of the honeypot passwords gets used. This idea has been modified by Herley and Florencio to protect online banking accounts from password brute-force attacks. According to the study, for each user incorrect login try with some passwords lead to honeypot accounts, i.e., malicious behavior is recognized. For instance, there are 108 possibilities for an eight-digit password and let system links 10,000 wrong password to honeypot accounts, so the adversary performing the brute-force attack 10,000 times more likely to access a honeypot account than the genuine account. Use of decoys for building theft-resistant was introduced by Bojinov et al. in called as Kamouflage. In this model, the fake password sets are stored with the real user password set to mask the real passwords, thereby forcing an adversary to carry out a considerable amount of work before getting the Right information. Recently, Juels and Rivest have presented the honeyword mechanism to detect an adversary who attempts to login with cracked passwords. Basically, for each username a set of sweet-words is generated such that only one element is the correct password and the others are honeywords (decoy passwords). Hence, when an adversary tries to enter into the system with a honeyword, an alarm is triggered to notify the administrator about a password leakage.

In this study, we analyze the honeyword approach and give some remarks about the security of the system. Furthermore, we point out that the key term for this method is the generation algorithm of the honeywords such that they shall be identical from the correct passwords.

2. RELATED WORK

2.1 The Use of Deception Techniques: Honeypots and Decoys (2006). [2]

Deception techniques have the demonstrated ability to increase attacker workload and reduce attacker effectiveness. The most critical work that must be done in order to make progress is the systematic study of the

effectiveness of deception techniques against combined systems with people and computers. Modern defensive computer deceptions are in their infancy, but they are moderately effective, even in this simplistic state. This article has summarized a great deal of information on the history of honeypots and decoys for use in defense of computer systems.

2.2 Protecting financial institutions from Brute-force attacks (2008). [3]

This show that is simple to ensure that a brute-force attacker will encounter hundreds or even thousands of honeypot accounts for every real break-in. activity in the honeypots provides the data by which the bank learns the attacker attempts to tell real from honeypot accounts, and his cash out strategy. Examine the problem of protecting online banking accounts from password brute-forcing attacks. In a brute-force attack repeated credential pairs are tried in an attempt to gain access to an account. The simplest is directed against a single account: the attacker tries all possible passwords for one user ID until one succeeds.

2.3 Kamouflage: Loss resistant Password Management (2008). [4]

Introduce Kamouflage: new architecture for building the resistant password managers. An attacker who steals a laptop or cell phone with a Kamouflage based password manager is forced to carry out a considerable amount of online work before obtaining any user credentials. Replacement for the built-in Firefox password manager, and provide performance measurements and the results from experiments with large real-world password sets to evaluate the feasibility and effectiveness of our approach. This presented a system to secure the password database on a mobile device from attacks that are often ignored by deployed password managers. Kamouflage is well suited to become a standard architecture for password managers.

2.4 Password Cracking Using Probabilistic Context-Free Grammars (2009). [5]

This grammar allows us to generate word-mangling rules, and from them, password guesses to be used in password cracking. This approach seems to provide a more effective way to crack passwords as compared to traditional methods. Approach was able to crack 28%. Our approach was able to crack 28% to 129% more passwords than John the Ripper, a publicly available standard password cracking program.

2.5 Improving Security using Deception (2013). [6]

We explore complex relationships among protection techniques ranging from denial and isolation, to degradation and obfuscation, through negative information and deception, ending with adversary attribution and counter-operations. This outlined a new classification scheme for deception techniques in cyber security. have shown how some of these techniques have been known and used for many years, but that the field is under-developed. This explained how systems can be augmented to use deception and false information to protect them and their data, to degrade attacks, to expose attackers, to enhance attribution, and possibly to be used to damage or degrade attacker capabilities.

2.6 Understanding password Database Compromises (2013). [8]

It forces the attacker to brute force the hashes one at a time, instead of attacking them as a group. Offering the benefit of flexibility, with the ability to provide resources almost instantaneously as necessary to avoid site shutdown. In High profile website intrusions, wherein user login through credentials and other data were compromised. A study was undertaken to research information posted on the web concerning recent, high profile website intrusions.

3. PROBLEM DEFINITION

Suggest an alternative approach that selects the honeywords from existing user passwords in order to provide realistic honeyword – a perfectly flat honeyword generation method.

4. EXISTING SYSTEM

In previous system, Discloser of password files is a severe problem that has affected millions of users and companies like Yahoo, RockYou, LinkedIn, eHarmony and Adobe since leaked passwords make the users target of many possible cyber-attacks. These recent events have demonstrated that the weak password storage methods are currently in place on many web sites.

5. PROPOSED SYSTEM

Here we focus on the security issue and deal with fake passwords or accounts as a simple and cost effective solution to detect compromise of passwords. Honeypot is one of the methods to identify occurrence of a password database breach. In this approach, the administrator purposely creates deceit user accounts to lure adversaries and detects a password disclosure, if any one of the honeypot passwords get used. In this paper we have proposed a novel honeyword generation approach which reduces the storage overhead and also it addresses majority of the drawbacks of existing honeyword generation techniques. Proposed model is based on use of honey words to detect password-cracking. We propose to use indexes that map to valid passwords in the system. The contribution of our approach is twofold. First, this method requires less storage compared to the original study.

Within our approach passwords of other users are used as the fake passwords, so guess of which password is fake and which is correct becomes more complicated for an adversary.

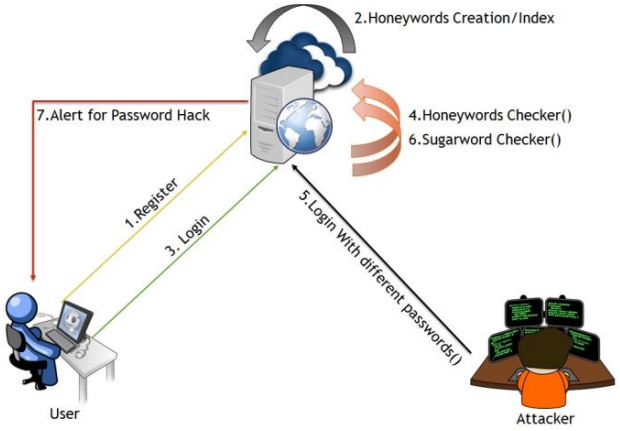

Fig 5.1. Working of Proposed System.

6. IMPLEMENTATION

6.1. Algorithm

In the proposed system, in order to generate honeyword in different areas following algorithm is implemented. With the help of this algorithm, system can generate decoy password and provide alert system with unique honeyword and key-bit mapping technique.

Inputs:

1. T fake user accounts (honey pots).
2. Index value between [1; N].
3. Index list, which is not previously assign to user.

Procedure:

Step 1: Honeypot creation: fake user account 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.
Create two password file f1 and file f2.
F1 Store username and honey index set < Hui, xi> where Hui is honeypot account
F2 keeps the index number and the corresponding hash of the password (create the hash of the password),
< Ci; H (pi) >

Step 2: Generation of honey index set
In Step 1 we insert honey index set in file F1 but don't know how to create that we use honey index generator algorithm
Gen (k; SI) →ci; Xi
Generate Xi

a. Select xi randomly selecting k-1 numbers from SI and also randomly picking a number Ci SI.
b. Ui; Ci pair is delivered to the honey checker and F1, F2 files are updated.

Step 3: Honey checker
Set: Ci, Ui: Sets correct password index ci for the user Ui
Check: Ui, j: Checks whether Ci for Ui is equal to given j. Returns the result and if equality does not hold, notifies system a honeyword situation.

Step 4: Encryption

We have a user message (password) space M which contains all possible messages. We map these messages to a seed space S through the use of a distribution-transforming encoder (DTE). The seed space is simply 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. The size of the seed range of m is directly proportional to how probable 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 cumulative distribution function (CDF) of M and some information on the ordering of messages. Additionally, the seed space must be large enough so that even the message with smallest probability in the message space is assigned at least one seed. With this information, we can find the cumulative probability range corresponding the message m and map it to the same percentile seed range in S.

6.2. Mathematical model

Let,

S→{I,O}

Where, I→{Input}

I→{H, U, A, AD, P, E, K, HE}

Where, H→{Honeywords}

U→{Users}

A→{Attacker}

AD→{Admin}

P→{Password}

E→{Email}

K→{Key}

HE→{Honey encryption}

O →{Output}

Considering that we have database 'D' and 'n' number of attribute such as user name, user id etc. D→{A j A 2 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 enters 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 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.

GET (D, X, SERVER): Admin get all information about the user account from server.

PUT (X, ATK, SERVER): Here admin will upload attacker's information on server.

6.3 Honeyword Generation Methods

a. Chaffing-by-Tweaking

b. Chaffing-with-a-Password-Model

c. Chaffing with "Tough Nuts"

Method	DoS Resistance	Flatness	Storage Cost
Tweaking	*Weak*	*Weak*	hN*
Password-model	*Strong*	*Strong+*	khN
Our model	*Strong*	*Strong+*	4kN+hN+4N

Table 6.3.1 Comparison of Honeyword generation Method

6.4 Test Cases and Test Results

Test Case Title	Honeyword generation
Objective	System will generate the honeywords using some methods such as chaffing by tweaking, chaffing with password model, chaffing with tough nuts.
Expected Result	Successful.

Table 6.4.1 Test Case - Honeyword generation

Test Case Title	Notification
Objective	System will notify user for malicious login attempt.
Expected Result	Successful.

Table 6.4.2 Test Case – Notification

Test Case Title	Login Screen - Sign up
Objective	Click on sign up button then check all required/ mandatory fields with leaving all fields blank.
Expected Result	All mandatory fields should display with symbol "*". Instruction line "* field(s) are mandatory" should be displayed .

Table 6.4.3 Test Case - Login Screen- Sign up

Test Case Title	Create a Password - Text Box Confirm Password - Text Box
Objective	Check the validation message for Password and Confirm Password field.
Expected Result	Correct validation message should be displayed accordingly or "Password and confirm password should be same" in place of "Password mismatch".

Table 6.4.4 Test Case - Text Box

7. CONCLUSIONS

We have study carefully the security factors of the honeywords system and introduce a number of defect that need to be fitted with before successful realization of the scheme. In this respect, we have pointed out that the strong point of the honeywords system directly depends on the generation algorithm finally, we have presented a new approach to make the generation algorithm as close as to human nature by generating honeywords with randomly selecting passwords that belong to other users in the system. We present a standard approach to securing personal and business data in the system. We propose monitoring data access patterns by marking user behavior to determine if and when a malicious insider illegally accesses someone's documents in a system service. Decoy documents stored in the system along with 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 barrage the malicious insider with fake information in order to dilute or divert the user' s real data. Such preventive attacks that rely on disinformation technology could provide unusual 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 difficult for an adversary in getting the passwords in plaintext form a leaked password hash file. Hence, by developing such methods both of two securities goal – 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*. 13.2 (2016): 284-295.

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

[3] C. Herley and D. Florencio. "Protecting financial institutions from brute-force attacks." *Microsoft Research One Microsoft Way Redmond*, WA, 2008.

[4] H. Bojinov, E. Bursztein, X. Boyen, and D. Boneh . "Kamouflage: Lossresistant Password Management."

Computer Security - ESORICS 2008. Springer, 2010, pp. 286-302.

[5] 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. IEEE,* 2009, pp.391-405.

[6] M. H. Almeshekah , E. H. Spafford, and M. J. Atallah. "Improving Security using Deception, 2013." *Center for Education and Research Information Assurance and Security, Purdue University,* Tech. Rep. CERIAS Tech Report 2013- 13, 2013.

[7] The Dangers ofWeak Hashes. "If Your Password is 123456, Just Make It Hack me." *SANS Institute InfoSec Reading Room,* Tech. Rep., 2013.

[8] D. Mirante and C. Justin. "Understanding password Database Compromises." *Department of Computer Science and Engineering Polytechnic Institute of NYU*, 2013.

[9] Genc, Ziya Alper, S¨uleyman Kardas, and Mehmet Sabir Kiraz. "Examination of a New Defense Mechanism: Honeywords." *IACR Cryptology ePrint Archive* 2013 (2013): 696.

[10] Malone, David, and Kevin Maher. "Investigating the distribution of password choices." *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012.

[11] Almeshekah, Mohammed H, Eugene H. Spafford, and Mikhail J. Atallah. "Improving security using deception." *Center for Education and Research Information Assurance and Security, Purdue University,* Tech. Rep. CERIAS Tech Report 13 (2013).

[12] Juels, Ari, and Ronald L. Rivest. "Honeywords: Making password-cracking detectable." *Proceedings of the 2013 ACM SIGSAC conference on Computer and communications security*. ACM, 2013.

[13] Weir Matt et al. "Password cracking using probabilistic context-free grammars*." Security and Privacy, 2009. 30th IEEE Symposium on.* IEEE, 2009.

Appendix B

Certificates

INTERNATIONAL JOURNAL OF ADVANCE
RESEARCH AND INNOVATIVE IDEAS IN EDUCATION

★★★

CERTIFICATE

of

PUBLICATION

The Board of International Journal of Advance Research and Innovative Ideas in Education is hereby Awarding this Certificate to

MAHESH ARUN UMALE

In Recognition of the Publication of the Paper Entitled

IMPROVING SECURITY AND ACHIEVING FLATNESS TO THE USER-PASSWORD BY USING HONEYWORDS

Published in E-Journal

Volume-2 Issue-6 2016

Paper Id : 3449
ISSN(O) : 2395-4396

www.ijariie.com

N Patel

Editor In Chief

Sinhgad Technical Education Society's

Sinhgad Institutes

SKN SINHGAD INSTITUTE OF TECHNOLOGY AND SCIENCE, LONAVALA

Department of Computer Engineering

INTERNATIONAL ENGINEERING SCIENCES SUMMIT IESS

Certificate

This is to certify that Mr./Ms. Mahesh Umale

has attended/presented/published paper titled "Improving security and achieving flatness to the user password by using Honeyword" in International Engineering Sciences Summit (IESS '17) at SKN Sinhgad Institute of Technology and Science, Lonavala held on 16th and 17th March 2017.

Prof. Venkatesan N.
Co-convener

Prof. Neha Jamdar
Co-convener

Prof. V.D. Thombre
Convener

Dr. M.S. Rohokale
Patron

Co-sponsored by

TECHNOGEEKS