A

Project Report

on

ENHANCING SECURITY USING HONEYWORD

Submitted in Partial Fulfillment of
the Requirements for the Degree
of

Bachelor of Engineering

in

Computer Engineering

to

North Maharashtra University, Jalgaon

Submitted by

Bhagyashri Dhanraj Suryawanshi
Akshay Vilas Patil
Jeevan Bhagavan Patil
Paresh Tayade
Dipak Rajput

Under the Guidance of

Mrs.Shital A.Patil

DEPARTMENT OF COMPUTER ENGINEERING
SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY,
BAMBHORI, JALGAON - 425 001 (MS)

2016 - 2017

SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY, BAMBHORI, JALGAON - 425 001 (MS)

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the project entitled *Enhancing security using honeyword*, submitted by

Bhagyashri Dhanraj Suryawanshi
Akshay Vilas Patil
Jeevan Bhagavan Patil
Paresh Tayade
Dipak Rajput

in partial fulfillment of the degree of *Bachelor of Engineering* in *Computer Engineering* has been satisfactorily carried out under my guidance as per the requirement of North Maharashtra University, Jalgaon.

Date: October 9, 2016
Place: Jalgaon

Mrs.Shital A.Patil
Guide

Prof. Dr. Girish K. Patnaik
Head

Prof. Dr. K. S. Wani
Principal

Acknowledgements

The project on Enhancing Security using Honeyword is standing on the pillars of contribution of many people. We would like to express gratitude towards our beloved Principal Dr. K. S. Wani for introducing us to this research work. We would also like to thank our HOD Dr.G. K. Patnaik for encouraging us to enthusiastically accomplish this project. We would also extend my gratitude towards our Guide Mrs. Shital A.Patil for guiding us and showing the right path to successfully reach our destination. Also, We would like to thank all the faculty and staff members for extending their helping hand directly or indirectly. Last but not the least, We would also like to thank our dear parents for having their blessings on me and motivating me endlessly.

Bhagyashri Dhanraj Suryawanshi
Akshay Vilas Patil
Jeevan Bhagavan Patil
Paresh Tayade
Dipak Rajput

Contents

List of Figures

Abstract

Disclosure of password files is a servere security concerns making millions of vulnerable to cyber attack. A cyber security is nothing but cracking the user's password and stolen files. A bruteforce attack on the stolen hashfiles can be easily recover the user's password. Honeyword (Decoy Password) method is used to detect attack against hashed password database. A secure server can distinguish on user's real password among Honeyword and set off an alarm whenever a honeyword is used. Thus the Honeyword Technique is used to detect the unauthorised attempt and protect from stolen.

Chapter 1

Introduction

Introduction chapter will introduce the work, It will focusses exactly on what is the area of project and explains what is actually be done in this work. All ideas about project work are cleared here.

In the authentication process it becomes difficult to handle security of passwords that's why password became the most important asset to login. But users choose weak passwords (for easy to remember) that can be predicted by the attacker using brute force, dictionary, rainbow table attacks etc. Hence it becomes much easier to crack a password . 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 illegitimate user authentication. So Honeywords plays an important role to defense against stolen password files.

1.1 Background

This study focuses on 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 . This idea has been modified by Herley and Florencio to protect online banking accounts from password brute-force attacks[1]. 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 item for this method is the generation algorithm of the honeywords such that they shall be indistinguishable from the correct passwords[2]. Therefore, we propose a new approach that uses passwords of other users in the system for honeyword sets, i.e., realistic honeywords are provided. Moreover, this technique also reduces the storage cost compared with the honeyword method in section.

1.2 Motivation

Theft of Password Hash Files is a major security concern worldwide.Attacker compromises the system ephemerally and steals passwords hashes. Once the hash file is stolen ,by using password cracking techniques it is easy to capture most of the plain text passwords. Adversary almost always succeeds and is often undetected. The motivation behind this project is making password cracking detectable.

1.3 Problem Definition

In the authentication process it becomes difficult to handle security of passwords that's why password became the most important asset to login. Honeywords plays an important role to defense against stolen password files. The user enter's a password to login to a system. Server first check whether or not hash value for that password is in the list. if not then login is denied. Otherwise system check to varify if it is honeyword or correct password. The index of the honeyword is delivered to the honeychecker in an authenticated secure communication. The honeychecker checks whether the index of honeyword is equal to the index of correct password or not.If the equality holds it returns true otherwise it returns false. and it may raise an alarm depending on the security policy the system[3].

1.4 Scope

The main goal of project is to validate whether data access is authorised and to protects the user's real data from attackers. The proposed project is useful to all types of user's , because the data i.e the information is protected from the unauthorised person means from attacker.

1.5 Objective

In the project a simple method for improving the security of hashed password is provided.only additional honeywords (false passwords) associated with each user accounts. An attacker who steals a file of hashed password and inverts the hash function cannot tell if he has found the password or a honeywords The attempted use of honeyword for login send alert message to real user.An auxiliary server means honeychecker can distinguish the password from honeyword for login routine. And will send alert message to real user if honeyword is submitted.

1.6 Organization of the report

The system provide more security to client data which protect data from unauthorized access or attack. Firstly,Chapter 1 describes an Introduction. Chapter 2 describes System Analysis for gathering and interpreting facts, diagnosing problems and using the facts that improve the system. Chapter 3 describes Software Requirement Specification through the various hardware, software, functional and non-functional requirements of the system. Chapter 4 describes System Design to provide understanding and procedural details necessary for implementing the system recommended in system study. Chapter 5 describes the conclusion that integrate the various issues, research etc.and future scope for making the new changes requests considered to modify project scope.

1.7 Summary

In this chapter, an overview of the problem statement along with its solution for the work contained in this dissertation is provided. In the next chapter, related work in the area of communication architecture for disaster rescue operations is presented.

Chapter 2

System Analysis

System analysis is the study of states of interacting entities, including computer system analysis. The field closely related to requirement analysis or operation research. The chapter briefly discuss "System analysis" of Honeyword system with is's related work.

The related work and attempts are reviewed in section 2.1, them proposed system of the project is put in section 2.2. Next, section 2.3 explains the different feasibility studies related to the project . All risk involved in project are given in section 2.4, section 2.5 gives the project scheduling. Effort allocation is explained in section 2.6.

2.1 Literature Survey

Passwords are a notoriously weak authentication mechanism. Users frequently choose poor passwords. An adversary who has stolen a file of hashed passwords can often use brute-force search to find a password p whose hash value H(p) equals the hash value stored for a given user's password, thus allowing the adversary to impersonate the user. An effective approach is needed in the scenario where an adversary has stolen the password hash files[4]. 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 . 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 attempts 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 hit 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 conceal the real passwords, thereby forcing an adversary to carry out a considerable amount of online

work before getting the correct 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 sweetwords is constructed 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[5].

- The Honeyword generation methods, such as tweaking and take-a-tail or hybrid of them are previously implemented. The tweaking method alters the characters at specified positions. Take-a-tail method appends some digits at the end.
- The proposed system uses case alteration method in which only the case of specific characters is altered i.e.from uppercase to lowercase and vice-a-versa.
- The existing methods of honeyword generation are effective to certain extent which can be applied to any set of characters while altering passwords. However tweaking a semantically significant password in various different forms would lose the semantic significance of other honeywords. This makes it easy for an advisery to guess actual password from given set of honeywords. Moreover the existing methods need memory overhead and use of complex algorithm.

2.2 Proposed System

The proposed system called case alteration takes full advantage of the case sensitive nature of password authentication systems. It uses the actual password as a seed to generate decoy passwords or honeywords. The actual password may contain lowercase characters, uppercase characters or combination of both. A set of honeywords is generated such that the position of the characters remains same for each honeyword but some lowercase characters are converted into respective uppercase characters and some uppercase characters are converted into respective lowercase characters. This approach does not alter the semantic significance of the generated passwords if any. The case sensitive nature of authentication system can distinguish between an actual password and honeyword. For example, Let an actual password be Password. It contains 1 uppercase letter and 7 lowercase letters. The proposed case alteration system generate different number of honeywords by altering cases of some characters of the word Password. The generated honeywords may contain more than one uppercase characters and/or more than one lowercase characters. Some examples of the generated honeywords using the proposed case alteration system are,password,PASSWORD,PassWORD,paSSWORD,PaSsWoRd,pAsSwOrD,PASSword,etc. For

a password of length alphabetic characters, 2 to raise n honeywords can be generated by using this simple approach. It uses permutation of 2 cases for every alphabetic character namely uppercase and lowercase. This approach maintains semantic significance of the honeywords and hence makes it difficult for attacker to guess the actual password as all the honeywords appear semantically similar. Further embellishments like substitution of special symbol for some characters and vice- a versa can be added for generating more honeywords. E.g. 1l,2z,3E,4A, 5S,6G, 7T,8g, 9q,0O. This takes care of digits contained in passwords. While generating honeywords it makes sure that the honeywords look similar in appearance without altering semantic significance if any. This takes care of worst case scenario for the proposed approach. i.e. A password containing all digits like phone numbers, etc.

2.3 Feasibility study

Feasibility study aims to uncover the strengths and weakness of the existing system and the threats presented . The feasibility analysis shows the developers all the aspects of the project and they can know that whether the project is practically possible to develop worth limited resources and time. A feasibility study could be used to test a proposal for the new system. There are various types of feasibilities available that depends on different factors like technical, economical and operational feasibility.

2.3.1 Economical feasibility

The project is economically feasible as it requires open source softwares meaning most of the resources required for the development of this system are free to use. This makes it very much feasible in accordance to economy.

2.3.2 Operational feasibility

The proposed system needs no involvement of end user. It only implements security at back end servers maintaining password hash files and sets off security alarm. No additional components other than these are required, thus the proposed system is operationally feasible.

2.3.3 Technical feasibility

The technical feasibility deals with the technology and the tools used to develop the system. The proposed system can be implemented using current technologies available which are also open source hence free to use. It can also be improved by the technologies thus making it technically feasible.

2.4 Risk Analysis

In this section the analysis of the risk is discussed. Project Risk Analysis and Management is a process which enables the analysis and management of the risks associated with a project. Properly undertaken it will increase the likelihood of successful completion of a project to cost, time and performance objectives. The risk associated with the proposed system lies in the number of honeywords generated[7]. Another risk depends on length of passwords entered. Theoretically this model enables 2 raise to(-n) risk factor while guessing actual password from generated set of honeywords. Practically, it is advised to generate minimum 20 honeywords per user for assuring more than 95% security; as probability of guessing acual password out of 20 honeywords is equal to 5%, thus secutity is 95% in this case. In special cases where length of the password is less than 5, the number of honeywords generated will be less than 20. In such cases probability of risk increases drastically. However the risk cannot exceed 50% even in worst case scenario where password consists of a single character.

2.4.1 Project Risk

Threaten the project plan. That is, if project risks become real, it is likely that project schedule will slip and that costs will increase. Project risks identify potential budgetary, schedule, personnel (standing and organization), resource, customer, and requirements problems and their impact on a software project. In our project, project risk occurs if our requirement of technical member means technical team is unavailable according to our project plan and estimation and if our project is not completed within time in this situation project risk can occurs.

2.4.2 Technical Risk

Threaten the quality and timeliness of the software to be produced. If a technical risk becomes a reality, implementation may become difficult or impossible. Technical risks identify potential design, implementation, interface, verification, and maintenance problems. In addition, specification ambiguity, technical uncertainty, technical obsolescence, and "leading edge" technology are also risk factors. Technical risks occur because the problem is harder to solve than we thought it would be. In our project if any module of our web site is not worked properly according to our expectation then technical risk may occur.

2.5 Project Scheduling

Software project scheduling is an activity that distributes estimated effort across the planned project duration by allocating the effort to specific software engineering tasks. It is important to note, however, that the schedule evolves over time. During early stages of project planning, a macroscopic schedule is developed. That type of schedule identifies all major software engineering activities and the product functions to which they are applied. As the project gets under way, each entry on the macroscopic schedule is refined into a detailed schedule. Here, specific software tasks required to accomplish an activity are identified and scheduled. Scheduling for software engineering projects can be viewed from two rather different perspectives. In the first, an end-date for release of a computer-based system has already and irrevocably been established. The software organization is constrained to distribute effort within the prescribed time frame. The second view of software scheduling assumes that rough chronological bounds have been discussed but that the end-date is set by the software engineering organization. Effort is distributed to make best use of resources and an end-date is defined after careful analysis of the software. Unfortunately, the first situation is encountered far more frequently than the second.

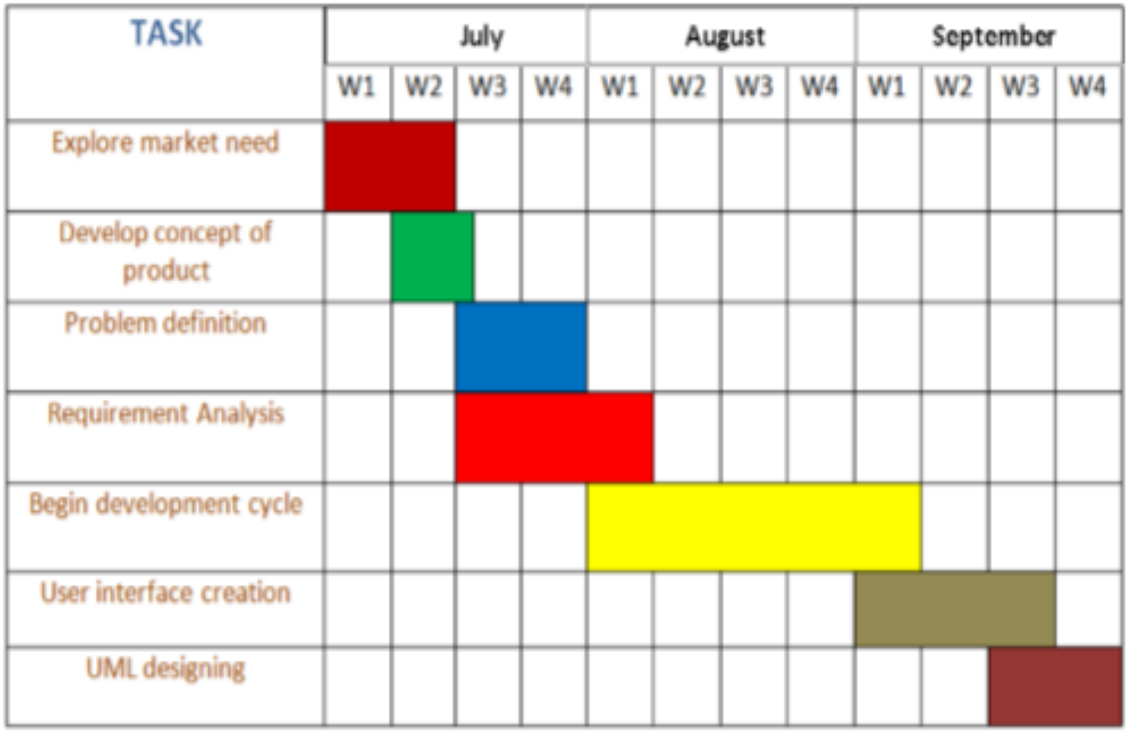

Figure 2.1: Gantt chart

2.6 Effort allocation

Project means team work; Project is developed by combination of effort of team. So whole project is divided into modules and number of modules is allotted to team members. After completion of each module, it will be link from one module to another module to form a complete project. That effort allocation should be used as a guideline only. The characteristics of each project must dictate the distribution of effort. Work expended on project planning rarely accounts for more than 23 percent of effort, unless the plan commits an organization to large expenditures with high risk. Requirements analysis may comprise 10 to 25 percent of project effort. Effort expended on analysis or prototyping should increase in direct proportion with project size and complexity. A range of 20 to 25 percent of effort is normally applied to software design. Time expended for design review and subsequent iteration must also be considered.

Table 2.1: Effort Allocation

Activity	Bhagyashri	Akshay	Jeevan	Paresh	Deepak
Project Planning	21%	21%	21%	21%	15%
Requirement Gathering	20%	19%	19%	21%	21%
Design	21%	22%	21%	19%	17%

2.7 summary

In this chapter system analysis of project is described briefly. In the next chapter system requirement specification is described.

Chapter 3

System Requirement Specification

The chapter focuses on the various requirements of the system.Section 3.1 describes the hardware requirements of the system. The software requirements of the system are discussed in section 3.2. Section 3.3 describes the functional requirements of the system. Non-functional requirement of system are discussed in 3.4. Section 3.5 describes other requirements and constraints of system. Finally the last section is of summary.

3.1 Hardware requirements

- Processor :Pentium IV and above.
- Display Type :VGA and above.
- Memory(RAM) :256MB.
- Storage Memory :1GB.

3.2 Software requirements

- Operating system : Windows 7/8.
- System Type : 64-bit/32-bit operating system.
- Front end : Java.
- Java version : Jdk1.6.0
- Back end :mysql.
- Web server : Apache Tomcat 6.0

3.3 Functional requirements

Function Requirements include both functional and non functional requirements aspects related to the project.Installed distribution must provide the option for development of distribution and support.Various phases of development of project.Thus a pre installed Java,Mysql and Winedit 8 which will include other supporting software for process of developments[6].

3.4 Non-Functional requirements

Non-Functioanl requirements include various requirements but the most prominent one are access to Unified Audi Portal web pages and search engine for accessing the web pages.

3.5 Summary

In this Chapter ,Hardware requirements, Software requirements ,functional and non functional requirements are explained.In next chapter the system design is described through various UML diagrams.

Chapter 4

System Design

Design is an activity concerned with making major decisions, often of a structural nature. It shares with programming a concern for abstracting information representation and processing sequences, but the level of detail is quite different at the extremes. Design builds coherent, well planned representations of programs that concentrate on the interrelationships of parts at the higher level and the logical operations involved at the lower levels.Software design is the first of the three technical activities-designs, Coding and test which are required to build and verify the software. Section 4.1 describes the system architecture of the proposed system. E-R Diagrams are discussed in section 4.2. Section 4.3 describes the database design of the project.Data flow diagrams are discussed in section 4.4. The UML diagrams are discussed in section 4.5. Finally, the last section is of summary.

4.1 System Architecture

Fig shows a schematic view of the architecture of entire system. The architecture consist of two server first is identification server and second is verification server. Both of these performs authentication process by dividing it in two parts i.e Identification and verification.

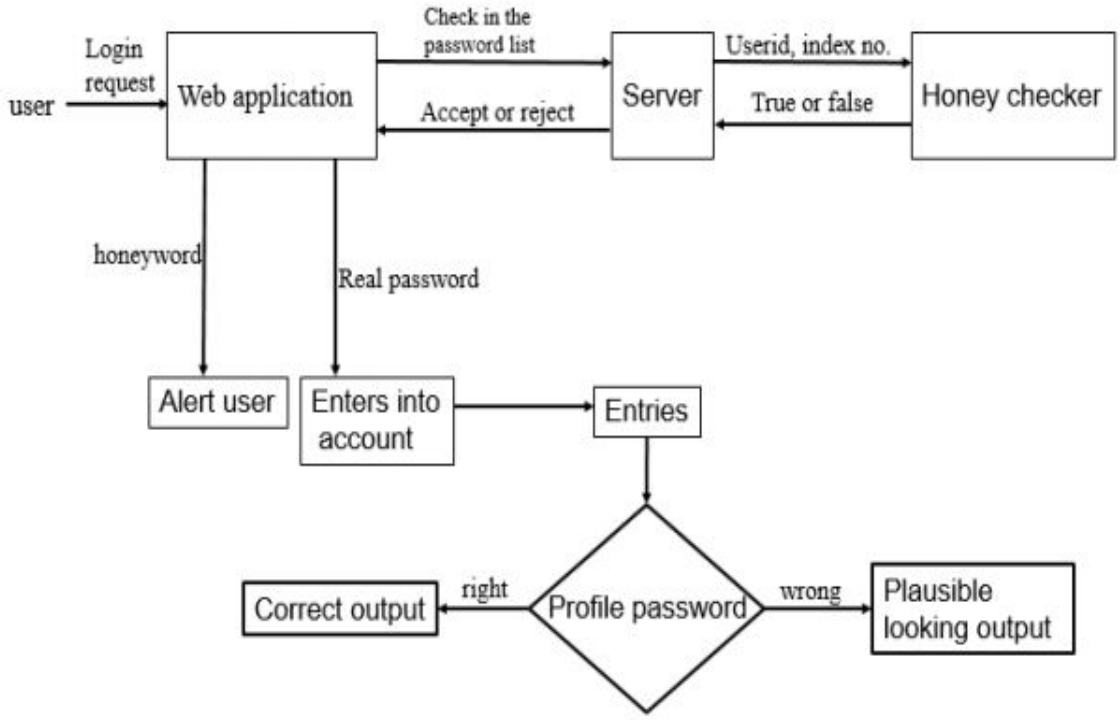

Figure 4.1: Architecture of proposed system

4.2 E-R Diagram

In software engineering, an entity relationship model (ER model) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities (things) and the relationships that can exist among them. Entity-relationship modeling was developed by Peter Chen and published in a 1976 paper. variants of the idea existed previously, and have been devised subsequently such as super type and subtype data entities and commonality relationships. ER diagram shows the relationship between various entities involved in the system. Entity relationship diagram for mindmetrics is shown in following fig

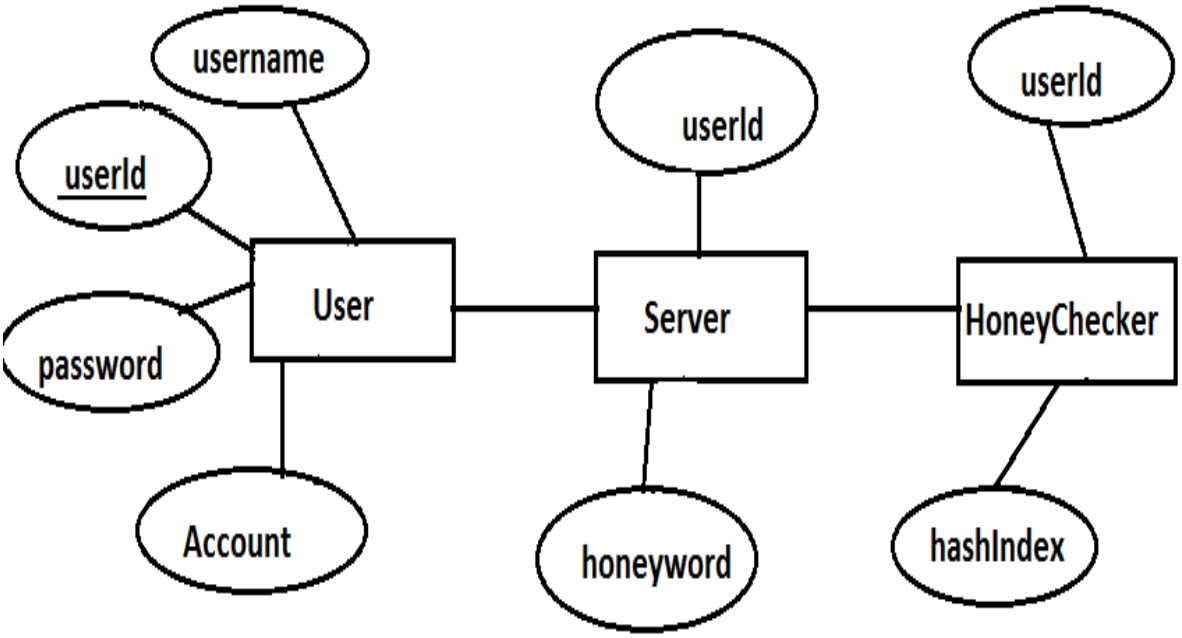

Figure 4.2: ER Diagram for Honeyword system

4.3 Data flow diagram

A DFD is a graphical technique that depicts the information ow and the transformation that we have applied as the data moves from input to output. The data ow diagram also known as data ow graph or bubble chart. A data ow diagram may be used to represent a system or software at any level of abstraction. The data ow diagram can be completed using only four simple notations i.e. special symbols or icons and the annotation that with a special system. A data ow diagram (DFD) is a graphical technique that describes information about ow and that are applied as data moves from input to output. The DFD is also called as data ow graph or bubble chart. Named circles show the processes in DFD or named arrows entering or leaving the bubbles represent bubbles and data ow. A rectangle represents a source or sink and is not originate or consumer of data. Data ow diagrams are the basic building blocks that define the ow of data in a system to the particular destination and difference in the ow when any transformation happens. It makes whole procedure like a good document and makes simpler and easy to understand for both programmers and non-programmers by dividing into the sub process. The data ow diagram serves two purposes:

- To provide an indication of how data are transform as the moves through the system.

- To depict the function that transforms the data flow.

A level 0 DFD, also called a fundamental system model (FSM) or a context model. It represents the entire software elements as a single bubble with input and output data indicated by incoming and outgoing arrows respectively. Figure 4.1shows the Level 0 DFD of the system.

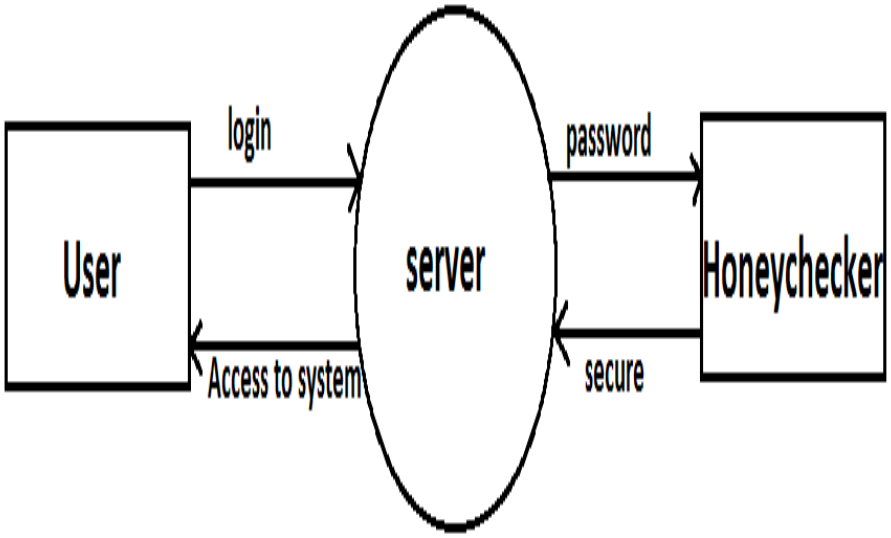

Figure 4.3: Level 0 DFD for Honeyword system

Level 1 DFD contains additional processes and information ow paths, as the level 0 DFD is partitioned to reveal more detail. Level 1 DFD might contain 5 - 6 bubbles with interconnecting arrows.Figure 4.2 shows the Level 1 DFD of the system.

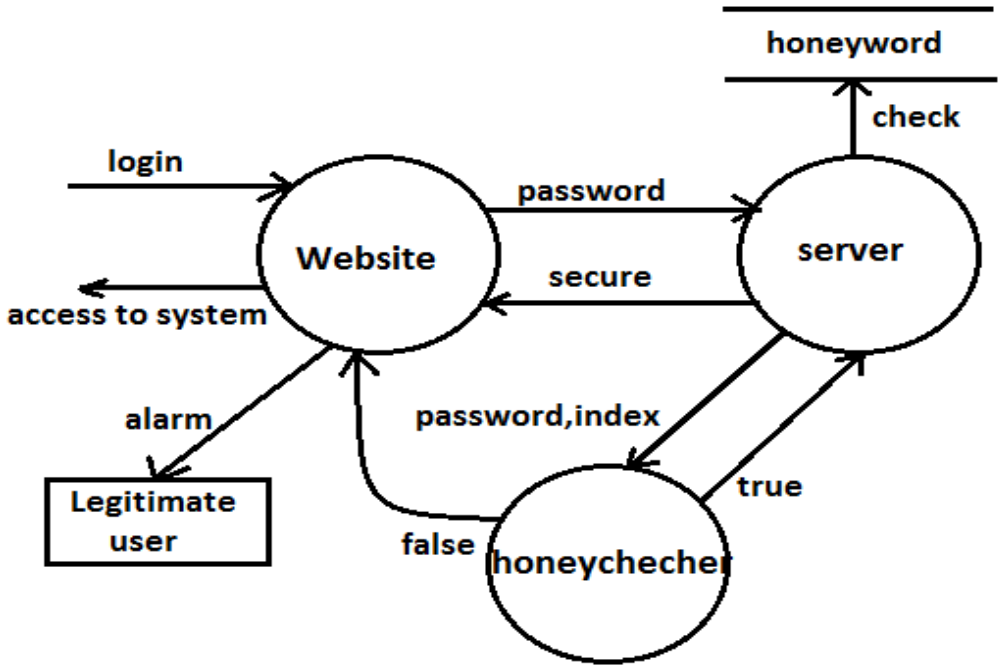

Figure 4.4: Level 1 DFD for Honeyword system

4.4 Interface Design

The interface design describes how the software communicates within itself, with systems that inter operate with it, and with humans who use it. An interface implies a flow of information (e.g., data and control) and a specific type of behavior. Therefore, data and control flow diagrams provide much of the information required for interface design.

4.4.1 User Interface design

The overall process for designing a user interface begins with the creation of differen models of system function (as perceived from the outside). The human- and computer-oriented tasks that are required to achieve system function are then delineated; design issues that apply to all interface designs are considered; tools are used to prototype and ultimately implement the design model; and the result is evaluated for quality.

4.5 UML Diagrams

The UML is a language for: Visualizing: Structures which are transient can be represented using the UML Specifying: The UML addresses the specification of all the important analysis, design and implementation decisions that must be made in developing and deploying a software intensive system. Constructing: The UML is not a visual programming language,

but its models can be directly connected to a variety of programming languages. Documenting: The UML addresses the documentation of a system's architecture and all of its details.

4.5.1 Usecase Diagram

A Use case diagram shows a set of use cases and actors and their relationships. Use case diagrams address the static use case view of a system. These diagrams are especially important in organizing and modeling the behaviors of a system. The Use Case diagram of the proposed system is shown in figure 4.3.

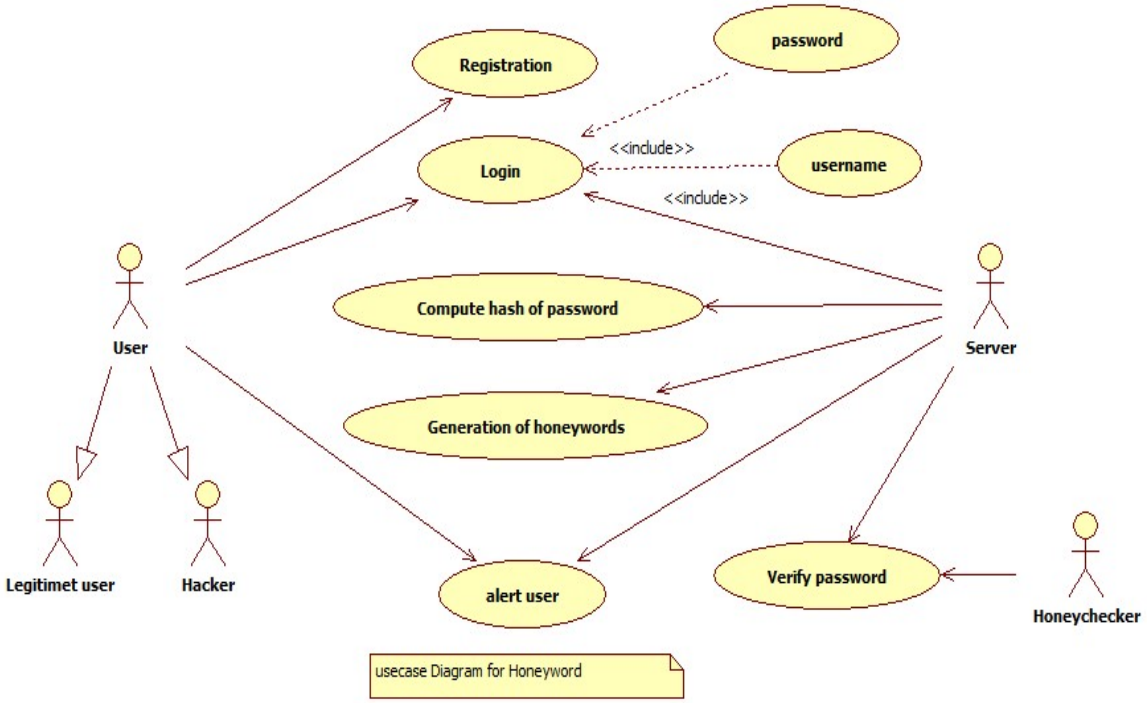

Figure 4.5: UseCase diagram for Honeyword system

4.5.2 Class Diagram

A Class diagram shows a set of classes, interfaces and collaborations and their relationships. These diagrams are the most common diagram found in modeling object-oriented systems. Class diagram address the static design view of a system. Figure 4.4 shows the class diagram for the proposed system.

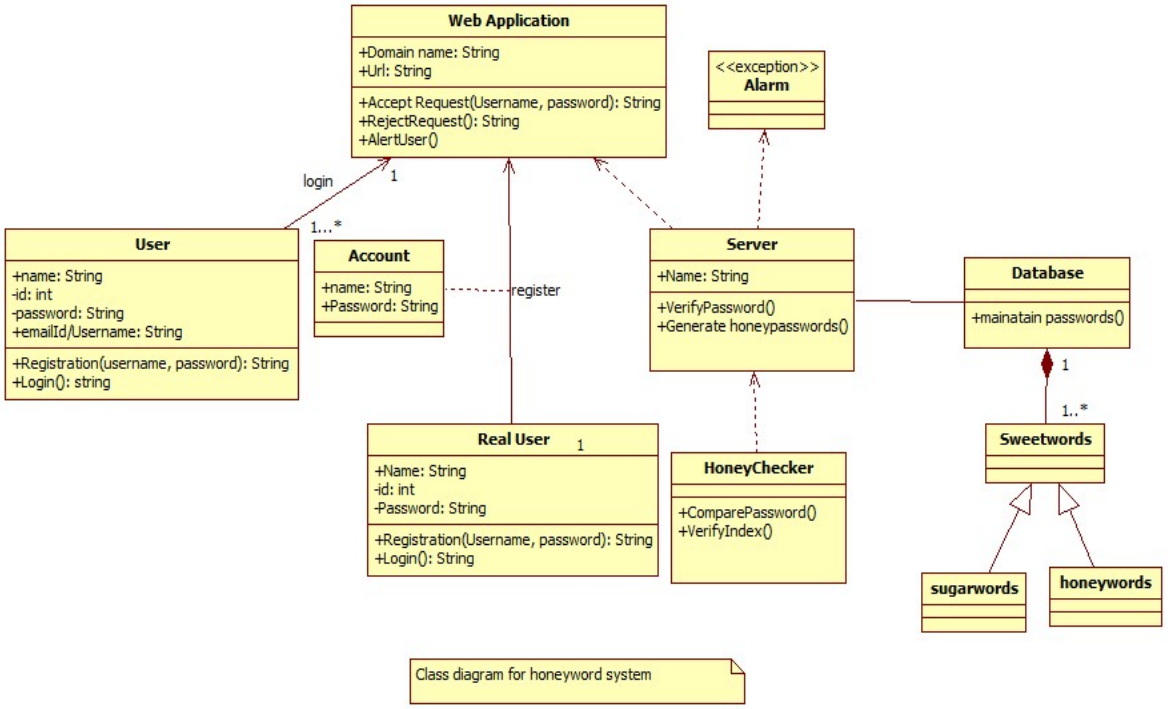

Figure 4.6: Class diagram for Honeyword system

4.5.3 Sequence Diagram

Both sequence and collaboration diagrams are kinds of interaction diagrams. An interaction diagram shows an interaction, consisting of a set of objects and their relationships. They address the dynamic view of a system.

- A sequence diagram is an interaction diagram that emphasizes the time-ordering of messages.
- A collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive massages. Sequence diagram and collaboration diagrams are isomorphic i.e one can be transformed into other.

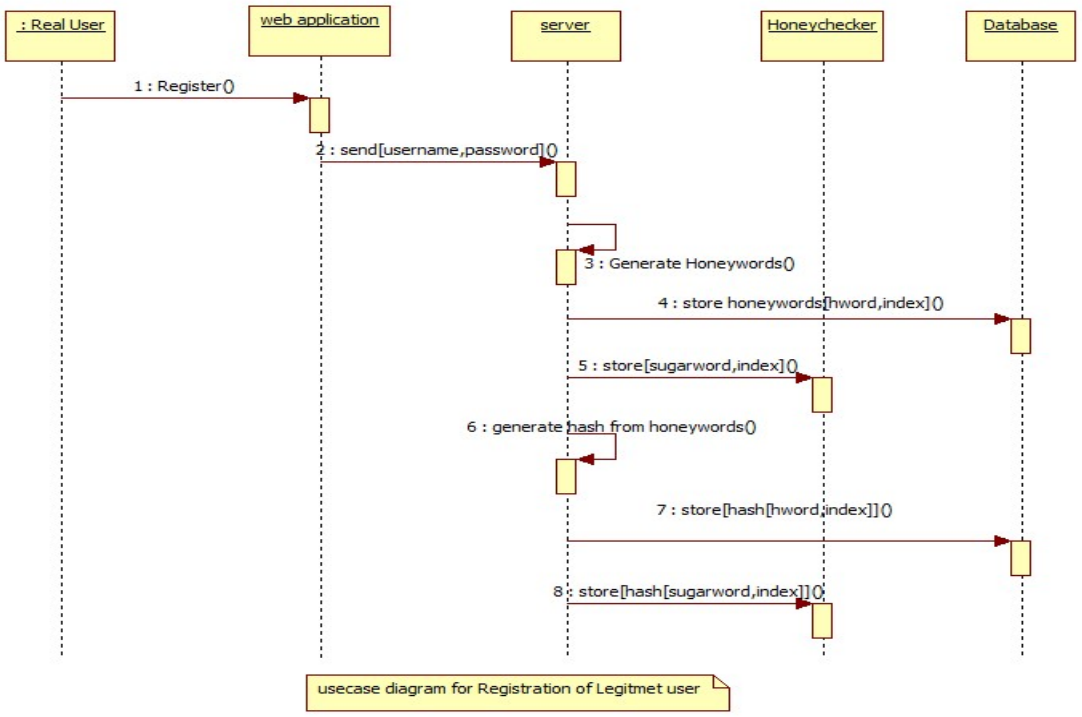

Figure 4.7: Sequence diagram for Registration of User

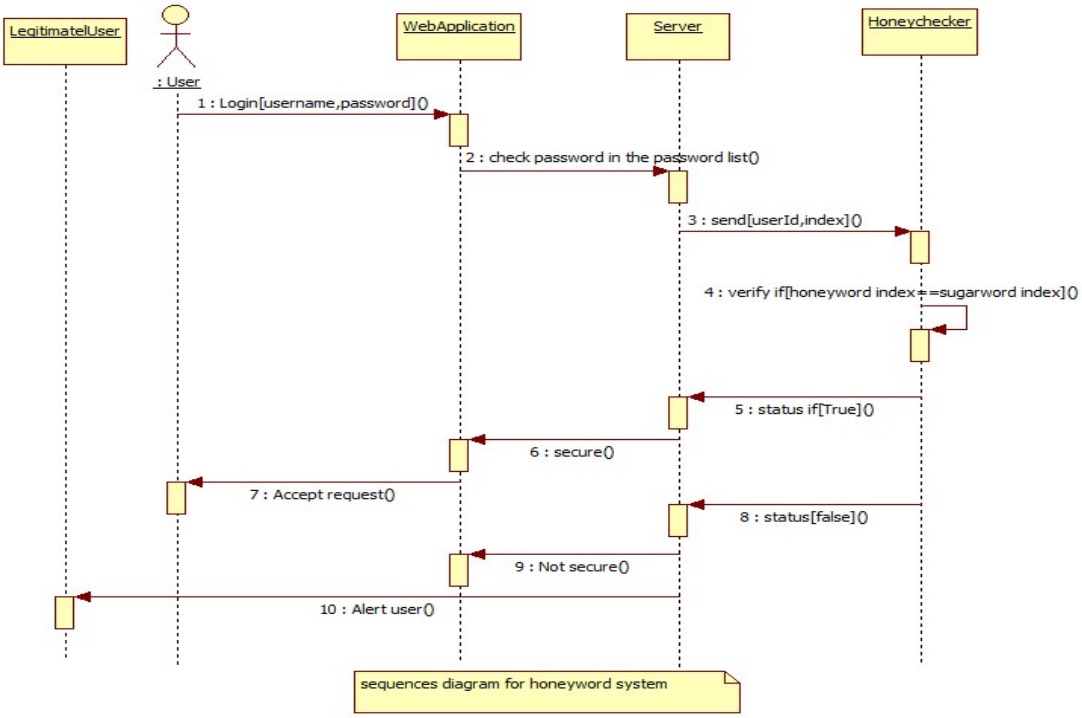

Figure 4.8: Sequence diagram for Honeyword System

4.5.4 State Chart Diagram

A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.State diagrams are used to give an abstract description of the behavior of a system. This behavior is analyzed and represented as a series of events that can occur in one or more possible states. Hereby "each diagram usually represents objects of a single class and track the different states of its objects through the system.

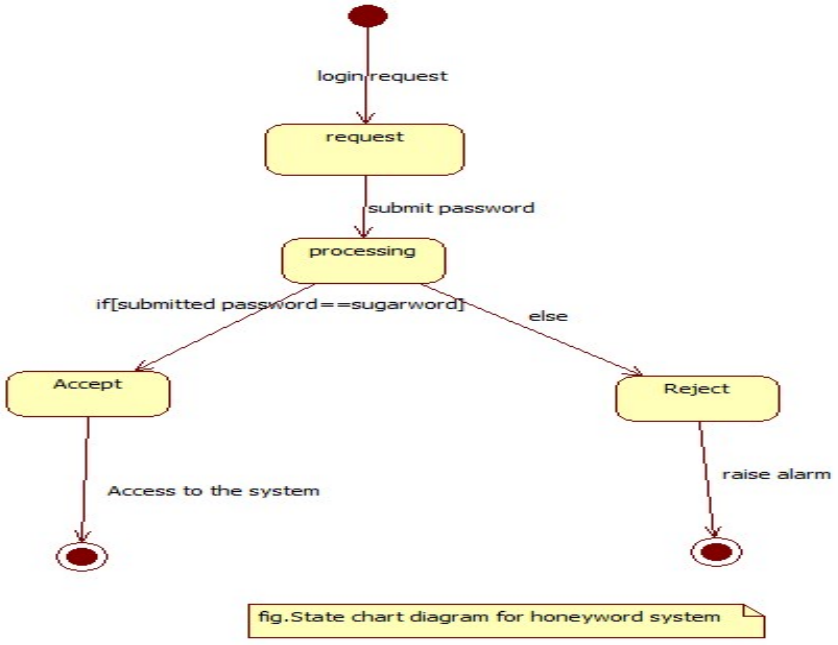

Figure 4.9: State chart diagram for Honeyword System

4.5.5 Component Diagram

A component diagram shows the organization and dependencies among a set of components. Component diagrams address the static implementation view of a system. The component diagram for the proposed system is shown in figure 4.9.

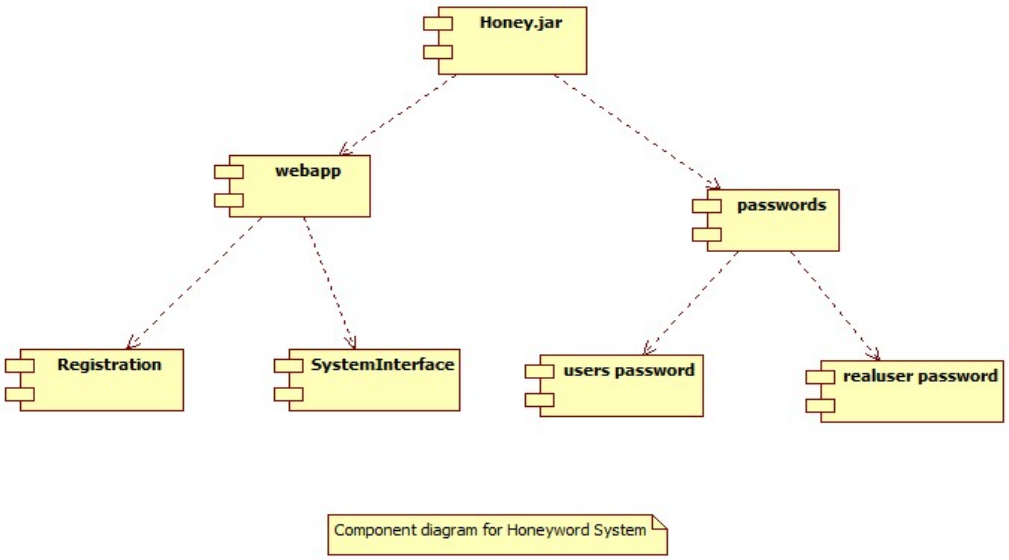

Figure 4.10: Component diagram for Honeyword system

4.5.6 Deployment Diagram

A deployment diagram shows the configuration of run-time processing nodes and the components that live on them. Deployment diagram address the static deployment view of an architecture. Deployment diagram for the proposed system is shown in figure.

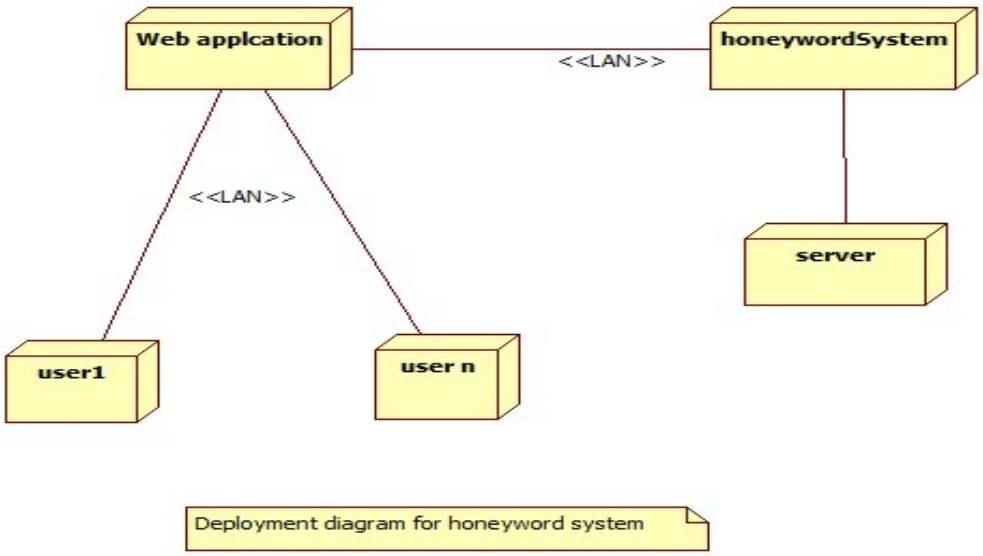

Figure 4.11: Deployment diagram for Honeyword system

4.6 Summary

In this chapter, System design is described. In the next chapter conclusion is presented

Chapter 5

Conclusion

Honeywords provide many benefit. Published password files provide attackers with insight into how users compose their passwords. Attackers can then refine their models of user password selection and design faster password cracking algorithms . Thus every breach of a password server has the potential to improve future attacks. Some honeyword generation strategies, particularly chaffing ones, obscure actual user password choices, and thus complicate model building for would-be hash crackers. It may even be useful to muddy attackers knowledge of users composition choices intentionally by drawing some honeywords from slightly perturbed probability distributions.Despite their benefits over common methods for password management, honeywords arent a wholly satisfactory approach to user authentication. They inherit many of the well known drawbacks of passwords and something-you-know authentication more generally. Eventually, passwords should be supplemented with stronger and more convenient authentication methodsor give way to better authentication methods completely, as recently predicted by the media

Bibliography

[1] D. Mirante and C. Justin, Understanding password database compromises, Dept. of Comput. Sci. Eng. Polytechnic Inst. of NYU, New York

[2] M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek, Password cracking using probabilistic context-free grammars, in Proc. 30th IEEE Symp. Security Privacy, 2009

[3] A. Juels and R. L. Rivest. Honeywords: Making password cracking detectable. Unpublished draft.

[4] J. Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In IEEE Symp. Security and Privacy, 2012.

[5] D. Elser and M. Pekrul. Inside the password-stealing business: the who and how of identity theft, 2009.

[6] L. Spitzner. Honeytokens: The other honeypot. Symantec SecurityFocus, July 2003.L. Spitzner. Honeytokens: The other honeypot. Symantec SecurityFocus, July 2003.L. Spitzner. Honeytokens: The other honeypot. Symantec SecurityFocus, July 2003.

[7] M. Bakker and R. van der Jagt. GPU-based pass- word cracking. Technical report, Univ. of Amster- dam, 2010.