Analysis of User Behaviour Tracking, Detection And Elimnation of Thief

A PROJECT REPORT

Submitted by

AUVULA MOHAN	110319104003
C.LOKASAI REEDY	110319104026
RAVELLA NARSIMHA RAO	110319104035
CIRIVELLA SIVA SAI	110319104039

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

GRT INSTITUTE OF ENGINEERING AND TECHNOLOGY
(Affiliated To Anna University, Chennai-600 025)

ANNA UNIVERSITY::CHENNAI 600 025

MAY 2023

GRT INSTITUTE OF ENGINEERING AND TECHNOLOGY

Tiruttani-631209.

ANNA UNIVERSITY::CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report''**ANALYSIS OF USER BEHAVIOUR TRACKING, DETECTION AND ELIMINATION OF THIEF"**is the bonafide work of **AUVULA MOHAN [110319104003] , C.LOKASAI REEDY [110319104026], CIRIVELLA SIVA SAI [110319104039] RAVELLA NARSIMHA RAO [110319104035]** who carried out the project work under my supervision.

SIGNATURE

Dr.N.KAMAL

HEAD OF THE DEPARTMENT

Department Of Computer Science And Engineering

GRT Institute of Engineering and and Technology Tiruttani.

SIGNATURE

Mrs.V.AARTHI
SUPERVISOR
Assistant Professor

Department Of Computer Science And Engineering

GRT Institute of Engineering and and Technology Tiruttani.

Certified that the candidate were examined in Viva-voice Examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First and foremost I place this project work on the feet of GOD ALMIGHTY and PARENTS who is the power of strength in each step of progress towards the successful completion of our project.

We are very grateful to **Dr.S.ARUMUGAM, M.E, Ph.D.,** Principal for providing us with consistent guidance and motivation to execute a real time project to learn and experience the project work in an environment to complete our project successfully.

Our sincere thanks to **Dr.N.KAMAL, M.E., Ph.D.,** Professor and Head, Department of Computer Science and Engineering for giving us the wonderful opportunity to do the project and providing the required facilities to fulfill our work.

We are highly indebted and thankful to our project coordinator **Mr.D.ABDUL KAREEM, M.E.,** Professor and Head, Department of Computer Science and Engineering for his immense support in doing the project.

We are very grateful to our internal guide **Mr.AARATHI**, Assistant Professor, Department of Computer Science and Engineering for guiding us with her suggestions to complete our project.

We also dedicate equal and grateful acknowledgment to all the respectable members of the faculty and lab in-charges of the Department of Computer Science and Engineering, friends and our families for their motivation, encouragement and continuous support.

ABSTRACT

In the big data era, the volume of digital data increases explosively. A recent report indicates that the data we create and copy are doubling in size every two years, and will reach 175 zettabytes by 2025. As the amount of data has increased exponentially, users suffer from critical problems in data management. With the significant development of cloud storage, people are increasingly outsourcing their data to cloud servers, which enables them to efficiently manage their data without deploying infrastructures and maintaining local devices, Social networks Accounts are tracked & detected. One fundamental aspect of this paradigm shifting is to centralize the data from the users' infrastructure to the cloud. Users can acquire almost unlimited storage capability from the cloud, and their cost for infrastructure purchase and storage maintenance can be released. Although storing data remotely to the cloud brings appealing benefits to users, it also brings new challenging security threats towards users' outsourced data. Since users lose ultimate control over their data, the correctness of their outsourced data becomes the major concern of cloud users. If hacker attacks the Genuine user then our allows the attacker to proceed further until our system captures all the important information about the attacker. We generate Honey words based on the user info provided and the original password is converted into another format and stored along with the Honey words. We deploy Intermediate server, Shopping server for purchase and Cloud server for maintaining user account details. Attacker who knows the E mail account of original user can easily reset the password of the cloud server. Attacker is invited to do attack in this Project, so as to find him out very easily. Now attacker logins into the purchase portal, where he is been tracked unknowingly & he is allowed to do purchase. Server identifies the attacker and sends the info to the Original owner and also it blocks the attacker even doing transaction from his original account. The main aim of the project is detect the attacker in shopping purchase portal. Identify the hacker and can fetch the hacker details. Constructing a Rat Trap for the Hacker

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1
INRODUCTION

1.1 PROJECT INTRODUCTION:

Cloud storage services have rapidly become increasingly popular. Users can store their data on the cloud and access their data anywhere at any time. Because of user privacy, the data stored on the cloud is typically encrypted and protected from access by other users. Considering the collaborative property of the cloud data, attribute-based encryption (ABE) is regarded as one of the most suitable encryption schemes for cloud storage. There are numerous ABE schemes that have been proposed, including.

Most of the proposed schemes assume cloud storage service providers or trusted third parties handling key management are trusted and cannot be hacked; however, in practice, some entities may intercept communications between users and cloud storage providers and then compel storage providers to release user secrets by using government power or other means. In this case, encrypted data are assumed to be known and storage providers are requested to release user secrets. As an example, in 2010, without notifying its users, Google released user documents to the FBI after receiving a search warrant. In 2013, Edward Snowden disclosed the existence of global surveillance programs that collect such cloud data as emails, texts, and voice messages from some technology companies. Once cloud storage providers are compromised, all encryption schemes lose their effectiveness. Though we hope cloud storage providers can fight against such entities to maintain user privacy through legal avenues, it is seemingly more and more difficult.

Cloud storage services have rapidly become increasingly popular. Users can store their data on the cloud and access their data anywhere at any time. Because of

user privacy, the data stored on the cloud is typically encrypted and protected from access by other users. Considering the collaborative property of the cloud data, attribute-based encryption (ABE) is regarded as one of the most suitable encryption schemes for cloud storage.

Password Segmentation Algorithm can be used to segment password and user attributes, and later the correlation between a segmented password and user attributes is achieved [1]. Cloud Data can be splitted into multiple blocks. TPA will verify the data integrity and on the overlay Block chain technology which is used for secured data storage in the cloud server [2]. Various different algorithms can be used to identify the load for every Virtual Machine like Bayesian Stackelberg game and Risk assessment framework is used to identify the Attacks [3]. We need to deploy multi user data access with highly secured multi encryption and decryption Policies for secured data access [10]. We use label and pseudorandom function to encrypt message, which significantly reduce the computations on the servers and enable us to use homomorphic MACs technology to realize verifiable computations naturally [11]. We need to focus on a mobile cloud scenario where data is stored within a geographically-limited area A by a community of mobile devices, For ease of illustration, we consider a single file. A user within A can download F from the mobile devices via direct communication links without accessing the network infrastructure [12]. The concept which is proposed to deal with secured data storing and sharing in multiple cloud providers. Big data based geo dispersed data storage services. Secured data storing using encryption Techniques [13]. To ensure the integrity of the data stored in the cloud, many data integrity auditing schemes can be used. Biometric data can be used (e.g. iris scan, fingerprint) as the user's fuzzy private key to avoid using the hardware token. Meanwhile, the scheme can still effectively complete the data integrity auditing [14]. Encrypted keyword using block chain can be integrated for the Data security in this Project [15]

1.2 OBJECTIVE OF PROJECT

We see that attack prevention may take place either at the server or the client-side. A proxy is typically placed in front of the server to examine the server's responses before they reach a user's browser. A modified browser or a library that is securelydownloaded from the server checks the responses for potential attacks.

1.3 DOMAIN INTRODUCTION

Cloud computing is a fast-growing technology that has established itself in the next generation of IT industry and business. Cloud computing promises reliable software, hardware, and IaaS delivered over the Internet and remote data centers. Cloud services have become a powerful architecture to perform complex large-scale computing tasks and span a range of IT functions from storage and computation to database and application services. The need to store, process, and analyze large amounts of datasets has driven many organizations and individuals to adopt cloud computing . A large number of scientific applications for extensive experiments are currently deployed in the cloud and may continue to increase because of the lack of available computing facilities in local servers, reduced capital costs, and increasing volume of data produced and consumed by the experiments. In addition, cloud service providers have begun to integrate frameworks for parallel data processing in their services to help users access cloud resources and deploy their programs .

Cloud computing "is a model for allowing ubiquitous, convenient, and on-demand network access to a number of configured computing resources (e.g., networks, server, storage, application, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". Cloud computing has a number of favorable aspects to address the rapid growth of economies and technological barriers. Cloud computing provides

total cost of ownership and allows organizations to focus on the core business without worrying about issues, such as infrastructure, flexibility, and availability of resources . Moreover, combining the cloud computing utility model and a rich set of computations, infrastructures, and storage cloud services offers a highly attractive environment where scientists can perform their experiments . Cloud service models typically consist of PaaS, SaaS, and IaaS.

PaaS, such as Google′s Apps Engine, Salesforce.com, Force platform, and Microsoft Azure, refers to different resources operating on a cloud to provide platform computing for end users.

SaaS, such as Google Docs, Gmail, Salesforce.com, and Online Payroll, refers to applications operating on a remote cloud infrastructure offered by the cloud provider as services that can be accessed through the Internet.

IaaS, such as Flexiscale and Amazon′s EC2, refers to hardware equipment operating on a cloud provided by service providers and used by end users upon demand.

The increasing popularity of wireless networks and mobile devices has taken cloud computing to new heights because of the limited processing capability, storage capacity, and battery lifetime of each device. This condition has led to the emergence of a mobile cloud computing paradigm. Mobile cloud facilities allow users to outsource tasks to external service providers. For example, data can be processed and stored outside of a mobile device . Mobile cloud applications, such as Gmail, iCloud, and Dropbox, have become prevalent recently. Juniper research predicts that cloud-based mobile applications will increase to approximately 9.5$ billion by 2014. Such applications improve mobile cloud performance and user experience. However, the limitations associated with wireless networks and the intrinsic nature of mobile devices have imposed computational and data storage restrictions

Cloud Setup

Cloud Service Provider will contain the large amount of data in their Data Storage. Also the Cloud Service provider will maintain the all the User information to authenticate the User when are login into their account. The User information will be stored in the Database of the Cloud Service Provider. Also the Cloud Server will redirect the User requested job to the Resource Assigning Module to process the User requested Job. The Request of all the Users will process by the Resource Assigning Module. To communicate with the Client and the with the other modules of the Cloud Network, the Cloud Server will establish connection between them. For this Purpose we are going to create an User Interface Frame. Also the Cloud Service Provider will send the User Job request to the Resource Assign Module in Fist in First out (FIFO) manner.

Private Cloud

Private clouds are dedicated to one organization and do not share physical resources. The resource can be provided in-house or externally. A typical underlying requirement of private cloud deployments are security requirements and regulations that need a strict separation of an organization's data storage and processing from accidental or malicious access through shared resources. Private cloud setups are challenging since the economical advantages of scale are usually not achievable within most projects and organizations despite the utilization of industry standards. The return of investment compared to public cloud offerings is rarely obtained and the operational overhead and risk of failure is significant.

Public Cloud

Public clouds share physical resources for data transfers, storage, and processing. However, customers have private visualized computing environments and isolated storage.Security concerns, which entice a few to adopt private clouds or custom deployments, are for the vast majority of customers and

projects irrelevant. Visualization makes access to other customers' data extremely difficult.

Hybrid Cloud

The hybrid cloud architecture merges private and public cloud deployments. This is often an attempt to achieve security and elasticity, or provide cheaper base load and burst capabilities. Some organizations experience short periods of extremely high loads, e.g. as a result of seasonality like black Friday for retail, or marketing events like sponsoring a popular TV event. These events can have huge economic impact to organizations if they are serviced poorly.

The hybrid cloud provides the opportunity to serve the base load with in-house services and rent for a short period a multiple of the resources to service the extreme demand. This requires a great deal of operational ability in the organization to seamlessly scale between the private and public cloud. Tools for hybrid or private cloud deployments exist like Eucalyptus for Amazon Web Services. On the long-term the additional expense of the hybrid approach often is not justifiable since cloud providers offer majorTo ensure the integrity of the data stored in the cloud, many data integrity auditing schemes can be used. Biometric data can be used (e.g. iris scan, fingerprint) as the user's fuzzy private key to avoid using the hardware token. Meanwhile, the scheme can still effectively complete the data integrity auditing [14]. Encrypted keyword using block chain can be integrated for the Data security in this Project [15].

CHAPTER 2

LITERATURE SURVEY

1.Some Remarks on Honeyword BasedPassword-Cracking Detection

Imran Erguler
TUBITAK BILGEM, Gebze, Kocaeli, Turkey
imran.erguler@tubitak.gov.tr

ABSTRACT

Recently, Juels and Rivest proposed honeywords (decoy passwords) to detect attacks against hashed password databases. For eachuser account, the legitimate password is stored with several honeywordsin order to sense impersonation. If honeywords are selected properly, anadversary who steals a _le of hashed passwords cannot be sure if it isthe real password or a honeyword for any account. Moreover, enteringwith a honeyword to login will trigger an alarm notifying the administrator about a password _le breach. At the expense of increasing storagerequirement by 20 times, the authors introduce a simple and effectivesolution to detection of password file disclosure events. In this study, wescrutinize the honeyword system and present some remarks to highlightpossible weak points. Also, we suggest an alternative approach that selects honeywords from existing user passwords in the system to providerealistic honeywords { a perfectly at honeyword generation method {and also to reduce storage cost of the honeyword scheme.

2. Prover and Verifier Based Password Protection: PVBPP

Priyanka Naik1*, Sugata Sanyal2
Computer Science Department, Manipal Institute of Technology, Manipal, India
ppnaik1890@gmail.com
Corporate Technology Office, Tata Consultancy Services, Mumbai, India
sugata.sanyal@tcs.com
*Corresponding Author

Abstract
In today's world password are mostly used for authentication. This makes them prone to various kinds of attacks like dictionary attacks. A dictionary attack is a method of breaking the password by systematically entering every word in a dictionary as a password. This attack leads to an overload on the server leading to denial of service attack. This paper presents a protocol to reduce the rate of dictionary attack by using a prover and a verifier system. This system makes it difficult for the attacker to prove it as a valid user by becoming computationally intensive. The rate of attempts is also reduced and thus restricting the Denial of Service attack.

3.The Dangers of Weak Hashes

GIAC GWEB Gold Certification
Author: Kelly Brown, kbrown@aboutweb.com
Advisor: Robert Vandenbrink
Accepted: November 15, 2013

ABSTRACT

There have been several high publicity password leaks over the past year including LinkedIn,Yahoo, and eHarmony. While you never want to have vulnerabilities that allow hackers to getaccess to your password hashes, you also want to make sure that if the hashes are compromisedit is not easy for hackers to generate passwords from the hashes. As these leaks havedemonstrated, large companies are using weak hashing mechanisms that make it easy to crackuser passwords. In this paper I will discuss the basics of password hashing, look at passwordcracking software and hardware, and discuss best practices for using hashes securely.

4.Improving Security using Deception

Mohammed H. Almeshekah, Eugene H. Spafford and Mikhail J. Atallah

November 11, 2013

Abstract

As the convergence between our physical and digital worlds continues at a rapid pace, much of our information is becoming available online. In this paper we develop a novel taxonomy of methods and techniques that can be used to protect digital information. We discuss how information has been protected and show how we can structure our methods to achieve better results. 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. We present analysis of these relationships and discuss how they can be applied at different scales within organizations. We also identify some of the areas that are worth further investigation. We map these protection techniques against the cyber kill-chain model and discuss some findings.

5.Kamouage: Loss-Resistant PasswordManagement

Hristo Bojinov1, Elie Bursztein1, Xavier Boyen2, and Dan Boneh1
1 Stanford University
2 Universit_e de Li_ege, Belgium

Abstract

We introduce Kamouage: a new architecture for buildingtheft-resistant password managers. An attacker who steals a laptop orcell phone with a Kamouage-based password manager is forced to carryout a considerable amount of online work before obtaining any user credentials. We implemented our proposal as a replacement for the built-inFirefox password manager, and provide performance measurements andthe results from experiments with large real-world password sets to evaluate the feasibility and effectiveness of our approach. Kamouage is wellsuited to become a standard architecture for password managers on mobile devices.

6.Honeywords:Making Password-Cracking Detectable

Ari Juels
RSA Labs
Cambridge, MA 02142
ajuels@rsa.com
Ronald L. Rivest
MIT CSAIL
Cambridge, MA 02139rivest@mit.edu
May 2, 2013
Version 2.0

Abstract

We suggest a simple method for improving the security of hashed passwords: the maintenance of additional honeywords" (false passwords) associated with each user'saccount. An adversary who steals able of hashed pass-words and inverts the hash function cannot tell if he hasfound the password or a honeyword. The attempted useof a honeyword for login sets o_ an alarm. An auxiliary server (the honeychecker") can distinguish the user password from honeywords for the login routine, and will setan alarm if a honeyword is submitted.

7.The science of guessing: analyzing an anonymized corpus of 70 million passwords

Joseph Bonneau
Computer Laboratory
University of Cambridge
jcb82@cl.cam.ac.uk

ABSTRACT

We report on the largest corpus of user-chosenpasswords ever studied, consisting of anonymized passwordhistograms representing almost 70 million Yahoo! users, mitigatingPrivacy concerns while enabling analysis of dozens ofsubpopulations based on demographic factors and site usagecharacteristics. This large data set motivates a thorough statisticaltreatment of estimating guessing difficulty by samplingfrom a secret distribution. In place of previously used metricssuch as Shannon entropy and guessing entropy, which cannotbe estimated with any realistically sized sample, we developpartial guessing metrics including a new variant of guessworkparameterized by an attacker's desired success rate. Our newmetric is comparatively easy to approximate and directlyrelevant for security engineering. By comparing passworddistributions with a uniform distribution which would provideequivalent security against different forms of guessing attack,we estimate that passwords provide fewer than 10 bits ofsecurity against an online, trawling attack, and only about 20bits of security against an optimal offline dictionary attack.We find surprisingly little variation in guessing difficulty;every identifiable group of users generated a comparablyweak password distribution. Security motivations such as theregistration of a payment card have no greater impact thandemographic factors such as age and nationality. Even proactiveefforts to nudge users towards better password choiceswith graphical feedback make little difference. More surprisingly,even seemingly distant language communities choose thesame weak passwords and an attacker never gains more thana factor of 2 efficiency gain by switching from the globallyoptimal dictionary to a population-specific lists.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

Cloud storage services have become increasingly popular. Because of the importance of privacy, many cloud storage encryption schemes has not much of security to store the data safely.

3.1.1DISADVANTAGES:

- Security is very low, so that the attacker can easily hack the passwords of the users and can do anything.
- Passwords can be hacked using guessing attacks

3.2 PROPOSED SYSTEM:

We include two processes. 1. As per the Paper Compromised Social networks Accounts are tracked & detected. 2. If hacker attacks the Genuine user then our allows the attacker to proceed further until our system captures all the important information about the attacker. We generate Honeywords based on the user info provided and the original password is converted into another format and stored along with the Honeywords. We deploy Intermediate server, Shopping server for purchase and Cloud server for maintaining user account details. Attacker who knows the E mail account of original user can easily reset the password of the cloud server. Attacker is invited to do attack in this Project, so as to find him out very easily. Now attacker logins into the purchase portal, where he is been tracked unknowingly & he is allowed to do purchase. Server identifies the attacker and sends the info to the Original owner and also it blocks the attacker even doing transaction from his original account.

3.2.2 ADVANTAGES:

- The attackers are not able to guess and hack the passwords.
- It provide high security to data owner
- Hacker ip address and address sent original user via email

CHAPTER 4

HONEY WORDS DIAGRAMS

4.1 ARCHITECTURE DIAGRAM:

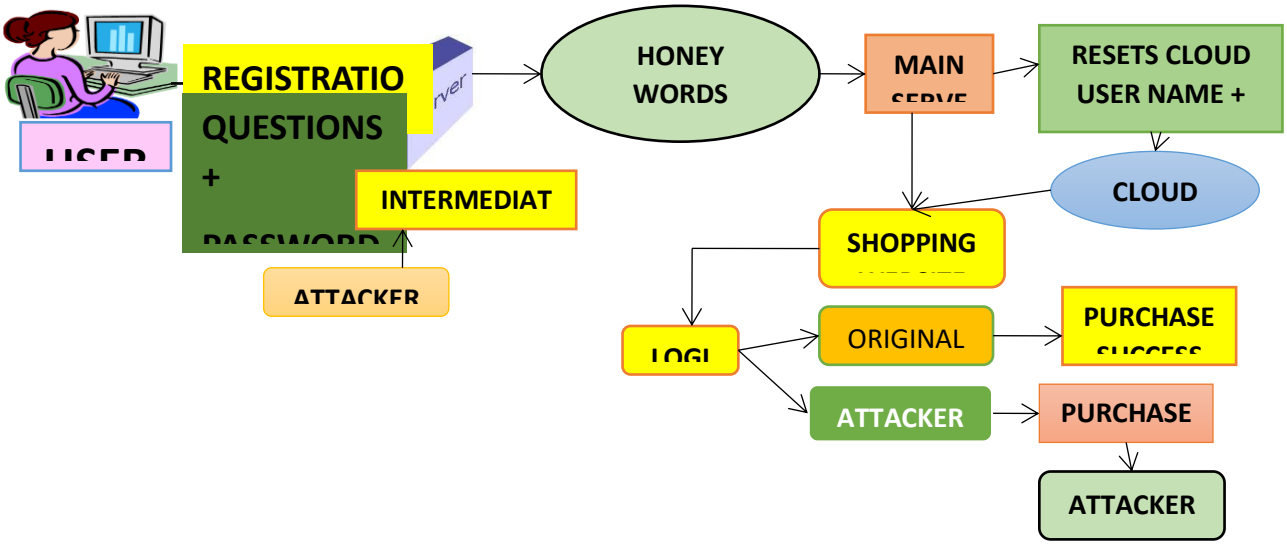


Fig 4.1.1 SYSTEM ARCHITECTURE

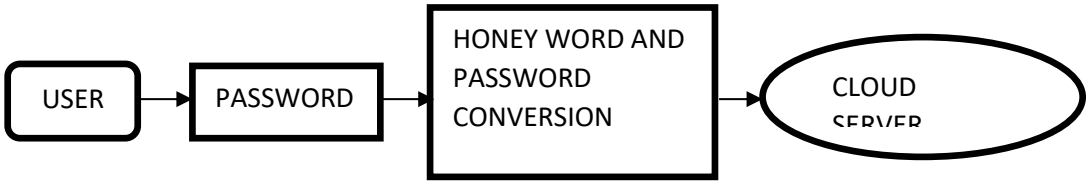
The overall architecture represents the working procedure of the entire system. User will register their details on the portal with two email ids like primary and secondary account. while user login with correct user id password they can purchase the product with normal portal or else if any one try to login user id with fake password by trying more than three system will automatically divert them into fake portal and fetch the fake user's delivery address and IP of the system.

4.2 DATA FLOW DIAGRAM:

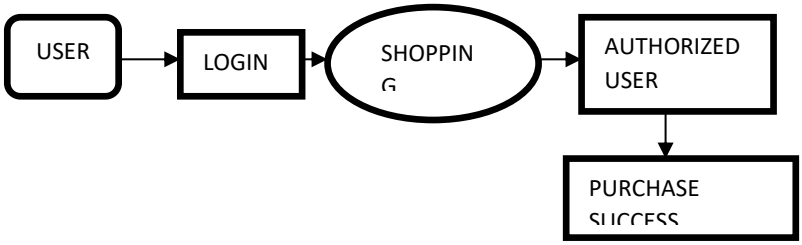
DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. A process receives input data and produces output with a different content or form. Processes can be as simple as collecting input data and saving in the database. A data store or data repository is used in a data-flow diagram to represent a situation when the system must retain data because one or more processes need to use the stored data in a later time.

LEVEL 1:

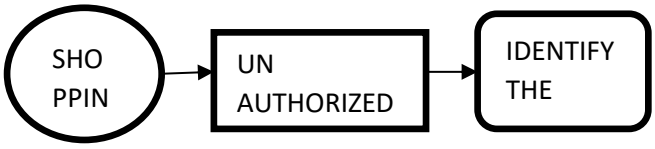
LEVEL 2:



LEVEL 3:



LEVEL 4:



4.3 UML DIAGRAMS:

UML is simply anther graphical representation of a common semantic model. UML provides a comprehensive notation for the full lifecycle of object-oriented development.

ADVANTAGES

- To represent complete systems (instead of only the software portion) using object oriented concepts
- To establish an explicit coupling between concepts and executable code
- To take into account the scaling factors that are inherent to complex and critical systems
- To creating a modeling language usable by both humans and machines

UML defines several models for representing systems

- The class model captures the static structure
- The state model expresses the dynamic behavior of objects
- The use case model describes the requirements of the user
- The interaction model represents the scenarios and messages flows
- The implementation model shows the work units
- The deployment model provides details that pertain to process allocation

4.4 USECASE DIAGRAM

Use case diagrams overview the usage requirement for system. they are useful for presentations to management and/or project stakeholders, but for actual development you will find that use cases provide significantly more value because they describe “the meant” of the actual requirements. A use case describes a sequence of action that provide something of measurable value to an action and is drawn as a horizontal ellipse.

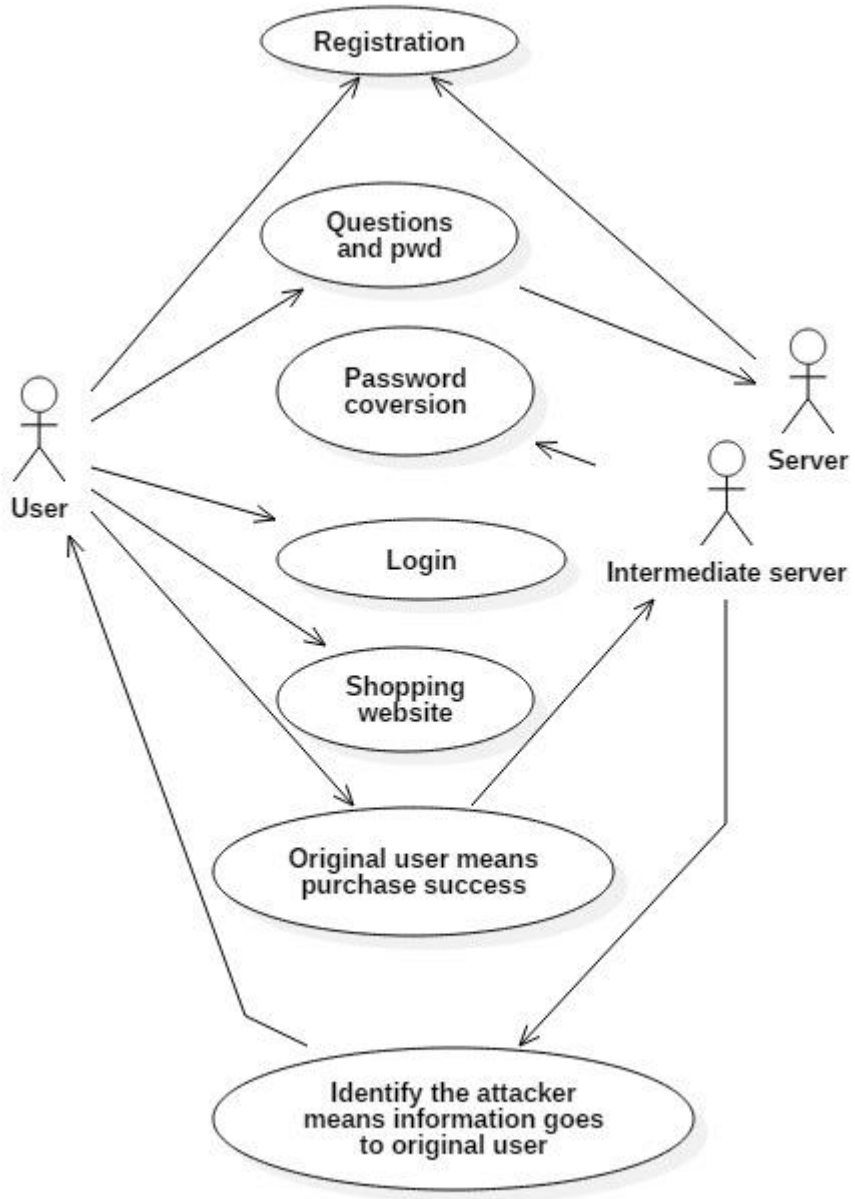


FIG 4.4 USE CASE DIAGRAM

4.5 SEQUENCE DIAGRAM

Sequence diagram model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and commonly used for both analysis and design purpose. Sequence diagram are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system.

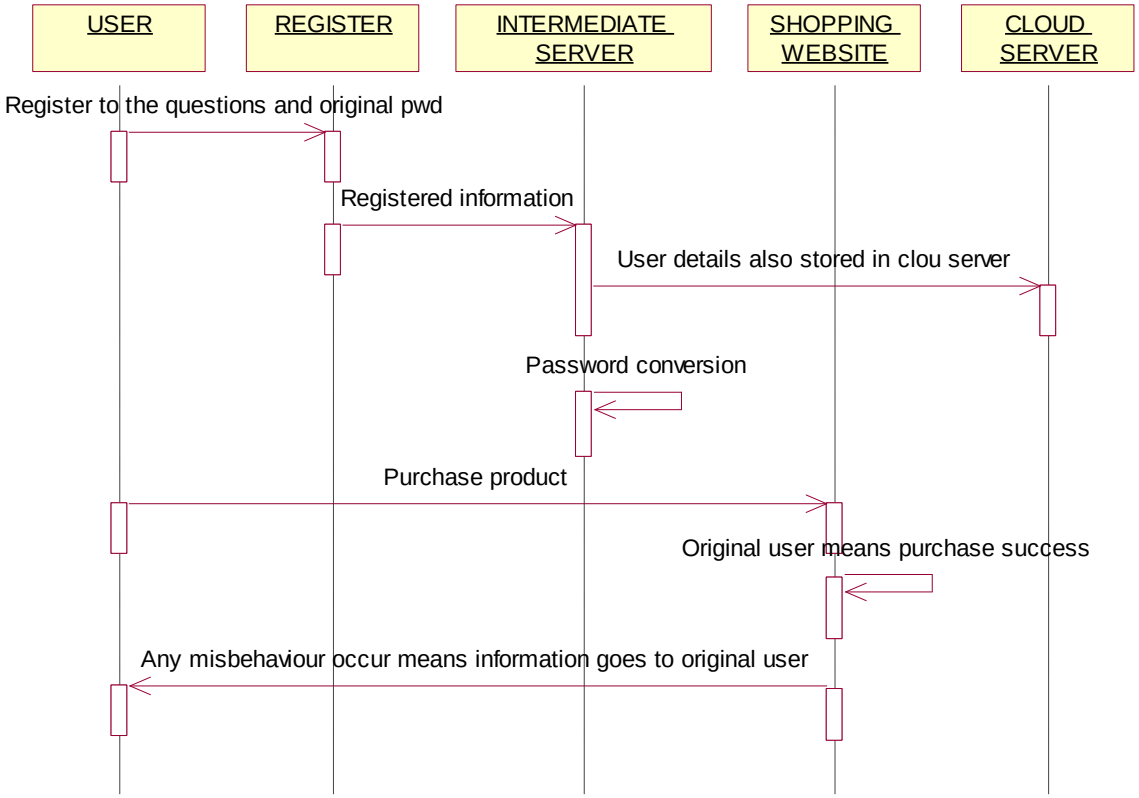


FIG 4.5 SEQUENCE DIAGRAM

4.6 ACTIVITY DIAGRAM:

Activity diagram are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. The activity diagrams can be used to describe the business and operational step-by-step

workflows of components in a system. Activity diagram consist of Initial node, activity final node and activities in between.

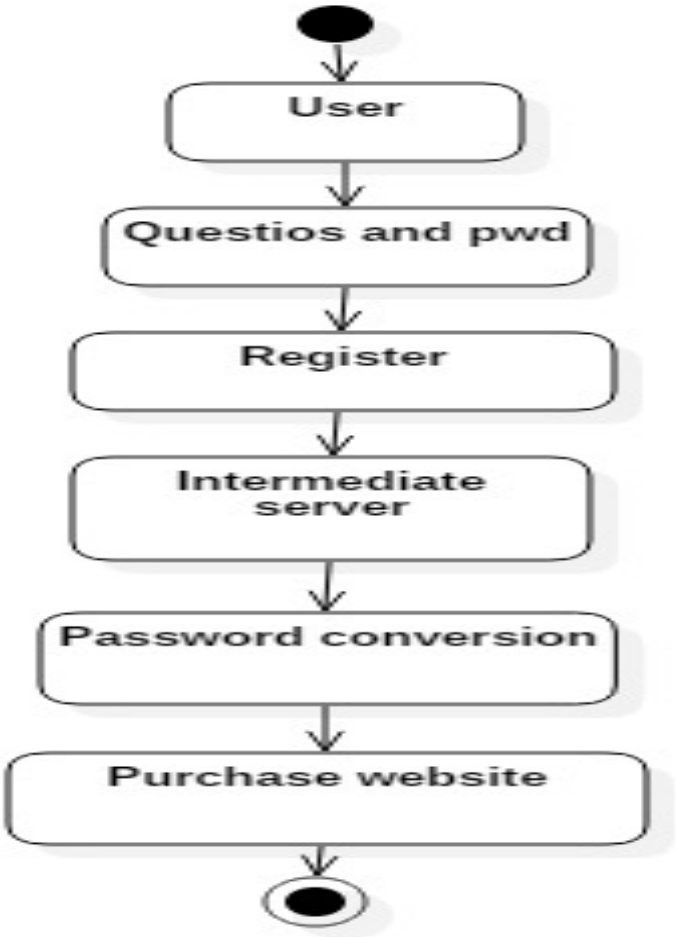


FIG 4.6 ACTIVITY DIAGRAM

4.7 COLLABORATION DIAGRAM

Another type of interaction diagram is the collaboration diagram. A collaboration diagram represents a collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchange among the objects within the collaboration to achieve a desired outcome.

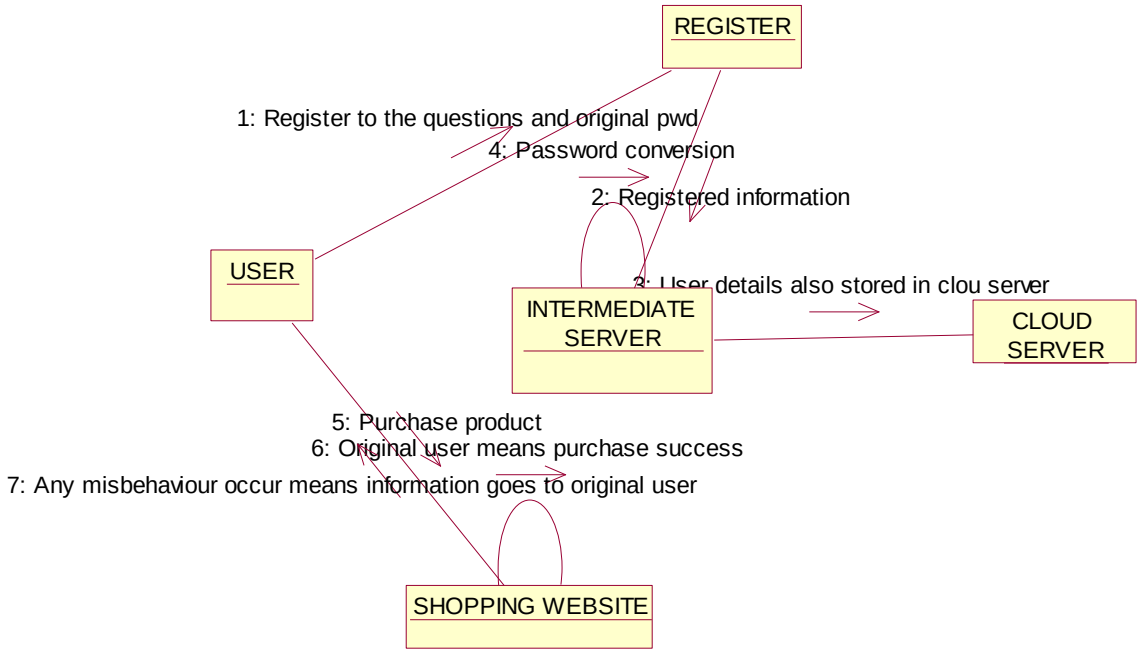


FIG 4.7 COLLBORATION DIAGRAM

4.8 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- OPERATIONAL FEASIBILITY

4.9 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

4.10 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

4.11 OPERATIONAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4.12 PROBLEM DEFINITION AND METHODOLOGY

This chapter deals with the problem definition of the project and methodology used in this project. Problem definition describes the detailed information about the project. Methodology that has been adapted in this project is discussed in methodology.

In previous system there is no identification of hacker, also does not detect the hacker ip address. Considering the modeling syntax method, one can concludethat the honeyword system loses its effectiveness against such passwords, i.e., the correct password has become noticeably recognized by an adversary. In fact, this problem seems an inherent weakness of randomly replacement based honey word methods. Since character groups or individual characters are replaced by picked character/characters, the content integrity of such passwords would be broken and the correct password becomes quite salient.

CHAPTER 5

SYSTEM SPECIFICATIONS

LANGUAGE SPECIFICATION:

This chapter describes the requirement analysis in accordance with the input and the resources and it also describes the implementation of the project with the technology used.

REQUIREMENT ANALYSIS

Requirement analysis determines the requirements of a new system. This project analyses on product and resource requirement, which is required for this successful system. The product requirement includes input and output requirements it gives the wants in term of input to produce the required output. The resource requirements give in brief about the software and hardware that are needed to achieve the required functionality.

5.1Hardware Environment

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the systems do and not how it should be implemented.

- Hard disk : 120 GB
- Monitor : 15’ color with vgi card support
- Ram : Minimum 256 MB
- Processor : Pentium iv and above (or) equivalent

- Processor speed : Minimum 500 MHZ

5.2 Software Environment

The software requirements are the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's and tracking the team's progress throughout the development activity.

- Operating system : Windows XP
- Languages : Java
- Data Base : Mysql
- IDE : Net Beans

OVERVIEW OF THE MYSQL DATABASE MANAGEMENT SYSTEM

WHAT IS MYSQL?

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site (http://www.mysql.com/) provides the latest information about MySQL software.

- MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by MySQL AB. MySQL AB is a commercial company, founded in 1995 by the MySQL developers. It is a second generation Open Source company that unites Open Source values and methodology with a successful business model.
- The MySQL® software delivers a very fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) database server. MySQL

Server is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software. MySQL is a registered trademark of MySQL AB.

- The MySQL software is Dual Licensed. Users can choose to use the MySQL software as an Open Source product under the terms of the GNU General Public License or can purchase a standard commercial license from MySQL AB.

The company name was given after co-founder Monty Widenius's daughter, My, the SQL is "Structured Query Language.", AB is Sweedish for limited partnership.

• MYSQL IS A DATABASE MANAGEMENT SYSTEM.

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

• MySQL databases are relational.What is MySQL?

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and "pointers" between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of "MySQL" stands for "Structured Query Language". SQL is the most common standardized language used to access

databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL-92” refers to the standard released in 1992, “SQL:1999” refers to the standard released in 1999, and “SQL:2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time

MySQL Overview

What does it do?

- MySQL is a database management system.

A database is a structured collection of data. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server.

- **MySQL is a relational database management system.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of “MySQL” stands for “Structured Query Language.”

- **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. If you want to redistribute altered versions MySQL, other licences are available

- **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

- **A large amount of contributed MySQL software is available.**

MySQL is a widely supported database, and very likely that most of your applications supports it.

MySQL Architecture

Three layer model:

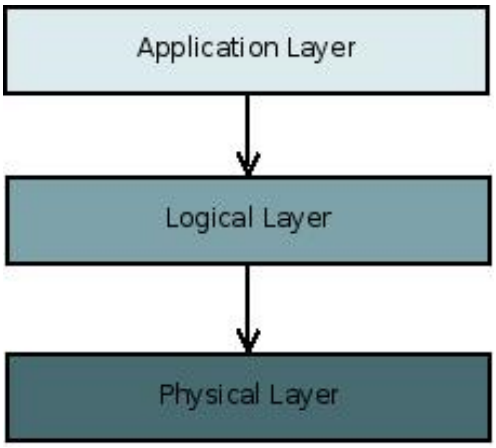


FIG 5.2 THREE LAYER MODEL

- The application layer contains common network services for connection handling, authentication and security. This layer is where different clients interact with MySQL these clients can written in different API's:.NET, Java, C, C++, PHP, Python, Ruby, Tcl, Eiffel, etc...

- The Logical Layer is where the MySQL intelligence resides, it includes functionality for query parsing, analysis, caching and all built-in functions (math, date...). This layer also provides functionality common across the storage engines.

- The Physical Layer is responsible for storing and retrieving all data stored in "MySQL". Associated with this layer are the storage engines, which MySQL interacts with very basic standard API's. Each storage engine has it strengths and weakness, some of this engines are MyISAM, InnoDB, CSV, NDB Cluster, Falcon, etc...

SOME INTERNAL COMPONENTS:

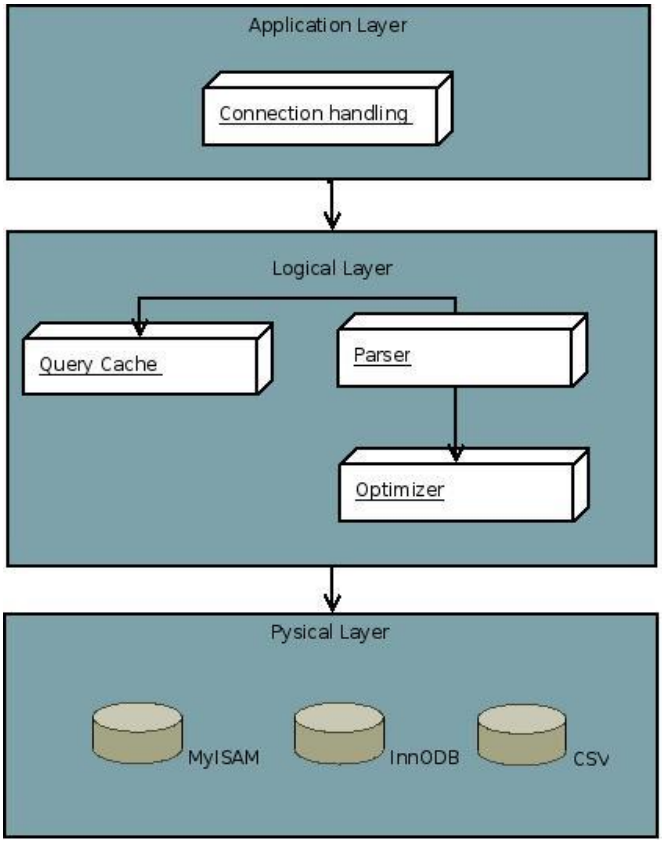


FIG 5.3 INTERNAL COMPONENTS

Each client connection gets its own thread within the server process.

When clients (applications) connect to the MySQL server, the server needs to authenticate them.

Before even parsing the query, though, the server consults the query cache, which only stores SELECT statements, along with their result sets.

The storage engine does affect how the server optimizes query.

Storage engines:

MySQL supports several storage engines that act as handlers for different table types. MySQL storage engines include both those that handle transaction-safe tables and those that handle non-transaction-safe tables.

Feature	MyISAM	Memory	InnoDB	Archive	NDB
Storage limits	256TB	RAM	64TB	None	384EB
Transactions	No	No	Yes	No	Yes
Locking granularity	Table	Table	Row	Row	Row

MySQL Files:

In Linux the configuration file is typically located at /etc/mysql/my.cnf, but may vary for the different Linux flavours, this also applies to the database files them selves which are located in the /var/lib/MySQL.

Regardless of the storage engine, every MySQL table you create is represented, on disk, by a .frm file, which describes the table's format (i.e. the table definition). The file bears the same name as the table, with a .frm extension. The .frm format is the same on all platforms but in the description of the .frm format that follows, the examples come from tables created under the Linux operating system.

/var/lib/mysql/db.frm #Table definition

/var/lib/mysql/db.MYD #MyISAM data file

/var/lib/mysql/db.MYI #MyISAM Index file

/var/lib/mysql/ibdata1 #Innodb data file

MySQL Cluster Overview

MySQL Cluster is a technology that enables clustering of in-memory databases in a shared-nothing system. The shared-nothing architecture enables the system to work with very inexpensive hardware, and with a minimum of specific requirements for hardware or software.

MySQL Cluster is designed not to have any single point of failure. In a shared-nothing system, each component is expected to have its own memory and disk,

and the use of shared storage mechanisms such as network shares, network file systems, and SANs is not recommended or supported.

MySQL Cluster integrates the standard MySQL server with an in-memory clustered storage engine called NDB (which stands for “Network DataBase”). In our documentation, the term NDB refers to the part of the setup that is specific to the storage engine, whereas “MySQL Cluster” refers to the combination of one or more MySQL servers with the NDB storage engine. A MySQL Cluster consists of a set of computers, known as hosts, each running one or more processes. These processes, known as nodes, may include MySQL servers (for access to NDB data), data nodes (for storage of the data), one or more management servers, and possibly other specialized data access programs. The relationship of these components in a MySQL Cluster is shown here:

5.3 N-Tier Architecture

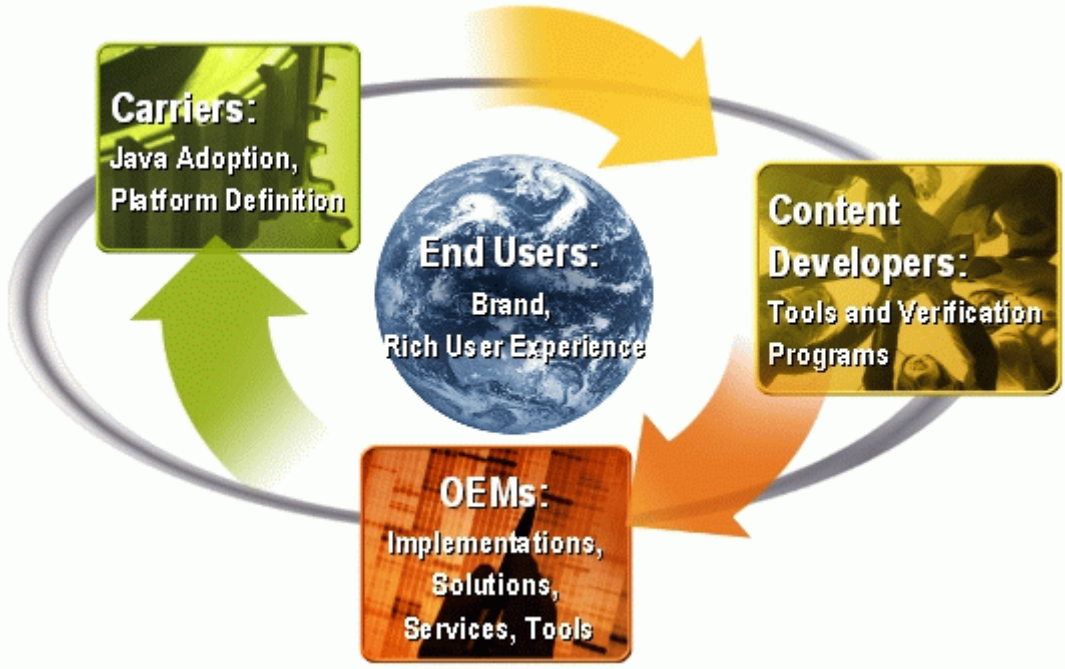


FIG 5.3 N-Tier Architecture

5.3.1 Java

The Java platform is the ideal platform for network computing. Running across all platforms -- from servers to cell phones to smart cards -- Java technology unifies business infrastructure to create a seamless, secure, networked platform for your business. The Java platform benefits from a massive community of developers and supporters that actively work on delivering Java technology-based products and services as well as evolving the platform through an open, community-based, standards organization known as the Java Community Process program. You can find Java technology in cell phones, on laptop computers, on the Web, and even trackside at Formula One Grand Prix races. The fact is today, you can find Java technology just about everywhere!

5.3.2 Business Benefits

- A richer user experience - Whether you're using a Java technology-enabled mobile phone to play a game or to access your company's network, the Java platform provides the foundation for true mobility. The unique blend of mobility and security in Java technology makes it the ideal development and deployment vehicle for mobile and wireless solutions.
- The ideal execution environment for Web services - The Java and XML languages are the two most extensible and widely accepted computing languages on the planet, providing maximum reach to everyone, everywhere, every time, to every device and platform.
- Enabling business from end to end - Java offers a single, unifying programming model that can connect all elements of a business infrastructure.

CHAPTER 6

6.1 FEATURES OF JAVA

Java is a programming language originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code(class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is general-purpose, concurrent, class-based, and object-oriented, and is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere". Java is considered by many as one of the most influential programming languages of the 20th century, and widely used from application software to web application.

The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1995. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of their Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java and GNU Class path.

6.1.1 JAVA PROGRAMMING LANGUAGE

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented

- Portal
- Distributed
- High performance

Each of the preceding buzzwords is explained in *The Java Language Environment*, a white paper written by James Gosling and Henry Chilton. In the Java programming language, all source code is first written in plain text files ending with the .java extension. Those source files are then compiled into .class files by the javac compiler. A .class file does not contain code that is native to your processor; it instead contains *byte codes* — the machine language of the Java Virtual Machine (Java VM). The java launcher tool then runs your application with an instance of the Java Virtual Machine.

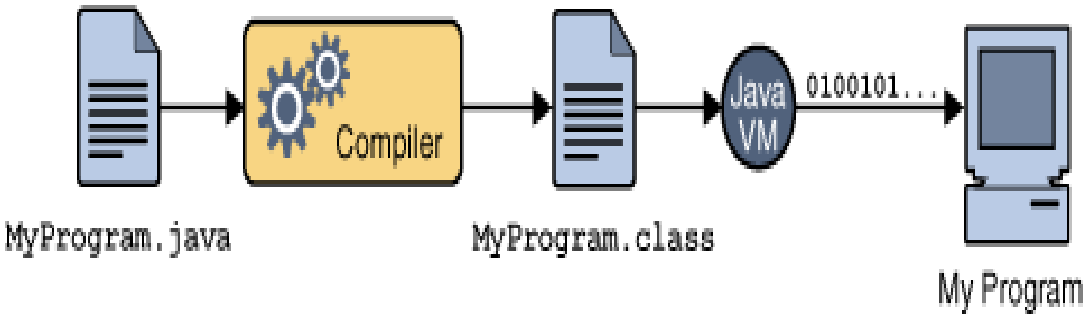


Fig 6.1.1 Software Development Process.

Because the Java VM is available on many different operating systems, the same .class files are capable of running on Microsoft Windows, the Solaris Operating System (Solaris OS), Linux, or Mac OS. Some virtual machines, such as the Java Hotspot virtual machine, perform additional steps at runtime to give your application a performance boost. This includes various tasks such as finding performance bottlenecks and recompiling (to native code) frequently used sections of code.

6.2 JAVA PLATFORM

A *platform* is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Microsoft Windows, Linux, Solaris OS, and Mac OS. Most platforms can be described as a combination of the operating system and underlying hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The *Java Virtual Machine*
- The *Java Application Programming Interface* (API)

6.2.1 JAVA VIRTUAL MACHINE

A **Java virtual machine** (**JVM**) is a virtual machine that can execute Java byte code. It is the code execution component of the Java software platform. A Java virtual machine is a program which executes certain other programs, namely those containing Java byte code instructions. JVMs are most often implemented to run on an existing operating system, but can also be implemented to run directly on hardware. A JVM provides an environment in which Java byte code can be executed, enabling such features as automated exception handling, which provides root-cause debugging information for every software error (exception), independent of the source code.

A JVM is distributed along with a set of standard class libraries that implement the Java application programming interface (API). These libraries, bundled together with the JVM, form the Java Runtime Environment (JRE). JVMs are available for many hardware and software platforms. The use of the same byte code for all JVMs on all platforms allows Java to be described as a writeonce, runanywhere programming language, versus writeonce,

compileanywhere, which describes cross-platform compiled languages. Thus, the JVM is a crucial component of the Java platform.

6.2.2 JAVA API

The API is a large collection of ready-made software components that provide many useful capabilities. It is grouped into libraries of related classes and interfaces; these libraries are known as *packages*.

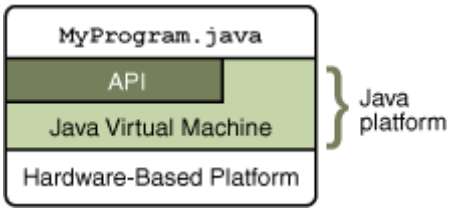


Fig 6.2.2 API and JVM

As a platform-independent environment, the Java platform can be a bit slower than native code. However, advances in compiler and virtual machine technologies are bringing performance close to that of native code without threatening portability.

6.3 JAVA RUNTIME ENVIRONMENT

The Java Runtime Environment, or JRE, is the software required to run any application deployed on the Java Platform. End-users commonly use a JRE in software packages and Web browser plug-in. Sun also distributes a superset of the JRE called the Java 2 SDK (more commonly known as the JDK), which includes development tools such as the Java compiler, Javadoc,Jar and debugger.

One of the unique advantages of the concept of a runtime engine is that errors (exceptions) should not 'crash' the system. Moreover, in runtime engine environments such as Java there exist tools that attach to the runtime engine and every time that an exception of interest occurs they record debugging information

that existed in memory at the time the exception was thrown (stack and heap values). These Automated Exception Handling tools provide 'root-cause' information for exceptions in Java programs that run in production, testing or development environments.

6.4 JAVASERVER PAGE

Java Server Pages (JSPs) are server-side Java EE components that generate responses, typically HTML pages, to HTTP requests from clients. JSPs embed Java code in an HTML page by using the special delimiters <% and %>. A JSP is compiled to a Java *servlet*, a Java application in its own right, the first time it is accessed. After that, the generated servlet creates the response.

6.5 SWING APPLICATION

Swing is a graphical user interface library for the Java SE platform. It is possible to specify a different look and feel through the pluggable look and feel system of Swing. Clones of Windows, GTK+ and Motif are supplied by Sun. Apple also provides an Aqua look and feel for Mac OS X. Where prior implementations of these looks and feels may have been considered lacking, Swing in Java SE 6 addresses this problem by using more native GUI widget drawing routines of the underlying platforms.

This example Swing application creates a single window with "Hello, world!" inside:

```
// Hello.java (Java SE 5)
 import javax.swing.*;
 public class Hello extends JFrame {
PublicHello () {
```

```
setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);

add(new JLabel("Hello, world!"));

pack();

 } public static void main(String[] args) {

new Hello().setVisible(true);

}}
```

The first import includes all of the public classes and interfaces from the javax.swing package. The Hello class extends the Frame class; the Frame class implements a window with a title bar and a close control. The Hello**()** constructor initializes the frame by first calling the super class constructor, passing the parameter "hello", which is used as the window's title. It then calls the setDefaultCloseOperation **(**into**)**method inherited from Frame to set the default operation when the close control on the title bar is selected to Window Constants. EXIT ON CLOSEthis causes the JFrame to be disposed of when the frame is closed (as opposed to merely hidden), which allows the JVM to exit and the program to terminate.

Next, the layout of the frame is set to a Border Layout; this tells Swing how to arrange the components that will be added to the frame. A Label is created for the string "Hello, world!" and the add (Component) method inherited from the Container super class is called to add the label to the frame. The pack () method inherited from the Window super class is called to size the window and lay out its contents. The main **()** method is called by the JVM when the program starts. It instantiates a new Hello frame and causes it to be displayed by calling the set Visible **(**Boolean**)** method inherited from the Component super class with the Boolean parameter true. Once the frame is displayed, exiting the main method does not cause the program to terminate because the AWT event dispatching

thread remains active until all of the Swing top-level windows have been disposed.

6.6 JAVA PLATFORM, STANDARD EDITION

Java Platform, Standard Edition or **Java SE** is a widely used platform for programming in the Java language. It is the Java Platform used to deploy portable applications for general use. In practical terms, Java SE consists of a virtual machine, which must be used to run Java programs, together with a set of libraries (or "packages") needed to Allow the use of file systems, networks, graphical interfaces, and so on, from within those programs. Some of the general purpose packages in J2SE are as follows

- Java.lang
- Java.io
- Java.math
- Java.net
- Java.util

6.7 INTRODUCTION TO APPACHE TOM CAT

Tomcat, developed by Apache (**www.apache.org**), is a standard reference implementation for Java servlets and JSP. It can be used standalone as a Web server or be plugged into a Web

Server like Apache, Netscape Enterprise Server, or Microsoft Internet Information Server. There are many versions of Tomcat. This tutorial uses Tomcat 5.5.9 as an example. The tutorial should also apply to all later versions of Tomcat. To start Tomcat, you have to first set the JAVA_HOME environment variable to the JDK home directory using the following command. The JDK home directory is where your JDK is stored. On my computer, it is c:\Program Files\jdk1.6.0_24.

The apache tomcat server is an open source, java-based web application container that was created to servlets and java server page (JSP) web applications. It was created under the apache-Jakarta subproject; however, due to its popularity, it is now hosted as separate apache project, where it is supported and enhanced by a group of volunteers from the open source java community.

The Tomcat Manager Web Application

The Tomcat manager web application is packaged with the tomcat server. It is installed in the context path of manager and provides the basic functionality to manage web applications running in the tomcat server from any web browser. Some of the provided functionality includes the ability to install, start, stop, remove, and report on web applications. Covers the details of the tomat manager web application.

The Server

The first container element referenced in this snippet is the <Server> element. It represents the entire Catalina servlet engine and is used as a top-level element for a single Tomcat instance. The <Server> element may contain one or more <Service> containers.

The Service

The next container element is the <Service> element, which holds a collection of one or more <Connector> elements that share a single <Engine> element. N-number of <Service> elements may be nested inside a single <Server> element.

The Connector

The next type of element is the <Connector> element, which defines the class that does the actual Handling requests and responses to and from a calling client application.

The Engine

The third container element is the <Engine> element. Each defined <Service> can have only one <Engine> element and this single <Engine> component handles all requests received by all of the defined <Connector> components defined by a parent service.

The Host

The <Host> element defines the virtual hosts that are contained in each instance of a Catalina <Engine>. Each <Host> can be a parent to one or more web applications, with each being represented by a <Context> component.

The Context

The <Context> element is the most commonly used container in a Tomcat instance. Each <Context> element represents an individual web application that is running within a defined <Host>. There is no limit to the number of contexts that can be defined within a <Host>.

Java Web Applications

The main function of the Tomcat server is to act as a container for Java web applications. Therefore, before we can begin our Tomcat-specific discussion, a brief introduction as to exactly what web applications are is in order. The concept of a web application was introduced with the release of the Java servlets specification 2.2. According to this specification, "a web application is a collection of servlets, html pages, classes, and other resources that can be bundled and run on multiple containers from multiple vendors." What this really means is that a web application is a container that can hold any combination of the following list of objects:

- Servlets
- Java Server Pages (JSPs)
- Utility classes
- Static documents, including HTML, images, JavaScript libraries, cascading style sheets, and so on
- Client-side classes
- Meta-information describing the web application

One of the main characteristics of a web application is its relationship to the Servlets Context. Each web application has one and only one Servlets Context. This relationship is controlled by the servlets container and guarantees that no two web applications will clash when accessing objects in the Servlets Context.

CHAPTER 7

MODULES:

- A modular design reduces complexity, facilities change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different part of system. Software with effective modularity is easier to develop because function may be compartmentalized and interfaces are simplified. Software architecture embodies modularity that is software is divided into separately named and addressable components called modules that are integrated to satisfy problem requirements.

- Modularity is the single attribute of software that allows a program to be intellectually manageable. The five important criteria that enable us to evaluate a design method with respect to its ability to define an effective modular design are: Modular decomposability, Modular Comps ability, Modular Understandability, Modular continuity, Modular Protection.
- The following are the modules of the project, which is planned in aid to complete the project with respect to the proposed system, while overcoming existing system and also providing the support for the future enhancement.

7.1 MODULES:

1. USER REGISTRATION
2. SERVER
3. HONEY WORDS GENERATION
4. INTERMEDIATE SERVER DEPLOYMENT & SHOPPING SERVER DEPLOYMENT
5. PASSWORD HACKING PROCESS
6. IDENTIFICATION OF ATTACKERS & AVOIDANCE OF DDOS ATTACK

7.2MODULES DESCRIPTION:

1. USER REGISTRATION

In this module we are going to create a User application by which the User is allowed to access the data from the Server. Here first the User wants to create an account and then only they are allowed to access the Network. Once the User creates an account, they are allowed to login into their account to access the application. Based on the User's request, the Server will respond to the User. All the User details will be stored in the Database of the Server. In this module bank user details are registered with the fields like username, password, and personal details with some set of questions and answers. These details are saved into the server. After proper registration only the user cal allowed to login into the server.

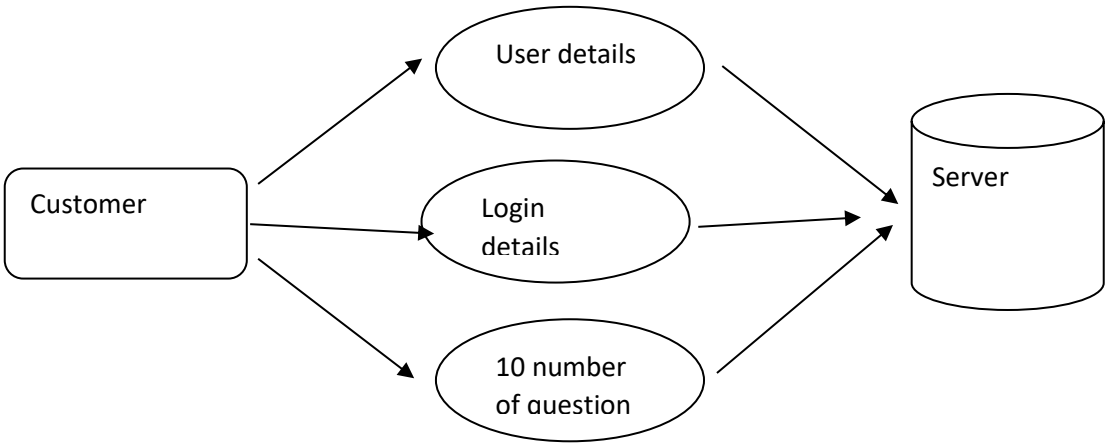


FIG 7.2.1 USER REGISTRATION

2. SERVER

In this Server module server will deployed to access the database and web based application. Server will verify the users and generates honey word for save the users password. In case illegal actions happened means server will generates alert and intimate it to user. The Server will monitor the entire User's information in their database and verify them if required. Also the Server will store the entire User's information in their database. Also the Server has to establish the connection to communicate with the Users. The Server will update the each

User's activities in its database. The Server will authenticate each user before they access the Application. So that the Server will prevent the Unauthorized User from accessing the Application.

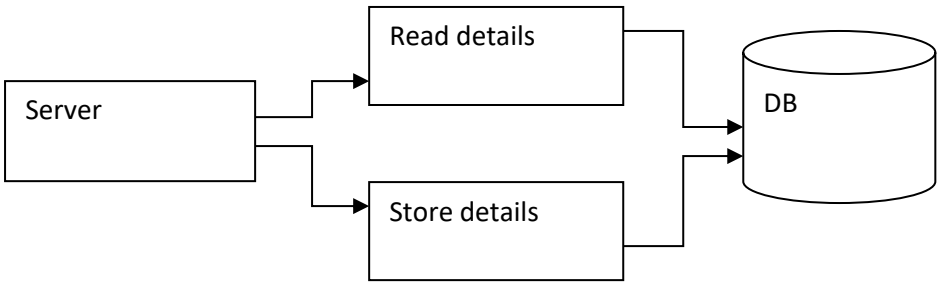


FIG 7.2.2 SERVER FLOW CHART

3. HONEY WORDS GENERATION

Password files have got a lot of security problem that has affected millions of users as well as many companies. Password file is generally stored in encrypted format, if a password file is stolen or theft by using the password cracking techniques and decryption technique it is easy to capture most of the plaintext and encrypt passwords.SoIn this module we deployed honey word creations. That is the user's password and registered questions are combined and then it will generate a key as unknown Name.

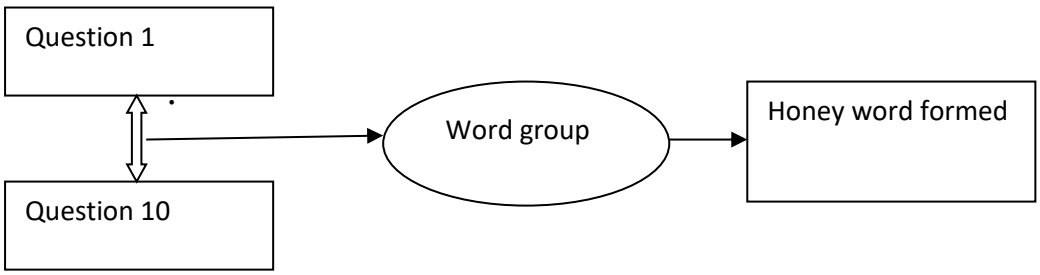


FIG 7.2.3 HONEY WORDS GENERATION

4. INTERMEDIATE SERVER & SHOPPING SERVER DEPLOYMENT

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

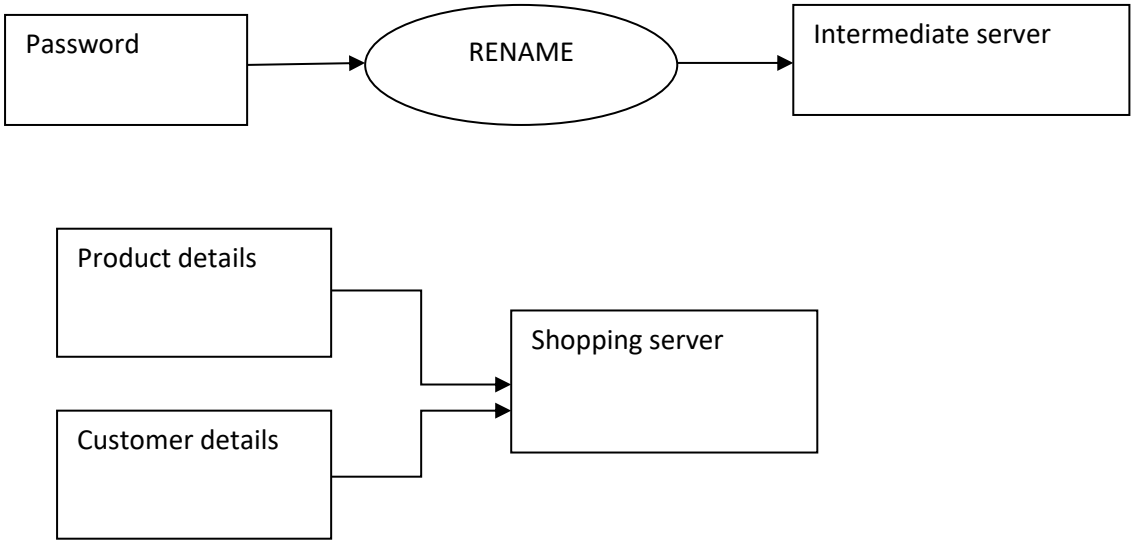


FIG 7.2.4 SERVER DEPLOYMENT

5. PASSWORD HACKING PROCESS

Hacking is the process of recovering passwords from data that has been stored in or transmitted by a computer system. A common approach is to repeatedly try guesses for the password. Another common approach is to say that you have "forgotten" the password and then change it. Password Hacking is blocked in this Module. Because we modifies the users original passwords into unknown Name and saved into server.

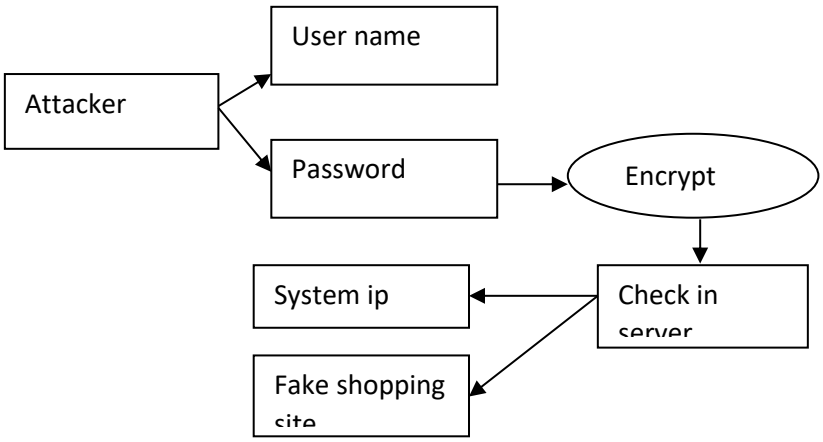


FIG 7.5 PASSWORD HACKING PROCESS

6.IDENTIFICATION OF ATTACKERS & REMOVE DDOS ATTACKS

A distributed denial of service (DDos) attack is an attempt to make an online service unavailable by overwhelming it with traffic from multiple sources. They target a wide variety of important resources, from banks to news websites and present a major challenge to making sure people can publish and access important information. If there is anybody trying with wrong password or any illegal action means server will block that action and intimate to the Specified Users. If the same request comes from same user or from different user's means server will blocks that actions also. This is done in DDOS attack.

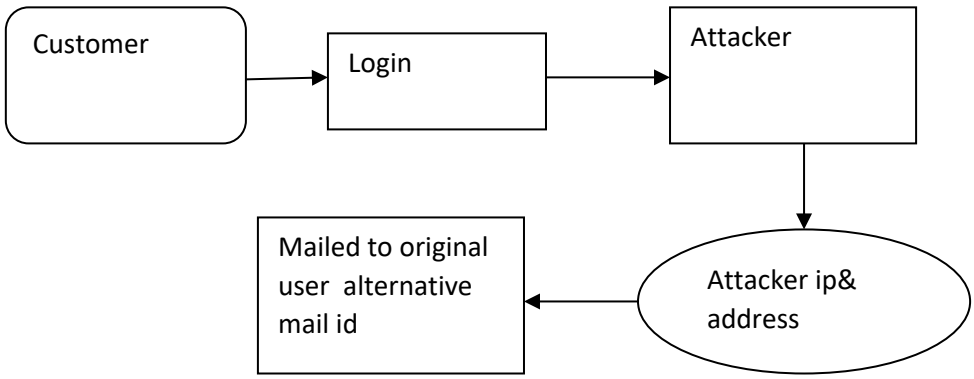


FIG7.6 IDENTIFICATION OF ATTACKERS & REMOVE DDOS ATTACKS

CHAPTER 8

8. APPENDIX

A1 SOURCE CODING:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
packagecom.nura.servlet.user;

importcom.google.gson.Gson;
importcom.nura.dao.impl.UserDtlsDAOImpl;
importcom.nura.dao.impl.UserLikeDAOImpl;
importcom.nura.entity.UserDetails;
importcom.nura.security.JuliusCeaser;
importcom.nura.security.RandomGenerator;
importjava.io.File;
importjava.io.FileWriter;
importjava.io.IOException;
importjava.io.PrintWriter;
importjava.util.ArrayList;
importjava.util.List;
importjavax.servlet.ServletException;
importjavax.servlet.http.HttpServlet;
importjavax.servlet.http.HttpServletRequest;
```

```
importjavax.servlet.http.HttpServletResponse;
importjavax.servlet.http.HttpSession;

/**
 *
 * @author ArunRamya
 */
public class UserRegistration extends HttpServlet {

    /**
    * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
    * methods.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throwsServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
HttpSession session = request.getSession();
try (PrintWriter out = response.getWriter()) {

        /* TODO output your page here. You may use following sample code. */
UserDetailsuserDtls = new UserDetails();
```

```
userDtls.setUserName(request.getParameter("user_name"));
userDtls.setAddress(request.getParameter("address"));
userDtls.setCity(request.getParameter("city"));
userDtls.setPassword(request.getParameter("pwd"));
userDtls.setPhoneNumber(request.getParameter("mobile"));
userDtls.setPostalCode(request.getParameter("zip"));
userDtls.setState(request.getParameter("state"));
userDtls.setUserRole("Customer");
userDtls.setCcNo(request.getParameter("ccno"));

userDtls.setAltMailId(request.getParameter("alt_mail_id"));

        //Get questions and answers
userDtls.setQ1(request.getParameter("q1"));
userDtls.setQ2(request.getParameter("q2"));
userDtls.setQ3(request.getParameter("q3"));
userDtls.setQ4(request.getParameter("q4"));
userDtls.setQ5(request.getParameter("q5"));
userDtls.setQ6(request.getParameter("q6"));
userDtls.setQ7(request.getParameter("q7"));
userDtls.setQ8(request.getParameter("q8"));
userDtls.setQ9(request.getParameter("q9"));
userDtls.setQ10(request.getParameter("q10"));

List<String>dropVal = new ArrayList<>();
dropVal.add(request.getParameter("q1"));
dropVal.add(request.getParameter("q2"));
dropVal.add(request.getParameter("q3"));
```

```
dropVal.add(request.getParameter("q4"));
dropVal.add(request.getParameter("q5"));
dropVal.add(request.getParameter("q6"));
dropVal.add(request.getParameter("q7"));
dropVal.add(request.getParameter("q8"));
dropVal.add(request.getParameter("q9"));
dropVal.add(request.getParameter("q10"));

   //Generate sec password
intrandVal = new RandomGenerator().randVal();
        String secKey = JuliusCeaser.encodeWord(request.getParameter("pwd"),
randVal);
userDtls.setKeyIndex(randVal);

dropVal.add(secKey);

Gsongson = new Gson();
        String json = gson.toJson(dropVal);
System.out.println("JSON data =>" + json);

String[] getLikes = request.getParameterValues("prd_category");
UserLikeDAOImpl _userLikeImpl = new UserLikeDAOImpl();
for (String likes : getLikes) {
System.out.println("Like:-" + likes);
            //Save user likes
com.nura.entity.UserLikesusrLikes = new com.nura.entity.UserLikes();
usrLikes.setUserId(request.getParameter("user_name"));
usrLikes.setUserLike(likes);
```

```
            _userLikeImpl.saveUserLike(usrLikes);


        }
long _usrObjSaved = new UserDtlsDAOImpl().saveUserDtlsRObj(userDtls);
if (_usrObjSaved> 0) {


FileWriterfileWriter        =new FileWriter(new
File(constants.Constants.DROP_FILE_LOCATION));
fileWriter.write(json);
fileWriter.flush();
fileWriter.close();


com.nura.dropbox.AuthenticateUser.upload(constants.Constants.DROP_FILE_
LOCATION, "" + _usrObjSaved);


response.sendRedirect("respPage.jsp?msg=User Details saved");
        } else {
response.sendRedirect("respPage.jsp?msg=User Details not saved. Contact
admin");
        }
      }
   }


   // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on
the + sign on the left to edit the code.">
   /**
    * Handles the HTTP <code>GET</code> method.
    *
    * @param request servlet request
```

```
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throwsServletException, IOException {
processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throwsServletException, IOException {
processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
```

```
     *
     * @return a String containing servlet description
     */
    @Override
public String getServletInfo() {
return "Short description";
    }// </editor-fold>

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
packagecom.nura.servlet.user;

importcom.google.gson.Gson;
importcom.nura.dao.impl.UserDtlsDAOImpl;
importcom.nura.dropbox.AuthenticateUser;
importcom.nura.entity.UserDetails;
importjava.io.BufferedReader;
importjava.io.File;
importjava.io.FileReader;
importjava.io.IOException;
importjava.io.PrintWriter;
importjava.util.List;
importjava.util.logging.Level;
```

```
importjava.util.logging.Logger;
importjavax.servlet.RequestDispatcher;
importjavax.servlet.ServletException;
importjavax.servlet.http.HttpServlet;
importjavax.servlet.http.HttpServletRequest;
importjavax.servlet.http.HttpServletResponse;
importjavax.servlet.http.HttpSession;

/**
 *
 * @author ArunRamya
 */
public class ValidateUser extends HttpServlet {

    /**
    * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
    * methods.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
throwsServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
```

```
HttpSession session = request.getSession();
RequestDispatcherredirectPage = null;
longuserId = 0l;
        String data = null;
        String userPwd = null;
        //Get login attempt
intloginAttempt = 0;

try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
System.out.println(request.getParameter("emalid"));
UserDtlsDAOImpl _userImpl = new UserDtlsDAOImpl();
UserDetails _userDtls = _userImpl.getUserDetails(request.getParameter("emalid"));
userId = _userDtls.getUserId();
System.out.println("User verfication for id =>" + userId);
userPwd = request.getParameter("password");

            //Downloading the file
            File file = AuthenticateUser.download("" + userId, "user.json");

FileReaderfileReader = new FileReader(file);
BufferedReaderbrReader = new BufferedReader(fileReader);
            String line = null;
while ((line = brReader.readLine()) != null) {
data = line;
            }
System.out.println("Drop box data =>" + data);
```

```
try {
loginAttempt = Integer.parseInt(request.getParameter("retryCount"));
System.out.println("Login attempt from urlval =>" + loginAttempt);
        } catch (Exception ex) {
           //ex.printStackTrace();
loginAttempt = 0;
        }
if (loginAttempt>constants.Constants.HACK_RETRY_LIMIT) {
if (_userDtls != null) {
session.setAttribute("User", _userDtls);
session.setAttribute("UserType", "hacker");
session.setAttribute("srch_type", "All");
                //response.sendRedirect("products.jsp");
redirectPage = request.getRequestDispatcher("products.jsp");
redirectPage.forward(request, response);
            }
        } else {
System.out.println("Login attempt val=>" + loginAttempt);
if (request.getParameter("emalid").equals("admin")
&&request.getParameter("password").equals("admin")) {
response.sendRedirect("adminPage.jsp");
            } else {
                _userDtls =
_userImpl.getUserDetails(request.getParameter("emalid"),
request.getParameter("password"));
if (_userDtls != null) {
session.setAttribute("User", _userDtls);
session.setAttribute("UserType", "geniuine");
```

```
session.setAttribute("srch_type", "All");
                    //response.sendRedirect("products.jsp");
redirectPage = request.getRequestDispatcher("products.jsp");
redirectPage.forward(request, response);
                } else {


if (data.contains(userPwd)) {
redirectPage = request.getRequestDispatcher("loginPage.jsp?msg=Invalid Credentials&retryCount=" + ++loginAttempt);
redirectPage.forward(request, response);
                    //response.sendRedirect("loginPage.jsp?msg=Invalid Credentials&retryCount=" + ++loginAttempt);
                } else {
redirectPage = request.getRequestDispatcher("loginPage.jsp?msg=Invalid Credentials&retryCount=0");
redirectPage.forward(request, response);
                    //response.sendRedirect("loginPage.jsp?msg=Invalid Credentials&retryCount=" + 0);
                }
            }
        }
    } catch (Exception ex) {


if (data.contains(userPwd)) {
redirectPage = request.getRequestDispatcher("loginPage.jsp?msg=Invalid Credentials&retryCount=" + ++loginAttempt);
redirectPage.forward(request, response);
            //response.sendRedirect("loginPage.jsp?msg=Invalid Credentials&retryCount=" + ++loginAttempt);
```

```
        } else {
redirectPage = request.getRequestDispatcher("loginPage.jsp?msg=Invalid Credentials&retryCount=0");
redirectPage.forward(request, response);
            //response.sendRedirect("loginPage.jsp?msg=Invalid Credentials&retryCount=" + 0);
        }
      }
   }

   // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
   /**
    * Handles the HTTP <code>GET</code> method.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
   @Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throwsServletException, IOException {
processRequest(request, response);
   }

   /**
    * Handles the HTTP <code>POST</code> method.
```

```
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throwsServletException, IOException {
processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
public String getServletInfo() {
return "Short description";
    }// </editor-fold>

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
```

```
 * and open the template in the editor.
 */
packagecom.nura.servlet.product;

importcom.nura.dao.impl.ProductDtlsDAOImpl;
importcom.nura.entity.ProductDtls;
importjava.io.IOException;
importjava.io.PrintWriter;
importjavax.servlet.ServletException;
importjavax.servlet.http.HttpServlet;
importjavax.servlet.http.HttpServletRequest;
importjavax.servlet.http.HttpServletResponse;

/**
 *
 * @author ArunRamya
 */
public class ViewImage extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
```

```
    */
protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
throwsServletException, IOException {
        // response.setContentType("image/jpg");
try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */


ProductDtlsprdDtls = new ProductDtlsDAOImpl().getProductDtls("HEALTH &
BEAUTY").get(0);
byte[] content = prdDtls.getPrdImg();
response.setContentType(getServletContext().getMimeType(prdDtls.getPrdNa
me()));
response.setContentLength(content.length);
response.getOutputStream().write(content);
        }
    }


    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on
the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throwsServletException, IOException {
processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throwsServletException, IOException {
processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
public String getServletInfo() {
```

```
return "Short description";
    }// </editor-fold>

}



/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
packagecom.nura.servlet.product;

importcom.nura.dao.impl.ProductDtlsDAOImpl;
importcom.nura.entity.ProductDtls;
importjava.io.IOException;
importjava.io.InputStream;
importjava.io.PrintWriter;
importjavax.servlet.ServletException;
importjavax.servlet.annotation.MultipartConfig;
importjavax.servlet.http.HttpServlet;
importjavax.servlet.http.HttpServletRequest;
importjavax.servlet.http.HttpServletResponse;
importjavax.servlet.http.Part;

/**
 *
```

```
 * @author ArunRamya
 */
@MultipartConfig
public class ProductUpload extends HttpServlet {


    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throwsServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
try (PrintWriter out = response.getWriter()) {

ProductDtlsprdDtls = new ProductDtls();
prdDtls.setCategory(request.getParameter("prd_category"));
prdDtls.setProductName(request.getParameter("prd_name"));
prdDtls.setProductQty(request.getParameter("prd_qty"));
prdDtls.setRelatedProduct(request.getParameter("prd_rel"));
        //Getting image bytes
for (Part parts : request.getParts()) {
```

```
InputStreaminputStream = parts.getInputStream();
byte[] getImgytes = new byte[inputStream.available()];
inputStream.read(getImgytes);
prdDtls.setPrdImg(getImgytes);
            String fileName = getFileName(parts);
System.out.println("File name:-" + fileName);
prdDtls.setPrdName(fileName);
        }
prdDtls.setProductPrice(request.getParameter("prd_price"));
if (new ProductDtlsDAOImpl().saveProductDtls(prdDtls)) {
response.sendRedirect("respPage.jsp?msg=Image
Uploaded&page=adminPage.jsp");
        } else {
response.sendRedirect("respPage.jsp?msg=Image                    not
Uploaded&page=adminPage.jsp");
        }

    }
  }

  // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on
the + sign on the left to edit the code.">
  /**
   * Handles the HTTP <code>GET</code> method.
   *
   * @param request servlet request
   * @param response servlet response
   * @throws ServletException if a servlet-specific error occurs
   * @throws IOException if an I/O error occurs
```

```
     */
    @Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throwsServletException, IOException {
processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throwsServletException, IOException {
processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
```

```
    @Override
public String getServletInfo() {
return "Short description";
    }// </editor-fold>

    // Extract file name from content-disposition header of file part
private String getFileName(Part part) {
final String partHeader = part.getHeader("content-disposition");
System.out.println("***** partHeader: " + partHeader);
for (String content : part.getHeader("content-disposition").split(";")) {
if (content.trim().startsWith("filename")) {
returncontent.substring(content.indexOf('=') + 1).trim()
                        .replace("\"", "");
            }
        }
return null;
    }
}



/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
packagecom.nura.servlet.product;

importjava.io.IOException;
```

```
importjava.io.PrintWriter;
importjavax.servlet.ServletException;
importjavax.servlet.http.HttpServlet;
importjavax.servlet.http.HttpServletRequest;
importjavax.servlet.http.HttpServletResponse;
importjavax.servlet.http.HttpSession;

/**
 *
 * @author ArunRamya
 */
public class ProductSessionMgmt extends HttpServlet {

    /**
    * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
    * methods.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
throwsServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
HttpSession session = request.getSession();
```

```
try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            String searchType = request.getParameter("srch_type");
session.setAttribute("srch_type", searchType);
response.sendRedirect("products.jsp");
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throwsServletException, IOException {
processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
```

```
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throwsServletException, IOException {
processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
public String getServletInfo() {
return "Short description";
    }// </editor-fold>

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
```

```
 */
packagecom.nura.servlet.product;

importcom.nura.dao.impl.ProductDtlsDAOImpl;
importcom.nura.dao.impl.PurchaseDtlsDAOImpl;
importcom.nura.entity.UserDetails;
importcom.nura.mail.JavaEmail;
importcom.nura.mail.SendMail;
importjava.io.IOException;
importjava.io.PrintWriter;
importjava.net.InetAddress;
importjava.util.ArrayList;
importjava.util.List;
importjavax.servlet.ServletException;
importjavax.servlet.http.HttpServlet;
importjavax.servlet.http.HttpServletRequest;
importjavax.servlet.http.HttpServletResponse;
importjavax.servlet.http.HttpSession;

/**
 *
 * @author ArunRamya
 */
public class OrderPlacement extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
```

```
    * methods.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */
protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
throwsServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
HttpSession session = request.getSession();
try (PrintWriter out = response.getWriter()) {
        /* TODO output your page here. You may use following sample code. */
UserDetailsuserDtls = (UserDetails) session.getAttribute("User");
        //Get user type hacker or genuine
        String userType = (String) session.getAttribute("UserType");
if (userType.equals("hacker")) {
            //Send mail to orginal user
InetAddressipAddr = InetAddress.getLocalHost();
System.out.println(ipAddr.getHostAddress());
            String addr = request.getParameter("delivery_addr");
            String hostAddr = ipAddr.getHostAddress();

SendMail.main(userDtls.getAltMailId(), "IP detected " + hostAddr
                    + " and Address of the hacker :- " + addr, "Security Breach
Detected change password");
```

```
        } else {
            List<com.nura.entity.PurchaseDtls>purDtlsList = new ArrayList<>();
if (session.getAttribute("cart") != null) {
purDtlsList.addAll((List<com.nura.entity.PurchaseDtls>)
session.getAttribute("cart"));
ProductDtlsDAOImpl _impl = new ProductDtlsDAOImpl();
PurchaseDtlsDAOImpl _purImpl = new PurchaseDtlsDAOImpl();
for (com.nura.entity.PurchaseDtlspurDtls : purDtlsList) {
                    _impl.updateQty(purDtls.getPrdId(), purDtls.getQty());
                    //Saving purchase details in DB
                    _purImpl.savePurchaseDtls(purDtls);
                }
            }
        }
        List<com.nura.entity.PurchaseDtls> temp = new ArrayList<>();
session.setAttribute("cart", null);
response.sendRedirect("respPage.jsp?msg=Order placed");
      }
   }
   // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on
the + sign on the left to edit the code.">
   /**
    * Handles the HTTP <code>GET</code> method.
    *
    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
```

```
     */
    @Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throwsServletException, IOException {
processRequest(request, response);
    }
   @Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throwsServletException, IOException {
processRequest(request, response);
    }
Override
public String getServletInfo() {
return "Short description";
    }// </editor-fold
}
```

A2 SCREENSHOTS

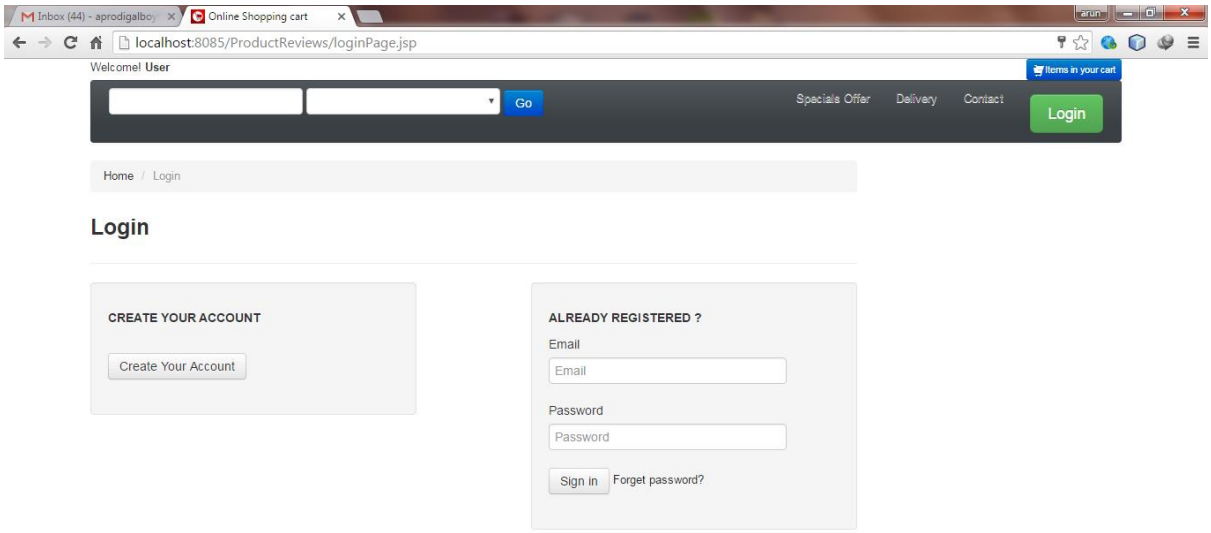


FIG 9.1 LOGIN ACCOUNT

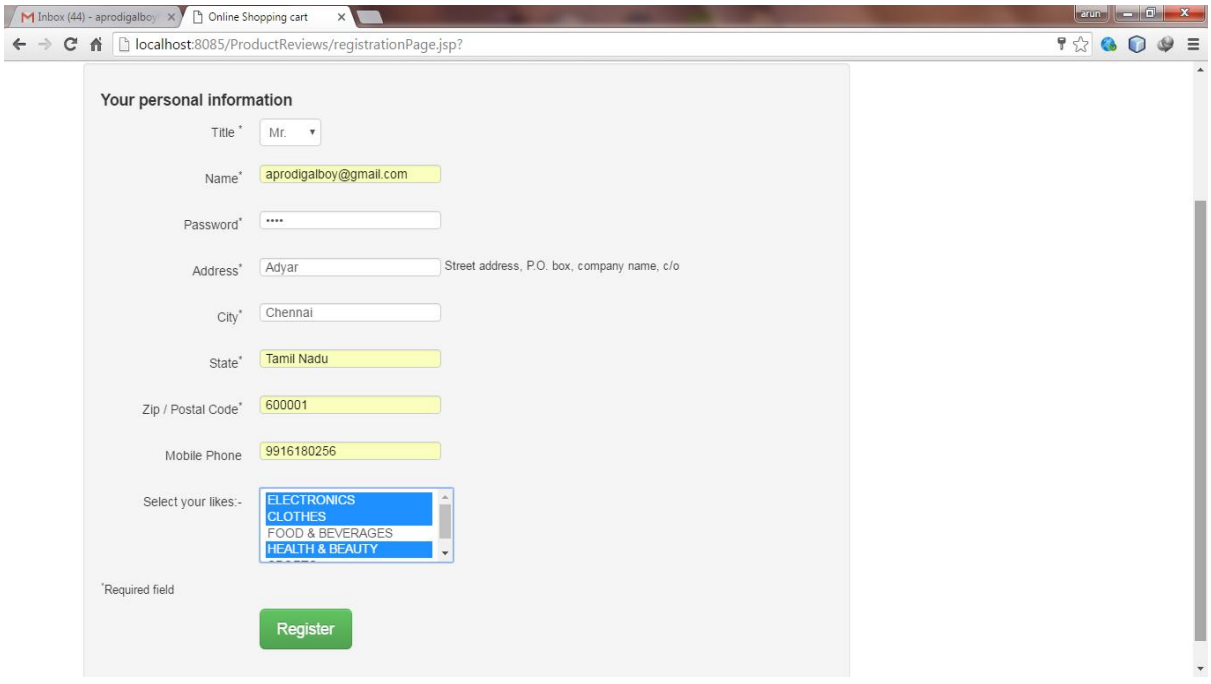


FIG 9.2 CREATE ACCOUNT

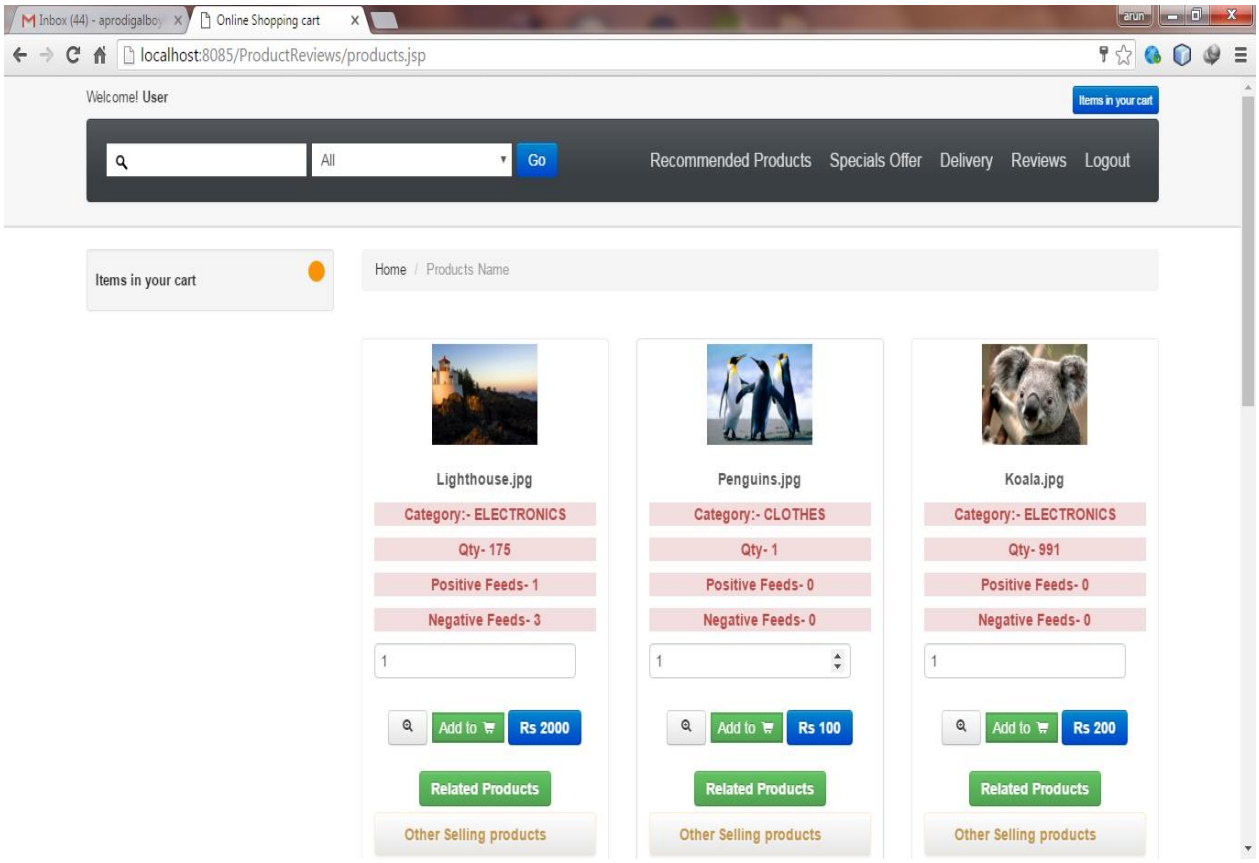


FIG 9.3 SHOPPING CART

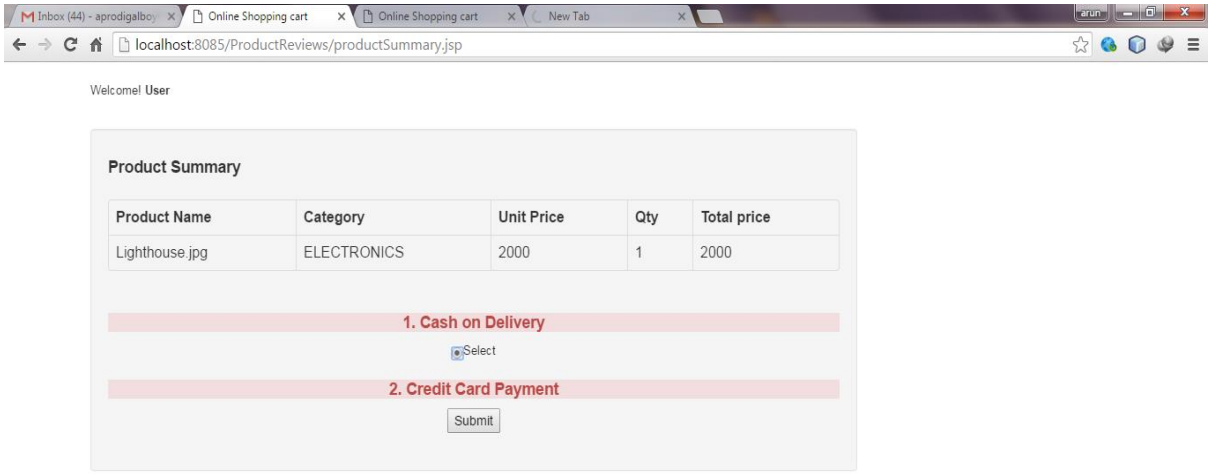


FIG 9.4 USER DETAILS

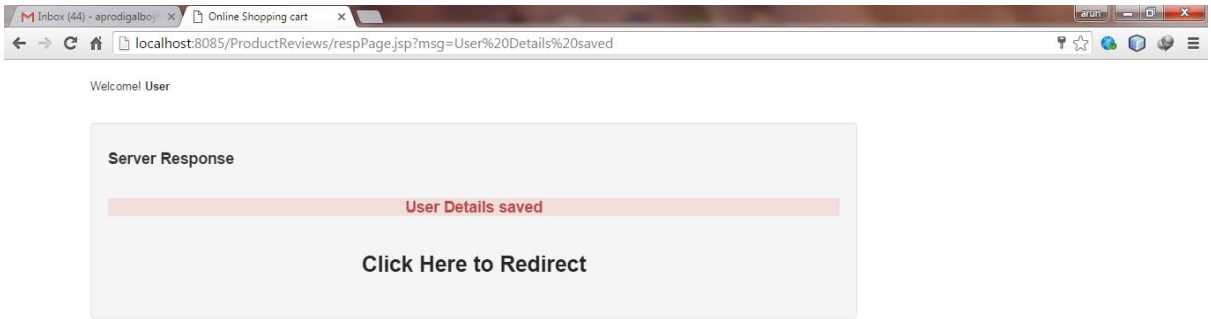


FIG 9.5 SERVER RESPONSE

CHAPTER 10

CONCLUSIONS:

In this project, we have to implement secured purchasing system in online. Honey words are generated based on the user info provided and the original password is converted into another format and stored along with the Honey words. If identify the hacker means , Hacker's IP Address, E mail ID, Phone number and postal Address are tracked and stored. Finally these details are sent to original user alternative mail id.

FUTURE ENHANCEMENT

1. E mail Id and Mobile number registered in any real time server of the hacker are to be identified and their accounts are to be automatically deleted.

2. Location detection of IP address is achieved and automatically tracked.

3. Same user's Id should be deleted from all the Social and any other public Domains so that same user cannot be entitled to access any other services.

4. We need to fetch any other Original IDs so that exact person can be tracked and blocked completely.

CHAPTER 11

REFERENCES:

[1] D. Mirante and C. Justin, "Understanding password database compromises,"Dept. of Comput. Sci. Eng. Polytechnic Inst. of NYU, NewYork, NY, USA: Tech. Rep. TR-CSE-2013-02, 2013.

[2] A. Vance, "If your password is 123456, just make it hackme," NewYork Times, Jan. 2010.

[3] K. Brown, "The dangers of weak hashes," SANS InstituteInfoSec 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, "Passwordcracking using probabilistic context-free grammars," in Proc. 30thIEEE Symp. Security Privacy, 2009, pp. 391–405.

[5] F. Cohen, "The use of deception techniques: Honeypots anddecoys," Handbook Inform. Security, vol. 3, pp. 646–655, 2006.

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

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

[8] 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