Semester Project Progress Report

on

Honeypots: Tracking Hackers

Submitted in partial fulfillment for award of

BACHELOR OF TECHNOLOGY

Degree

In

COMPUTER SCIENCE & ENGINEERING

2022-2023

Under the guidance of:

Ms. Kumud Alok

Submitted By:

Sanju Tomer(2000330109010)

Yugal Teotia(2000330109012)

Mohd Nadir(2000330109006)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAJ KUMAR GOEL INSTITUTE OF TECHNOLOGY
5 K.M. STONE DELHI-MEERUT ROAD, GHAZIABAD

Affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow

February 2023

Raj Kumar Goel Institute of Technology Ghaziabad

ISO 9001:2015 Certified

5th KM. STONE, DELHI-MEERUT ROAD, GHAZIABAD (U.P)-201003

Department of Computer Science & Engineering

Project Progress Report

1. Course : Bachelor of Technology
2. Semester : VIIIth
3. Branch : Computer Science & Engineering
4. Project Title : HoneyPots: Tracking Hackers
5. Details of Students:

S. No.	Roll No.	Name	Role as	Signature
1	2000330109010	Sanju Tomer	Team Leader	
2	2000330109012	Yugal Teotia	Coder, Report	
3	2000330109006	Mohd. Nadir	Designer, Tester	

6. SUPERVISOR: Ms. Kumud Alok

Remarks from Project Supervisor:

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

…………………………………………………………………………………

SYNOPSIS

A honeypot is a network-attached system set up as a decoy to lure cyber attackers and detect, deflect and study hacking attempts to gain unauthorized access to information systems. The function of a honeypot is to represent itself on the internet as a potential target for attackers usually, a server or other high-value asset and to gather information and notify defenders of any attempts to access the honeypot by unauthorized users.

Honeypot systems often use hardened operating systems (OSes) where extra security measures have been taken to minimize their exposure to threats. They are usually configured so they appear to offer attackers exploitable vulnerabilities. For example, a honeypot system might appear to respond to Server Message Block (SMB) protocol requests used by the WannaCry ransomware attack and represent itself as an enterprise database server storing consumer information.

Large enterprises and companies involved in cyber security research are common users of honeypots to identify and defend against attacks from advanced persistent threat (APT) actors. Honeypots are an important tool that large organizations use to mount an active defense against attackers or for cyber security researchers who want to learn more about the tools and techniques attackers use.

The cost of maintaining a honeypot can be high, in part because of the specialized skills required to implement and administer a system that appears to expose an organization's network resources, while still preventing attackers from gaining access to any production systems.

Till now there has not been any development made in the field of intrusion and its detection. And what's more no organization has enabled itself against any kind of such intrusion as well as detection i.e.very few of them are immune against such catastrophes. The hazards of such malicious activities are numerous most prominent of them is the Data Leakage."Honeypots: Tracking Hackers" are designed to mimic systems that an intruder would like to break into but limit the intruder from having access to an entire network. If a "Honeypots: Tracking Hackers" is successful, the intruder will have no idea that s/he is being tricked and monitored. Not only this in order to ensure proper security a record of the intruder's activities is kept, so

that we; can gain insight into attack methodologies to better protect our real production systems. Proposed "Honeypots: Tracking Hackers" is a web based program to promote awareness and to stop more websites to splurge into the victims of hacking. Also give them a platform to be connected with a helping website. It should showcase support in various types of hacking, which should be motivation for preventing hacking around the globe. We are providing information throughout the world 24*7. We are providing direct interaction with the administrator. Our project provides secure access of confidential data. It becomes Easy to manage the records. Removes cost of consultation. Availability of data on a click. While developing a project, it is very important to define the category of such project. As for as this application is concern, this application can be categorized in the category of RDBMS and OOPS because this application is built to perform and deliver the primary features of RDBMS and OOPS. Various features of Database are used to maintain the database. As this application is to be built using JSP, so all the basic and primary concepts of OOPS are used. It is a web application that can be run on internet or on any other network. Its back end is RDBMS (relational database management system).

TABLE OF CONTENTS

CHAPTER NO. **TITLE** **PAGE NO.**

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The term "Honeypots: Tracking Hackers" is often understood to refer as a bait which is use to lure any inter cyber malware may it be a hacker, an eavesdropper or any worm which is roaming in search of any such information (honey).

An alternative explanation for the term is a reflection of the sarcastic term for outhouses and other methods of collecting human waste in places that lack indoor plumbing. Honey being a euphemism for such waste, which is kept in a honey pot until it is picked up by a honey wagon and taken to a disposal area. In this usage, attackers are the equivalent of flies, drawn by the stench of waste.

1.1 OBJECTIVE AND SCOPE OF THE PROJECT

In the digital arms race that is **crackers vs. administrator**, tightening the existing security holes will only force the crackers to get better, while the administrator get complacent. They get the latest and greatest piece of ready-made software, and call themselves experts. What is bound to happen in the majority of the situations is that **a company will set a tracking program and never bother to spend the time it takes to maximize its effectiveness**.

Of course, the true answer is for administrator and software programmers who actually take a little pride in their work, and do their jobs properly. Also, it would help if software companies would take some responsibility when they find security holes in their product, and to update accordingly. System administrator should also feel obligated to keep their software current, and to make sure nobody within their company is given more access than they need. Hence our project “***Honeypots: Tracking Hackers***” is the sole answer to all such problems.

Proposed “***Honeypots: Tracking Hackers***” is a web based program to promote awareness and to stop more websites to splurge into the victims of hacking. Also give them a platform to be connected with a helping website. It should showcase support in various types of hacking, which should be motivation for preventing hacking around the globe.

- We are providing information throughout the world 24*7.
- We are providing direct interaction with the administrator.

1.2 MOTIVATION

Different types of honeypot can be used to identify different types of threats. Various honeypot definitions are based on the threat type that's addressed. All of them have a place in a thorough and effective cybersecurity strategy.

Email traps or spam traps place a fake email address in a hidden location where only an automated address harvester will be able to find it. Since the address isn't used for any purpose other than the spam trap, it's 100% certain that any mail coming to it is spam. All messages which contain the same content as those sent to the spam trap can be automatically blocked, and the source IP of the senders can be added to a denylist.

A **decoy database** can be set up to monitor software vulnerabilities and spot attacks exploiting insecure system architecture or using SQL injection, SQL services exploitation, or privilege abuse.

A **malware honeypot** mimics software apps and APIs to invite malware attacks. The characteristics of the malware can then be analyzed to develop anti-malware software or to close vulnerabilities in the API.

A **spider honeypot** is intended to trap webcrawlers ('spiders') by creating web pages and links only accessible to crawlers. Detecting crawlers can help we learn how to block malicious bots, as well as ad-network crawlers.

By monitoring traffic coming into the honeypot system, we can assess:

- where the cybercriminals are coming from
- the level of threat
- what modus operandi they are using
- what data or applications they are interested in
- how well our security measures are working to stop cyber attacks

1.3 EXISTING SOFTWARE

Cowrie – Cowrie is an SSH honeypot based off an earlier favourite called Kippo. It will emulate an interactive SSH server with customisable responses to commands. Another alternative is HonSHH which sits between a real SSH server and the attacker, MiTMing the connection and logging all SSH communications.

Dionaea is a multi-protocol honeypot that covers everything from FTP to SIP (VoIP attacks). Where it really excels is for SMB decoys. It can even simulate malware payload execution using LibEmu to analyse multi-part stagers.

Honeything emulates the TR-069 WAN management protocol, as well as a RomPager web-server, with vulnerabilities. Other IoT decoys can be created by emulating embedded telnet / FTP servers, for example with BusyBox.

ConPot emulates a number of operational technology control systems infrastructure. These include protocols like MODBUS, DNP3 and BACNET. It comes with a web-server that can emulate a SCADA HMI as well.

GasPot emulates a Veeder Root Gaurdian AST that is commonly used for monitoring in the oil and gas industry.

MongoDB-HoneyProxy emulates an insecure MongoDB database. Hackers regularly scan the interwebs looking for administrators who had an ‘oops moment’ and exposed their DB to the world.

ElasticHoney emulates an ElasticSearch instance, and looks for attempted remote code execution.

1.4 PROBLEM DEFINITION

Till now there hasn't been any development made in the field of **intrusion and its detection**. And what's more no organization has enabled itself against any kind of such intrusion as well as detection i.e. very few of them are immune against such catastrophes. The hazards of such malicious activities are numerous most prominent of them is the **Data Leakage.**

*"**Honeypots: Tracking Hackers**"* are designed to mimic systems that an intruder would like to break into but **limit the intruder from having access to an entire network**. If a "Honeypots: Tracking Hackers" is successful, the intruder will have no idea that s/he is being tricked and monitored.

Not only this in order to ensure proper security **a record of the intruder's activities is kept, so that we; can gain insight into attack methodologies to better protect our real production systems.**

Honey pots are usually programs that emulate services on a designated port, but once successfully cracked, offer no real power to the attacker. The "**Honeypots: Tracking Hackers**" program will then alert the admin that an attack is in progress, and will allow the admin to track the attacker's every move. Honeypots will also show the methods the attacker is using to gain entry, and what methods the attacker is using to cover his or her tracks.

While it is often a computer, a "**Honeypots: Tracking Hackers**" can take other forms, such as files or data records, or even unused **IP address** space. A honey pot that **masquerades as an open proxy** to monitor and record those using the system as bait.

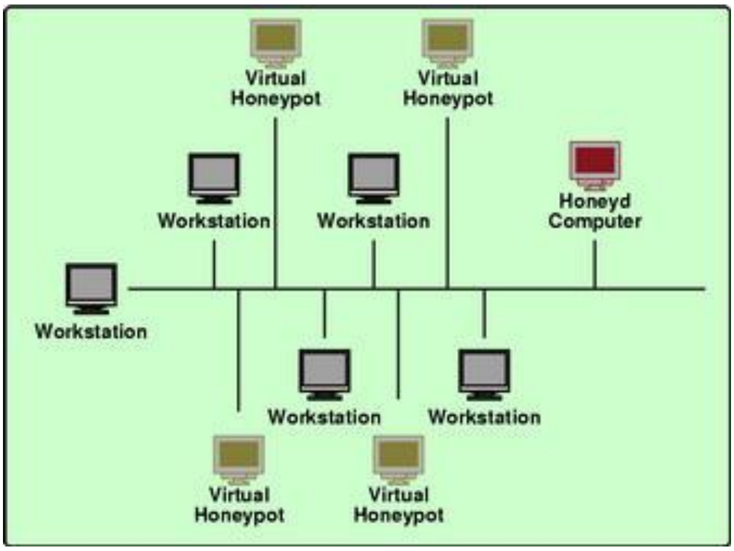


Figure 1.1 Block diagram of Honeypot System

1.5 BACKGROUND AND RELATED WORK

Most of the systems in use today, therefore, work with a set of so-called attack signatures, that describe attacking patterns in sufficient detail for identifying ongoing attacks automatically. However, the specification of such signatures usually needs to be done by experienced network security analysts by either monitoring an existing network and extracting the relevant information, this is the major loop hole in the whole process and using this glitch any outsider can or may gain access to company's relevant data.

Table. 1.1 Comparative Study of various honeypots systems

Sr.No.	Paper Name	Author	Year	Methodology
1	A survey of honeypot research: Trends and opportunities	Ronald M. Campbell, Keshnee Padayachee, Themba Masombuka	2015	Devices connected to computer networks and its threat
2	Recent Advances and Future Trends in Honeypot Research	Matthew L. Bringer, Christopher A. Chelmecki, and Hiroshi Fujinoki	2015	Honeypots, Alarm system, Computer hacking, Computer crime, Computer security
3	Honeypot Software and Data Analysis	Marcin Nawrocki, Matthias Wahlisch, Thomas C. Schmidt	2016	Extensive overview on honeypots. This includes not only honeypot software but also methodologies to analyse honeypot data.
4	Honeypot: A Trap for Attacker	Savita Paliwal	2017	Low level interaction honeypot, medium level interaction honeypot, high level interaction honeypot and functions of honeypots.

5	Honeypot: Tracking Hackers	Lance Spitzner, Addison-Wesley	2018	Tracking Hackers methods
6	Honeypot for Cyber security Threat Intelligence	Ren Rui Tan, Simon Eng, Casey How and Paul HJ Wu	2022	SME, Cybersecurity Threat Intelligence, Web Honeypot; Threat Intelligence Dashboard First Section

Ronald M. Campbell et. Al. [1] Studied that the number of devices connected to computer networks is increasing daily, and so is the number of network-based attacks. A honeypot is a system trap that is set to act against unauthorised use of information systems. The objective of this study was to survey the emergent trends in extant honeypot research with the aims of contributing to the knowledge gaps in the honeypot environment.

Matthew L. Bringer et. Al. [2] Summarizes 60 papers that had significant contribution to the field. In reviewing the literature, it became apparent that the research can be broken down into five major areas: 1. new types of honeypots to cope with emergent new security threats, 2. utilizing honeypot output data to improve the accuracy in threat detections, 3. configuring honeypots to reduce the cost of maintaining honeypots as well as to improve the accuracy in threat detections, 4. counteracting honeypot detections by attackers, and 5. legal and ethical issues in using honeypots.

Marcin Nawrocki et. Al. [3] This research includes not only honeypot software but also methodologies to analyse honeypot data.

Savita Paliwal et. Al. [4] As we have studied this paper introduces Honeypot, a new technology for Network Security. The paper deals with the basic aspects of Honeypots, their use in modern computer networks and their implementation in educational environments. It explains the different types i.e Production honeypot, Research honeypot, low level interaction honeypot, medium level interaction honeypot, high level interaction honeypot and functions of honeypots.

Lance Spitzner et. Al. [5] From the basics of shrink-wrapped honeypots that catch script kiddies to the detailed architectures of next-generation honeynets for trapping more sophisticated bad guys, this book covers it all....This book really delivers new information and insight about one of the most compelling information security technologies today.

Ren Rui Tan et. Al. [6] Studied that most of the Small and Medium-sized Enterprises (SMEs) have limited resources to guard their IT infrastructures because of high upfront costs and fully engaged manpower, resulting in lack of focus on cybersecurity. Though cybersecurity threat intelligence derived from honeypots can help businesses to mitigate cybersecurity risks, current implementations target huge enterprises that have in-house cybersecurity teams to digest the intelligence and implement countermeasures against emerging threats. In this paper, a honeypot mimicking SME web portals will be setup to acquire insights into cyberattacks.

CHAPTER 2

HARDWARE AND SOFTWARE REQUIREMENT

For software development very first and essential requirement is the availability of software and hardware. The software requires the following software and hardware for its successful implementation.

2.1 SOFTWARE REQUIREMENT

2.1.1 CLIENT SIDE (RECOMMENDED)

Any Operating System with Web Browser.

Internet or LAN connection

2.1.2 SERVER SIDE (RECOMMENDED)

Operating System	:	WINDOWS XP /98/2000/WIN 7/LINUX OR ANY OS
Scripting Language	:	JAVA SCRIPT
Front-End Language	:	JSP AND JDK1.6
Back-End	:	SQL-SERVER 2000
Mark-up Language	:	HTML4.0, DHTML
Web-Server	:	WEB-LOGIC or TOMCAT

2.2 HARDWARE REQUIREMENT

2.2.1 CLIENT SIDE (RECOMMENDED)

Processor	:	Pentium-4 or above
RAM	:	1GB
Hard Disk	:	160GB
Keyboard	:	Any
Mouse	:	Any

2.2.2 SERVER SIDE (MINIMUM)

Processor	:	Intel-C2D
RAM	:	4 GB
Hard Disk	:	500 GB
Keyboard	:	Any
Mouse	:	Any

CHAPTER 3

SDLC METHODOLOGIES

SDLC stands for Software Development Life Cycle models, and these are a variety of processes of design, development and testing that are used in the industry today. While there's no best or standout SDLC methodology, it's essential to be across the most common models that can be applied to projects within a company.

3.1.1 AGILE METHODOLOGY

Agile is a combination of an incremental and iterative approach, where the product is released on an ongoing cycle then tested and improved at each iteration. Fast failure is encouraged in agile methodology: the theory is that if we fail fast and early, we can solve minor issues before they grow into major issues.

Agile is one of the most common methodologies out there today but it's technically more of a framework than a distinct model. Within Agile, there are sub-models in place such as extreme programming (XP), Rapid Application Development (RAD), Kanban and Scrum methodology.

3.1.2 WATERFALL METHODOLOGY

The Waterfall methodology is one of the oldest surviving SDLC methodologies. It follows a straightforward approach: the project development team completes one phase at a time, and each phase uses information from the last one to move forward.

While this methodology does make the needs and outcomes clear, and gives each stage of the model a well-defined starting and ending point, there are downsides in Waterfall's rigidity. In fact, some experts believe the Waterfall model was never meant to be a working SDLC methodology for developing software because of how fixed it is in nature. Because of this, SDLC Waterfall methods are best used for extremely predictable projects.

3.1.3 DEV-OPS

DevOps is used by some of the biggest companies out there, such as Atlassian. A hybrid of Agile and Lean, DevOps evolved from the growing need for collaboration between operations and development teams throughout the SDLC process. In DevOps, both developers and operations teams work together to accelerate and innovate the deployment and creation of software. There are small but frequent updates and DevOps encourages continuous feedback, process improvement and the automation of previously manual processes.

DevOps methodology saves time and improves communication because both operations and development teams get to know about the potential obstructions at the same time. However, DevOps may open software to more security issues, as this approach generally favors speed over security.

3.1.4 SPIRAL

As one of the most flexible SDLC models out there, the Spiral model is used by the world’s leading software companies. Spiral enables project teams to build a highly customized product. Spiral methodology passes through four phases repeatedly until the project is finished: planning, risk analysis, engineering, and evaluation.

The biggest difference between Spiral and other methodologies is that it is focused on risk analysis, with each iteration it focuses on mitigating potential risks. The model also emphasizes customer feedback, and as the prototype build is done in small increments, cost estimation becomes easier.

3.1.5 ITERATIVE

The Iterative model is all about repetition. Instead of starting out with a comprehensive overview of the requirements, development teams build the software piece by piece and identify further requirements as they go along. As a result, each new phase in the Iterative model produces a newer, more-refined version of the software under development.

Iterative allows developers and testers to identify functional or design flaws early, and can easily adapt to the ever-changing needs of the client. Like Spiral, Iterative suits larger projects and requires more management and oversight to work well.

3.2 HOW IT AFFECT OUR MODEL

Honeypot is a complete package of all hacker tracking components. A honey pot is a trap set to detect, deflect, or in some manner counteract attempts at unauthorized use of information systems. Generally it consists of a computer, data, or a network site that appears to be part of a network, but is actually isolated, (un) protected, and monitored, and which seems to contain information or a resource of value to attackers. Iterative model applies to our project as we created feedback models. We created small models first such as admin module, login/sign-up module, customer module, etc and then tested separately. From those feed backs, we resolved issues and then create new models. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).

CHAPTER 4

REQUIREMENT SPECIFICATION AND ANALYSIS

The system requirements were analysed and defined before the development of the project began. The requirements were gathered from the stakeholders, which in this case, are the placement officers. The requirements were analysed and classified into three categories:

1. Functional requirements □
2. Non-functional requirements □
3. Technical requirements

Functional requirements are the requirements which define what the system should do. In this case, the functional requirements are as follows: -

1. The system should be able to keep track of all the hackers and their attacks.
2. The system should be able to keep track of all the users and their activities.
3. The system should be able to generate reports. □
4. The system should be able to generate reports of all activities of users.

Non-functional requirements are the requirements which define how the system should work. In this case, the non-functional requirements are as follows: □

1. The system should be user-friendly. □
2. The system should be able to handle a large amount of data. □
3. The system should be scalable. □
4. The system should be secure.
5. The system should be available 24/7.

Technical requirements are the requirements which define what technology should be used to develop the system.

In this case, the technical requirements are as follows:

1. Any Operating System
2. Web Browser.
3. Internet or LAN connection

4.1 OBJECTIVE

In the digital arms race that is crackers vs. administrator, tightening the existing security holes will only force the crackers to get better, while the administrator get complacent. They get the latest and greatest piece of ready-made software, and call themselves experts. What is bound to happen in the majority of the situations is that a company will set a tracking program and never bother to spend the time it takes to maximize its effectiveness.

Of course, the true answer is for administrator and software programmers who actually take a little pride in their work, and do their jobs properly. Also, it would help if software companies would take some responsibility when they find security holes in their product, and to update accordingly. System administrator should also feel obligated to keep their software current, and to make sure nobody within their company is given more access than they need. Hence our project “Honeypots: Tracking Hackers” is the sole answer to all such problems.

4.2 SCOPE

This website also provides full sort of security regarding the information of the customers who are linked to this website. In our view website will be successful way in spreading the awareness of “Honeypots: Tracking Hackers” among the customers.

This software can be easily upgraded in the future. And also include many more features for existing system. It is connected with the network for easily retrieved data and many location or many districts or cities in different states. Report on the different basis will be easily created on the demand.

4.3 PORTABILITY

Portability usually has 3 dimensons: Vertical (Development lifecycle), Horizontal (Platform/provider migration) or Deep (Replication). Vertical portability refers to consistency of software throught whole lifecycle, which can be ensured with Infrastructure as code. Horizontal stands for platform/dependency flexibility. With good Dependency management, single sourcing and good platform support plan, you can master the Horizontal dimension. Deep means how scalable and replicable your code/architecture/software is.

4.4 RELIABILITY

Software Reliability is an essential connect of software quality, composed with functionality, usability, performance, serviceability, capability, installability, maintainability, and documentation. Software Reliability is hard to achieve because the complexity of software turn to be high. While any system with a high degree of complexity, containing software, will be hard to reach a certain level of reliability, system developers tend to push complexity into the software layer, with the speedy growth of system size and ease of doing so by upgrading the software.

4.5 DATA INTEGRITY

Data integrity is one of the most important aspect of our Honeypots: Tracking Software.

As we have created a software which makes the information systems of any organization attack preventive and tracking.

Using honeypots we can track the details of attackers computer system like which OS is he using, what is his mac address, what is his ip, etc.

By using honeypots, even if attacker tries to steal or modify our database he will not be able to do so, as Honeypots main feature is Data Integrity.

CHAPTER 5

RISK ASSESSMENT

Risk analysis in software testing is an approach to software testing where software risk is analyzed and measured. Traditional software testing normally looks at relatively straight forward function testing (e.g. 2 + 2 = 4). A software risk analysis looks at code violations that present a threat to the stability, security, or performance of the code.

Software risk is measured during testing by using code analyzers that can assess the code for both risks within the code itself and between units that must interact inside the application. The greatest software risk presents itself in these interactions. Complex applications using multiple frameworks and languages can present flaws that are extremely difficult to find and tend to cause the largest software disruptions.

The main risks in this applications involve -

1. This project does not Edit the date of connection or store the date of transfer in case of connection transfer.
2. System date for the project is like as backbone for the human, i.e. proposed system is depends on system date so it must be correct.

5.1 RISK ASSESSMENT OF HONEYPOTS FROM DEVELOPER PERSPECTIVE

The main risks involved from the developer perspective are –

1. They can introduce risk to your environment. By risk, we mean that a honeypot, once attacked, can be used to attack, infiltrate, or harm other systems or organizations. Different honeypots have different levels of risk.

2. Some introduce very little risk, while others give the attacker entire platforms from which to launch new attacks. The simpler the honeypot, the less the risk. For example, a honeypot that merely emulates a few services is difficult to compromise and use to attack other systems.

3. In contrast, a honeypot that creates a jail gives an attacker an actual operating system with which to interact. An attacker might be able to break out of such a cage and then use the honeypot to launch passive or active attacks against other systems or organizations.

4. Risk is variable, depending on how one builds and deploys the honeypot.

5. Because of their disadvantages, honeypots cannot replace other security mechanisms such as firewalls and intrusion detection systems. Rather, they add value by working with existing security mechanisms. They play a part in your overall defenses.

5.2 RISK ASSESSMENT OF HONEYPOTS FROM USER PERSPECTIVE

The main risks involved from the user perspective are –

1. First risk of using honeypots is that they can lure attackers. This is because once attackers have accessed a honeypot; they may be motivated to instigate further attacks.

2. Attackers are typically relentless and if they discover that they have been duped, they are not likely to stop until they have gained access to the real thing.

3. The other risk of using honeypots is that they just add to the complexity of a network design. This means that the additional resources will incur extra costs through maintenance. Honeypots must also be kept up and running for them to be able to work effectively.

4. Any activity towards them must also be responded to and this adds to the tediousness of network maintenance. Yet another disadvantage is that a honeypot itself can be used as a launching point for attacks against either your network or another network. If this happens, the organization may find itself in a legal tussle with the affected parties.

CHAPTER 6

USECASE / ACTIVITY/ ER / DFD DIAGRAMS

6.1 USE CASE DIAGRAM

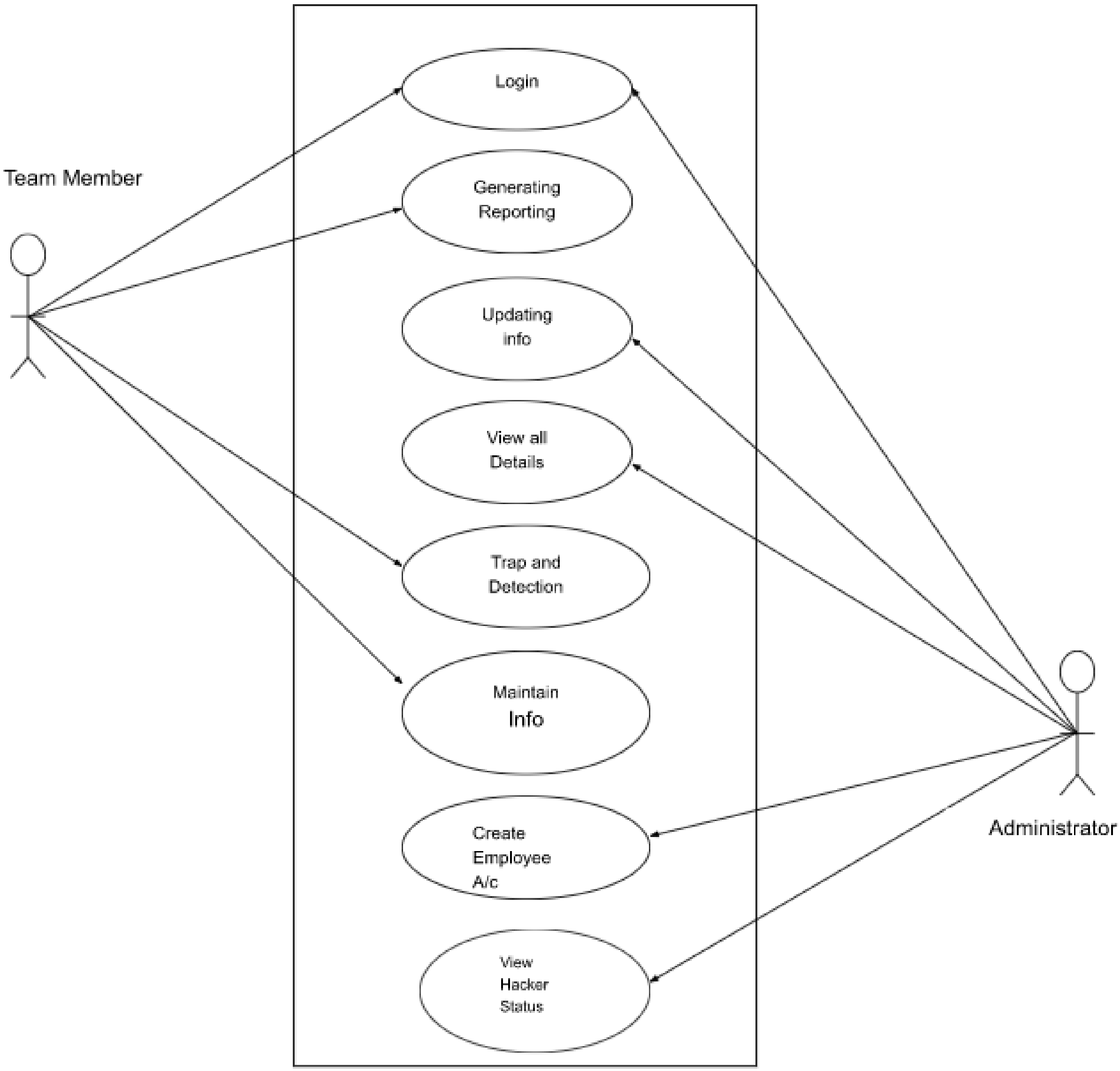


Figure 6.1 Use Case Diagram

6.2 ACTIVITY DIAGRAMS

6.2.1 LOGIN MODULE

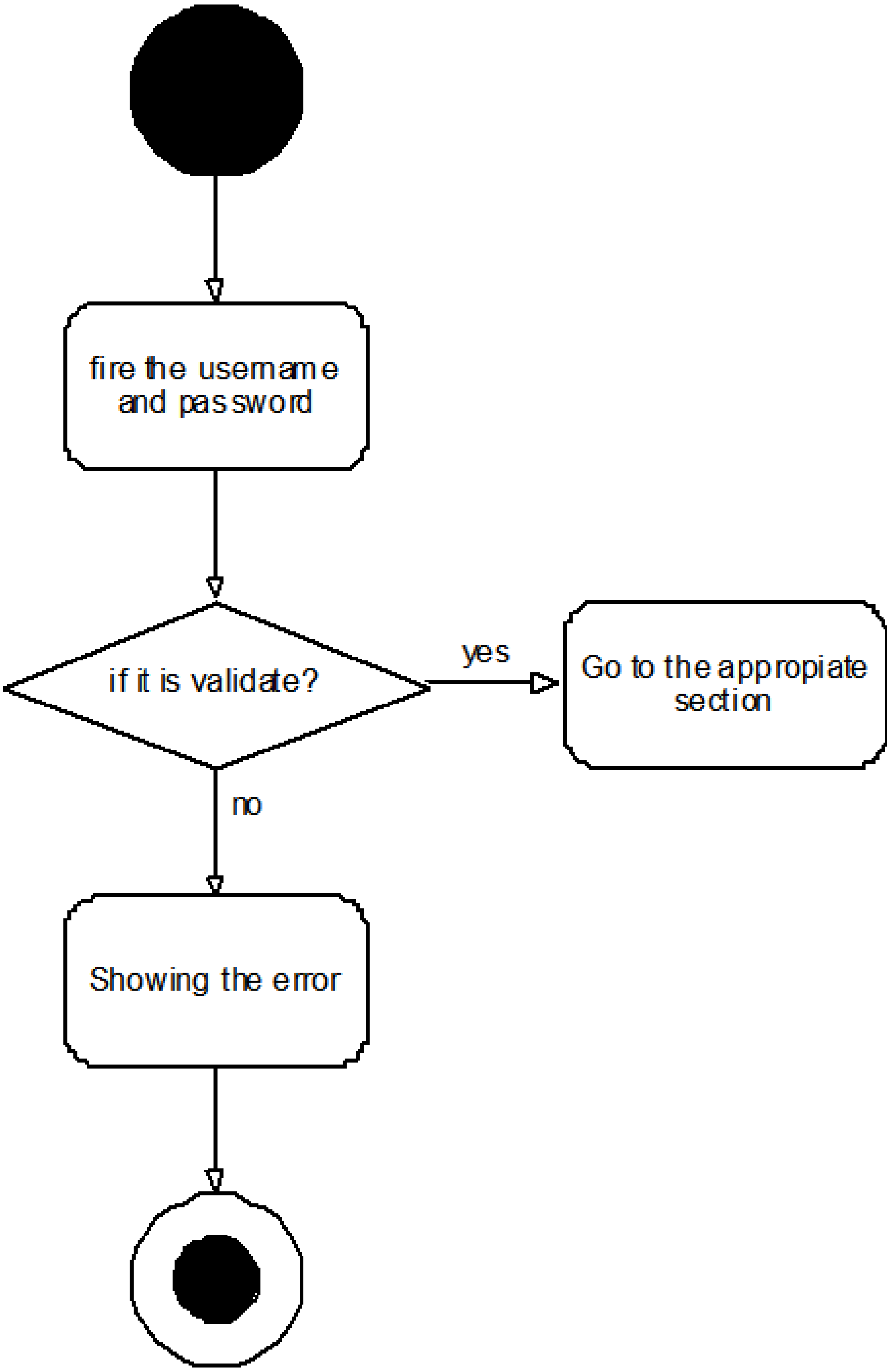


Figure 6.2 Login Module

6.2.2 ADMIN MODULE

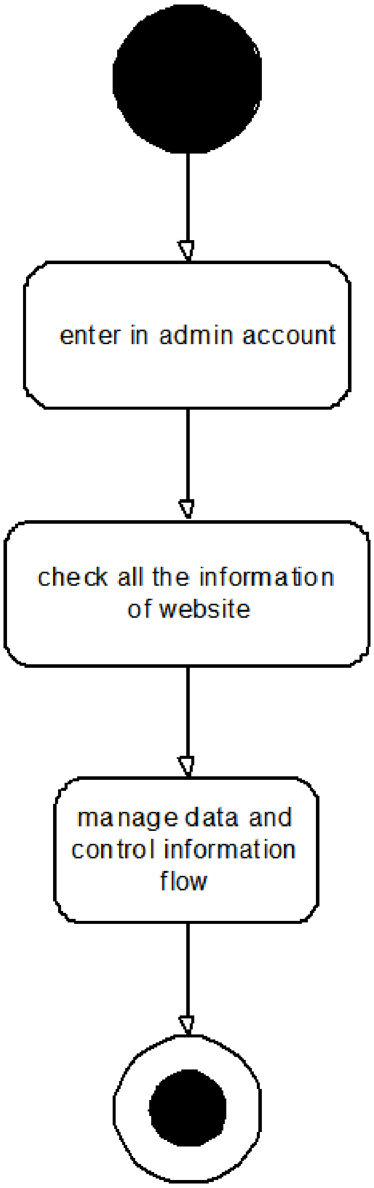

enter in admin account
check all the information of website
manage data and control information flow

Figure 6.3 Admin Module

6.2.3 HACKER MODULE

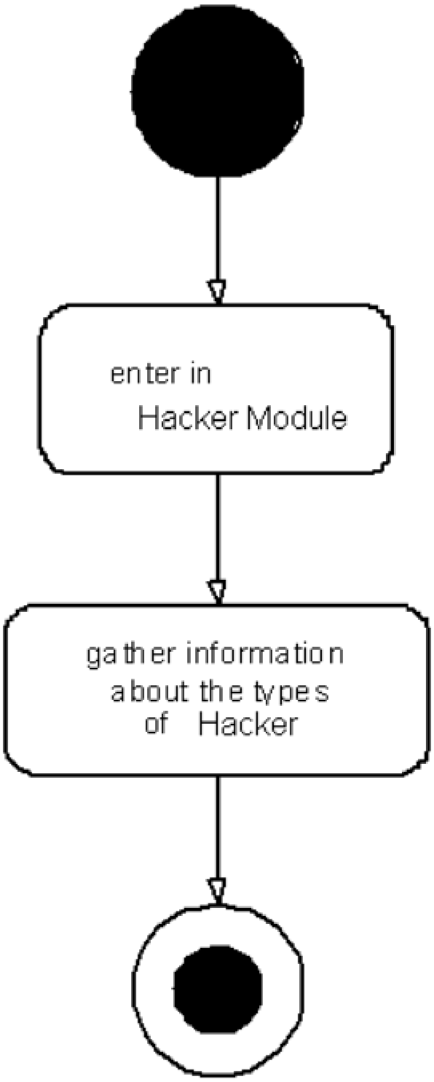


Figure 6.4 Hacker Module

6.2.4 CUSTOMER MODULE

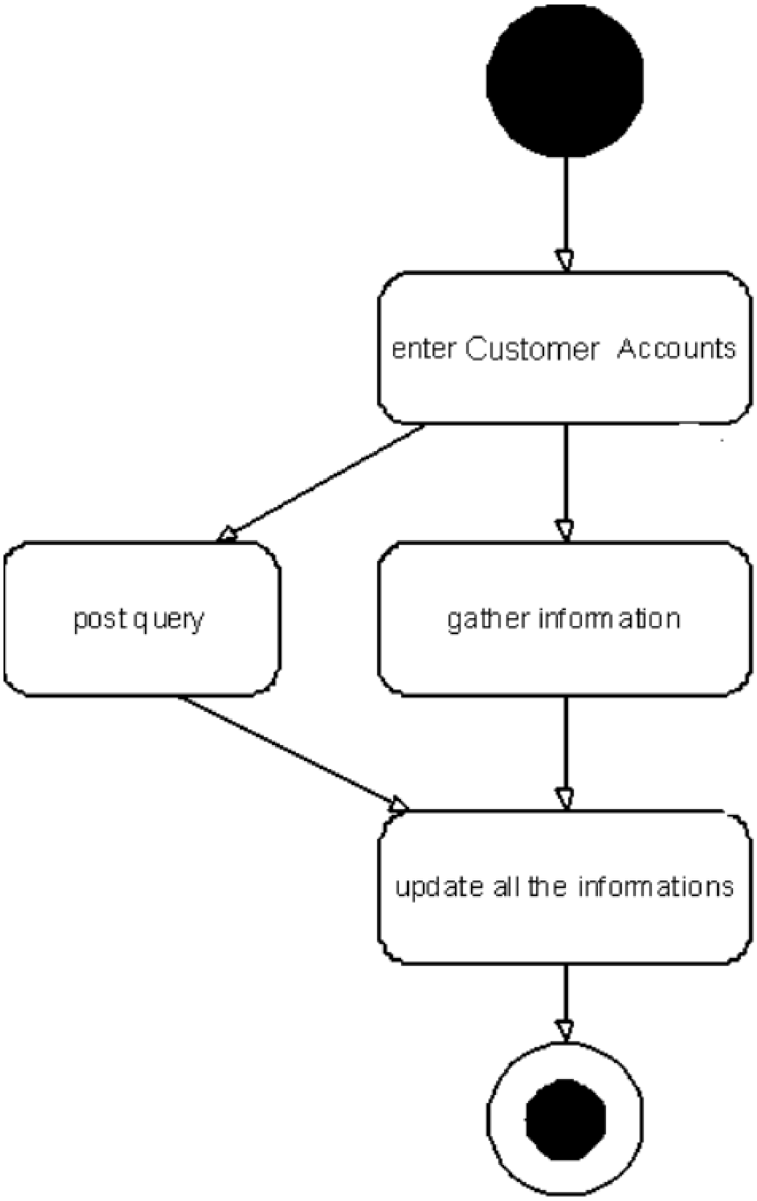


Figure 6.5 Customer Module

6.3 ER DIAGRAM

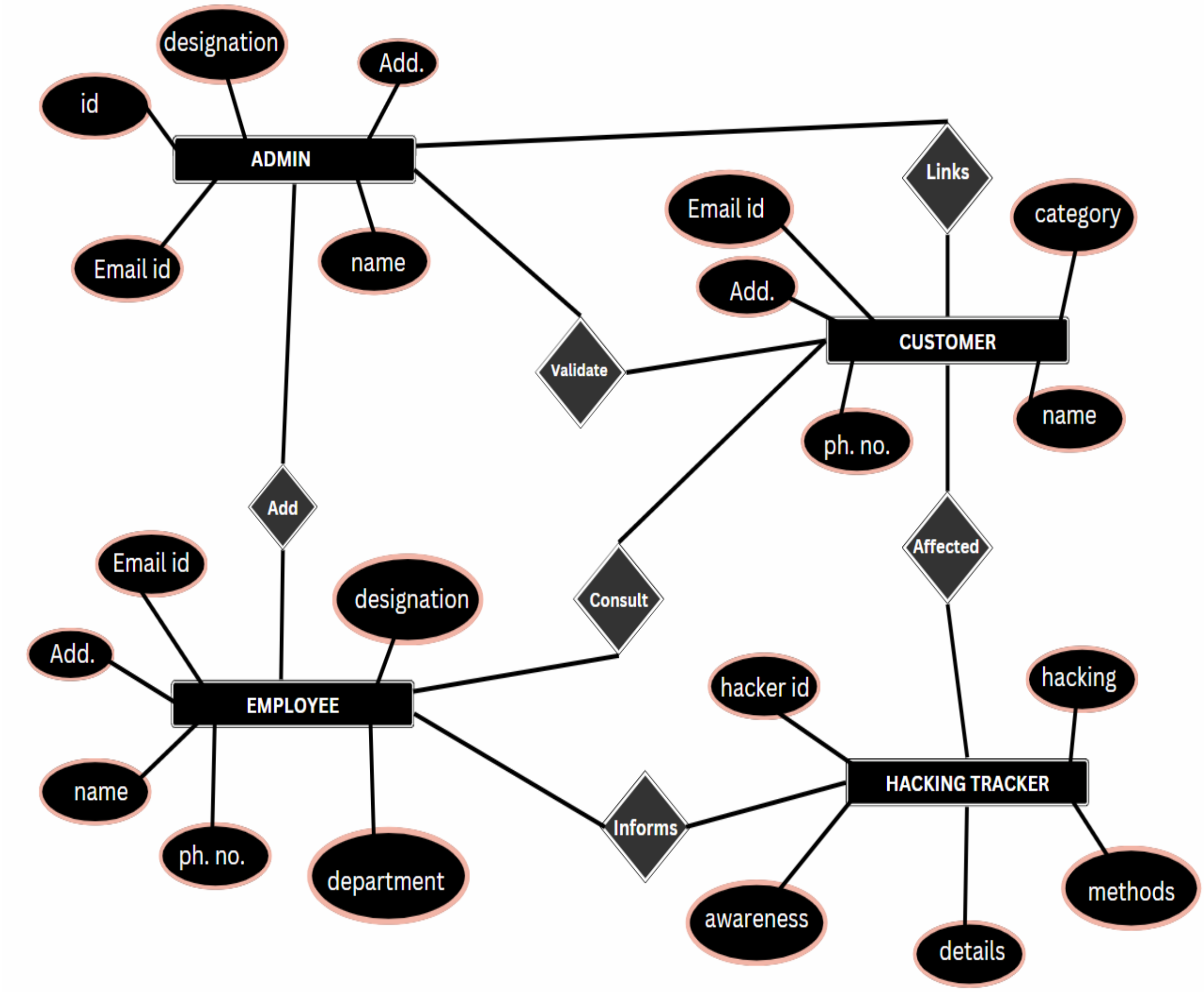


Figure 6.6 ER Model for Honeypots: Tracking Hackers

6.4 DFD DIAGRAMS

6.4.1 ZERO LEVEL DFD FOR "HONEYPOTS: TRACKING HACKERS"

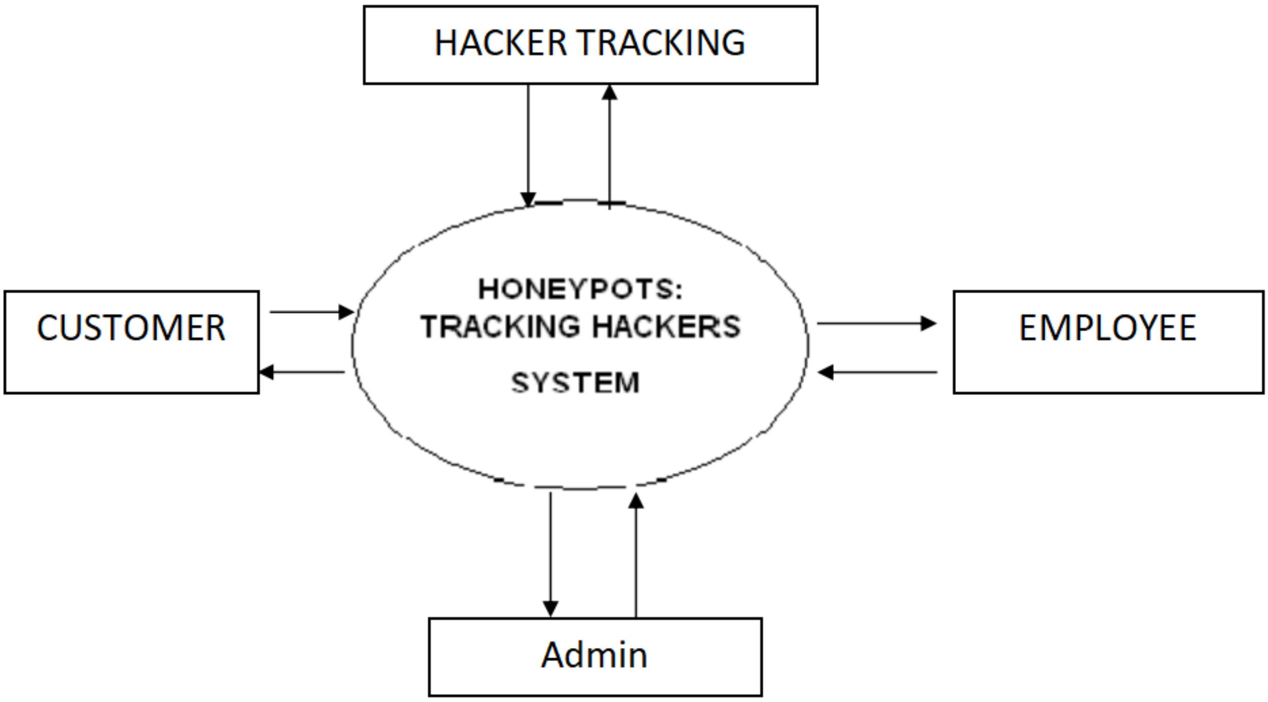


Figure 6.7 Zero-Level DFD

6.4.2 ONE LEVEL DFD FOR "HONEYPOTS: TRACKING HACKERS"

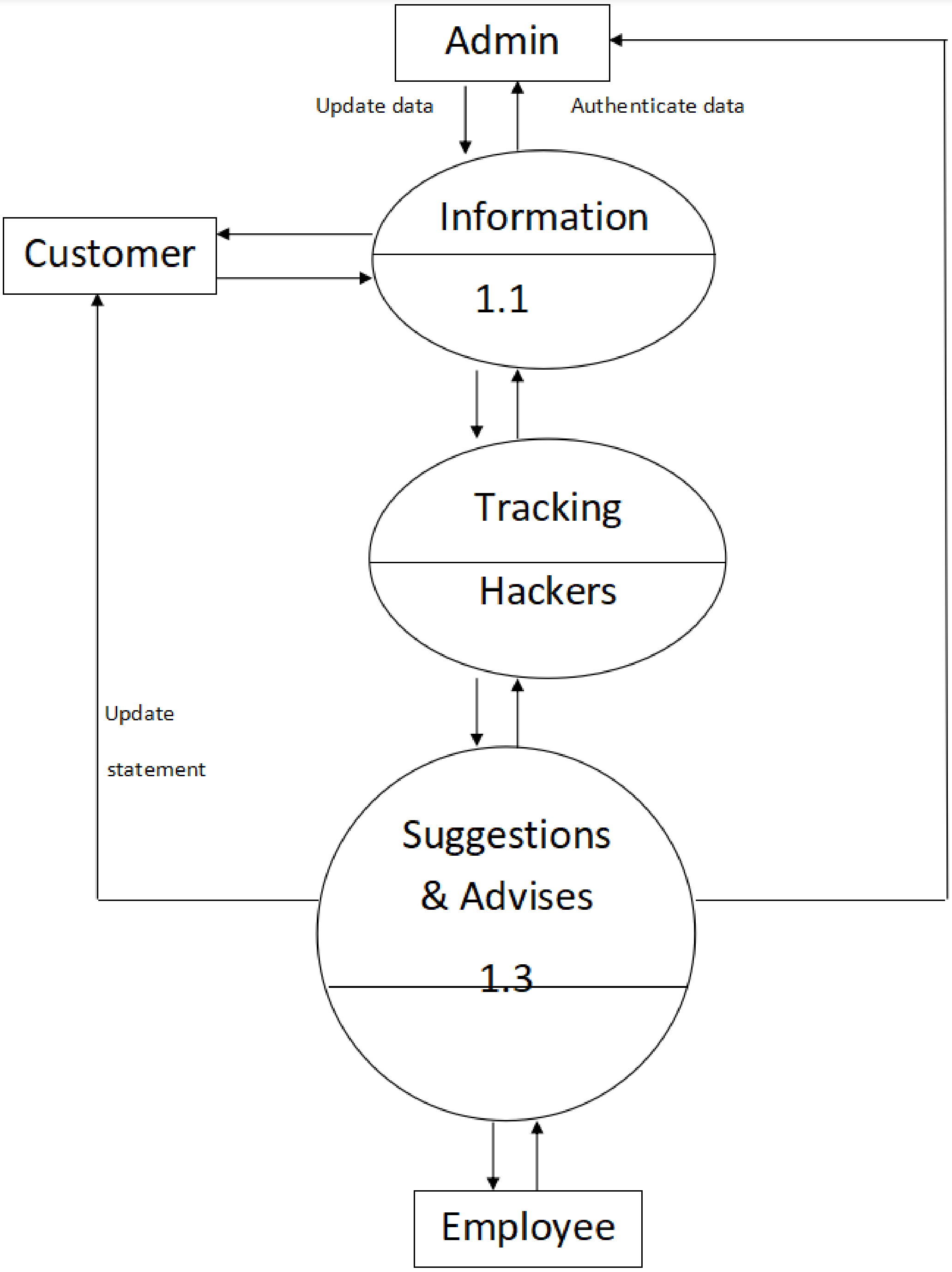


Figure 6.8 One-Level DFD

6.4.3 TWO LEVEL DFD FOR "HONEYPOTS: TRACKING HACKERS"

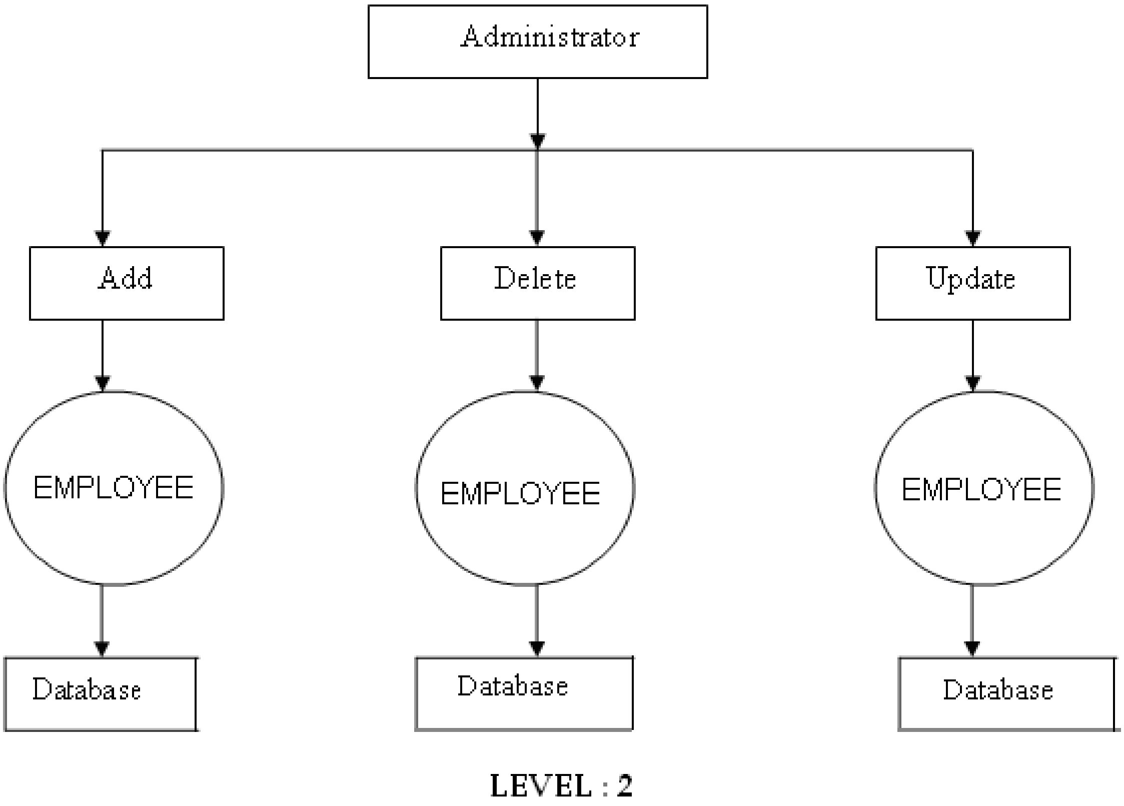
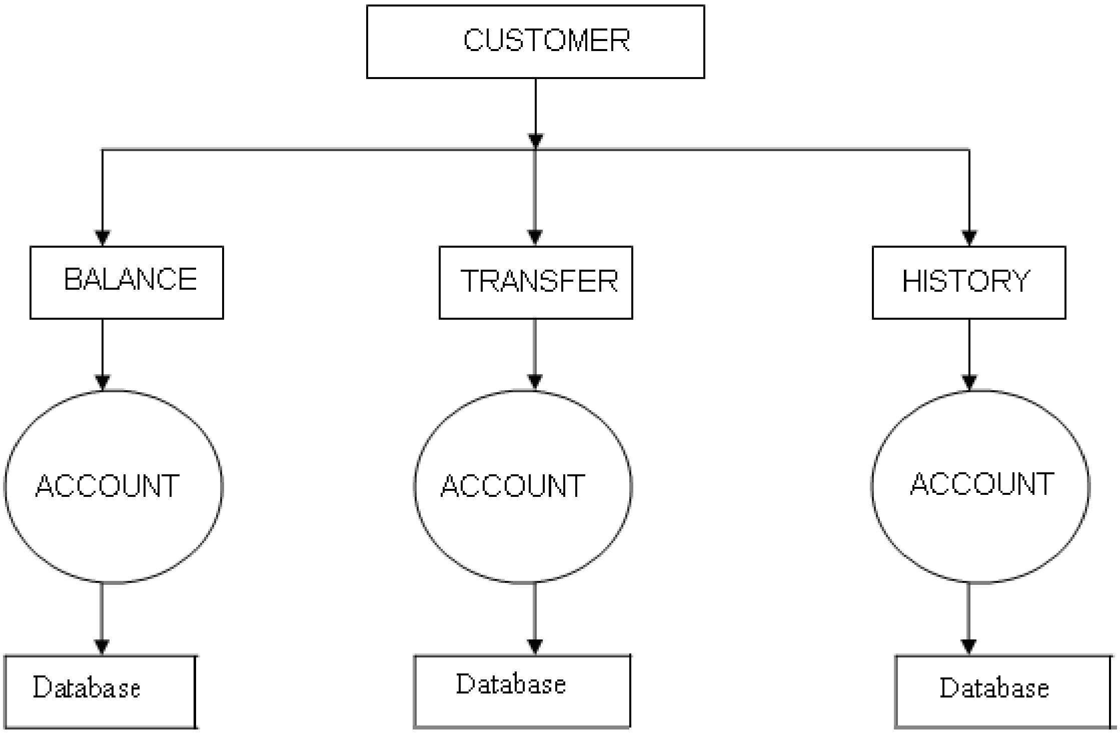


Figure 6.9 Two-Level DFD for Administrator module



LEVEL : 2

Figure 6.10 Two-Level DFD for Customer module

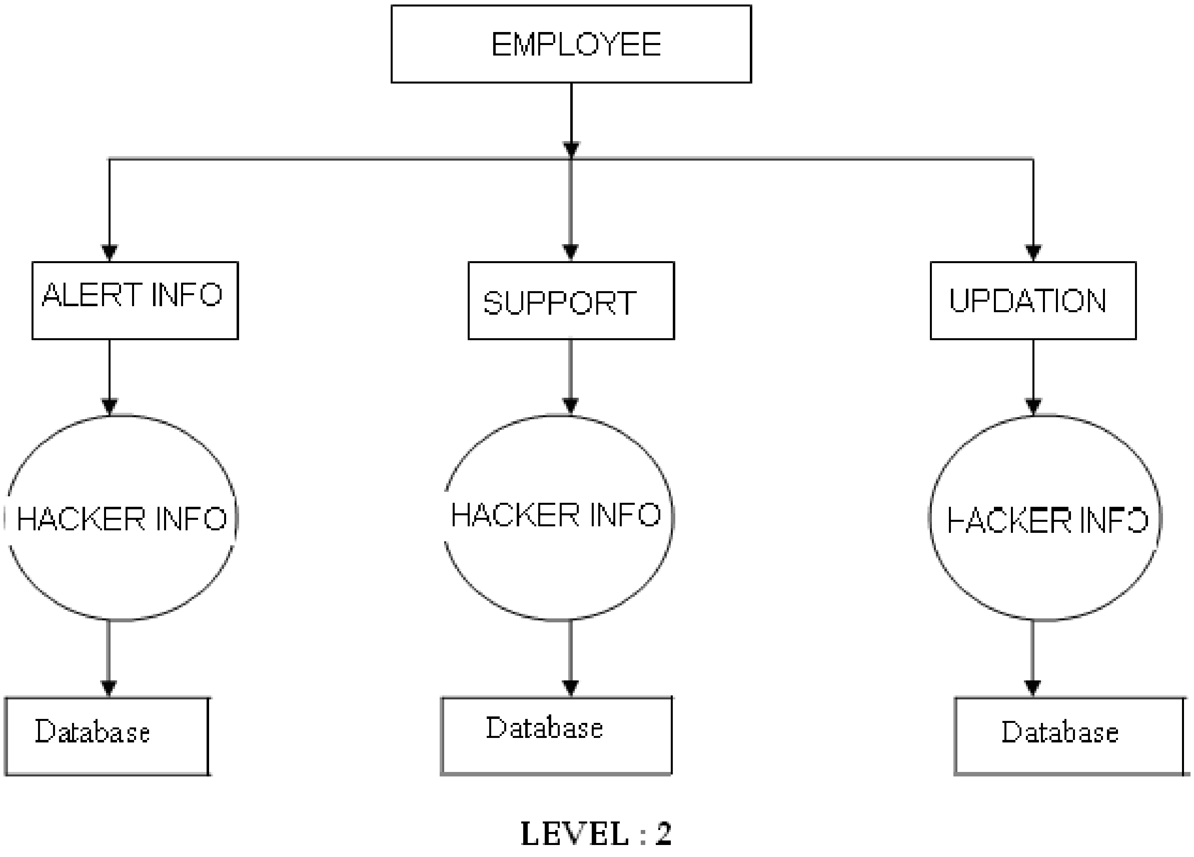


Figure 6.10 Two-Level DFD for Employee module

CHAPTER 7

PROJECT MODULES DESIGN / DATABASE TABLES

- Login Module
- Admin Module
- Customer Module
- Hacker Tracking Module
- Employee Module

7.1.1 LOGIN MODULE:

The system has two types of user logins. ADMIN is able to see all the queries and the referred solutions or remedies. It manages all the information uploaded and puts curb on it.

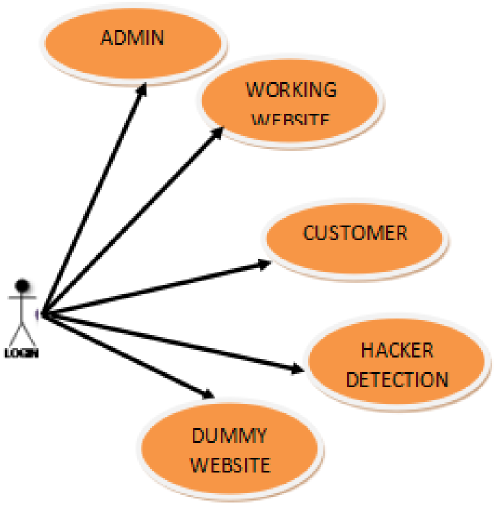


Figure 7.1 Login Module

7.1.2 ADMIN MODULE:

It includes:-

- Administrator designation

- Control over the database
- Updating information
- Authenticating customers

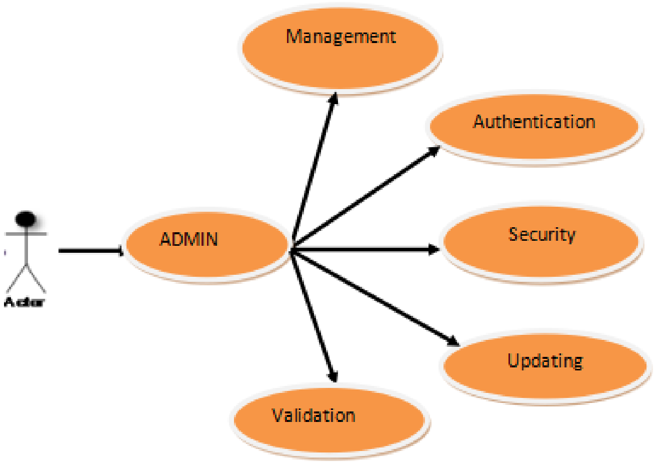


Figure 7.2 Admin Module

7.1.3 CUSTOMER MODULE:

It includes:-

- Customer login and information gathering
- Retrieving queries.
- Posting experiences

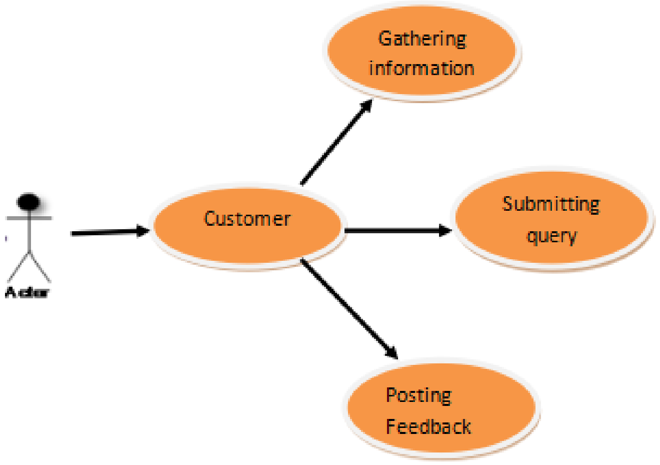


Figure 7.3 Customer Module

7.1.4 HACKER TRACKING MODULE:

It includes:-

- Hacker detail information

- Hacking information and precautions
- Prevention and other references

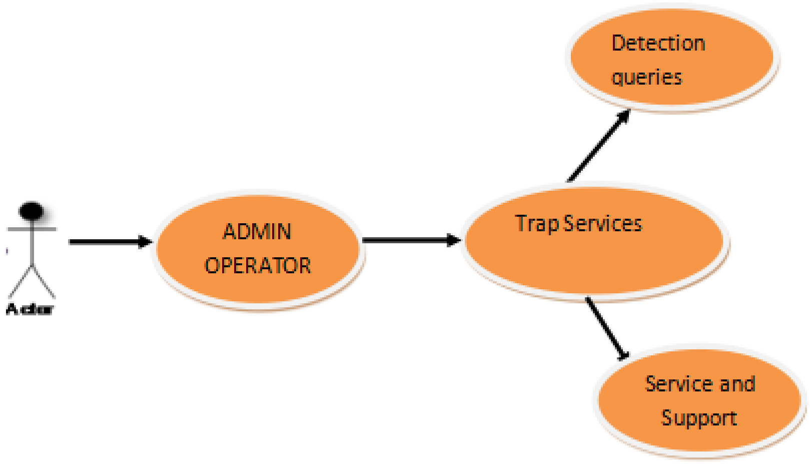


Figure 7.4 Hacker Tracking Module

7.1.5 EMPLOYEE MODULE:

This module has information like:-

- Organization information
- Event information
- Promotion of policies
- Guidance and suggestions

7.2 DATABASE TABLES

7.2.1 TABLE LOGIN

(

l_id varchar2(20) not null primary key,

m_password varchar2(20),

login_type varchar2(20),

reg_date date

);

7.2.2 TABLE ADMINISTRATOR

(

Admin_ID number (8) primary key,

```
l_id varchar2(20),
FirstName varchar2(30),
MiddleName varchar2(30),
LastName varchar2(30),
cellno varchar2(15),
Email varchar2(50),
AddressLine1 varchar2(50),
AddressLine2 varchar2(50),
City varchar2(20),
States varchar2(20),
Country varchar2(20)
);
```

7.2.3 TABLE CUSTOMER

```
(
ac_id number (8) not null,
accno varchar2(10)not null primary key,
cname1 varchar2(20),
cname2 varchar2(20),
cname3 varchar2(20),
vaddress1 varchar2(100),
ccity varchar2(50),
cstate varchar2(50),
ccountry varchar2(50),
cpincode varchar2(20),
l_id varchar2(20) not null ,
cpassword varchar2(20),
ccontactno varchar2(20),
cDOBdate varchar2(20),
cDOBmonth varchar2(20),
cDOByear varchar2(20),
cemail varchar2(100),
coccupation varchar2(20),
annualincome varchar2(20),
caccounttype varchar2(20),
cDDno varchar2(20),
DDamount number (8)
);
```

7.2.4 TABLE INFORMATION

```
(
inf_id number (8) not null primary key,
l_id varchar2(20) not null ,
password varchar2(20),
accno varchar2(10) not null ,
balance number (8)
);
```

7.2.5 TABLE HISTORY

```
(
h_id number (8) primary key,
dated varchar2(20),
type varchar2(20),
l_id varchar2(20) not null ,
accno1 varchar2(10) not null ,
amount number (8),
accno2 varchar2(10) not null ,
balance number (8)
);
```

7.2.6 TABLE HACKERINFO

```
 (
HIT number (8) primary key,
HID varchar2(50) not null,
LType varchar2(50),
Pass varchar2(50),
RequestMethod varchar2(100),
RequestURL varchar2(100),
RequestProtocol varchar2(100),
ServerName varchar2(100),
RequestPath varchar2(100),
ServerPort varchar2(100),
RemoteAddress varchar2(100),
RemoteHost varchar2(100),
Locale varchar2(100),
UserAgent varchar2(500),
Status varchar2(10),
Dated date
);
```

CHAPTER 8

TESTING AND EVALUATION

Testing and evaluation of honeypots play a critical role in ensuring their effectiveness and identifying any weaknesses or shortcomings. By subjecting honeypots to rigorous testing, organizations can validate their deployment, assess their capabilities, and make informed decisions regarding their implementation. Here are some key aspects to consider when testing and evaluating honeypots:

Functionality Testing: This involves verifying that the honeypot operates as intended and behaves convincingly as a legitimate system or service. It includes testing the emulation of protocols, services, and behaviors to ensure they accurately mimic real assets.

Attack Simulation: Simulating various types of attacks against the honeypot helps evaluate its ability to attract and capture malicious activity. It involves launching known attack techniques against the honeypot and analyzing the responses and interactions with the attackers.

Detection and Alerting: Assessing the honeypot's ability to detect and alert on malicious activities is crucial. This can involve evaluating the effectiveness of intrusion detection systems (IDS) and logging mechanisms within the honeypot to ensure they capture and report relevant events accurately.

False Positive and False Negative Testing: Determining the rate of false positives (legitimate activity flagged as malicious) and false negatives (malicious activity not detected) is important for assessing the honeypot's accuracy. By analyzing the effectiveness of the honeypot in differentiating between genuine and malicious activities, organizations can fine-tune its configuration and minimize false alarms.

Performance and Scalability: Evaluating the performance and scalability of honeypots is necessary to ensure they can handle anticipated levels of traffic and attacker interactions.

Stress testing the honeypot by generating high volumes of network traffic or launching multiple simultaneous attacks helps identify any bottlenecks or performance limitations.

Forensic Analysis: Conducting forensic analysis on the data captured by the honeypot allows organizations to gain insights into attacker techniques, intentions, and potential vulnerabilities. Analyzing the captured data can provide valuable intelligence for incident response and security improvements.

Vulnerability Assessment: Regular vulnerability assessments on the honeypot itself help identify any weaknesses or potential vulnerabilities that attackers may exploit. This includes assessing the underlying operating system, software, and configurations to ensure they are patched and hardened.

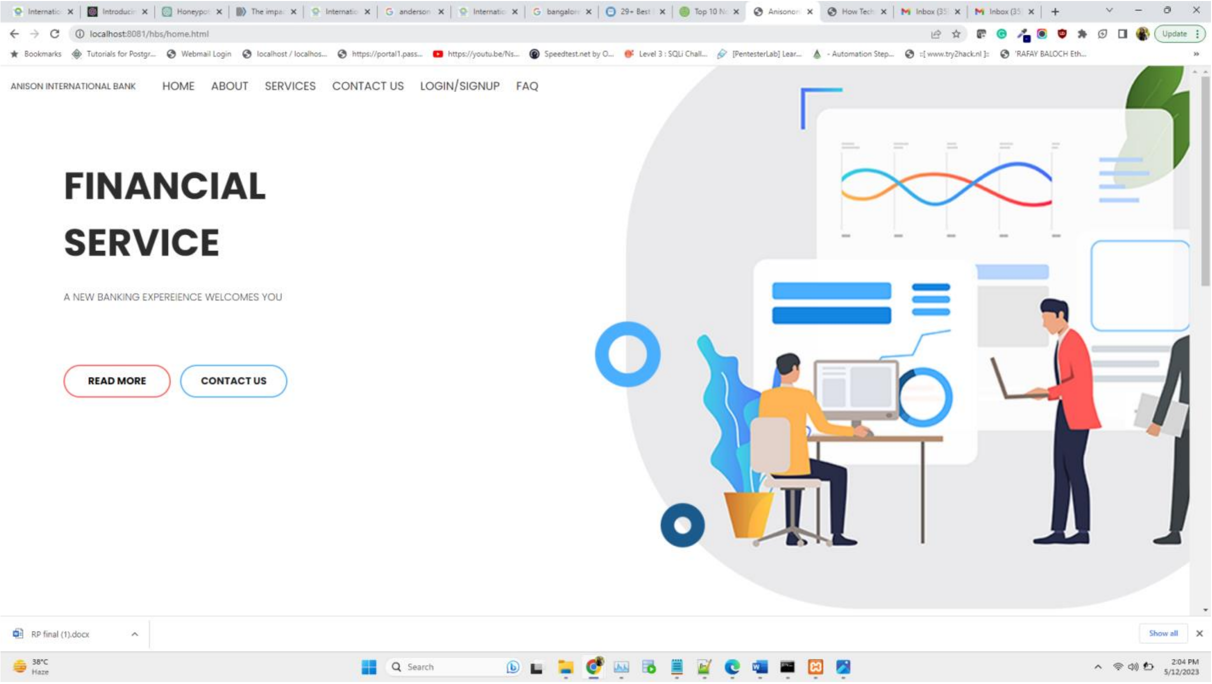
Compliance and Legal Considerations: Evaluating the honeypot's compliance with relevant laws, regulations, and ethical guidelines is essential. Organizations must ensure that the honeypot deployment aligns with legal requirements and does not infringe upon any privacy or data protection regulations.

Continuous Monitoring and Updates: Testing and evaluation of honeypots should be an ongoing process. Regular monitoring and updates are necessary to ensure the honeypot remains effective against evolving threats and accurately emulates current attack vectors.

By conducting thorough testing and evaluation, organizations can identify potential weaknesses and make informed decisions to improve the effectiveness of their honeypots. This process helps organizations stay ahead of attackers, enhance their security posture, and gain valuable insights into emerging threats and attack techniques.

CHAPTER 9

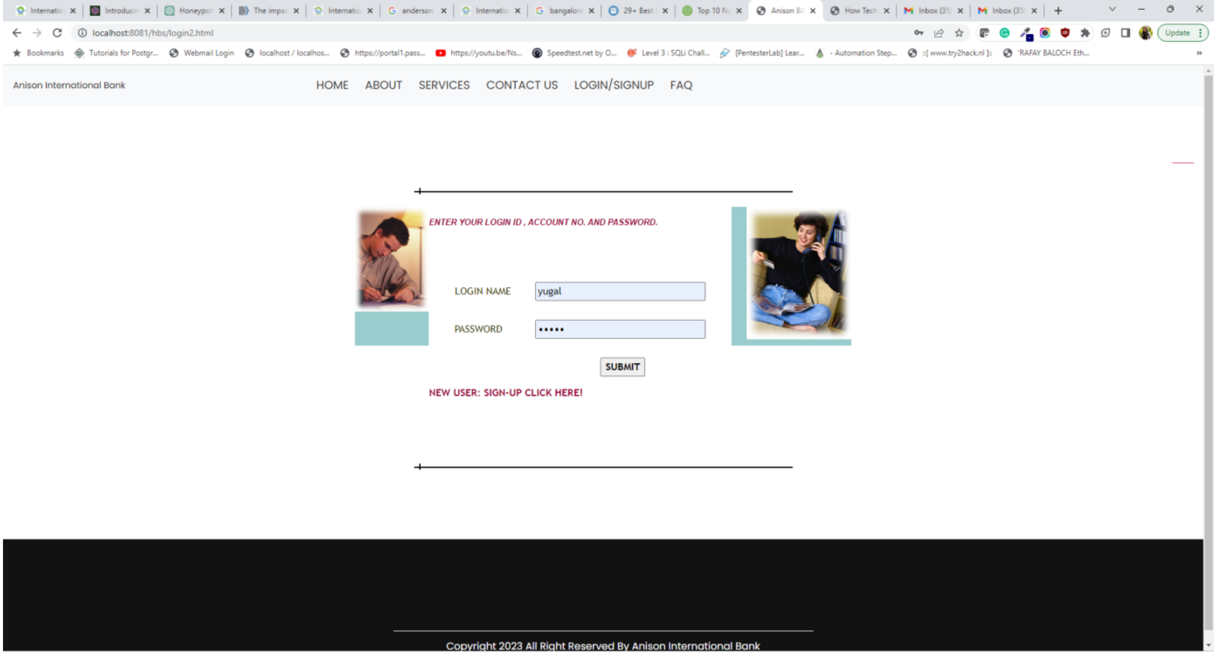
PROJECT SNAPSHOTS



FINANCIAL SERVICE

A NEW BANKING EXPEREIENCE WELCOMES YOU

Figure 9.1 Home Page



ENTER YOUR LOGIN ID , ACCOUNT NO. AND PASSWORD.

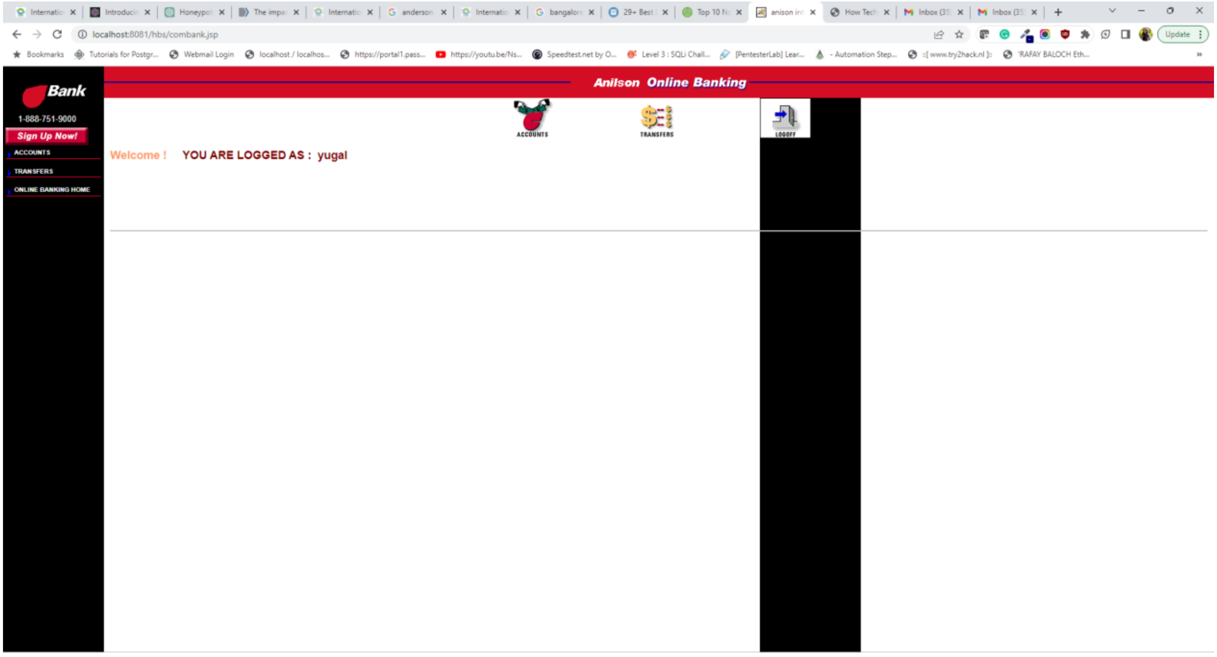
LOGIN NAME: yugal

PASSWORD: •••••

NEW USER: SIGN-UP CLICK HERE!

Copyright 2023 All Right Reserved By Anison International Bank

Figure 9.2 Login Page

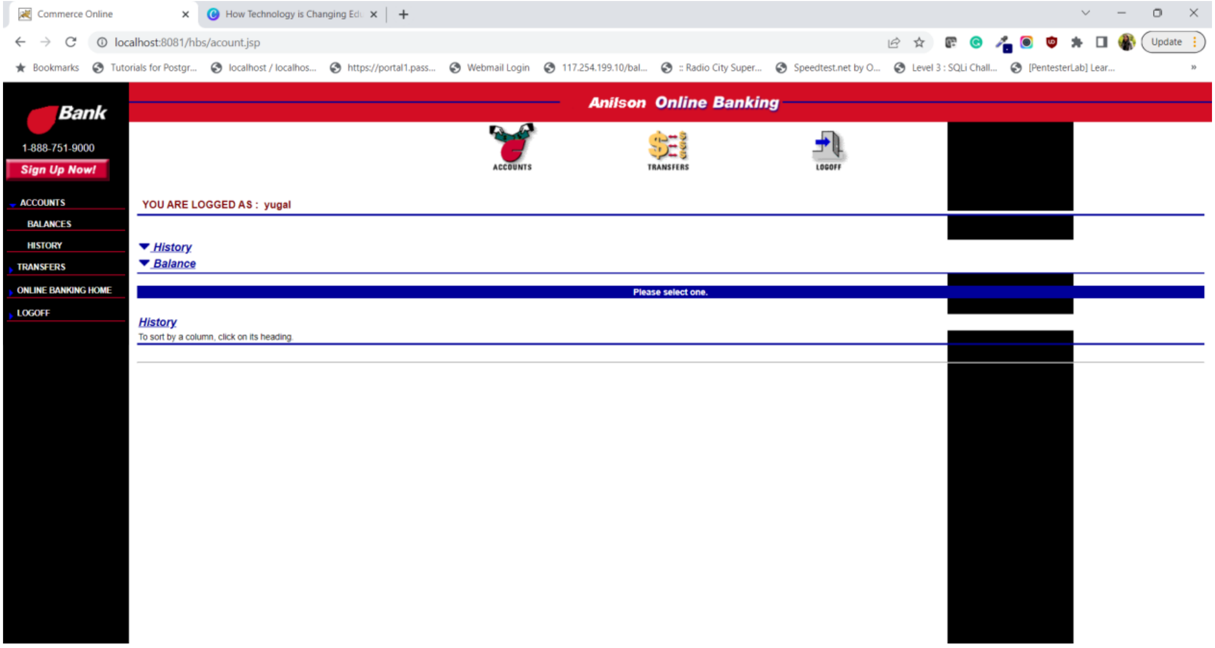


localhost:8081/hbs/combank.jsp

Anilson Online Banking

Welcome ! YOU ARE LOGGED AS : yugal

Figure 9.3 Logged-in Page



localhost:8081/hbs/acount.jsp

Anilson Online Banking

YOU ARE LOGGED AS : yugal

History

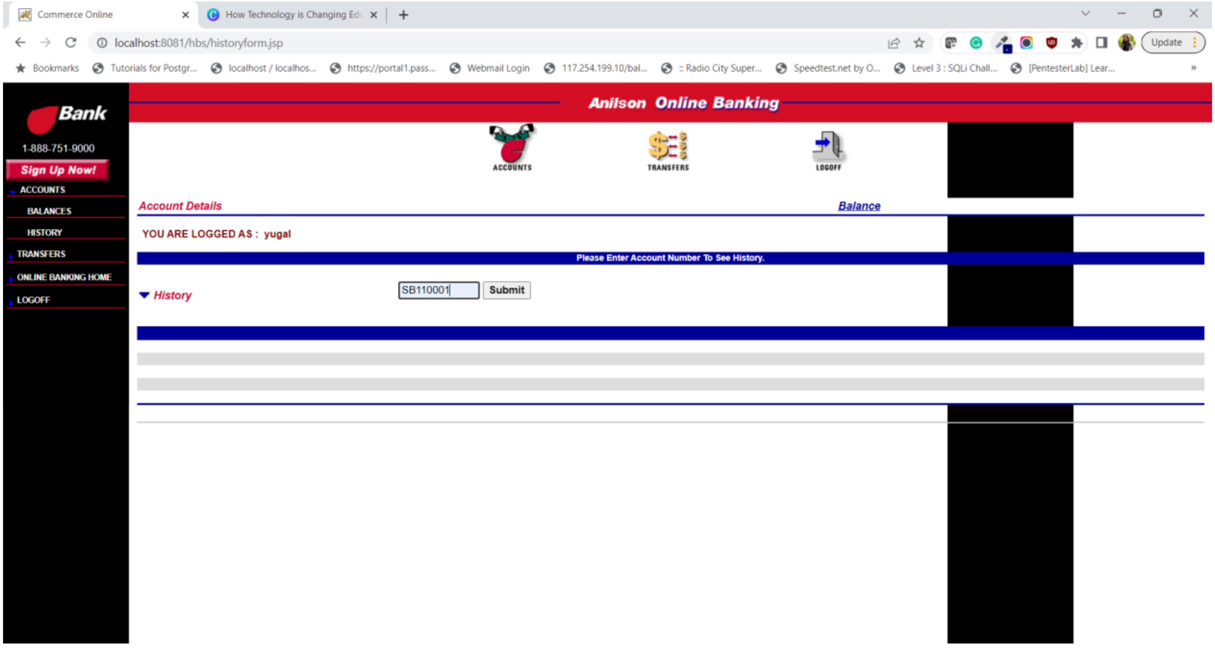
Balance

Please select one.

History

To sort by a column, click on its heading

Figure 9.4 Account Information Page

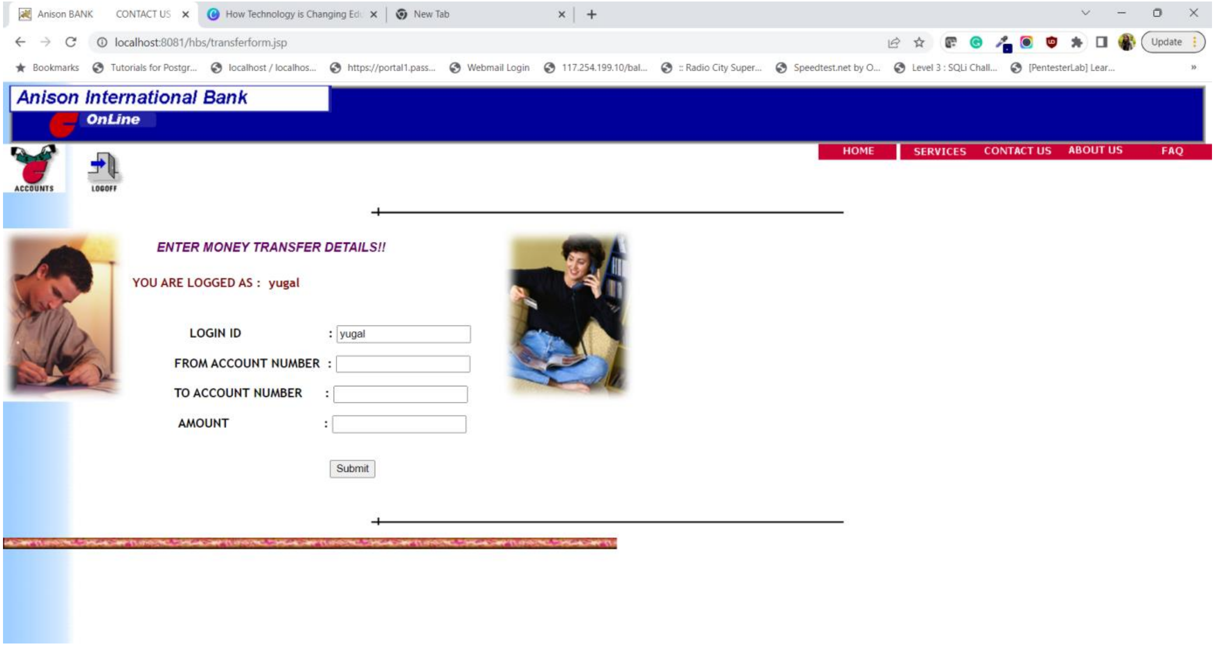


YOU ARE LOGGED AS : yugal

Please Enter Account Number To See History.

SB110001

Figure 9.5 History Page


Anison BANK
CONTACT US
How Technology is Changing Ed
New Tab
localhost:8081/hbs/transferform.jsp
Anison International Bank
OnLine
HOME
SERVICES
CONTACT US
ABOUT US
FAQ
ACCOUNTS
LOGOFF
Submit

ENTER MONEY TRANSFER DETAILS!!

YOU ARE LOGGED AS : yugal

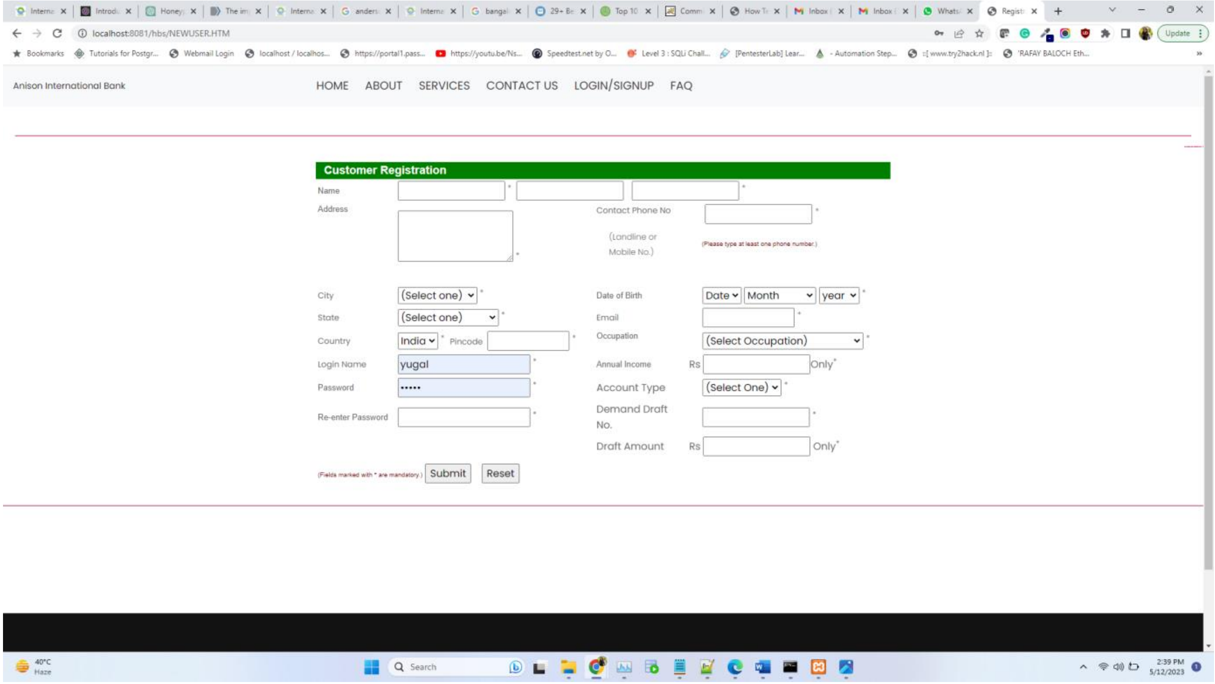
LOGIN ID : yugal

FROM ACCOUNT NUMBER :

TO ACCOUNT NUMBER :

AMOUNT :

Figure 9.6 Transaction Page

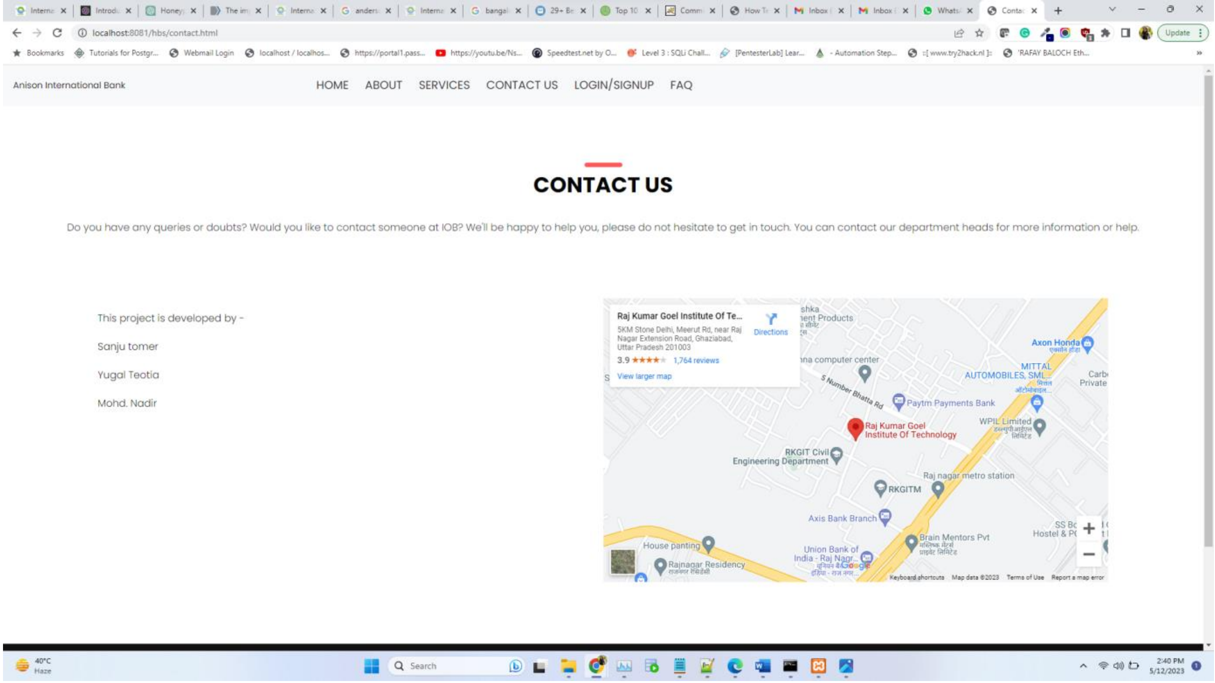


Customer Registration

Name	*, *		
Address	*	Contact Phone No (Landline or Mobile No.)	* (Please type at least one phone number.)
City	(Select one) *	Date of Birth	Date Month year *
State	(Select one) *	Email	*
Country	India * Pincode *	Occupation	(Select Occupation) *
Login Name	yugal *	Annual Income	Rs Only*
Password	••••• *	Account Type	(Select One) *
Re-enter Password	*	Demand Draft No.	*
		Draft Amount	Rs Only*

(Fields marked with * are mandatory.) Submit Reset

Figure 9.7 Registration Page



CONTACT US

Do you have any queries or doubts? Would you like to contact someone at IOB? We'll be happy to help you, please do not hesitate to get in touch. You can contact our department heads for more information or help.

This project is developed by -

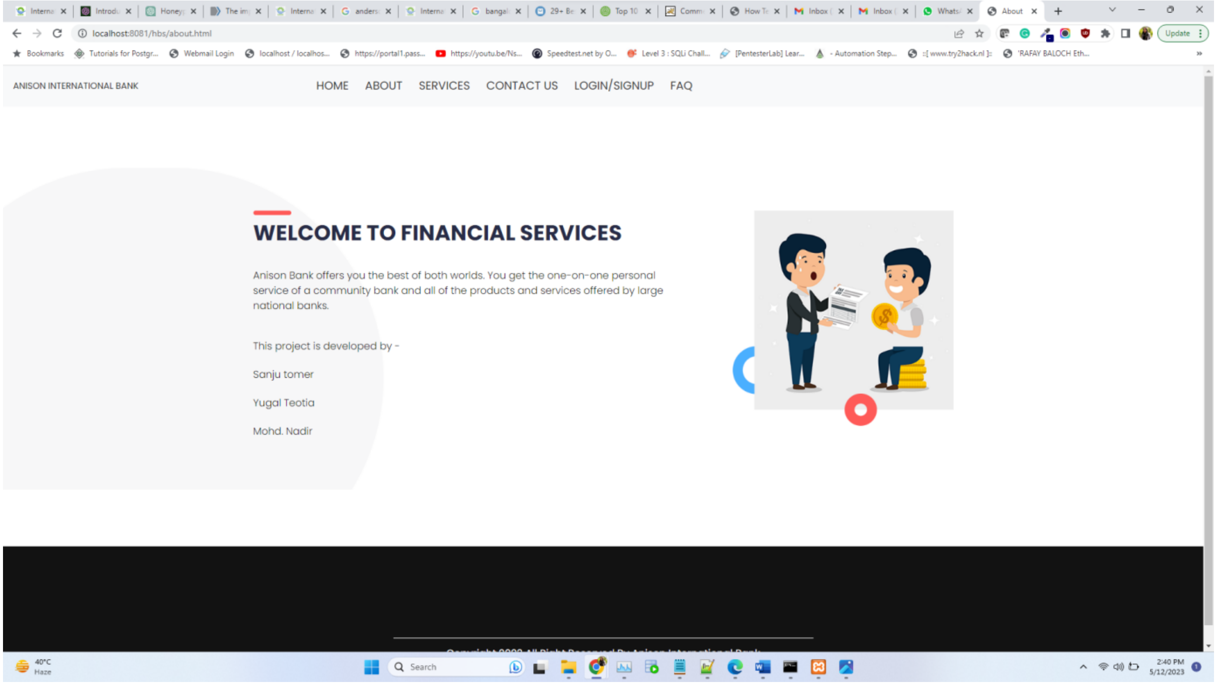
Sanju tomer

Yugal Teotia

Mohd. Nadir

Raj Kumar Goel Institute Of Te...
5KM Stone Delhi, Meerut Rd, near Raj Nagar Extension Road, Ghaziabad, Uttar Pradesh 201003
3.9 ★★★★ 1,764 reviews
View larger map

Fiagure 9.8 Contact Page



localhost:8081/hbs/about.html

ANISON INTERNATIONAL BANK

HOME ABOUT SERVICES CONTACT US LOGIN/SIGNUP FAQ

WELCOME TO FINANCIAL SERVICES

Anison Bank offers you the best of both worlds. You get the one-on-one personal service of a community bank and all of the products and services offered by large national banks.

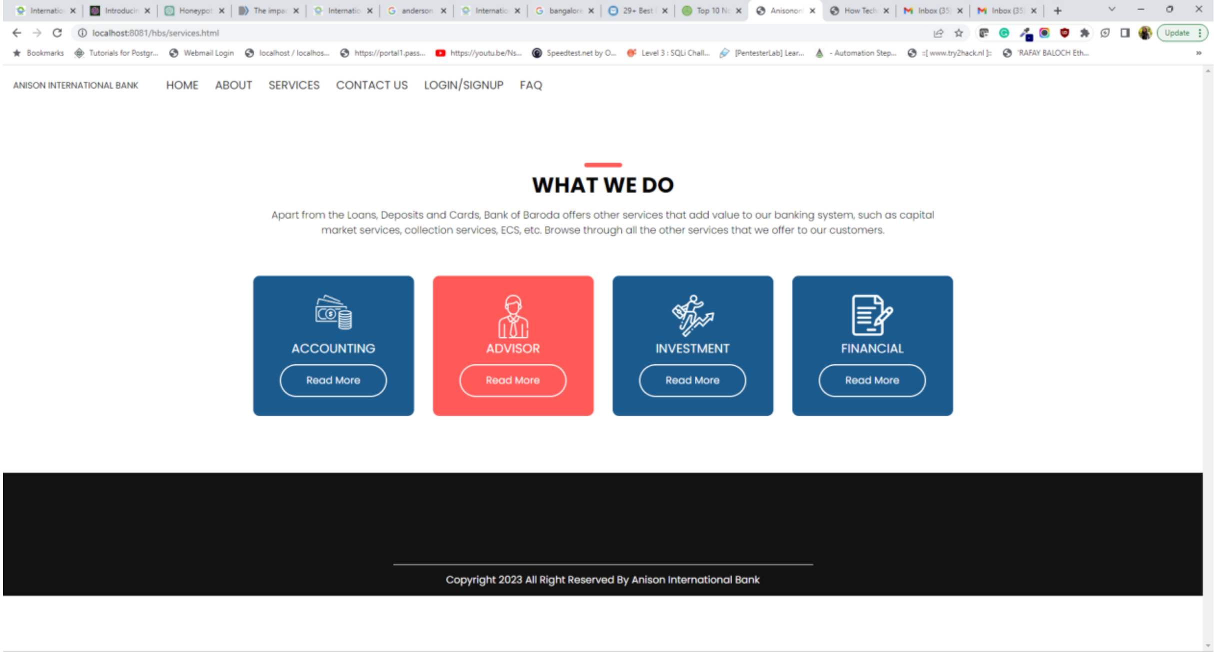
This project is developed by -

Sanju tomer

Yugal Teotia

Mohd. Nadir

Figure 9.9 About Page



localhost:8081/hbs/services.html

ANISON INTERNATIONAL BANK

HOME ABOUT SERVICES CONTACT US LOGIN/SIGNUP FAQ

WHAT WE DO

Apart from the Loans, Deposits and Cards, Bank of Baroda offers other services that add value to our banking system, such as capital market services, collection services, ECS, etc. Browse through all the other services that we offer to our customers.

Copyright 2023 All Right Reserved By Anison International Bank

Figure 9.10 Services Page

CHAPTER 10

LIMITATIONS

While honeypots offer numerous benefits in cybersecurity, they also have certain limitations that need to be considered. These limitations include:

False Sense of Security: Honeypots can create a false sense of security if not properly implemented and monitored. Attackers may specifically target and exploit honeypots, which could divert their attention from real assets. Organizations must understand that honeypots are just one element of a comprehensive security strategy and should not solely rely on them for protection.

Resource Intensive: Deploying and maintaining honeypots can be resource-intensive. Each honeypot requires careful setup, monitoring, and maintenance to ensure it remains effective. Organizations need to allocate sufficient resources, including time, personnel, and infrastructure, to manage and update their honeypots regularly.

Legality and Ethics: The use of honeypots raises legal and ethical considerations. Depending on the jurisdiction, there may be laws and regulations that govern the creation and use of honeypots. Organizations must ensure they comply with all relevant laws and respect ethical guidelines to avoid any legal or reputational consequences.

Network Complexity: In complex network environments, deploying honeypots can be challenging. Organizations with large-scale networks may find it difficult to accurately mimic their real systems and services, which could impact the effectiveness of honeypots in attracting attackers. Additionally, integrating honeypots into existing network architectures can introduce complexities that need to be carefully managed.

Maintenance and Updates: Honeypots need regular maintenance and updates to remain effective. Attackers are constantly evolving their techniques, and honeypots must be kept up to date to accurately mimic the latest vulnerabilities and attack vectors. Failure to update honeypots may result in outdated information and ineffective threat detection.

Insider Threats: Honeypots are typically designed to attract external attackers, but they may not effectively detect or mitigate insider threats. Malicious insiders with legitimate access to the network may bypass or avoid interacting with honeypots, making them less effective in detecting internal threats.

Potential Impact on Legitimate Users: If not properly isolated or configured, honeypots can potentially impact legitimate users and services. Attackers targeting honeypots may inadvertently disrupt or compromise adjacent systems, leading to unintended consequences. Careful consideration must be given to the network segmentation and isolation of honeypots to minimize the risk to genuine users.

It is essential for organizations to be aware of these limitations and carefully evaluate whether honeypots align with their specific security needs and resources. Proper planning, implementation, and ongoing management are crucial to maximize the benefits of honeypots while mitigating their limitations.

CHAPTER 11

FUTURE SCOPE

The future scope of honeypots, which are decoy systems designed to attract and monitor malicious activity, is quite promising. As cyber threats continue to evolve and grow in sophistication, honeypots offer several potential developments and applications. Here are some future directions for honeypot technology:

Advanced Threat Detection: Honeypots can be enhanced to provide more comprehensive threat detection capabilities. By analyzing the behavior of attackers and capturing their techniques, organizations can gain valuable insights into emerging attack vectors and develop stronger security measures to counteract them.

Deception Technology: Honeypots can serve as a fundamental component of deception technology, which aims to mislead and divert attackers from real assets. Future honeypots may become even more sophisticated in mimicking legitimate systems and services, making them harder to detect and increasing their effectiveness in diverting malicious actors.

Internet of Things (IoT) Security: As the number of IoT devices increases, so does the need for effective security measures. Honeypots can be specifically tailored to mimic vulnerable IoT devices, attracting attackers and capturing their activities. This data can help identify vulnerabilities in IoT systems and enable organizations to develop more secure and resilient IoT architectures.

Threat Intelligence: Honeypots can play a vital role in generating threat intelligence by collecting valuable data on attacker techniques, trends, and motivations. This information can be analyzed and shared with security communities, enabling organizations to stay updated on the latest threats and improve their overall security posture.

Cloud-Based Honeypots: With the widespread adoption of cloud computing, honeypots can be deployed in cloud environments to monitor and mitigate threats specific to cloud infrastructure. Cloud-based honeypots can provide valuable insights into attacks targeting cloud services, allowing organizations to enhance their cloud security strategies.

Machine Learning and Artificial Intelligence: Integrating machine learning and artificial intelligence algorithms with honeypots can improve their capabilities in real-time threat detection and response. By leveraging these technologies, honeypots can identify patterns, anomalies, and new attack vectors, enabling automated and proactive defense mechanisms.

Collaboration and Information Sharing: Honeypot networks can be established to facilitate collaboration and information sharing among organizations and security researchers. This collective effort can lead to a broader understanding of threats, improved incident response capabilities, and the development of more effective countermeasures.

Industrial Control Systems (ICS) Security: Honeypots can be customized to mimic industrial control systems, such as those used in critical infrastructure. By attracting attackers targeting these systems, organizations can gain insights into potential vulnerabilities and enhance the security of vital infrastructure components.

It's important to note that as honeypot technology evolves, so do the techniques employed by malicious actors. It is crucial to continually update and adapt honeypots to stay ahead of emerging threats and ensure their effectiveness in the ever-changing landscape of cybersecurity.

CONCLUSION

Honeypots continue to hold significant value in the field of cyber security and offer promising future prospects. These deceptive systems play a crucial role in attracting, monitoring, and analyzing malicious activity, providing organizations with valuable insights into emerging threats and vulnerabilities.

The future of honeypots lies in their ability to evolve and adapt to the changing landscape of cyber threats. Advanced threat detection, deception technology, and IoT security are just a few areas where honeypots can make a substantial impact. By integrating machine learning and artificial intelligence algorithms, honeypots can enhance real-time threat detection and response capabilities.

Furthermore, honeypots can contribute to the collective knowledge of the security community by generating valuable threat intelligence and facilitating collaboration and information sharing. This collaborative effort can lead to improved incident response strategies, stronger countermeasures, and a better understanding of evolving attack techniques.

As organizations increasingly rely on cloud infrastructure and industrial control systems, honeypots can be customized and deployed in these domains to identify vulnerabilities and enhance their security. Additionally, honeypots can serve as an essential component of comprehensive cybersecurity strategies, complementing other defense mechanisms and strengthening overall resilience.

However, it is essential to recognize that honeypots must continually evolve to keep pace with the ever-changing tactics of malicious actors. Regular updates, customization, and continuous monitoring are crucial to maintain the effectiveness of honeypots in identifying and mitigating emerging threats.

In conclusion, honeypots are poised to play a vital role in the future of cybersecurity. With their ability to gather threat intelligence, divert and deceive attackers, and enhance incident response capabilities, honeypots offer significant value in the ongoing battle against cyber threats.

REFERENCES

[1]. M. Spitzner, "Honeypots: Tracking Hackers", IEEE Security & Privacy [2003], vol. 1, no. 2, pp. 15-19.

[2]. L. R. Alejos and J. Lopez, "A Survey of Honeypot Technologies", Journal of Information Assurance and Security [2012], vol. 7, no. 3, pp. 180-190.

[3]. R. L. Thomas and T. V. Lakshman, "Honeypots: A Security Strategy for Detecting Intrusions", International Journal of Computer Science and Security [2008], vol. 2, no. 1, pp. 14-20.

[4]. A. M. Alazab, R. Layton, and S. Venkatraman, "Survey on Honeypot-based Intrusion Detection Systems", Computers & Security [2011], vol. 30, no. 5, pp. 372-391.

[5]. N. Provos, "A Virtual Honeypot Framework", Proceedings of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots'07) [2007], Cambridge, MA, USA.

[6]. J. B. Caballero, "Catching the Kill Chain: Honeypots as Early Warning Systems", Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC'06) [2006], Miami Beach, FL, USA.

[7]. S. Bhatia and R. K. Sharma, "Honeypots: A Comprehensive Review", Proceedings of the 7th International Conference on Security of Information and Networks (SIN'14) [2014], Glasgow, Scotland, UK.

[8]. J. Honek and V. Krmicek, "Honeytrap – An Advanced Honeypot Framework", Journal of Digital Forensics, Security and Law [2011], vol. 6, no. 3, pp. 55-68.

[9]. G. Vigna, R. A. Kemmerer, and C. Kruegel, "Honeyd: Creating Virtual Honeynets with Virtual Machine Technology", Proceedings of the 9th USENIX Security Symposium [2000], Washington, D.C., USA.

[10]. A. Al-Aziz and M. N. H. Siddique, "An Evaluation of Honeypot Solutions: A Survey", International Journal of Computer Applications [2014], vol. 94, no. 13, pp. 9-15.