

A

Major Project

On

**MITIGATING DATA THEFTS**

(Submitted in partial fulfilment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

BY

P. Sruthi Laya (177R1A05B0)

S. Abhinav Peter (177R1A05A8)

S. Aishwarya (177R1A05B1)

R. Divya Raja Lakshmi (177R1A05A3)

Under the Guidance of

**Mrs. D. Sandya Rani**

(Assistant Professor)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CMR TECHNICAL CAMPUS**

**UGC AUTONOMOUS**

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under

Section 2(f) & 12(B) of the UGC Act.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

**2017-2021**

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



## CERTIFICATE

This is to certify that the project entitled “MITIGATING DATA THEFTS” being submitted by **P. SRUTHI LAYA (177R1A05B0), S. ABHINAV PETER (177R1A05A8), S. AISHWARYA (177R1A05B1) & R. DIVYA RAJYA LAKSHMI (177R1A05A3)** in partial fulfilment of the requirements for the award of the degree of B. Tech in Computer Science and Engineering of the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafied work carried out by him/her under our guidance and supervision during the year 2020-2021.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

**Mrs. D. Sandya Rani**

**INTERNAL GUIDE**

**Dr. K. Srujan Raju**

**HOD**

**Dr. A. Raji Reddy**

**DIRECTOR**

**EXTERNAL EXAMINER**

Submitted for viva voce Examination held on \_\_\_\_\_

## ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We take this opportunity to express my profound gratitude and deep regard to my guide

**Mrs. D. Sandya Rani**, Assis. Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Mr. J. Narasimha Rao, Mr. B. P. Deepak Kumar, Mr. K. Murali, Dr. Suwarna Gothane and Mr. B. Ramji** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to the Head of the Department **Dr. K. Srujan Raju** for providing excellent infrastructure and a nice atmosphere for completing this project successfully.

We are obliged to our Director **Dr. A. Raji Reddy** for being cooperative throughout the course of this project. We would like to express our sincere gratitude to our Chairman Sri. **Ch. Gopal Reddy** for his encouragement throughout the course of this project

The guidance and support received from all the members of **CMR TECHNICAL CAMPUS** who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity thank our family for their constant encouragement without which this assignment would not be possible. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project.

**P. SRUTHI LAYA (177R1A05B0)**

**S. ABHINAV PETER (177R1A05A8)**

**S. AISHWARYA (177R1A05B1)**

**R. DIVYA RAJYA LAKSHMI (177R1A05A3)**

## **ABSTRACT**

Cloud computing promises to significantly change the way we use computers and access and store our personal and business information. With these new computing and communications paradigms arise new data security challenges. Existing data protection mechanisms such as encryption have failed in preventing data theft attacks, especially those perpetrated by an insider to the cloud provider. We propose a different approach for securing data in the cloud using offensive decoy technology. We monitor data access in the cloud and detect abnormal data access patterns. When unauthorized access is suspected and then verified using challenge questions, we launch a disinformation attack by returning large amounts of decoy information to the attacker. This protects against the misuse of the user's real data. Experiments conducted in a local file setting provide evidence that this approach may provide unprecedented levels of user data security in a Cloud environment.

## **LIST OF FIGURES**

<b>FIGURE NO</b>	<b>FIGURE NAME</b>	<b>PAGE NO</b>
Fig 3.1	Project Architecture	6
Fig 3.2	Use Case Diagram	8
Fig 3.3	Activity Diagram	9
Fig 3.4	Class Diagram	10
Fig 3.5	Sequence Diagram	11

## LIST OF SCREENSHOTS

<b>SCREENSHOT NO.</b>	<b>SCREENSHOT NAME</b>	<b>PAGE NO.</b>
Screenshot 5.1	Home Page	15
Screenshot 5.2	User Login Page	16
Screenshot 5.3	File Upload Page	17
Screenshot 5.4	Security Question Page	18
Screenshot 5.5	My Files Page	19
Screenshot 5.6	View Alert Page	20
Screenshot 5.7	Password Change Page	21
Screenshot 5.8	User Details Page	22
Screenshot 5.9	View Files Page	23
Screenshot 5.10	Upload Details Page	24

# TABLE OF CONTENTS

<b>ABSTRACT</b>	i
<b>LIST OF FIGURES</b>	ii
<b>LIST OF SCREENSHOTS</b>	iii
<b>1. INTRODUCTION</b>	1
1.1 PROJECT SCOPE	1
1.2 PROJECT PURPOSE	1
1.3 PROJECT FEATURES	1
<b>2. SYSTEM ANALYSIS</b>	2
2.1 PROBLEM DEFINITION	2
2.2 EXISTING SYSTEM	2
2.2.1 LIMITATIONS OF THE EXISTING SYSTEM	3
2.3 PROPOSED SYSTEM	3
2.3.1 ADVANTAGES OF PROPOSED SYSTEM	3
2.4 FEASIBILITY STUDY	3
2.4.1 ECONOMIC FESIBILITY	4
2.4.2 TECHNICAL FEASIBILITY	4
2.4.3 SOCIAL FEASIBILITY	4
2.5 HARDWARE & SOFTWARE REQUIREMENTS	5
2.5.1 HARDWARE REQUIREMENTS	5
2.5.2 SOFTWARE REQUIREMENTS	5
<b>3. ARCHITECTURE</b>	6
3.1 PROJECT ARCHITECTURE	6
3.1.1 DESCRIPTION	6
3.2 MODULES	7
3.3 USE CASE DIAGRAM	8
3.4 ACTIVITY DIAGRAM	9
3.5 CLASS DIAGRAM	10
3.6 SEQUENCE DIAGRAM	11
<b>4. IMPLEMENTATION</b>	12
4.1 SAMPLE CODE	12

<b>5. SCREENSHOTS</b>	15
<b>6. TESTING</b>	25
6.1 INTRODUCTION TO TESTING	25
6.2 TYPES OF TESTING	25
6.2.1 UNIT TESTING	25
6.2.2 INTEGRATION TESTING	25
6.2.3 FUNCTIONAL TESTING	26
6.3 TEST CASES	26
6.3.1 UPLOADING FILES	26
6.3.2 DETECTION	27
<b>7. CONCLUSION &amp; FUTURE SCOPE</b>	28
7.1 PROJECT CONCLUSION	28
7.2 FUTURE SCOPE	28
<b>8. BIBLIOGRAPHY</b>	29
8.1 REFERENCES	29
8.2 WEBSITES	29



# **1. INTRODUCTION**

# 1. INTRODUCTION

## 1.1 PROJECT SCOPE

The project is titled as “Mitigating Data Thefts”. This software provides facility for accessing a user’s profile online. The user needs to sign up or register using his/her respective details. He/she has to give two-way steps of security. One is the time limit which she/he can access the profile and the other one is the security question. Last but not the least password should also be given for authorizing.

## 1.2 PROJECT PURPOSE

This project has been developed to improve the security measures for any kind of applications. This System will help the important files or data to be saved from unknown hackers or illegitimate users. It is a different approach for securing data in the cloud using offensive decoy technology. We monitor data access in the cloud and detect abnormal data access patterns. When unauthorized access is suspected and then verified using challenge questions, we launch a disinformation attack by returning large amounts of decoy information to the attacker. This protects against the misuse of the user’s real data.

## 1.3 PROJECT FEATURES

The main feature of this project is security for your files. We have a user Login and an Admin login where an admin can keep track of number of authorized users and attackers. In this way we get to know who are trying to access our files with out our knowledge and can take respective measures towards it. The page is designed dynamically giving fast responses from server. The three measures we are using for our security purpose are password, security question and time limit where we can access. The user can come and visit his profile in his respective timing of access only. The admin updates the no of attackers and real users. The important and confidential data is secured in this way.

## **2. SYSTEM ANALYSIS**

## **2. SYSTEM ANALYSIS**

### **SYSTEM ANALYSIS**

System Analysis is the important phase in the system development process. The System is studied to the minute details and analysed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

#### **2.1 PROBLEM DEFINITION**

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

#### **2.2 EXISTING SYSTEM**

Existing data protection mechanisms such as encryption have failed in preventing data theft attacks, especially those perpetrated by an insider to the cloud provider. Much research in Cloud computing security has focused on ways of preventing unauthorized and illegitimate access to data by developing sophisticated access control and encryption mechanisms. However, these mechanisms have not been able to prevent data compromise.

### **2.2.1 LIMITATIONS OF EXISTING SYSTEM**

- Data theft
- Unauthorized and illegitimate access
- Prevention of data compromise is not achieved

### **2.3 PROPOSED SYSTEM**

We propose a completely different approach to securing the cloud using decoy information technology, that we have come to call Fog computing. We use this technology to launch disinformation attacks against malicious insiders, preventing them from distinguishing the real sensitive customer data from fake worthless data. The decoys, then, serve two purposes:

- (1) validating whether data access is authorized when abnormal information access is detected, and
- (2) confusing the attacker with bogus information.

#### **2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM**

- More agile
- Economical
- Easily maintained
- Centralization of infrastructure and data
- Efficient and reliable
- More secure

### **2.4 FEASIBILITY STUDY**

The feasibility of the project is analysed 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.

Three key considerations involved in the feasibility analysis are

- Economic Feasibility

- Technical Feasibility
- Social Feasibility

#### **2.4.1 ECONOMIC FEASIBILITY**

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

The costs conduct a full system investigation.

- The cost of the hardware and software
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

#### **2.4.2 TECHNICAL FEASIBILITY**

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

#### **2.4.3 BEHAVIORAL FEASIBILITY**

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible.

## 2.5 HARDWARE & SOFTWARE REQUIREMENTS

### 2.5.1 HARDWARE REQUIREMENTS

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

Processor	- Pentium –III
Speed	- 1.1 Ghz
RAM	- 256 MB (min)
Hard Disk	- 20 GB
Floppy Drive	- 1.44 MB
Key Board	- Standard Windows Keyboard
Mouse	- Two or Three Button Mouse
Monitor	- SVGA

### 2.5.2 SOFTWARE REQUIREMENTS

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements:

Operating System	: Windows95/98/2000/XP
Application Server	: Tomcat5.0/6.X
Front End	: HTML, Java
Scripts	: JavaScript.
Server side Script	: Java Server Pages.
Database	: MySQL
Database Connectivity	: JDBC.

# **3. ARCHITECTURE**



### 3. ARCHITECTURE

#### 3.1 PROJECT ARCHITECTURE

The Project Architecture shows the overview of the project. It shows how a user gets his data or file depending up on whether he/she is real user or attacker. The detailed architecture is explained below:

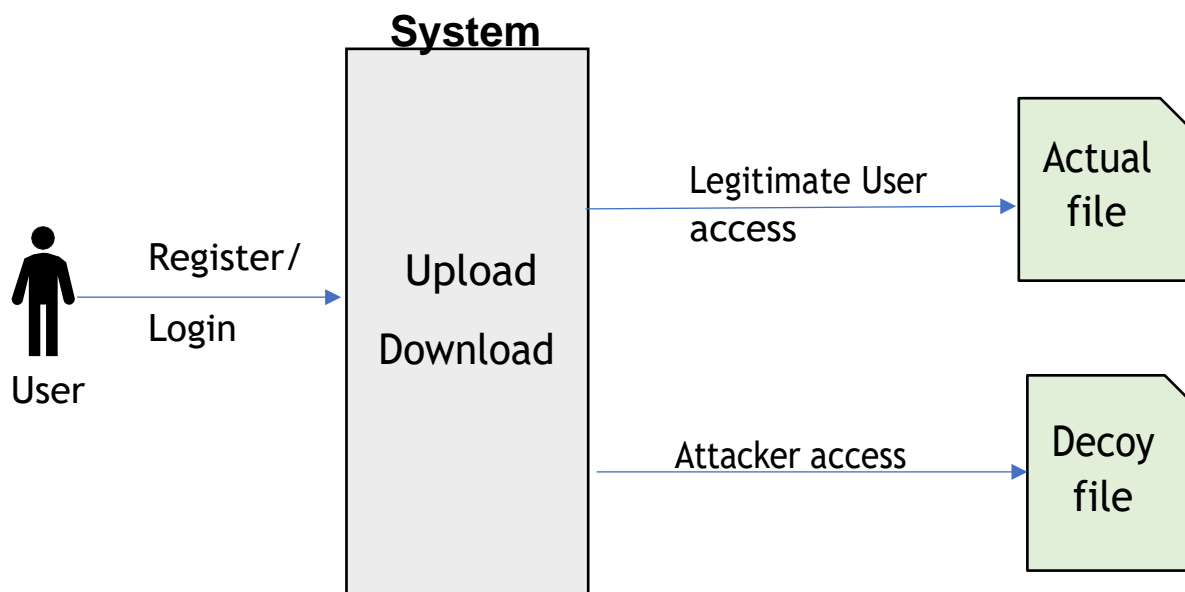


Fig 3.1 Project Architecture of Mitigating Data Thefts

##### 3.1.1 DESCRIPTION

The user tries to register or login to access his/her files. When unauthorized access is suspected and then verified using challenge questions, we launch a disinformation attack by returning large amounts of decoy information to the attacker. The user gets a file irrespective of a real user or attacker. But attacker gets a decoy file which he thinks he got an actual file but actually not. A legitimate user gets his/her actual file provided he passes through all three security measures. In this way our data is secured from attackers. Decoy file puts an attacker in to the belief that he/she got an actual file. This is method is ideal for securing important data.

## **3.2 MODULES**

### **3.2.1 ADMIN**

Admin checks overall process and verify whether it is an authorized or unauthorized user.

If it encounters authorized user, then it views user details and uploaded details.

In case it encounters unauthorized user, it throws decoy data.

### **3.2.2 USER**

User can register to the account and login.

He/she can upload details or even view uploaded details.

User has to answer the security questions to make them categorize as authorized or unauthorized user.

He/she can download files and view alerts.

### 3.3 USE CASE DIAGRAM

In the use case diagram, we have basically two actors who are the user and the administrator. The user can register, login, view related files, has to answer the security question, download file. The admin verifies the user, checks the user login time, knows whether it is attacker or not and sends decoy documents.

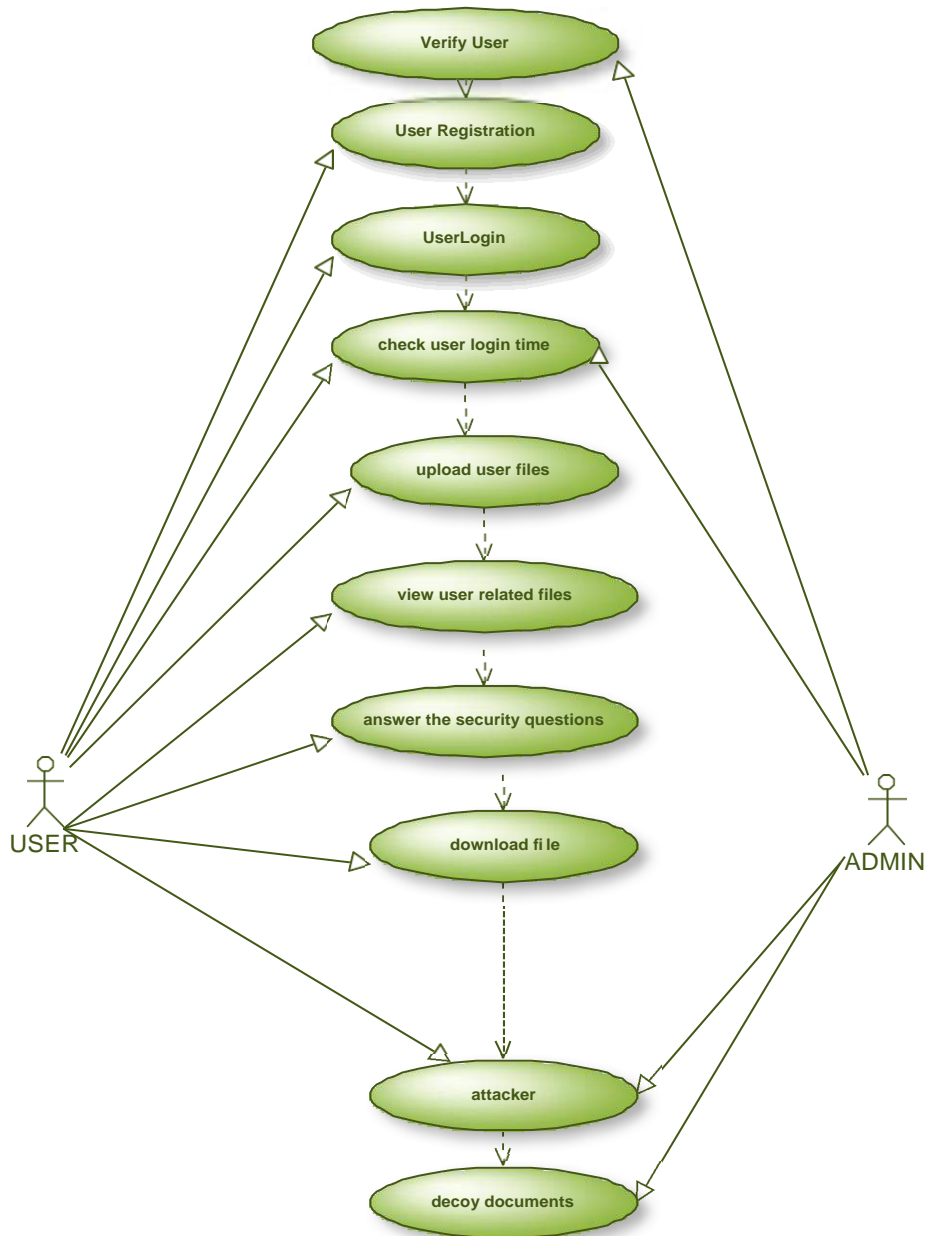


Fig 3.2 Use Case Diagram for Mitigating Data Thefts

### 3.4 ACTIVITY DIAGRAM

It describes about flow of activity states.

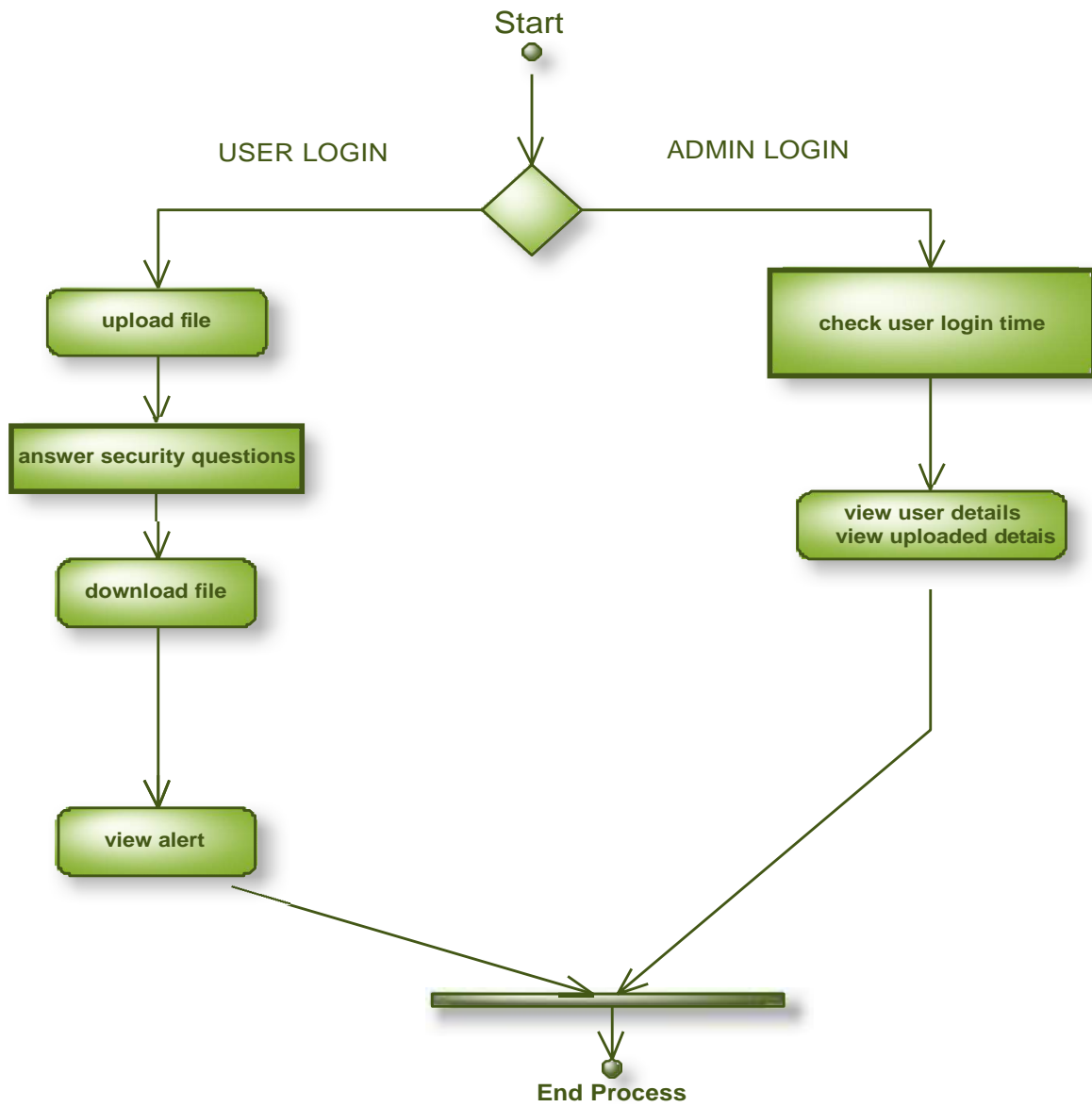


Fig 3.3 Activity Diagram for Mitigating Data Thefts

### 3.5 CLASS DIAGRAM

Class diagram is a collection of classes and objects.

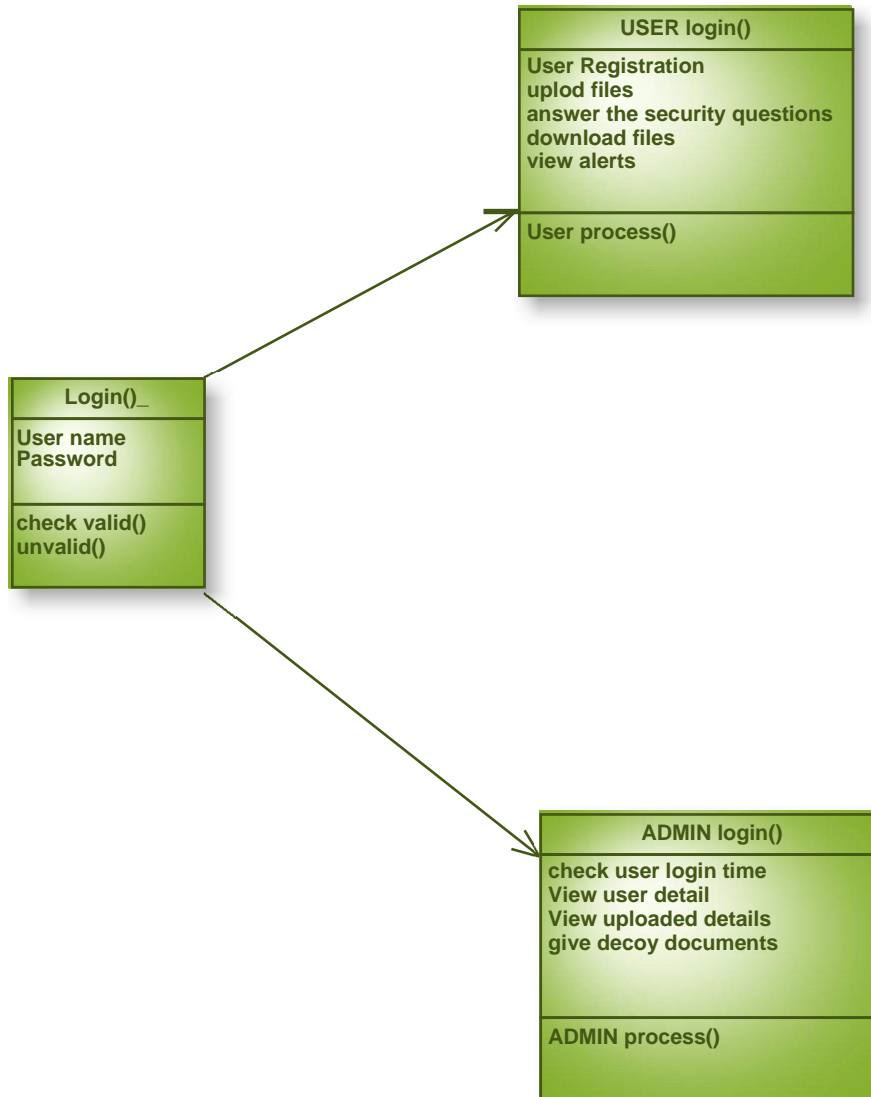


Fig 3.4 Class Diagram for Mitigating Data Thefts

### 3.6 SEQUENCE DIAGRAM

The sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. It shows the processes in a sequential manner between user and admin.

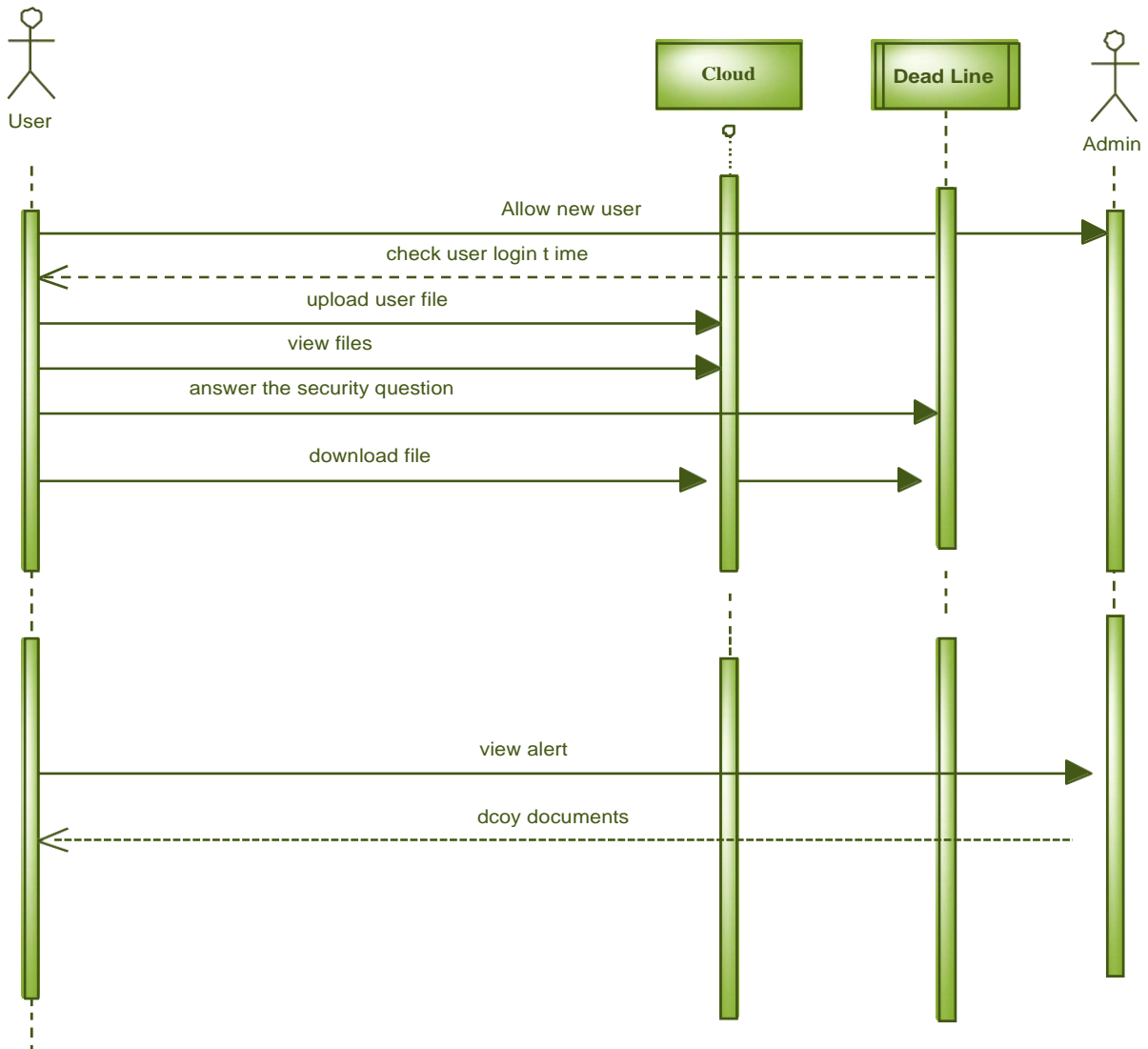


Fig 3.5 Sequence Diagram for Mitigating Data Thefts

# **4. IMPLEMENTATION**

## 4. IMPLEMENTATION

### 4.1 SAMPLE CODE

```

package databaseconnection;
import java.sql.*;
public class databasecon
{
    static Connection con;
    public static Connection getconnection()
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/employe","root","sk");
        }
        catch(Exception e)
        {
            System.out.println("class error");
        }
        return con;
    }
}

<% @ page
import="java.util.date.*,java.util.text.DateFormat.*,java.text.ParseException.*"%>

<% @page
import="com.oreilly.servlet.*,java.sql.*,java.lang.*,databaseconnection.*,java.text.SimpleDate
teFormat,java.util.*" %>

<% @ page import =
"java.util.Date,java.text.SimpleDateFormat,java.text.ParseException"%>

<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

```



```

</head>
<body>
  <%
SimpleDateFormat sdf1 = new SimpleDateFormat("HH:mm:ss ");
Calendar cal = Calendar.getInstance();
    String st=(sdf1.format(cal.getTime()));
String ans=request.getParameter("ans");
    session.setAttribute("ans",ans);
String name=(String)session.getAttribute("un");
String fname=null;
String fname1=null;
try
{
Class.forName("com.mysql.jdbc.Driver");
Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/foxcomputing","root","admin");
PreparedStatement ps=con.prepareStatement("select from1,to1 from signup where
name='"+name+"' ");
ResultSet rs=ps.executeQuery();
while(rs.next())
{
fname=rs.getString("from1");
fname1=rs.getString("to1");
}
}
catch(Exception e)
{
out.println(e.getMessage());
}%>
<%
SimpleDateFormat formater = new SimpleDateFormat("HH:mm:ss");

```

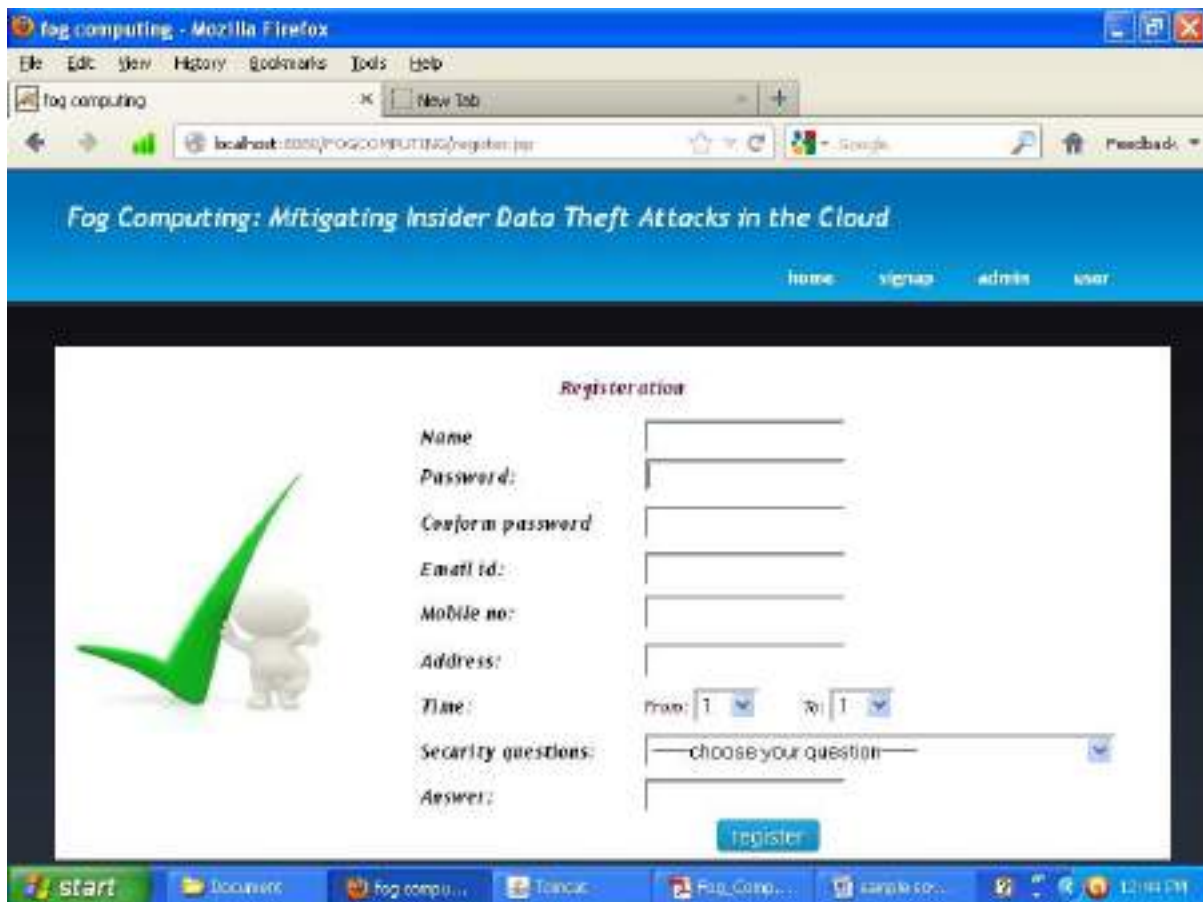
```
Date b=formater.parse(fname);  
Date c=formater.parse(fname1);  
Date a=formater.parse(st);  
if((a.after(b))&&(a.before(c)))  
{  
response.sendRedirect("ans_check.jsp");  
}  
else{  
response.sendRedirect("file_download.jsp");  
}  
%>  
</body>  
</html>
```

## **5. SCREENSHOTS**

## 5. SCREENSHOTS

### 5.1 HOME PAGE

This is the registration page where the user sign's up with his/her credentials.



Screenshot 5.1: Home Page of Mitigating Data Thefts

## 5.2 USER LOGIN PAGE

User Logins through credentials given during registration.



Screenshot 5.2: User Login Page of Mitigating Data Thefts

### 5.3 FILE UPLOAD PAGE

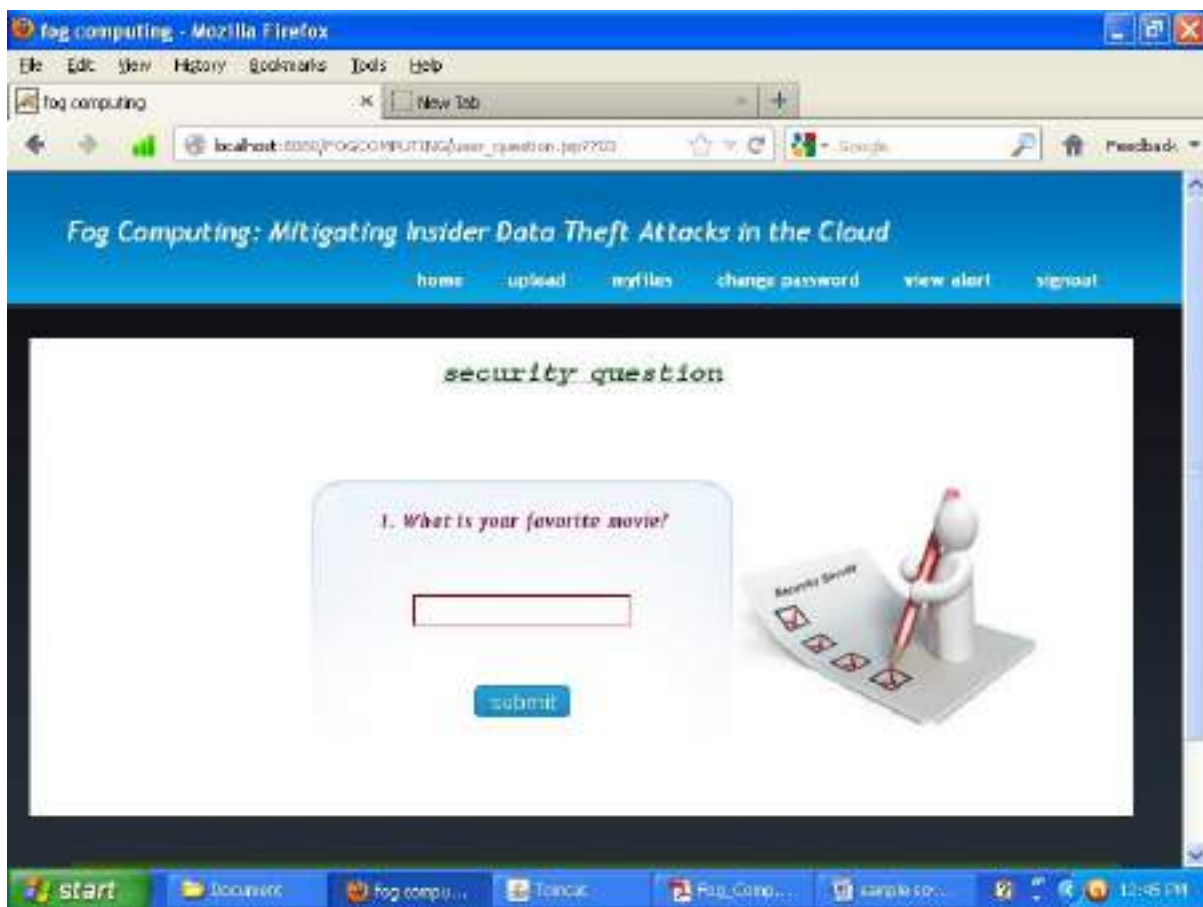
Uploading of data is done here by the user for future access.



Screenshot 5.3: File Upload Page of Mitigating Data Thefts

## 5.4 SECURITY QUESTION PAGE

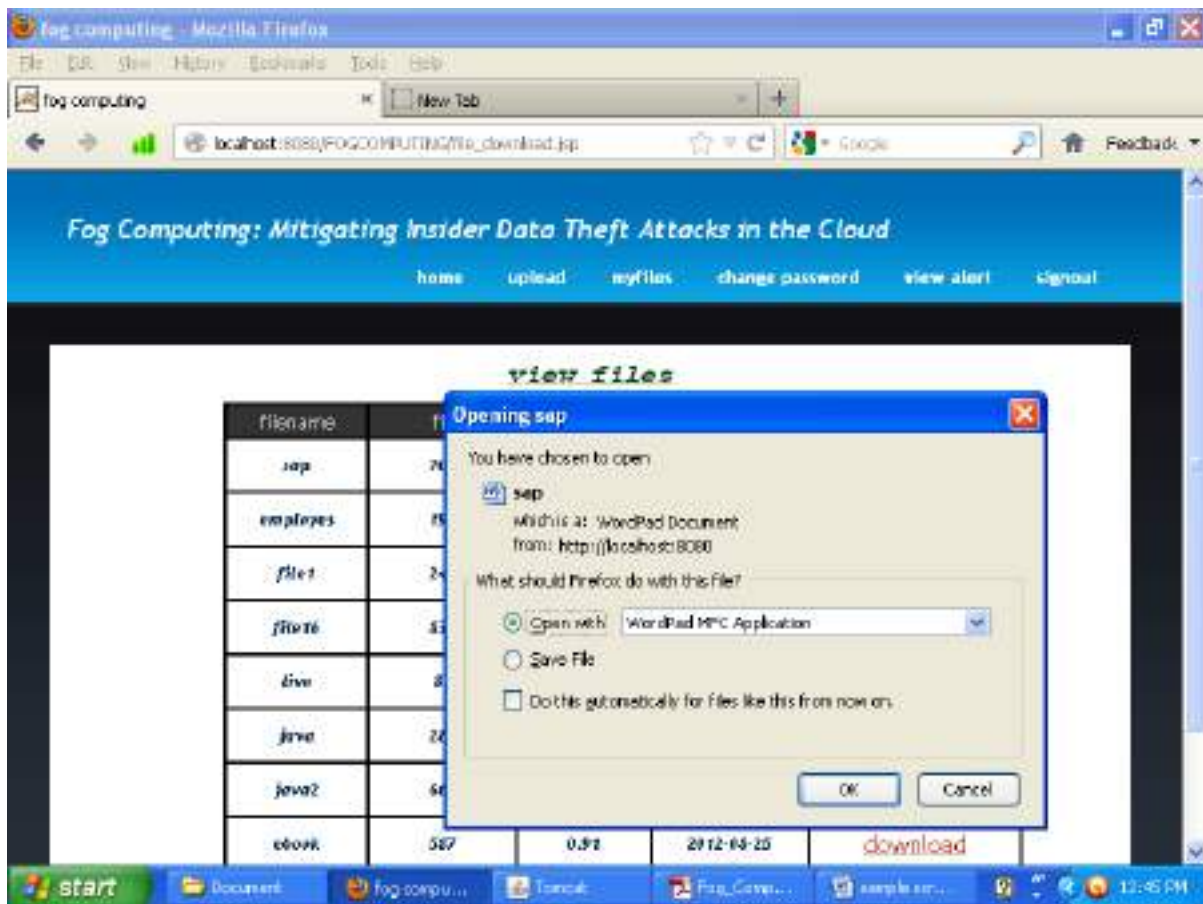
Here the user is asked security question for which he/she is distinguished as real user or hacker.



Screenshot 5.4: Security Question Page of Mitigating Data Thefts

## 5.5 MY FILES PAGE

The user can download the file through this page. Real user gets actual file and attacker gets decoy file.

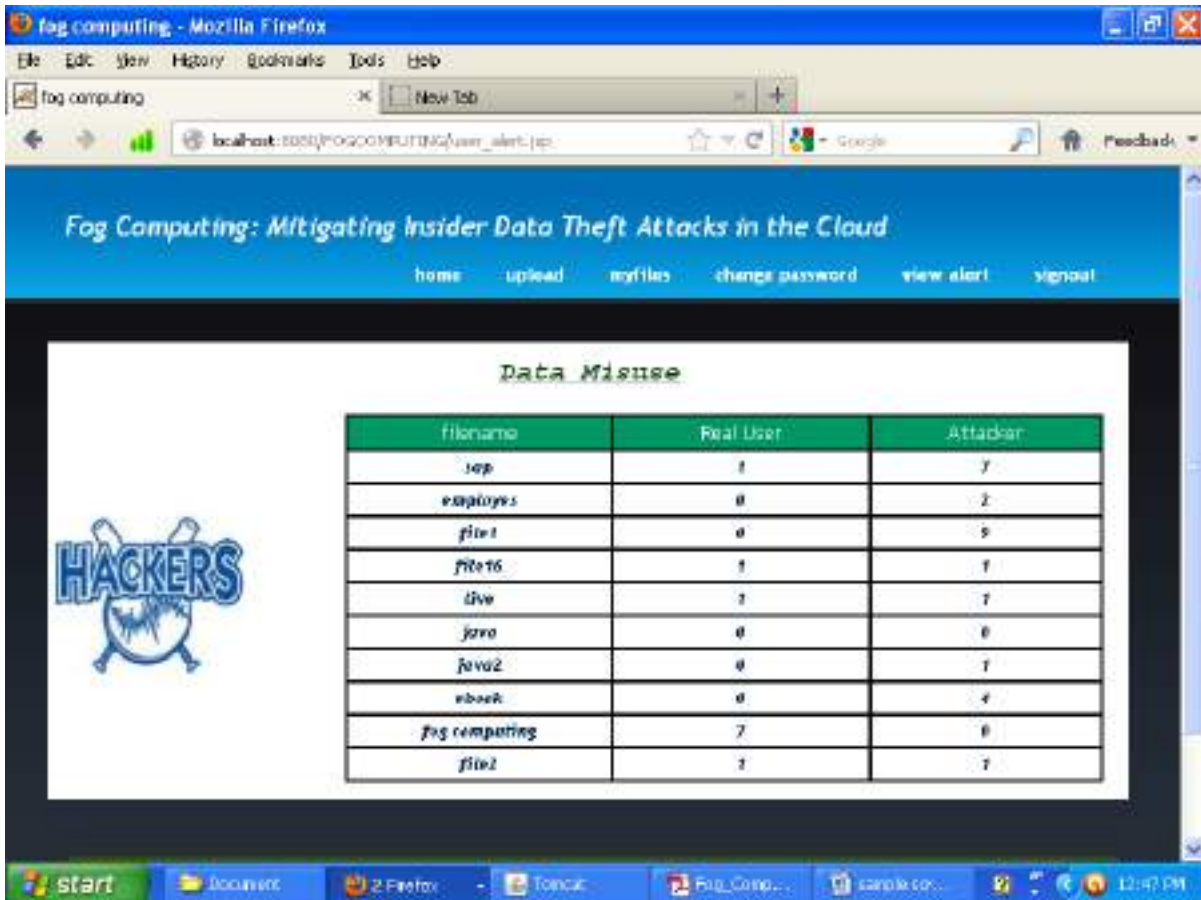


Screenshot 5.5: My Files Page of Mitigating Data Thefts



## 5.6 VIEW ALERT PAGE

In this page, the user or admin can see how many real users accessed your profile and how many attackers tried to access.



The screenshot shows a web browser window titled 'fog computing - Mozilla Firefox'. The address bar shows 'localhost:8080/fogcomputing/view\_alert.jsp'. The page header reads 'Fog Computing: Mitigating Insider Data Theft Attacks in the Cloud' and includes navigation links: home, upload, myfiles, change password, view alert, and signout. The main content area is titled 'Data Misuse' and contains a table with the following data:

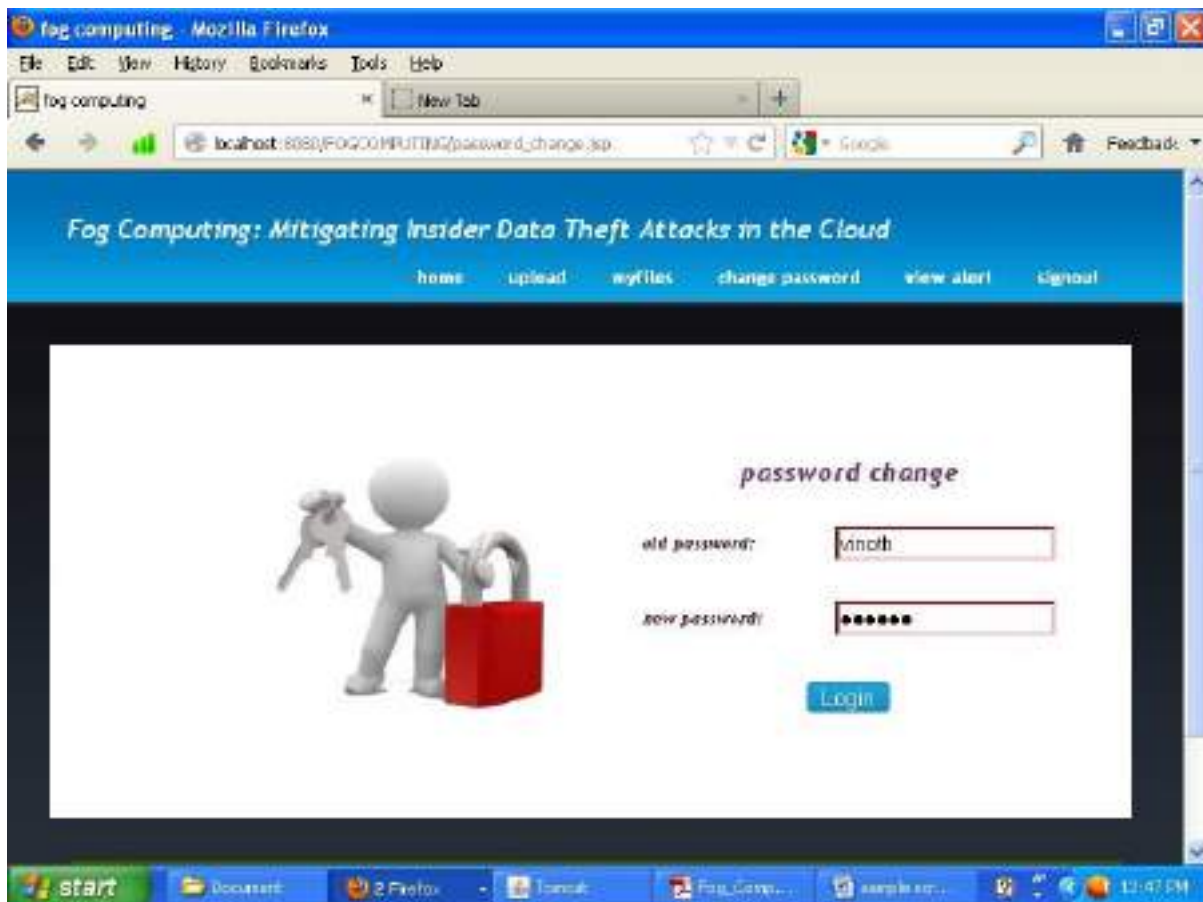
filename	Real User	Attacker
top	1	7
employees	0	2
file1	0	9
file16	1	1
live	1	1
java	0	0
java2	0	1
ebook	0	4
fog computing	7	0
file2	1	1

To the left of the table is a logo with the word 'HACKERS' in a stylized font above a globe icon.

Screenshot 5.6: View Alert Page of Mitigating Data Thefts

## 5.7 PASSWORD CHANGE PAGE

Here, the user can change his/her password if detects some other person using the credentials to access files.



Screenshot 5.7: Password Change Page of Mitigating Data Thefts

## 5.8 USER DETAILS PAGE

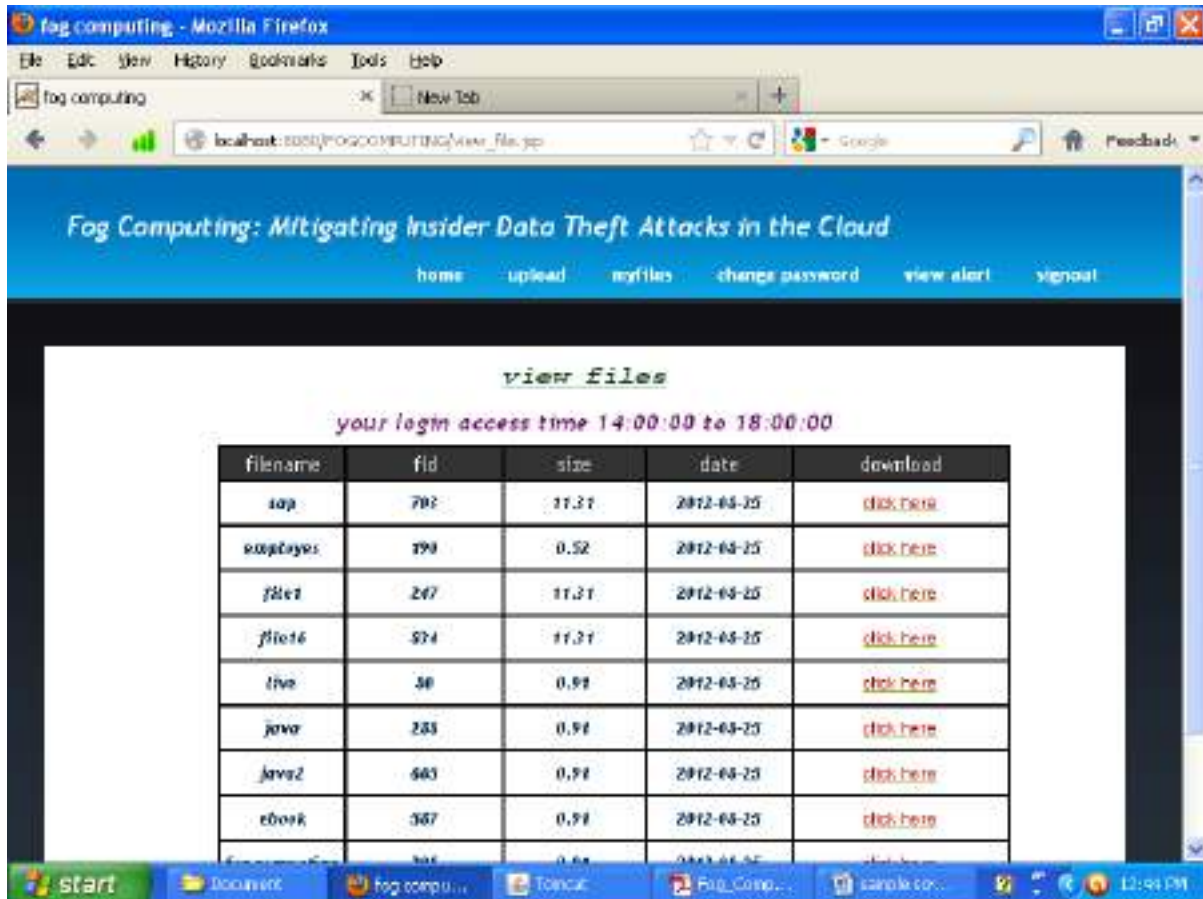
Admin keeps track of all the user details with their credentials.



Screenshot 5.8: User Details Page of Mitigating Data Thefts

## 5.9 VIEW FILES PAGE

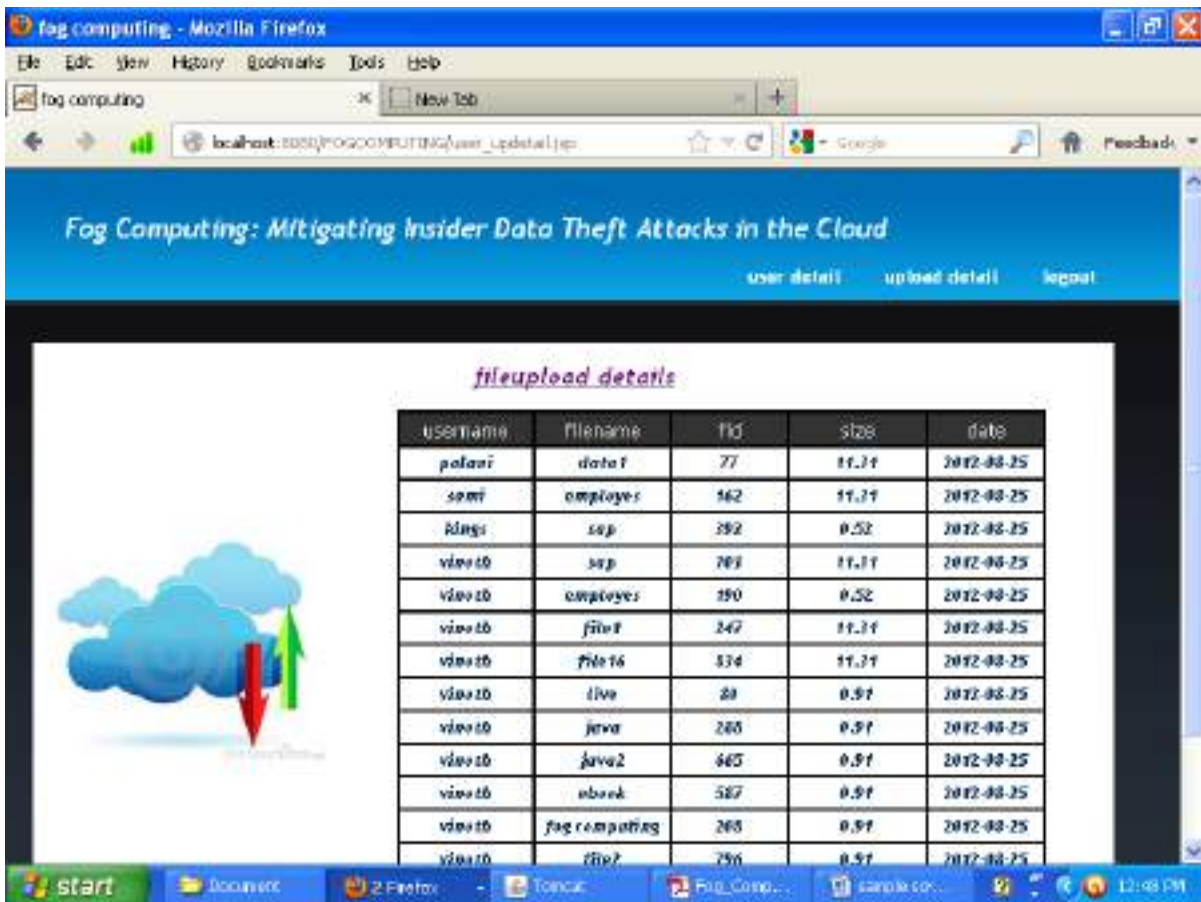
All files are viewed here by the user, whatever he/she uploaded.



Screenshot 5.9: View Files Page of Mitigating Data Thefts

## 5.10 UPLOAD DETAILS PAGE

Admin keeps track of all the upload details and its files



Fog Computing: Mitigating Insider Data Theft Attacks in the Cloud

user detail   upload detail   login

*fileupload details*

username	filename	fid	size	date
palani	data1	77	11.21	2012-08-25
sami	employee	162	11.21	2012-08-25
kingi	sep	392	0.52	2012-08-25
vinoth	sep	763	11.21	2012-08-25
vinoth	employee	350	0.52	2012-08-25
vinoth	file1	247	11.21	2012-08-25
vinoth	file16	334	11.21	2012-08-25
vinoth	live	88	0.91	2012-08-25
vinoth	java	265	0.91	2012-08-25
vinoth	java2	465	0.91	2012-08-25
vinoth	ebook	587	0.91	2012-08-25
vinoth	fog computing	265	0.91	2012-08-25
vinoth	file1	756	0.91	2012-08-25

Screenshot 5.10: Upload Details Page of Mitigating Data Thefts

# **6. TESTING**

## **6. TESTING**

### **6.1 INTRODUCTION TO TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### **6.2 TYPES OF TESTING**

#### **6.2.1 UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

#### **6.2.2 INTEGRATION TESTING**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### 6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.

## 6.3 TEST CASES

### 6.3.1 UPLOADING FILES

Test case ID	Test case name	Purpose	Test Case	Output
1	User uploads file	Use it for future access	The user uploads a particular file	Uploaded Successfully
2	User uploads 2 <sup>nd</sup> file	Use it for future access	The user uploads another file	Uploaded Successfully



**6.3.2 DETECTION**

Test case ID	Test case name	Purpose	Input	Output
1	Detection test 1	To check if the user is real or attacker.	If the user Logins in the wrong mentioned time or gives wrong security answer.	It predicts as attacker and sends decoy file
2	Detection test 2	To check if the user is real or attacker.	If the user logins in the correct mentioned time and gives correct security answer.	It predicts as a real user and sends actual file.

## **7. CONCLUSION**

## **7. CONCLUSION & FUTURE SCOPE**

### **7.1 PROJECT CONCLUSION**

We present a novel approach to securing personal and business data in the Cloud. We propose monitoring data access patterns by profiling user behavior to determine if and when a malicious insider illegitimately accesses someone's documents in a Cloud service. Decoy documents stored in the Cloud alongside the user's real data also serve as sensors to detect illegitimate access. Once unauthorized data access or exposure is suspected, and later verified, with challenge questions for instance, we inundate the malicious insider with bogus information in order to dilute the user's real data. Such preventive attacks that rely on disinformation technology, could provide unprecedented levels of security in the Cloud and in social networks model.

The constraints are met and overcome successfully. The system is designed as like it was decided in the design phase. The project gives good idea on developing a full-fledged application satisfying the user requirements.

The system is very flexible and versatile. Validation checks induced have greatly reduced errors. Provisions have been made to upgrade the software. The application has been tested with live data and has provided a successful result. Hence the software has proved to work efficiently.

### **7.2 FUTURE SCOPE**

In future we can add lot of security measures to prevent data thefts from unknown users. The software can be developed further to include a lot of modules because the proposed system is developed on the view of future. We can still make the application more dynamic in the future. We can connect to other databases by including them.

## **8. BIBLIOGRAPHY**

## **PROJECT LINK**

<https://github.com/yfuru-puck/Major.git>

## 8.BIBLIOGRAPHY

### 8.1 REFERENCES

[1] Cloud Security Alliance, “Top Threat to Cloud Computing V1.0 ,” March 2010.

[Online]. Available: <https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf>

[2] M. Arrington, “In our inbox: Hundreds of confidential Twitter documents,” July

2009. [Online] Available: <http://techcrunch.com/2009/07/14/in-our-inbox-hundreds-ofconfidential-twitter-documents/>

[3] B. M. Bowen and S. Hershkop, “Decoy Document Distributor:

<https://sneakers.cs.columbia.edu/ids/fog/>,” 2009. [Online]. Available:

<http://sneakers.cs.columbia.edu/ids/FOG/>

### 8.2 WEBSITES

<http://java.sun.com>

<http://www.sourceforge.com>

<http://www.networkcomputing.com/>

<http://www.roseindia.com/>

<http://www.java2s.com/>

# Cloud Computing: Mitigating Insider Data Theft Attacks in the Cloud

D. Sandya Rani<sup>1</sup>, P. Sruthi Laya<sup>2</sup>, S. Abhinav Peter<sup>3</sup>, S. Aishwarya<sup>4</sup>, R. Divya Raja Lakshmi<sup>5</sup>

<sup>1</sup>Assistant Professor, CSE, CMR Technical Campus, Hyderabad, India

<sup>2</sup>Student, CSE, CMR Technical Campus, Hyderabad, India

<sup>3</sup>Student, CSE, CMR Technical Campus, Hyderabad, India

<sup>4</sup>Student, CSE, CMR Technical Campus, Hyderabad, India

<sup>5</sup>Student, CSE, CMR Technical Campus, Hyderabad, India

**Abstract:** - Cloud computing promises to significantly change the way we use computers and access and store our personal and business information. With these new computing and communications paradigms arise new data security challenges. Existing data protection mechanisms such as encryption have failed in preventing data theft attacks, especially those perpetrated by an insider to the cloud provider.

We propose a different approach for securing data in the cloud using offensive decoy technology. We monitor data access in the cloud and detect abnormal data access patterns. When unauthorized access is suspected and then verified using challenge questions, we launch a disinformation attack by returning large amounts of decoy information to the attacker. This protects against the misuse of the user's real data. Experiments conducted in a local file setting provide evidence that this approach may provide unprecedented levels of user data security in a Cloud environment.

**Keywords:** Fog Computing, Insider Threat, User Behaviour Profiling, Decoys

## I. INTRODUCTION

Businesses, especially startups, small and medium businesses (SMBs), are increasingly opting for outsourcing data and computation to the Cloud. This obviously supports better operational efficiency, but comes with greater risks, perhaps the most serious of which are data theft attacks.

Data theft attacks are amplified if the attacker is a malicious insider. This is considered as one of the top threats to cloud computing by the Cloud Security Alliance [1]. While most Cloud computing customers are well-aware of this threat, they are left only with trusting the service provider when it comes to protecting their data. The lack of transparency into, let alone control over, the Cloud provider's authentication, authorization, and audit controls only exacerbates this threat. The Twitter incident is one example of a data theft attack from the Cloud. Several Twitter corporate and personal documents were ex-filtrated to technological website TechCrunch [2], [3], and customers' accounts, including the account of U.S. President Barack Obama, were illegally accessed [4], [5].

The attacker used a Twitter administrator's password to gain access to Twitter's corporate documents, hosted on Google's infrastructure as Google Docs. The damage was significant both for Twitter and for its customers.

While this particular attack was launched by an outsider, stealing a customer's admin passwords is much easier if perpetrated by a malicious insider. Rocha and Correia outline how easy passwords may be stolen by a malicious insider of the Cloud service provider [6]. The authors also demonstrated how Cloud customers' private keys might be stolen, and how their confidential data might be extracted from a hard disk. After stealing a customer's password and private key, the malicious insider get access to all customer data, while the customer has no means of detecting this unauthorized access. Much research in Cloud computing security has focused on ways of preventing unauthorized and illegitimate access to data by developing sophisticated access control and encryption mechanisms. However, these mechanisms have not been able to prevent data compromise. Van Dijk and Juels have shown that fully homomorphic encryption, often acclaimed as the solution to such threats, is not a sufficient data protection mechanism when used alone [7].

We propose a completely different approach to securing the cloud using decoy information technology, that we have come to call **Fog computing**. We use this technology to launch **disinformation attacks** against malicious insiders, preventing them from distinguishing the real sensitive customer data from fake worthless data. In this paper, we propose two ways of using Fog computing to prevent attacks such as the Twitter attack, by deploying decoy information within the Cloud by the Cloud service customer and within personal online social networking profiles by individual users.

## II. PROPOSED SYSTEM

Numerous proposals for cloud-based services describe methods to store documents, files, and media in a remote service that may be accessed wherever a user may connect to the Internet. A particularly vexing problem before such services are broadly accepted concerns guarantees for securing a user's data in a manner where that guarantees only the user and no one else can gain access to that data. The problem of providing security of confidential information remains a core security problem that, to

date, has not provided the levels of assurance most people desire. Many proposals have been made to secure remote data in the Cloud using encryption and standard access controls. It is fair to say all of the standard approaches have been demonstrated to fail from time to time for a variety of reasons, including insider attacks, mis-configured services, faulty implementations, buggy code, and the creative construction of effective and sophisticated attacks not envisioned by the implementers of security procedures [8]. Building a trustworthy cloud computing environment is not enough, because accidents continue to happen, and when they do, and information gets lost, there is no way to get it back. One needs to prepare for such accidents. The basic idea is that we can limit the damage of stolen data if we decrease the value of that stolen information to the attacker. We can achieve this through a 'preventive'

**disinformation attack.** We posit that secure Cloud services can be implemented given two additional security features:

- 1) **User Behavior Profiling:** It is expected that access to a user's information in the Cloud will exhibit a normal means of access. User profiling is a well-known technique that can be applied here to model how, when, and how much a user accesses their information in the Cloud. Such 'normal user' behavior can be continuously checked to determine whether abnormal access to a user's information is occurring. This method of behavior-based security is commonly used in fraud detection applications. Such profiles would naturally include volumetric information, how many documents are typically read and how often. These simple user-specific features can serve to detect abnormal Cloud access based partially upon the scale and scope of data transferred [9].
- 2) **Decoys:** Decoy information, such as decoy documents, honey files, honeypots, and various other bogus information can be generated on demand and serve as a means of detecting unauthorized access to information and to 'poison' the thief's ex-filtrated information. Serving decoys will confound and confuse an adversary into believing they have ex-filtrated useful information, when they have not. This technology may be integrated with user behavior profiling technology to secure a user's information in the Cloud. Whenever abnormal access to a cloud service is noticed, decoy information may be returned by the Cloud and delivered in such a way as to appear completely legitimate and normal. The true user, who is the owner of the information, would readily identify when decoy information is being returned by the Cloud, and hence could alter the Cloud's responses through a variety of means, such as challenge questions, to inform the Cloud security system that it has inaccurately detected an unauthorized access. In the case where the access is correctly identified as an unauthorized access, the Cloud security system would deliver unbounded amounts of bogus information to the adversary, thus securing the user's true data from unauthorized disclosure. The decoys, then, serve two purposes:

- (1) validating whether data access is authorized when abnormal information access is detected, and
- (2) confusing the attacker with bogus information.

We posit that the combination of these two security features will provide unprecedented levels of security for the Cloud. No current Cloud security mechanism is available that provides this level of security.

We have applied these concepts to detect illegitimate data access to data stored on a local file system by masqueraders, *i.e.*, attackers who impersonate legitimate users after stealing their credentials. One may consider illegitimate access to Cloud data by a rogue insider as the malicious act of a masquerader. Our experimental results in a local file system setting show that combining both techniques can yield better detection results, and our results suggest that this approach may work in a Cloud environment, as the Cloud is intended to be as transparent to the user as a local file system. In the following we review briefly some of the experimental results achieved by using this approach to detect masquerade activity in a local file setting.

### III. EXPERIMENTS AND DISCUSSION

#### A. Combining User Behavior Profiling and Decoy Technology for Masquerade Detection

1) **User Behavior Profiling:** Legitimate users of a computer system are familiar with the files on that system and where they are located. Any search for specific files is likely to be targeted and limited. A masquerader, however, who gets access to the victim's system illegitimately, is unlikely to be familiar with the structure and contents of the file system. Their search is likely to be widespread and untargeted.

Based on this key assumption, we profiled user search behavior and developed user models trained with a one-class modeling technique, namely one-class support vector machines. The importance of using one-class modeling stems from the ability of building a classifier without having to share data from different users. The privacy of the user and their data is therefore preserved.

We monitor for abnormal search behaviors that exhibit deviations from the user baseline. According to our assumption, such deviations signal a potential masquerade attack. Our previous experiments validated our assumption and demonstrated that we could reliably detect all simulated masquerade attacks using this approach with a very low false positive rate of 1.12% [9].

2) **Decoy Technology:** We placed traps within the file system. The traps are decoy files downloaded from a Fog computing site, an automated service that offers several types of decoy documents such as tax return forms, medical records, credit card statements, e-bay receipts, etc. [10]. The decoy files are downloaded by the legitimate user and placed in highly-conspicuous locations that are not likely to cause any interference with the normal user activities on the system. A masquerader, who is not familiar with the file system and its contents, is likely to access these decoy files, if he or she is in search for sensitive information, such as the bait information



embedded in these decoy files. Therefore, monitoring access to the decoy files should signal masquerade activity on the system. The decoy documents carry a keyed-Hash Message Authentication Code (HMAC), which is hidden in the header section of the document. The HMAC is computed over the file's contents using a key unique to each user. When a decoy document is loaded into memory, we verify whether the document is a decoy document by computing a HMAC based on all the contents of that document. We compare it with HMAC embedded within the document. If the two HMACs match, the document is deemed a decoy and an alert is issued.

The advantages of placing decoys in a file system are three-fold: (1) the detection of masquerade activity (2) the confusion of the attacker and the additional costs incurred to distinguish real from bogus information, and (3) the deterrence effect which, although hard to measure, plays a significant role in preventing masquerade activity by risk-averse attackers.

3) *Combining the Two Techniques*: The correlation of search behavior anomaly detection with trap-based decoy files should provide stronger evidence of malfeasance, and therefore improve a detector's accuracy. We hypothesize that detecting abnormal search operations performed prior to an unsuspecting user opening a decoy file will corroborate the suspicion that the user is indeed impersonating another victim user. This scenario covers the threat model of illegitimate access to Cloud data. Furthermore, an accidental opening of a decoy file by a legitimate user might be recognized as an accident if the search behavior is not deemed abnormal. In other words, detecting abnormal search and decoy traps together may make a very effective masquerade detection system. Combining the two techniques improves detection accuracy.

We use decoys as an oracle for validating the alerts issued by the sensor monitoring the user's file search and access behavior. In our experiments, we did not generate the decoys on demand at the time of detection when the alert was issued. Instead, we made sure that the decoys were conspicuous enough for the attacker to access them if they were indeed trying to steal information by placing them in highly conspicuous directories and by giving them enticing names. With this approach, we were able to improve the accuracy of our detector. Crafting the decoys on demand improves the accuracy of the detector even further. Combining the two techniques, and having the decoy documents act as an oracle for our detector when abnormal user behavior is detected may lower the overall false positive rate of detector.

We trained eighteen classifiers with computer usage data from 18 computer science students collected over a period of 4 days on average. The classifiers were trained using the search behavior anomaly detection described in a prior paper [9]. We also trained another 18 classifiers using a detection approach that combines user behavior profiling with monitoring access to decoy files placed in the local file system, as described above. We tested these classifiers using simulated masquerader data. Figure 1 displays the AUC scores achieved by both

detection approaches by user model<sup>1</sup>. The results show that the models using the combined detection approach achieve equal or better results than the search profiling approach alone.

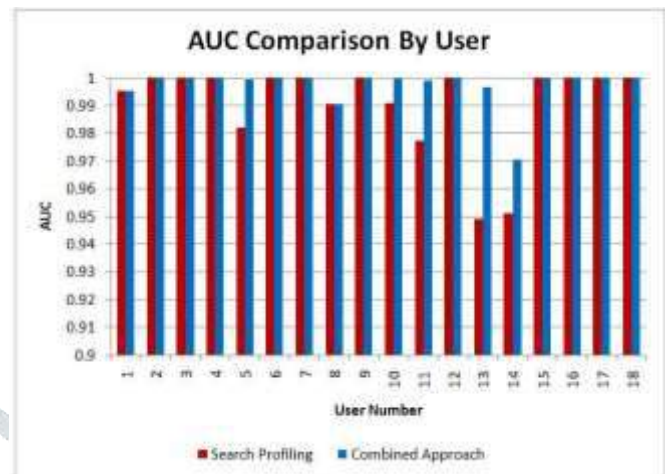


Fig. 1. AUC Comparison by User Model for the Search Profiling and Integrated Approaches

The results of our experiments suggest that user profiles are accurate enough to detect unauthorized Cloud access [9]. When such unauthorized access is detected, one can respond by presenting the user with a challenge question or with a decoy document to validate whether the access was indeed unauthorized, similar to how we used decoys in a local file setting, to validate the alerts issued by the anomaly detector that monitors user file search and access behavior.

#### IV. CONCLUSION

In this position paper, we present a novel approach to securing personal and business data in the Cloud. We propose monitoring data access patterns by profiling user behavior to determine if and when a malicious insider illegitimately accesses someone's documents in a Cloud service. Decoy documents stored in the Cloud alongside the user's real data also serve as sensors to detect illegitimate access. Once unauthorized data access or exposure is suspected, and later verified, with challenge questions for instance, we inundate the malicious insider with bogus information in order to dilute the user's real data. Such preventive attacks that rely on disinformation technology, could provide unprecedented levels of security in the Cloud and in social networks.

#### ACKNOWLEDGMENT

This material is based on work supported by the Defense Advanced Research Projects Agency (DARPA) under the ADAMS (Anomaly Detection at Multiple Scales) Program with grant award number W911NF-11-1-0140 and through the Mission-Resilient Clouds (MRC) program under Contract FA8650-11-C-7190. The views and conclusions contained in

this document is those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA. Professor Stolfo is founder of Allure Security Technology, Inc.

## REFERENCES

- [1] Cloud Security Alliance, “Top Threat to Cloud Computing V1.0,” March 2010. [Online]. Available: <https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf>
- [2] M. Arrington, “In our inbox: Hundreds of confidential twitter documents,” July 2009. [Online]. Available: <http://techcrunch.com/2009/07/14/in-our-inbox-hundreds-of-confidential-twitter-documents/>
- [3] D. Takahashi, “French hacker who leaked Twitter documents to TechCrunch is busted,” March 2010 [Online]. Available: <http://venturebeat.com/2010/03/24/french-hacker-who-leaked-twitter-documents-to-TechCrunch-is-busted/>
- [4] D. Danchev, “ZDNET: French hacker gains access to twitter’s admin panel,” April 2009. [Online]. Available: <http://www.zdnet.com/blog/security/french-hacker-gains-access-to-twiters-admin-panel/3292>
- [5] P. Allen, “Obama’s Twitter password revealed after French hacker arrested for breaking into U.S. president’s account,” March 2010. [Online]. Available: <http://www.dailymail.co.uk/news/article1260488/Barack-Obamas-Twitter-password-revealed-French-hacker-arrested.html>
- [6] F. Rocha and M. Correia, “Lucy in the sky without diamonds: Stealing confidential data in the cloud,” in *Proceedings of the First International Workshop on Dependability of Clouds, Data Centers and Virtual Computing Environments, Hong Kong, ser. DCDV ’11*, June 2011.
- [7] M. Van Dijk and A. Juels, “On the impossibility of cryptography alone for privacy-preserving cloud computing,” in *Proceedings of the 5th USENIX conference on Hot topics in security*, ser. HotSec’10. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–8. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1924931.1924934>
- [8] J. Pepitone, “Dropbox’s password nightmare highlights cloud risks,” June 2011.
- [9] M. Ben-Salem and S. J. Stolfo, “Modeling user search-behavior for masquerade detection,” in *Proceedings of the 14th International Symposium on Recent Advances in Intrusion Detection*. Heidelberg: Springer, September 2011, pp. 1–20.
- [10] B. M. Bowen and S. Hershkop, “Decoy Document Distributor: <http://sneakers.cs.columbia.edu/ids/fog/>,” 2009. [Online]. Available: <http://sneakers.cs.columbia.edu/ids/FOG/>
- [11] M. Ben-Salem and S. J. Stolfo, “Combining a baiting and a user search profiling techniques for masquerade detection,” in *Columbia University Computer Science Department, Technical Report # cucs-018-11*, 2011. [Online]. Available: <https://mice.cs.columbia.edu/getTechreport.php?techreportID=1468>



# Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

## Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

**P Sruthi Laya**

In recognition of the publication of the paper entitled

**Cloud Computing: Mitigating Insider Data Theft Attacks in the Cloud**

Published In JETIR ( www.jetir.org ) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 8 Issue 6 , June-2021 | Date of Publication: 2021-06-12

*Parisa P*

EDITOR

JETIR2106220

*[Signature]*

EDITOR IN CHIEF

Research Paper Weblink <http://www.jetir.org/view?paper=JETIR2106220>

Registration ID : 310533





# Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org **An International Scholarly Indexed Journal**

## Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

**S Abhinav Peter**

In recognition of the publication of the paper entitled

**Cloud Computing: Mitigating Insider Data Theft Attacks in the Cloud**

Published In JETIR ( www.jetir.org ) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 8 Issue 6 , June-2021 | Date of Publication: 2021-06-12

*Parisa P*

EDITOR

JETIR2106220

*[Signature]*

EDITOR IN CHIEF

Research Paper Weblink <http://www.jetir.org/view?paper=JETIR2106220>

Registration ID : 310533







# Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org **An International Scholarly Indexed Journal**

## Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

**S Aishwarya**

In recognition of the publication of the paper entitled

**Cloud Computing: Mitigating Insider Data Theft Attacks in the Cloud**

Published In JETIR ( www.jetir.org ) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 8 Issue 6 , June-2021 | Date of Publication: 2021-06-12

*Parisa P*

EDITOR

JETIR2106220

*[Signature]*

EDITOR IN CHIEF

Research Paper Weblink <http://www.jetir.org/view?paper=JETIR2106220>

Registration ID : 310533





# Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org **An International Scholarly Indexed Journal**

## Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

**R Divya Raja Lakshmi**

In recognition of the publication of the paper entitled

**Cloud Computing: Mitigating Insider Data Theft Attacks in the Cloud**

Published In JETIR ( www.jetir.org ) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 8 Issue 6 , June-2021 | Date of Publication: 2021-06-12

*Parisa P*

EDITOR

JETIR2106220

*[Signature]*

EDITOR IN CHIEF

Research Paper Weblink <http://www.jetir.org/view?paper=JETIR2106220>

Registration ID : 310533





# Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

## Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

**D Sandya Rani**

In recognition of the publication of the paper entitled

**Cloud Computing: Mitigating Insider Data Theft Attacks in the Cloud**

Published In JETIR ( www.jetir.org ) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 8 Issue 6 , June-2021 | Date of Publication: 2021-06-12

*Parisa P*

EDITOR

JETIR2106220

*[Signature]*

EDITOR IN CHIEF

Research Paper Weblink <http://www.jetir.org/view?paper=JETIR2106220>

Registration ID : 310533

