

WEB SECURITY  
BCI3001

TITLE- DATA LEAKAGE DETECTION USING  
GUILTY MODEL

Rakshith Sachdev 18BCI0109

Rohan Allen 18BCI0247

Rahul Balagopalan 18BCI0157

# INTRODUCTION

- ❖ Current statistics from different security association research firms and government organizations recommend that there has been a fast growth of data leak in past 8 years and as present world for the most part relies upon exchange of information i.e. transfer of data from one individual to another individual which is also known as distributary system. The data is sent from the distributor to the client are confidential so the data is distributed only between the distributor and the trusted third parties.
- ❖ The data sent by the distributor must be secured, private, confidential and must not be replicated as the data imparted with the trusted third parties are confidential and profoundly significant. In certain events the data distributed by the distributor are duplicated by different agents who cause an enormous harm to the institute and this process of losing the data is known as data leakage. The data leakage must be detected at an early stage in order to prevent the data form being open source. This project deals with shielding the data from being out sourcing by restricting the agents by using blacklisting so that it cannot be leaked

# ABSTRACT

- ❖ Data leakage is a serious security concern for every organization, the first step to solving this problem is to find the source of the data leakage. This project deals with shielding the data from being outsourced by restricting the agents by using blacklisting so that it cannot be leaked.
- ❖ A data distributor has given sensitive data to a set of supposedly trusted agents (third parties). Some of the data is leaked and found in an unauthorized place (e.g., on the web or somebody's laptop). The distributor must assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means. We propose data allocation strategies (across the agents) that improve the probability of identifying leakages. These methods do not rely on alterations of the released data (e.g., watermarks). In some cases we can also inject “realistic but fake” data records to further improve our chances of detecting leakage and identifying the guilty party

# LITERATURE SURVEY

PAPER TITLE	METHOD USED	LIMITATION	PROPOSED SYSTEM
<b>Fast Detection of Transformed Data Leaks,” in IEEE Transactions on Information Forensics and Security</b>	AlignDLD and Coll inter system  (prototype using virtual box)	Detects inadvertent leaks and not malicious leaks	Our system deals with both kinds of leaks
<b>Data Leakage Detection In cloud using watermarking technique</b>	Hybrid watermarking algorithms	Limited to image data set	Our system deals with any kind of data
<b>Data Leakage Detection in cloud computing environment</b>	Bell-Lapadula Model	Infeasible to extend to web environment where multiple users access data	Our system is feasible for multiple user access
<b>Privacy-Preserving Detection of Sensitive Data Exposure</b>	Fuzzy fingerprint technique	False positive and true positive yield the same fingerprints	Our system works on probability to avoid such cases
<b>Detection Method on the Privacy Leakage for Composite Services</b>	Detection Method on the Privacy Leakage for Composite Services	Infeasible in case of complex combinations	Our system might get cumbersome, but is feasible for complex combinations

# PROBLEM STATEMENT AND OBJECTIVE

- ❖ Throughout course of doing business, once in a while sensitive data must be given over to supposedly trusted third parties. For instance, a medical clinic may give patient records to specialists who will devise new treatments. Similarly, an organization may have partnerships with different organizations that require sharing client data. Another venture may redistribute its data processing, so data must be given to various other organizations. The owner of the data is the distributor and the supposedly trusted third parties are the agents. At that point further data will be given by the distributor to the trusted third party of the enterprise utilizing this application.
- ❖ We here aim to build an application that will monitor if on the off chance any data has been leaked by the agent of the enterprise. Additionally, here we ensure proper authentication among agents/users accessing the system so that data is accessed by only valid users. It likewise helps in discovering Guilt of Agent from the given set of agents which has leaked the data, who should be blacklisted, using Probability Distribution to find the guilt using the guilt model.

# SCOPE

- ❖ Data leakage is a serious security concern for every organization, the first step to solving this problem is to find the source of the data leakage.

The scope of this project is:

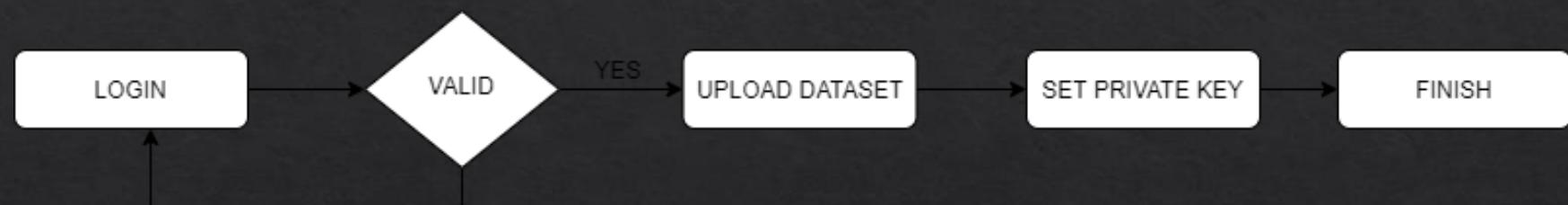
- ❖ To ensure authentication among agents/users accessing the system.
- ❖ To ensure that data is accessed by only valid users.
- ❖ To find the probable list of users who should be blacklisted by finding the probability of guilt using the guilt model.

# MODULES

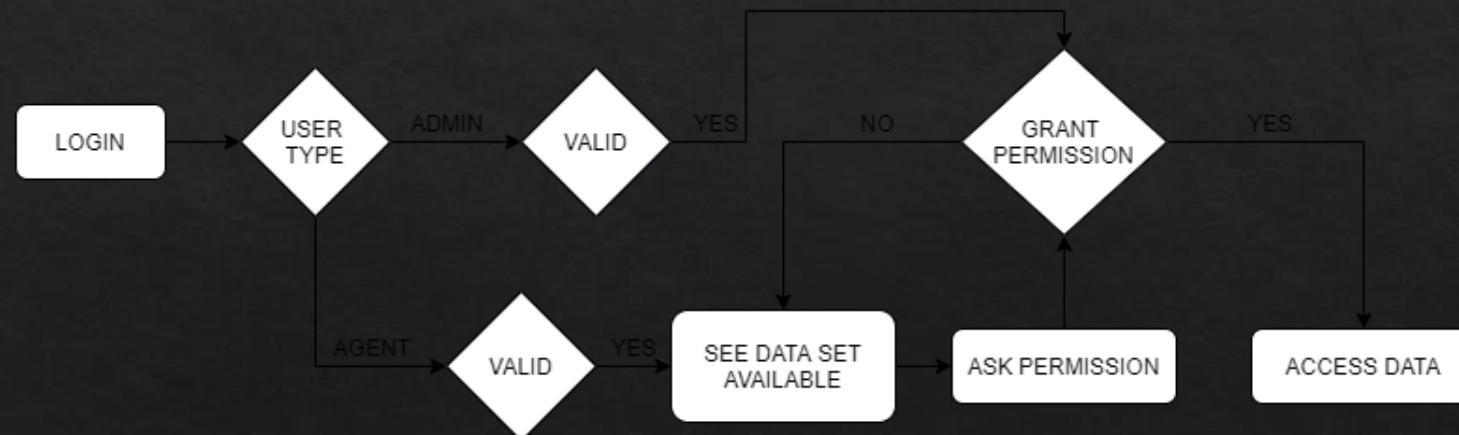
- ❖ 1. Admin Data Control: This module allows the admin to upload dataset to the database of the system (which can be seen by all users but cannot be accessed without permission) or share any data set to a particular user in private.
- ❖ 2. User Data File Access: This module allows users to send a request to the admin for a key in order to access the file available in the database of the system. It is only when the proper key is received, the user can access the data file.
- ❖ 3. Probability Of Guilt: This module analyses which user has the leaked file and sort the list of the probable leakers. Then using the guilt algorithm, the probability calculation is done keeping in mind a cookie jar analogy i.e if we catch Freddie with a single cookie, he can argue that a friend gave him the cookie. But if we catch Freddie with 5 cookies, it will be much harder for him to argue that his hands were not in the cookie jar. If the distributor sees “enough evidence” that an agent leaked data, he may stop doing business with him, or may initiate legal proceedings.
- ❖ 4. Managing the Users: In this module the admin can make changes to the authority of the users. In other words, he can black list the “known bad” by using the probability of the leaker calculated using the guilt model in order to ensure security of the system.

# MODULES DIAGRAM

## 1. Data control

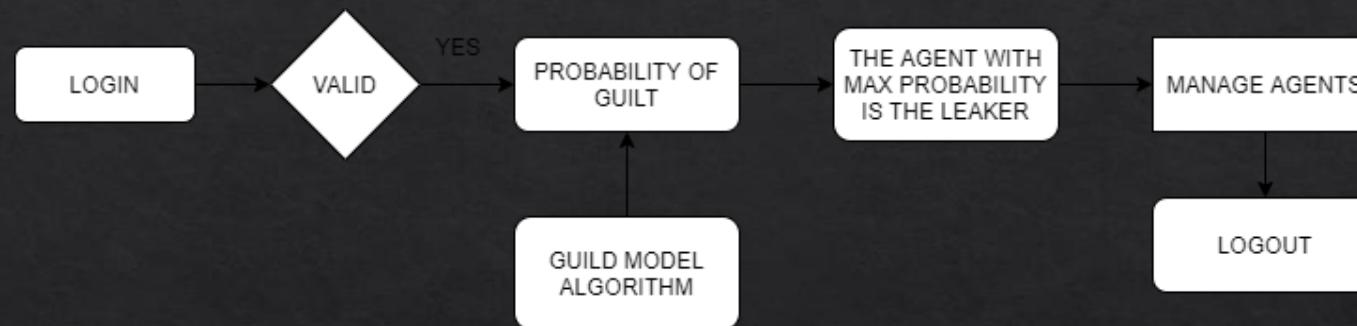


## 2. File access

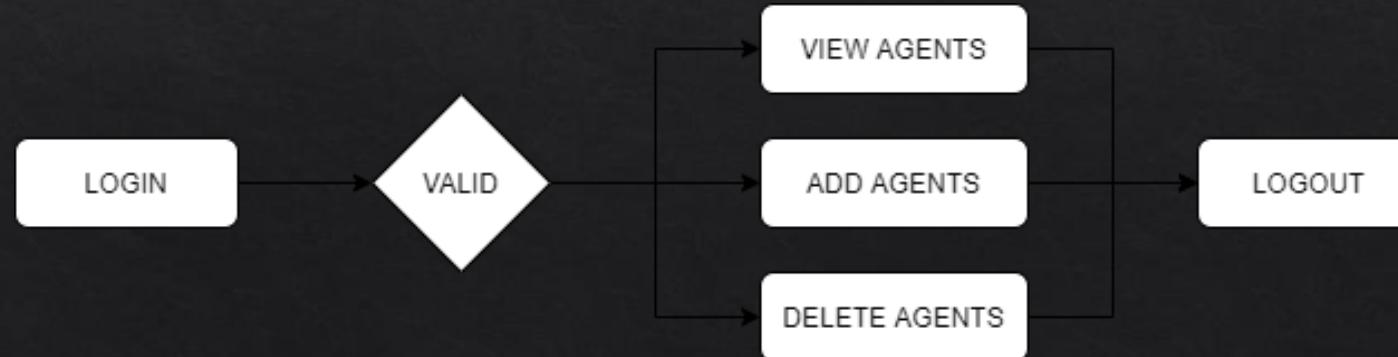


# MODULES DIAGRAM

## 3. Probability of guilt



## 4. Managing agents



# METHODOLOGY

- ❖ In this project we use unobtrusive techniques for detecting leakage of a set of objects or records. The model developed is used for assessing the “guilt” of agents. Algorithms are also provided for distributing objects to agents, in a way that improves the chances of identifying a leaker. Probability Of Guilt: This module analyses which user has the leaked file and sort the list of the probable leakers. Then using the guilt algorithm, the probability calculation is done keeping in mind a cookie jar analogy i.e if we catch Freddie with a single cookie, he can argue that a friend gave him the cookie. But if we catch Freddie with 5 cookies, it will be much harder for him to argue that his hands were not in the cookie jar. If the distributor sees “enough evidence” that an agent leaked data, he may stop doing business with him, or may initiate legal proceedings.

# METHODOLOGY

- ❖ When the distributor allocates data upon request from agents, there may be chance of asking same tuples by more than one agent. Here come overlap between more than one agent, while giving same tuple to more than one agent. When the data is released, distributor must able to assess who is guilt agent. Thus we are arrive solution for this overlap minimization

# METHODOLOGY

❖ Consider that sets T, R's and S are as follows:  $T = \{t_1, t_2, t_3\}$ ,  $R_1 = \{t_1, t_2\}$ ,  $R_2 = \{t_1, t_3\}$ ,  $S = \{t_1, t_2, t_3\}$ . For this situation, every one of the three of the distributor's objects have been leaked and show up in S. Consider how the target may have gotten object  $t_1$ , which was given to both agents. From Assumption 2, the target either guessed  $t_1$  or one of  $U_1$  or  $U_2$  leaked it. Knowing that the probability of the former event is p, so assuming that probability that each of the two agents leaked  $t_1$  is the same cases formed are:

- the target guessed  $t_1$  with probability p;
- agent  $U_1$  leaked  $t_1$  to S with probability  $(1 - p)/2$
- agent  $U_2$  leaked  $t_1$  to S with probability  $(1 - p)/2$

# METHODOLOGY

❖ Similarly, it is found that agent  $U_1$  leaked  $t_2$  to  $S$  with probability  $1 - p$  since he is the only agent that has this data object. Given these values, the probability that agent  $U_1$  is not guilty can be computed, namely that  $U_1$  did not leak either object:

$$\diamond \quad \Pr\{G'_1 | S\} = (1 - (1 - p)/2) \times (1 - (1 - p)) \quad (1)$$

❖ Hence, the probability that  $U_1$  is guilty is:

$$\diamond \quad \Pr\{G_1 | S\} = 1 - \Pr\{G'_1\} \quad (2)$$

❖ Consider the set of agents  $V_t = \{U_i | t \in R_i\}$  that have  $t$  in their data sets, now generalizing

❖(1) and (2) :

$$\diamond \quad \Pr\{U_i \text{ leaked } t \text{ to } S\} = \{ 1-p / |V_t| , \text{ if } U_i \in V_t \text{ and } 0, \text{ otherwise } \} \quad (3)$$

❖ Given that agent  $U_i$  is guilty if he leaks at least one value to  $S$ , with Assumption 1 and Equation 3 the probability  $\Pr\{G_i | S\}$  is computed, that agent  $U_i$  is guilty:

$$\diamond \quad \Pr\{G_i | S\} = 1 - \pi_{t \in S \cap R_i} (1 - (1 - p) / |Vt| )$$

# Module 1: Data control by the admin

## Management of all the dataset present on the database

The screenshot shows a web browser window titled "Data Leakage Detection" with the URL "localhost/data-leakage-detection/admin/m\_arti.php". The page has a dark theme with a top navigation bar containing "Home", "Upload Article", "View File", "Leak User", and "SendKey". On the right, it displays a welcome message "Welcome: admin" and a "Logout" link. The main content area is titled "Manage Article" and contains a table with the following data:

Article Subject	Article Key	File Name	Upload Date	Action
data1	key1	T1.txt	0000-00-00	<a href="#">data1</a>
data2	key2	T2.txt	0000-00-00	<a href="#">data2</a>
data3	key3	T3.txt	0000-00-00	<a href="#">data3</a>
data4	key4	T4.txt	0000-00-00	<a href="#">data4</a>
data5	key5	T5.txt	0000-00-00	<a href="#">data5</a>
data6	key6	T6.txt	0000-00-00	<a href="#">data6</a>
Surveillance Bot	3241	Surveillance Bot.png	0000-00-00	<a href="#">Surveillance Bot</a>
data7	rakshith	T7.txt	0000-00-00	<a href="#">data7</a>

# Uploading new dataset to the database

The screenshot shows a web browser window titled "Data Leakage Detection" with the URL "localhost/data-leakage-detection/admin/upload.php". The browser has tabs for "Home", "Upload Article" (which is highlighted in red), "View File", "Leak User", and "SendKey". The main content area is titled "Upload Article" and contains the following fields:

- Subject: data10
- Key: 1234
- File: Choose File (No file chosen)
- Upload File button

To the right of the form, under "Welcome: admin", there is a list of user names:

- Logout
- UserName: ROHAN ALLEN
- UserName: agent1
- UserName: agent2
- UserName: agent3
- UserName: agent4
- UserName: agent5
- UserName: agent6
- UserName: user1.0

# Sending the keys to as per requirement and trust between the admin and the agent

The screenshot shows a web browser window titled "Data Leakage Detection" with the URL "localhost/data-leakage-detection/admin/sendkey.php". The page has a dark theme with red text for certain labels.

Header navigation bar:

- Home
- Upload Article
- View File
- LeakFile
- SendKey

Main content area:

**Send Key**

UserRequest:

ID	UserName	Filename
34	agent1	data1
36	agent1	data3
37	agent1	data3
43	rohan18	data6

Welcome: admin

Logout

Send Key:

Send2:	rohan18
FileName:	data6
Key:	key6

Send clear

**View File And Key:**

FileName	key
data1	key1
data2	key2
data3	key3
data4	key4
data5	key5

# Module 2: Accessing of files by the user

## Asking for key for the required dataset to the admin

The screenshot shows a web browser window titled "Data Leakage Detection" with the URL "localhost/data-leakage-detection/user/view%20file.php". The page has a dark theme with red and white text. At the top, there is a navigation bar with links for "Home", "View msg", "View Articles" (which is highlighted in orange), and "View Key". On the right side, it displays a welcome message "Welcome: rohan18" and a "Logout" link. The main content area is titled "View Articles" and contains a table with five rows of data. Each row represents an article with columns for Article Name, Date, Detail, View, and Ask KEY. The "View" and "Ask KEY" columns are green, while the others are white. The "Ask KEY" column for each row contains a red link labeled "Click To ask". The bottom of the page shows a footer with the URL "localhost/data-leakage-detection/user/key.php?id=data1" and a link to "T5.txt".

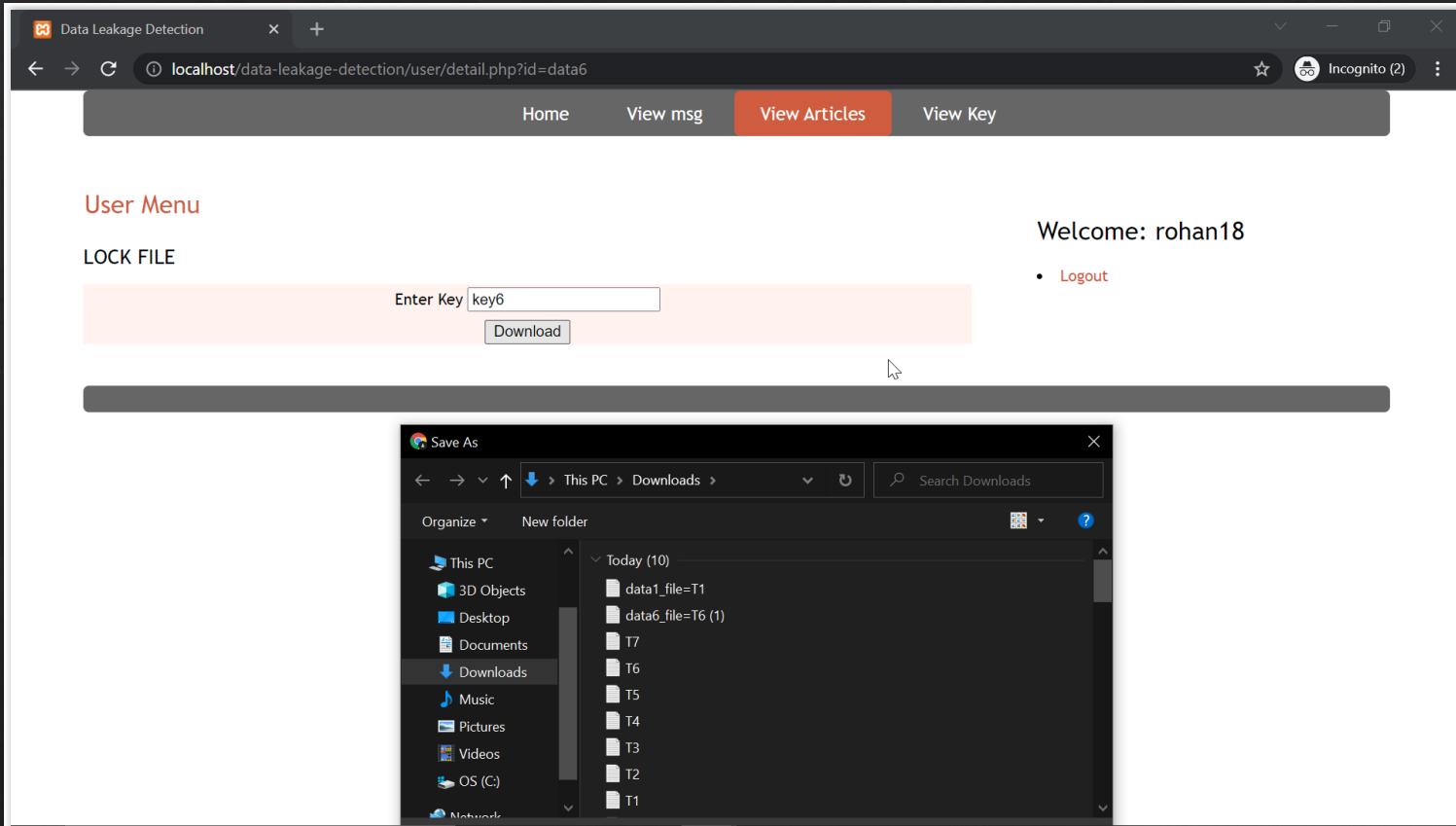
Article Name	Date	Detail	View	Ask KEY
data1	0000-00-00	T1.txt	Download: data1	<a href="#">Click To ask</a>
data2	0000-00-00	T2.txt	Download: data2	<a href="#">Click To ask</a>
data3	0000-00-00	T3.txt	Download: data3	<a href="#">Click To ask</a>
data4	0000-00-00	T4.txt	Download: data4	<a href="#">Click To ask</a>
	0000-	T5.txt	Download:	<a href="#">Click</a>

# Accessing the key send by the admin

The screenshot shows a web browser window titled "Data Leakage Detection". The address bar displays "localhost/data-leakage-detection/user/viewkey.php". The page has a dark theme with a navigation bar at the top containing "Home", "View msg", "View Article", and a red "View key" button. To the right of the navigation bar, it says "Welcome: rohan18" and includes a "Logout" link. The main content area is titled "View Keys" and contains a table with four rows of data. The table has three columns: "KeySender", "filename", and "Keys". The data is as follows:

KeySender	filename	Keys
admin	Surveillance Bot	3241
admin	data7	rakshith
admin	data6	key6
admin	data6	

# Downloading the dataset



## Module 3: Finding the probability of guilt of each agent when a particular dataset has been found with unauthorized entity

- ❖ Let all the agents have the following datasets:
- ❖ Agent1 = {t2, t4, t6, t8, t10, t12}
- ❖ Agent2 = {t6, t7, t8, t9, t10}
- ❖ Agent3 = {t1, t7, t9, t4, t5}
- ❖ Agent4 = {t1, t3, t5, t7, t9, t11}
- ❖ And the leaked dataset be S = {t1, t6, t9, t10}

# Then the guilt probability would be

The screenshot shows a web application titled "Data Leakage Detection" running on a local host. The URL in the address bar is `localhost/data-leakage-detection/admin/leakfile.php`. The page has a dark theme with a navigation bar at the top containing links for "Home", "Upload Article", "View File", "Leak User" (which is highlighted in red), and "SendKey". On the right side, there is a welcome message "Welcome: admin" and a "Logout" link. The main content area is titled "Leak User" and contains a table with the following data:

ID	User	Probability	Send msg
agent1	agent1	0.64	<a href="#">Click</a>
agent2	agent2	0.736	<a href="#">Click</a>
agent3	agent3	0.56	<a href="#">Click</a>
agent4	agent4	0	<a href="#">Click</a>
agent6	agent6	0	<a href="#">Click</a>
rohan18	ROHAN ALLEN	0	<a href="#">Click</a>
user1	name1	0	<a href="#">Click</a>

Below the table is a button labeled "Find Probability".

# Module 4: Managing of user which allows admin to blacklist any agent who is not trust worthy

Manage User

User Name	UserID	Password	EmailID	Delete
ROHAN ALLEN	rohan18	1234	rohan.vulnerable@gmail.com	ROHAN ALLEN
agent1	agent1	agent1	agent1@gmail.com	agent1
agent2	agent2	agent2	agent2@gmail.com	agent2
agent3	agent3	agent3	agent3@gmail.com	agent3
agent4	agent4	agent4	agent4@gmail.com	agent4
agent5	agent5	agent5	agent5@gmail.com	agent5
agent6	agent6	agent6	agent6@gmail.com	agent6
user1.0	user1.0	user1.0	user1.0@gmail.com	user1.0

Welcome: admin

- Logout

localhost/data-leakage-detection/admin/d\_user.php?id=ROHAN ALLEN

# CONCLUSION

- ◆ We here are successful to build an application that will monitor if any data has been leaked by the agent of the enterprise. It likewise helps in discovering possible Guilt of Agent from the given set of agents which has leaked the data using Probability Distribution.

## FUTURE WORK

- ❖ In future, along with finding the probability of guilt model, concept of a fake agent /data set can be implemented by adding fake data into the sensitive data set ,which will act as a watermark without changing the data itself .This technique will further ease the process of detecting leakage or a leaker .

# REFERENCES

- ❖ [1] X. Shu, et al., “Fast Detection of Transformed Data Leaks,” in IEEE Transactions on Information Forensics and Security, vol. 11, no. 3, pp. 528-542, 2016.
- ❖ [2] X. Shu, et al., “Privacy-preserving detection of sensitive data exposure,” IEEE Trans. Inf. Forensics Security, vol. 10, no. 5, pp. 1092-1103, 2015.
- ❖ [3] H. A. Kholidy, et al., “DDSGA: A data-driven semi-global alignment approach for detecting masquerade attacks,” IEEE Trans. Dependable Secure Comput., vol. 12, no. 2, pp. 164-178, 2015.
- ❖ [4] Panagiotis Papadimitriou, Student Member, IEEE, and Hector Garcia-Molina, Member, IEEE “Data Leakage Detection” IEEE Transactions on Knowledge and Data Engineering, Vol. 23, NO. 1, JANUARY 2011
- ❖ [5] Amir Harel, Asaf Shabtai, LiorRokach, and Yuval Elovici “M-Score: A Misuseability Weight Measure” IEEE Transactions ON Dependable And Secure Computing, Vol.9, NO. 3, MAY/JUNE 2012