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
Fast Detection of Transformed Data Leaks," in IEEE Transactions on Information Forensics and Security	AlignDLD and Coll inter system (prototype using virtual box	Detects inadvertent leaks and not malicious leaks	Our system deals with both kinds of leaks
Data Leakage Detection In cloud using watermarking technique	Hybrid watermarking algorithms	Limited to image data set	Our system deals with any kind of data
Data Leakage Detection in cloud computing environment	Bell-Lapadula Model	Infeasible to extend to web environment where multiple users access data	Our system is feasible for multiple user access
Privacy-Preserving Detection of Sensitive Data Exposure	Fuzzy fingerprint technique	False positive and true positive yield the same fingerprints	Our system works on probability to avoid such cases
Detection Method on the Privacy Leakage for Composite Services	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:

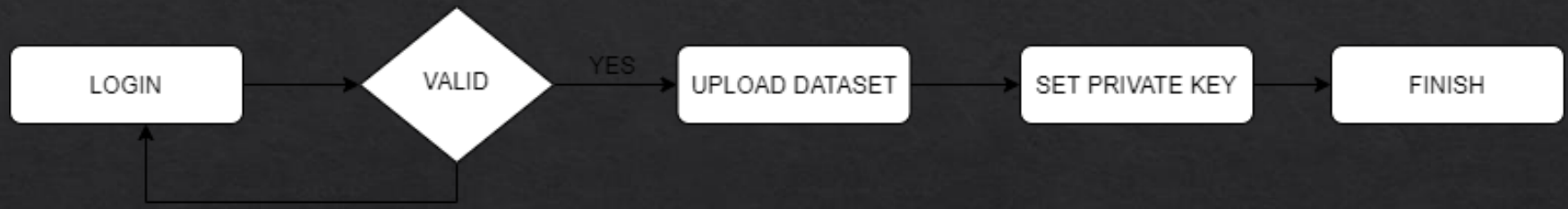
- To ensure authentication among agents/users accessing the system.
- To ensure that data is accessed by only valid users.
- To find the probable list of users who should be blacklisted by finding the probability of guilt using the guilt model.

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

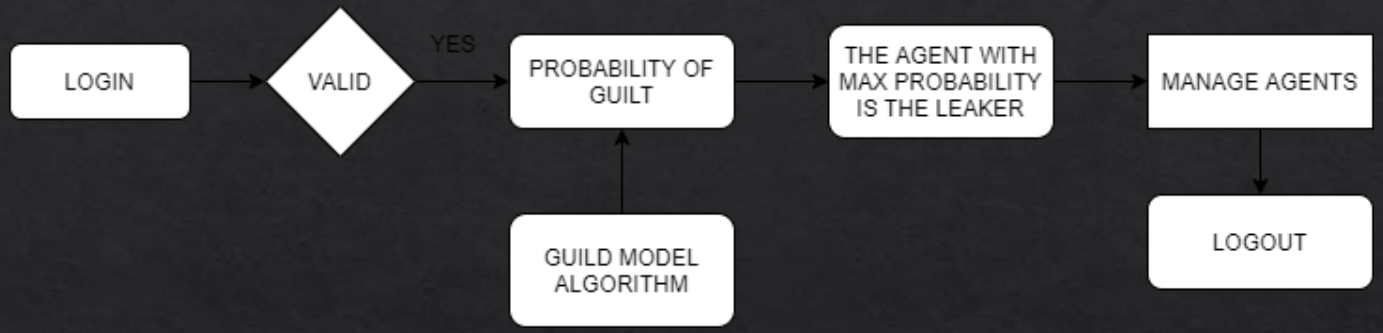
1. Data control



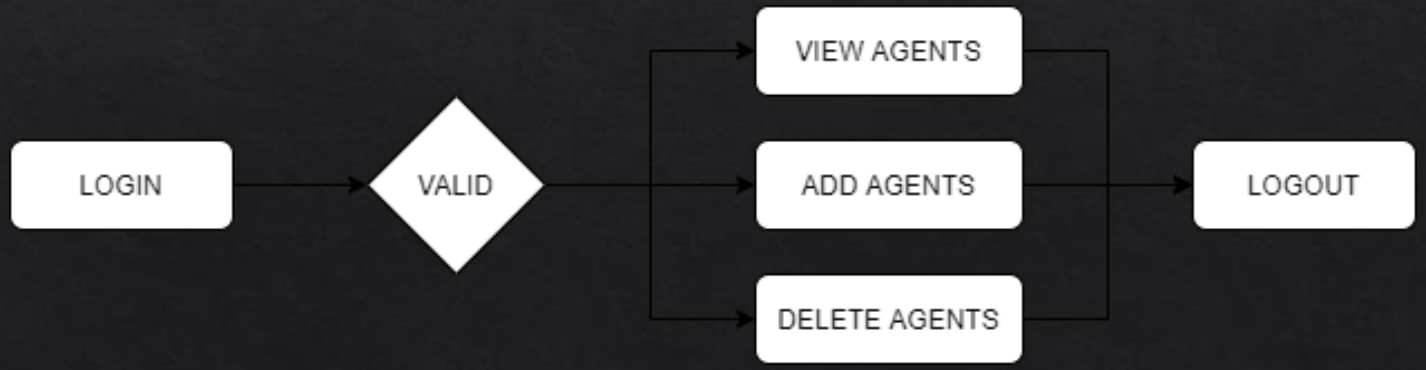
2. File access

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 that one agent. Here come overlap between more that 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 = {t1, t2, t3}, R1 = {t1, t2}, R2 = {t1, t3}, S = {t1, t2, t3}. 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:

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

- Hence, the probability that U_1 is guilty is:

  - $Pr\{G1 | S\} = 1 - Pr\{G'_1\}$ (2)

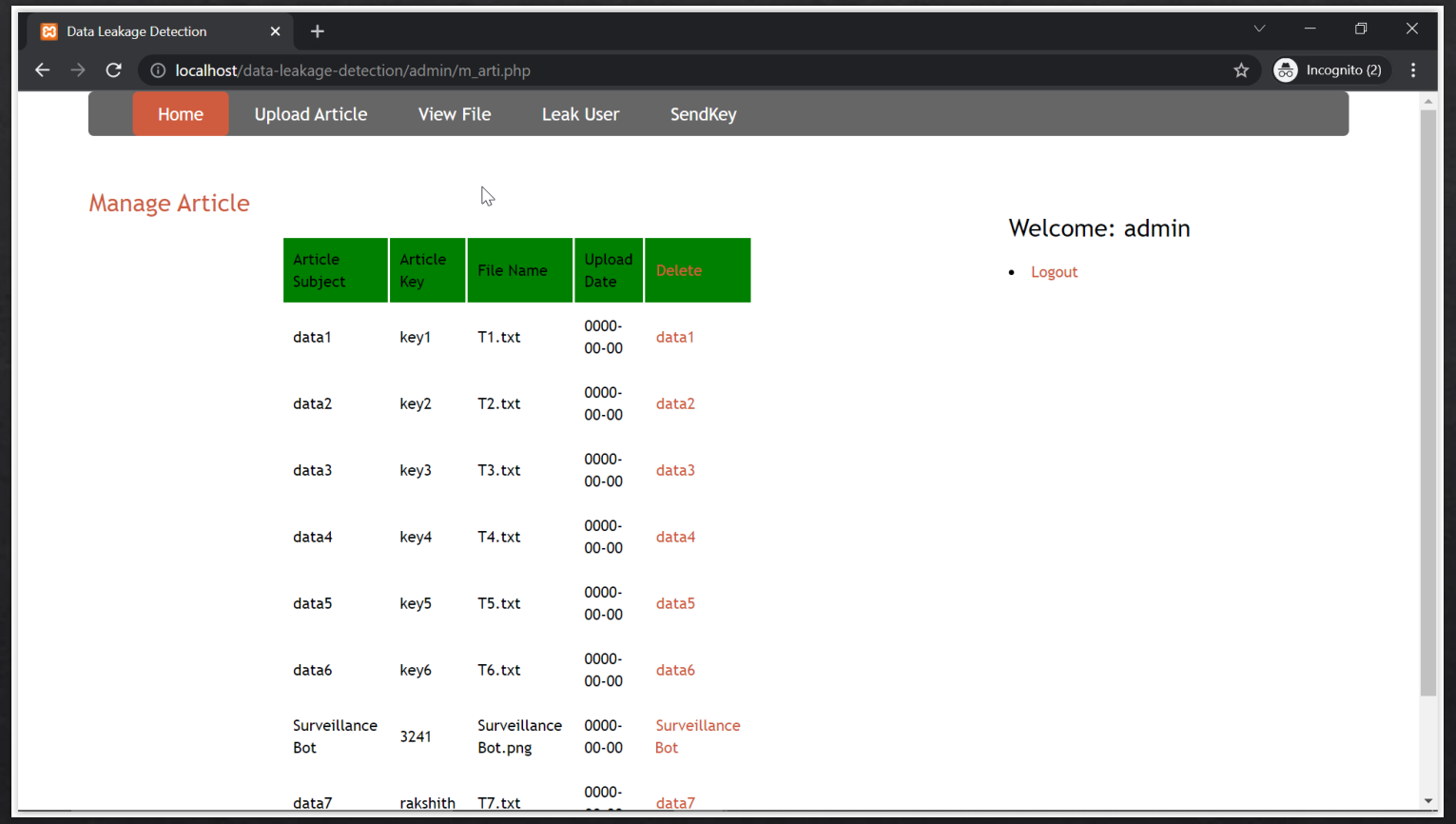
- Consider the set of agents $V_t = \{U_i | t \in R_i\}$ that have t in their data sets, now generalizing
- (1) and (2) :

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

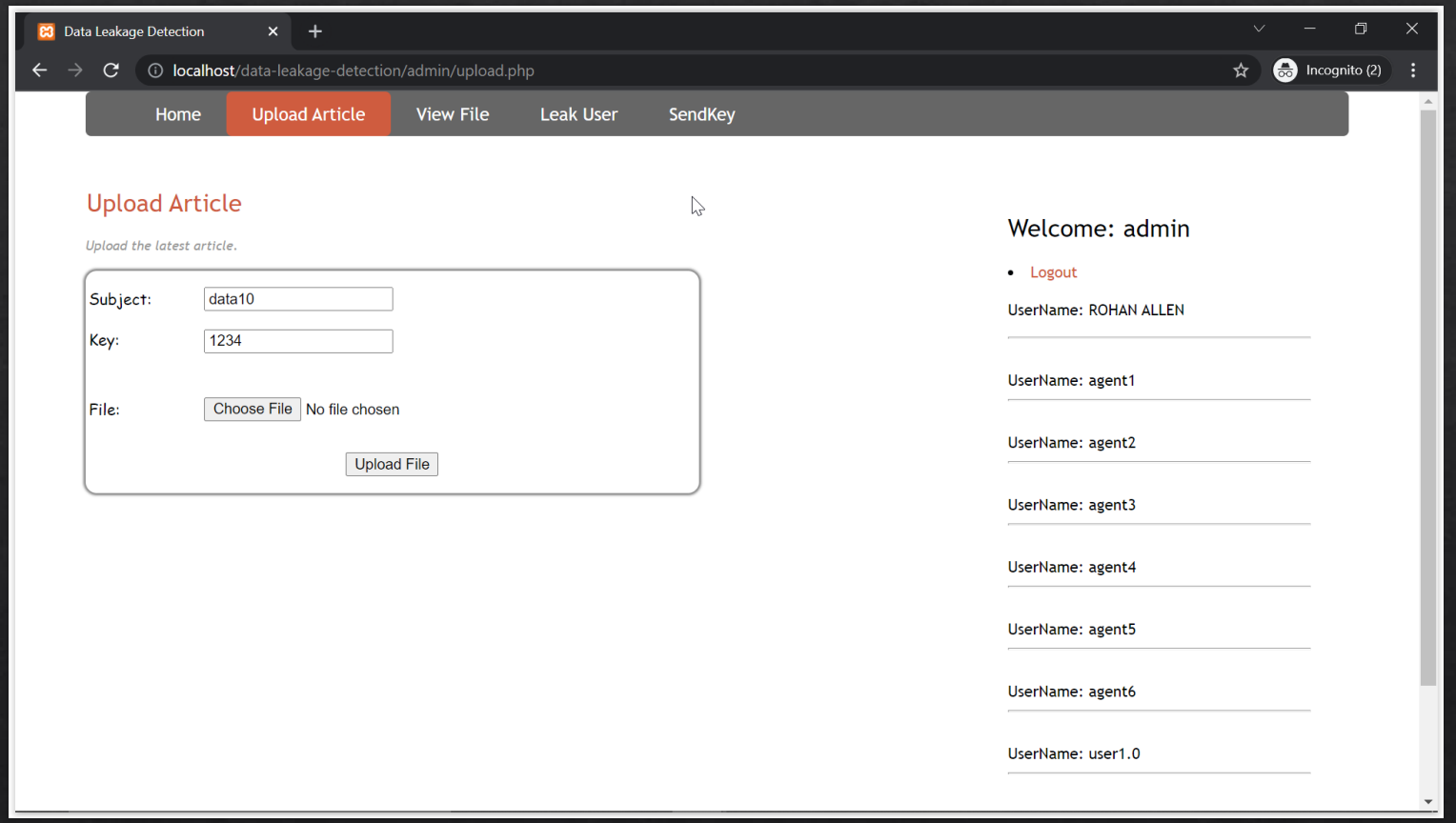
- Given that agent U_i is guilty if he leaks at least one value to S, with Assumption 1 and Equation 3 the probability $Pr\{G_i | S\}$ is computed, that agent U_i is guilty:

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

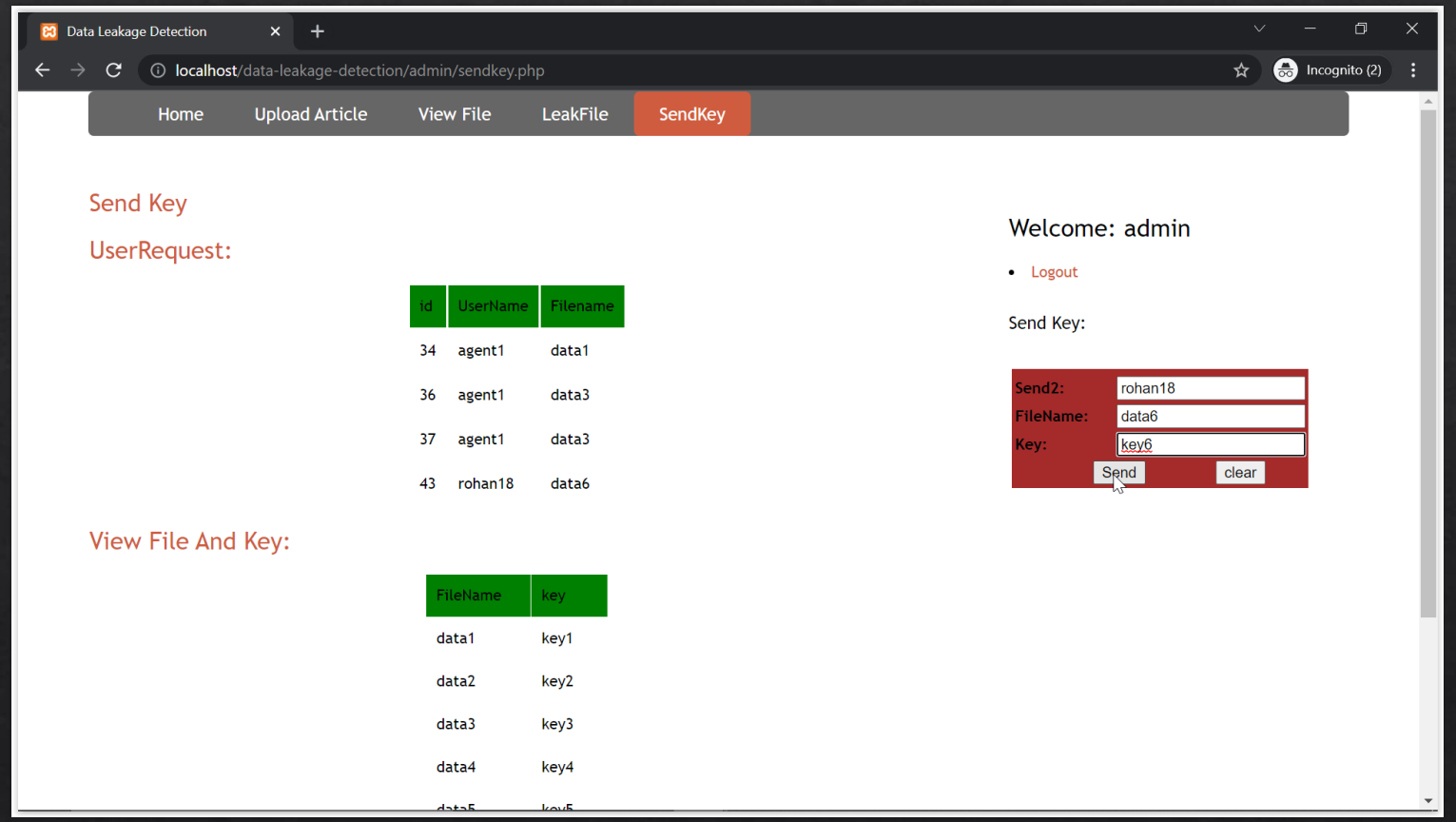
Module 1: Data control by the admin
Management of all the dataset present on the database



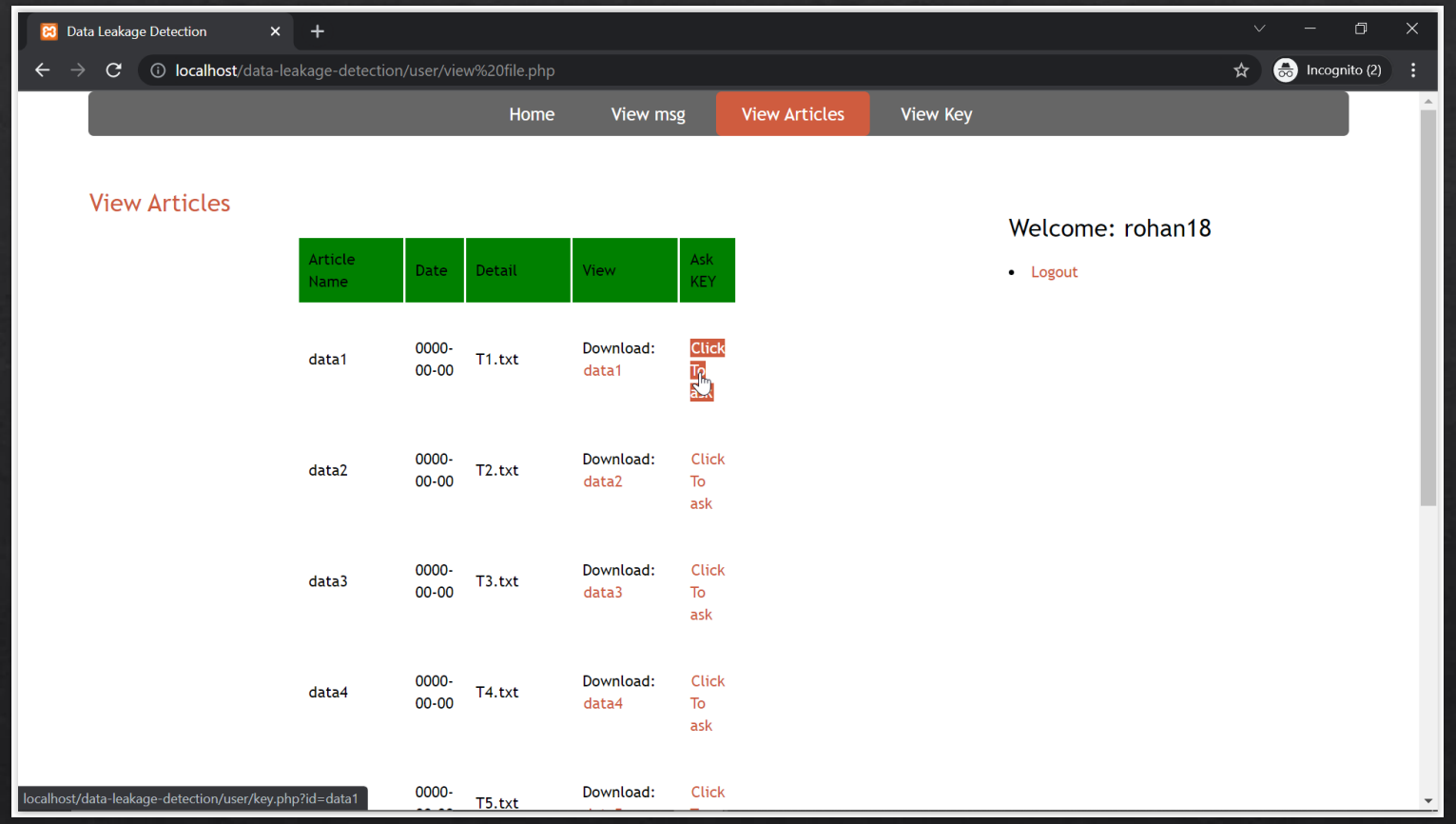
Uploading new dataset to the database



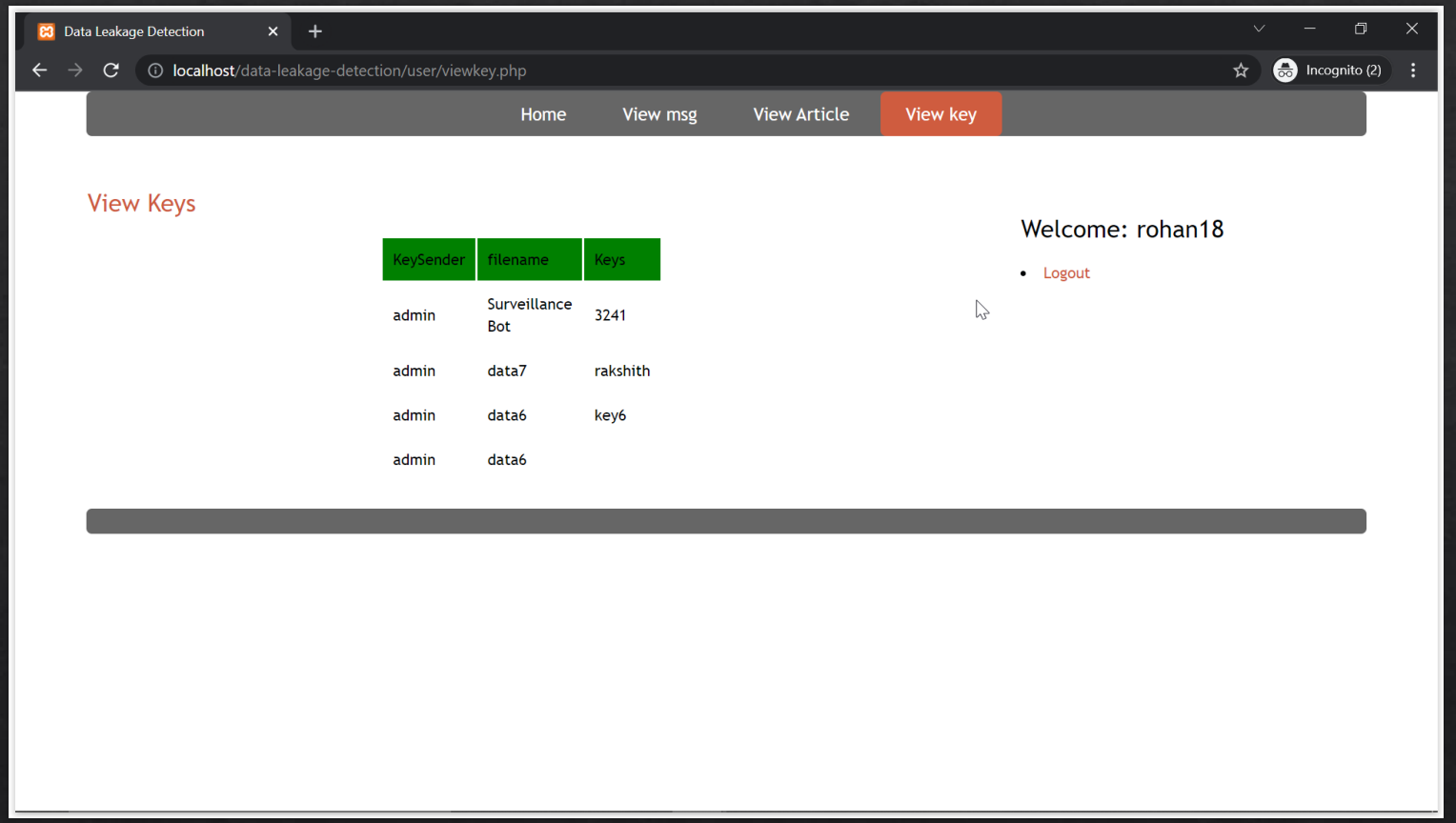
Sending the keys to as per requirement and trust between the admin and the agent



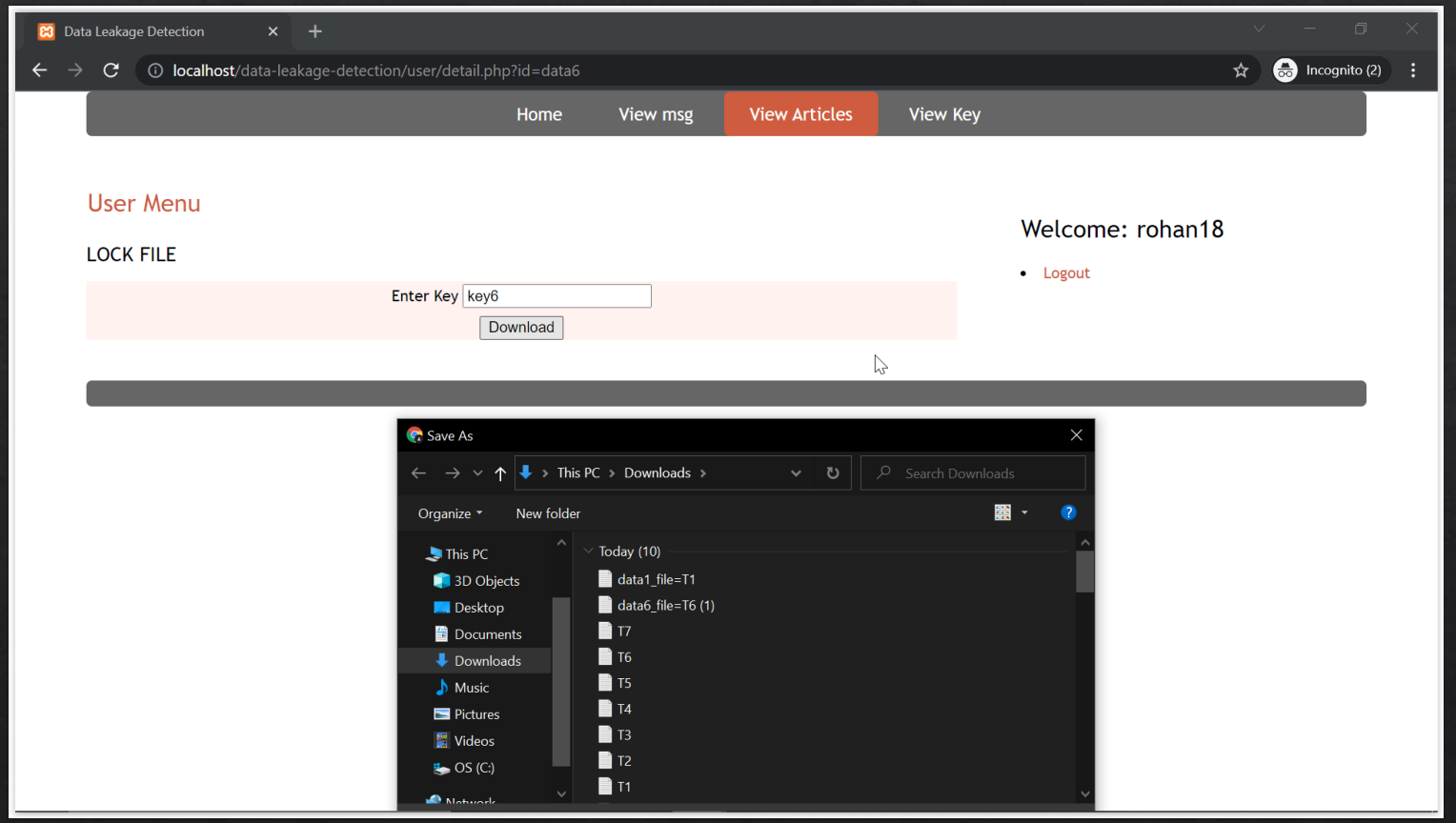
Module 2: Accessing of files by the user
Asking for key for the required dataset to the admin



Accessing the key send by the admin



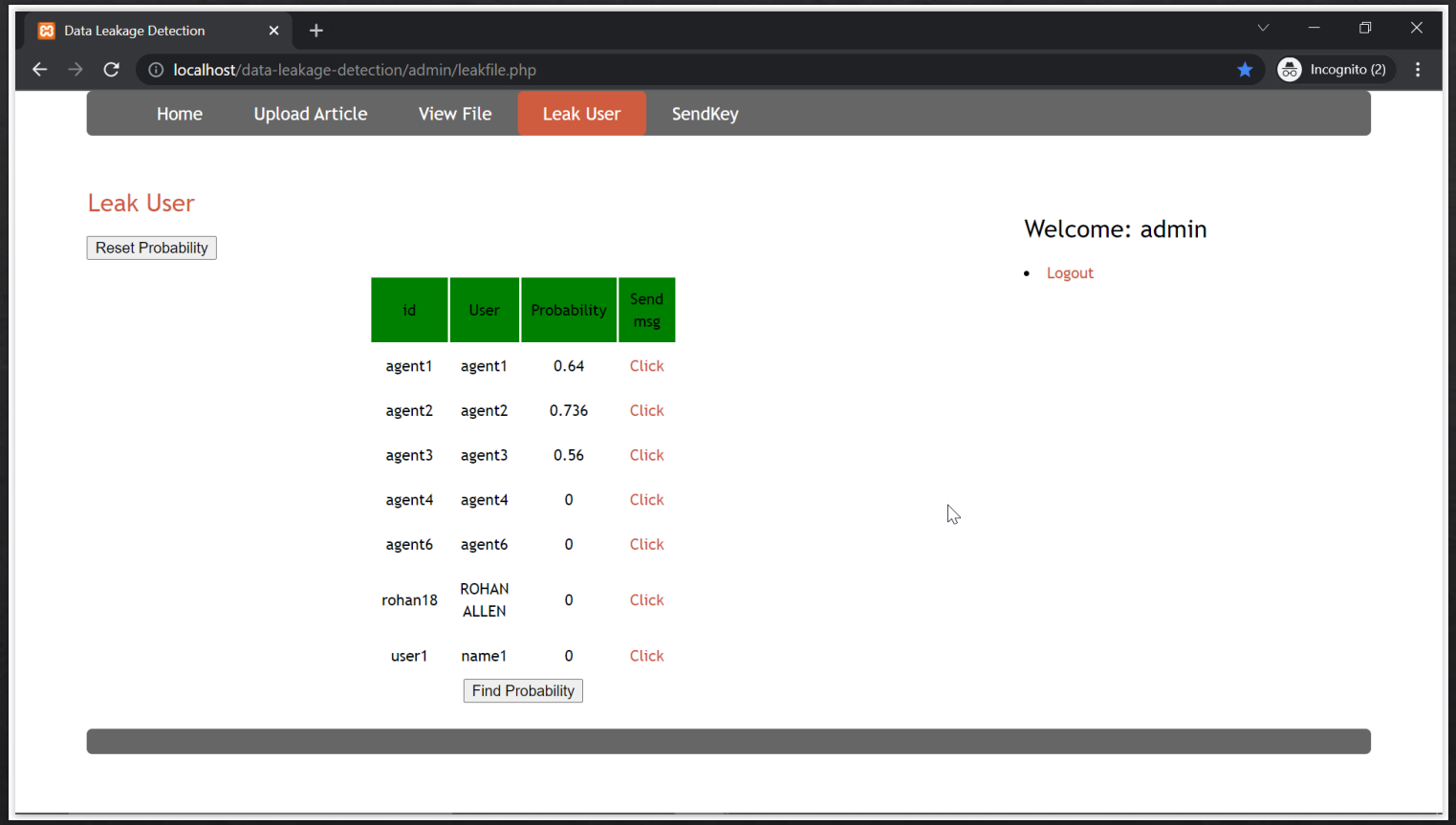
Downloading the dataset



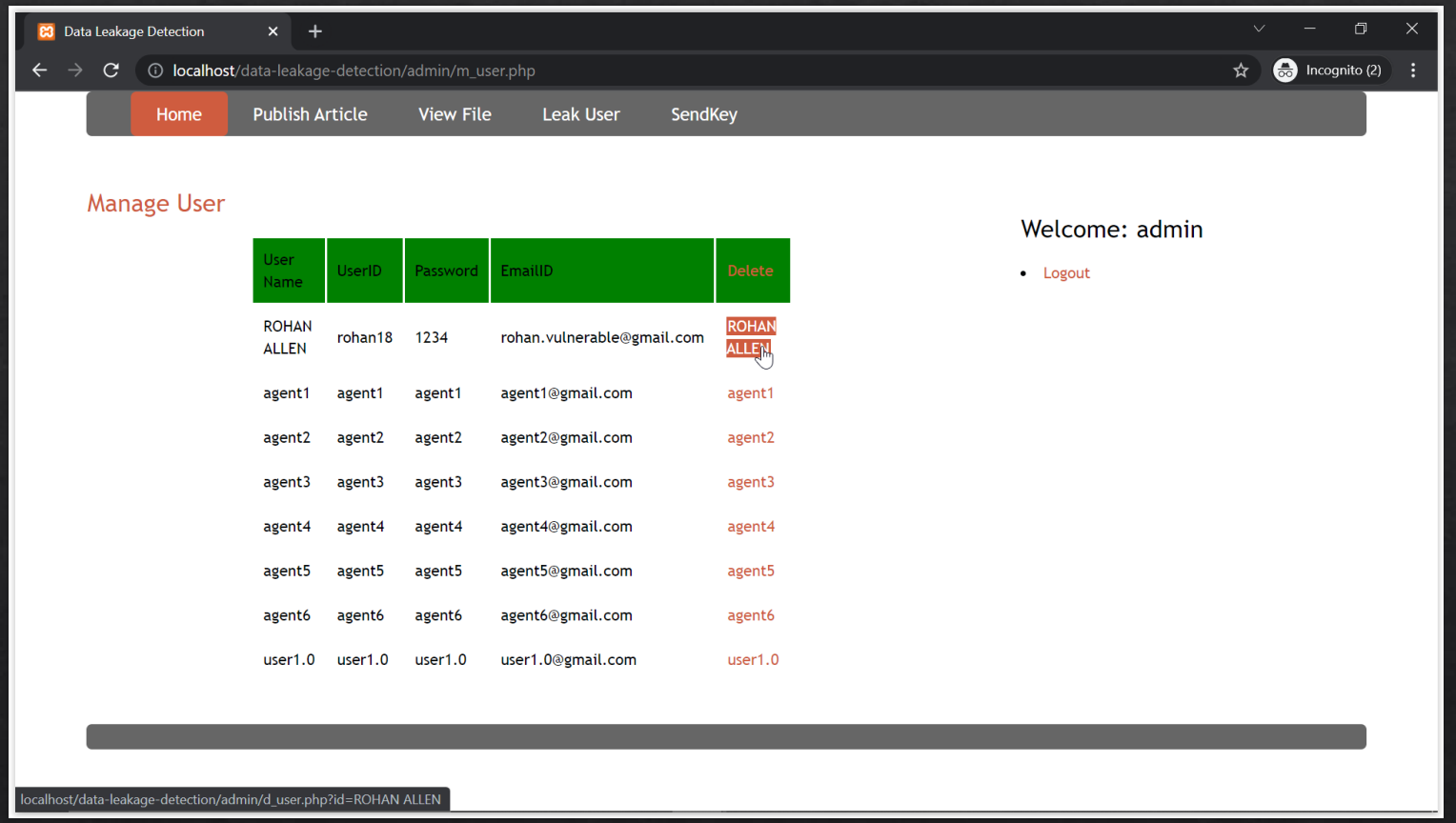
Module 3: Finding the probability of guilt of each agent when a particular dataset has been found with unauthorized entity

- 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



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



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