Data Leakage Detection using Guilty Agent Model

Muskan Gupta[1], Harshita Pundir[2]

[1]Vellore Institute of Technology, School of Computer Science and Engineering, Vellore - 632 014, Tamil Nadu, India
[2]Vellore Institute of Technology, School of Computer Science and Engineering, Vellore - 632 014, Tamil Nadu, India

---

Abstract: *Current statistics from various security research firms and government agencies recommend that there has been a rapid growth of data leaks over the past eight years and as of the present world, for the most part, depends on information exchange i.e., data transfer from one individual to another, also known as the distributary system. Data leakage is a serious security concern for every organization, and the first step to solving this problem is to find the source of the data leakage. This project works by saving the data from being outsourced by restricting the agents by using blacklisting so that it cannot be leaked.*

Keywords: Blacklisting, Data leakage detection, Guilty agent, Probability.

1. Introduction

Current statistics from various security research firms and government agencies recommend that there has been a rapid growth of data leak over the past eight years and as of the present world largely depends on the exchange of information i.e., transfer of data from one individual to another, also known as the distributary system. Information sent from the distributor to the client is confidential and therefore the information is only exchanged between the distributor and the trusted third parties. The information provided by the distributor should be secured, private, confidential and must not be duplicated as the information imparted with the trusted third parties is confidential and profoundly significant. In certain events, the data send by the distributor is duplicated by various agents which inflicts enormous harm to the facility and this process of data loss is known as data leakage. Data leakage should be detected early in order to prevent the data form becoming an open source. This project works by saving the data from being outsourced by restricting the agents by using blacklisting so that it cannot be leaked.

2. Problem statement and objective

Throughout course of doing business, once in a while sensitive information should be provided to reputable organizations. For instance, a medical clinic may give patient records to specialists who will need the data to devise new treatments. Similarly, an organization may have partnerships with different organizations that require sharing client data. Another venture maybe to redistribute it for data processing, so the information needs to be sent to various other organizations. The owner of the information is the distributor and the socalled 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 only by the valid users. It likewise helps in discovering Guilt of Agent from the given set of agents who has leaked the data, and therefore needs to be blacklisted, using Probability Distribution to find the guilt using the guilt model.

3. Literature Review

In paper [1], N. Kumar uses the Bell-Lapadula Model for Data Leakage detection where the server adds image logo to the archives. The logic behind the application of this technique is to embed classified message into the document in a computer-efficient manner. ASCII code is appended to documents and AES is used with SHA-512 to validate the hash algorithm. The main focus here is that only authorized clients will be able to access important documents. The created watermark is sent to the client with the public key certificates. This technique becomes infeasible to extend to web environment where multiple users access data.

In paper [2], Hybrid watermarking algorithms are used by R. Naik for leakage detection, the model is made out of three fundamental components, first is to extract the data being moved and to create a QR code utilizing data structures. At that point insert QR watermark into the cloud data utilizing frequency domain techniques. At last estimate if anyone has altered the information by looking at the current data features and finding the guilty agent who has leaked the data by extracting watermark and comparing data from watermark with agent's details. The issue faced is that Watermarks involve some modification of the original information and can sometimes be destroyed if the data recipient is malicious.

In paper [3], X. Shu has carried out the Fuzzy fingerprint technique. This framework satisfies high recognition

precision and finds transformed leakage showed up distinctively corresponding to the cutting-edge inspection structures. They parallelize their layout on pix making ready unit as well as exhibit the robust adaptability of their discovery arrangement required by a sizable affiliation. This method has proved to enhance the extent of accuracy in finding transformed data leaks when contrasted with the state-of the art set intersection methodology, but lacks in Time Consuming Process. The problem faced is that the false positive and true positive yield the same fingerprints

In paper [4], X. Shu uses AlignDLD and Coll inter system for detection. This algorithm is locked in for perceiving large and crucial data patterns. This distinguishing proof is combined with a sample algorithmic technique, which permits one to take a gander at the comparability of two solitarily tested successions. This structure achieves extraordinary exploration in the exactness in perceiving transformed leakage. This model provides considerable speedup and demonstrates high adaptability of the design, but on the other hand datamovement trail approach is not utilized. The disadvantage is that this algorithm only detects inadvertent leaks and not malicious leaks.

In paper [5], L. Zheng uses Security Label Based Dynamic Privacy Information Disclosure Detection. This method initially identifies if a session connection exists or not, and if positive, the privacy information disclosure detection is not required. If not, it is essential to verify the individual, to whom the service consumer belong to, and then dynamically test whether the services on the service chain coincides the read and write permissions of the private information. If both are satisfactory, i.e. returning positive depicts that the service consumer does not have a privacy information disclosure problem during the call. Otherwise, returning negative would infer that the service consumer faced a privacy information disclosure problem during the calling process. This method faces drawbacks like being infeasible in case of complex combinations.

4. Proposed Algorithm

To protect the sensitive data, most efficient way is to modify the data and make it “less sensitive”. At times, it is significant not to modify the original distributor’s data. Therefore, here software where the original sensitive data cannot be disturbed are considered. Generally, leakage detection is taken care of by watermarking, where a unique code is inserted in each distributed copy. On the off chance that replica is later found in the hands of an unauthorized user, the leaker can be identified. Watermarks involve few modification of the original data and can sometimes be removed if the recipient of the data is malevolent.

In this research we use unobtrusive techniques for detecting leakage of a set of records in the database. The model developed is used for calculating the “guilt” of agents. Algorithms are also provided for distributing objects to agents, in a way that it enhances the chances of identifying the leaker of the data. An option of adding “fake” objects to the distributed set into consideration. Such objects do not compare to genuine entities but seem realistic to the agents. It could be said, the fake objects go about as a sort of watermark for the entire set, without changing any individual members. On the off chance that it turns out an agent was given at least one fake object that were leaked, then the distributor can be surer that agent was guilty.

Say the distributor has the set M = {m_1, . . . , m_m}. The leaked set found out is A. Assumptions made in this implementation are:

1. For all m, m’ ∈ E such that m ≠ m’ the provenance of t is independent of the provenance of m’
2. An object m ∈ R can only be obtained by the target in one of two ways:
   - A single agent Z_i leaked t from its own R_i set; or
   - The target guessed (or obtained through other means) t without the help of any of the n agents.

Consider that sets M, O’s and E are as follows: M = {m1, m2, m3}, O1 = {m1, m2}, O2 = {m1, m3}, E = {m1, m2, m3}. For this situation, every one of the three of the distributor’s objects have been leaked and matches in with E. Consider how the target may have obtained object m_1, which was given to both agents. From Assumption 2, the target either fluked m_1 or one of Z_1 or Z_2 was responsibed for leaking it. Knowing that the probability of the former event is ‘pos’, so assuming that probability that each of the two agents leaked h_1 is the same cases formed are:

- the target guessed t_1 with probability p;
- agent Z_1 leaked t_1 to S with probability (1 − pos)/2
- agent Z_2 leaked t_1 to S with probability (1 − pos)/2

Similarly, it is found that agent Z_1 leaked t_2 to E with probability 1 − p since he/she is the only agent that has this particular object. Given these values, the probability that agent Z_1 is not guilty is computed using, namely that Z_1 did not leak either object:

$$P\ r\{K'_1|E\} = (1 - (1 - pos)/2) \times (1 - (1 - pos)) \quad (1)$$

Hence, the probability that Z_1 is guilty is:

$$P\ r\{K1|E\} = 1 - P\ r\{G'_1\} \quad (2)$$

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

$$P\,r\{Z_i \text{ leaked t to E}\} = \{ 1 - pos / |V_t| , \text{ if } Z_i \in V_t \text{ and } 0, \text{ otherwise } \} \quad (3)$$

Given that agent Z_i is guilty, if he leaks even one value to E, with Assumption 1 and Equation 3 the probability $P\,r\{K_i | E\}$ is computed, that agent Z_i is guilty:

$$P\,r\{K_i | E\} = 1 - \pi_{t \in S \cap Ri} (1 - (1 - pos)/ |Vt|) \quad (4)$$

5. Flowchart representation

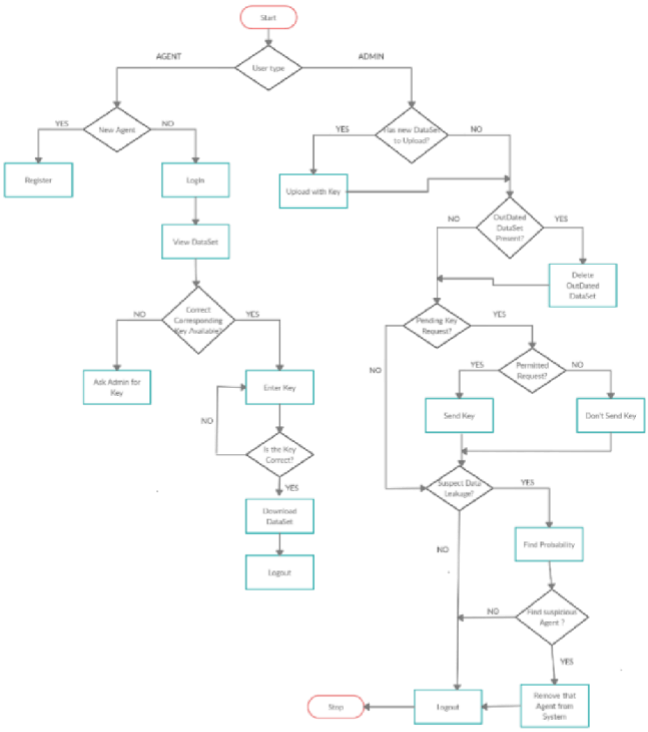

Figure 1: Flowchart of the proposed model

6. Module Structure

6.1 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.

6.2 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.

6.3 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 maintaining in mind the cookie jar analogy i.e., if we capture Freddie with a lone cookie, he can argue that a friend gave it to him. However, if we capture Freddie with five cookies, it will be much more difficult for him to prove that his hands were not in the cookie jar. If the distributor finds "enough proof" that an agent leaked the data, he may stop associating with him, or might also initiate legal proceedings against him.

6.4 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

7.1.1 Management of all the dataset present on the database

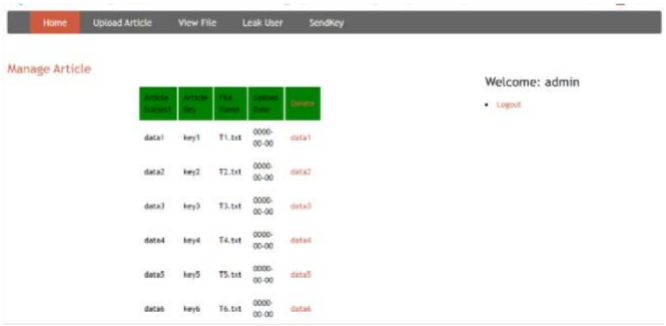

Figure 2: Screenshot of dataset management by admin

7.1.2 Uploading new dataset to the database

"known bad" by using the probability of the leaker calculated using the guilt model in order to ensure security of the system.

7. Implementation Screenshots

7.1 Data control by the admin

Figure 3: Screenshot of dataset upload by admin

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

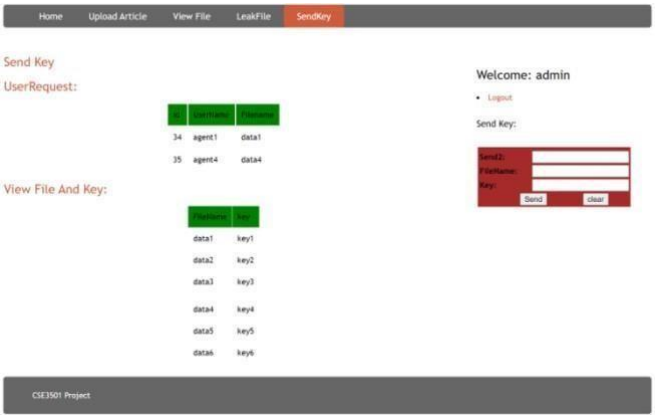

Figure 4: Screenshot of key sending facility by admin

7.2 Accessing of files by the user

7.2.1 Asking for key for the required dataset to the admin

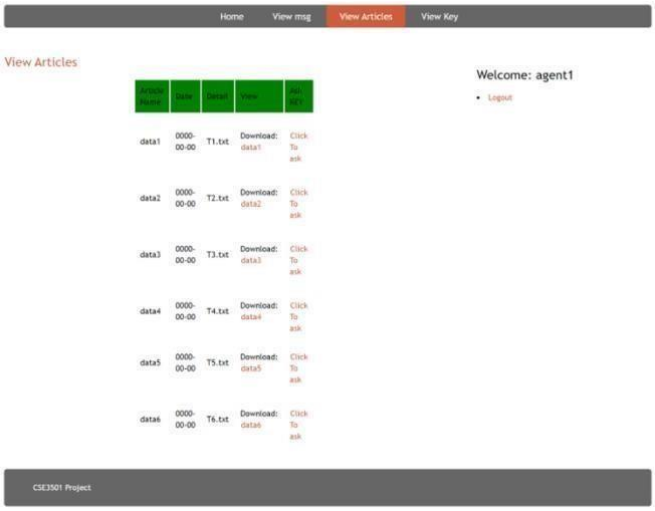

Figure 5: Screenshot of asking key from admin

7.2.2 Accessing the key send by the admin

Figure 6: Screenshot of accessing available key

7.2.3 Downloading the dataset

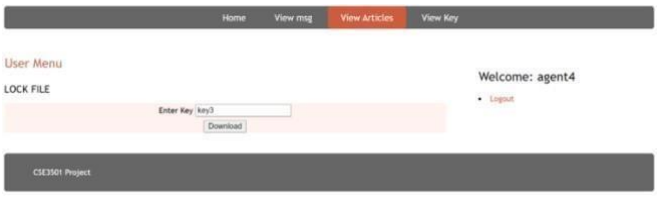

Figure 7: Screenshot of downloading permitted dataset

7.3 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={m2,m4,m6,m8,m10,m12}
Agent2={m6,m7,m8,m9,m10}
Agent3={m1,m7,m9,m4,m5}
Agent4={m1,m3,m5,m7,m9,m11}
And the leaked dataset be E={m1,m6,m9,m10}
Then the guilt probability would be

7.4 Managing of user which allows admin to blacklist any agent who is not trust worthy

Figure 8: Screenshot of guilt probability

Figure 9: Screenshot of blacklisting guilt agent

8.Results

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.

9. 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) N. Kumar, V. Katta, H. Mishra and H. Garg, "Detection of Data Leakage in Cloud Computing Environment," 2014 International Conference on Computational Intelligence and Communication Networks, Bhopal, 2014, pp. 803-807, doi: 10.1109/CICN.2014.172.R.
2) Caves, Multinational Enterprise and Economic Analysis, Cambridge University Press, Cambridge, 1982. (book style)
3) R. Naik and M. N. Gaonkar, "Data Leakage Detection in cloud using Watermarking Technique," 2019 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, Tamil Nadu, India, 2019, pp. 1-6, doi: 10.1109/ICCCI.2019.8821894.
4) X. Shu, D. Yao and E. Bertino, "Privacy-Preserving Detection of Sensitive Data Exposure," in IEEE Transactions on Information Forensics and Security, vol. 10, no. 5, pp. 1092-1103, May 2015, doi: 10.1109/TIFS.2015.2398363.
5) X. Shu, J. Zhang, D. D. Yao and W. Feng, "Fast Detection of Transformed Data Leaks," in IEEE Transactions on Information Forensics and Security, vol. 11, no. 3, pp. 528-542, March 2016, doi: 10.1109/TIFS.2015.2503271.
6) X. Zhu, C. Pan, L. Zheng and J. Sun, "Detection Method on the Privacy Leakage for Composite Services," 2019 International Conference on Networking and Network Applications (NaNA), Daegu, Korea (South), 2019, pp. 415-420, doi: 10.1109/NaNA.2019.00078.