Data Leakage Detection Through Guilty Agent

Mr. Gavali P. R. (PG Student) Prof. P. N. Kalavadekar (PG Guide)
SRES College of Engineering, Kopargaon, Affilated to University of Pune

Abstract

In real life sometime a valuable data or sensitive data is given to a set of supposedly trusted agents (third parties).if the data which is given to the some third parties found publically then we can say that the data might be leaked & also finding this guilty data is challenging task. To find such guilty data, traditionally people was used the watermarking technique. If this watermarked data was found on any site then they are claimed for its owenership.This Issus is solved by using data allocation strategies which are used to improve the performance of finding the guilty data. In this paper we are introducing some data allocation strategies that are useful to us for finding the leaked data. for finding this we can used the concept of fake object which looks like a real object but is actually fake. By using this method we can find the guilty parties as well as leaked data. in this paper we are also introduced Explicit & Implicit(sample) algorithms which are used for data allocations.

Key terms: guilty agents, data distributor, data leakage, fake object, sensitive data.

1. Introduction

Sometime people have been given some data to some trusted third parties. Their main aim is doing business & maintains relationship. this data is very sensitive & valuable. For example, some Industries may give their workers records to some trusted third party for updating. Similarly, for collaboration some companies require sharing of a customer data with other company.

Another enterprise may outsource its data processing, so data must be given to various other companies. The owner of the data is called the distributor and the supposedly trusted third parties the agents. The goal is to detect when the distributor's sensitive data has been leaked by agents, and if possible to identify the agent that leaked the data.

for example medical data given to the research scholars or students, a wide range of information that can include: political opinion; religious or other similar beliefs; memberships; physical or mental health details; personal life; or criminal or civil offences.Sensitive data can also include information that relates to you as a consumer, client, employee and it can be identifying information as well:

2. Related Work

Traditionally there are lots of methods has been done on the data leakage prevention. basically it has based on the trustworthiness [1] which is used to assess the trustiness of the customer & also used for Maintaining the log of all customer's request which is related to the data provenance problem [3] e.g. tracing the lineage of objects. Sometime we used the watermarking concept which is more relevant to the data allocation strategy [4],[5] for the purpose of establishing original ownership of distributed objects.

Some few years ago, leakage detection was handled by watermarking technique which is mostly used to identify a data owner and it also used to attacks where a pirate claims ownership of the data or weakens a merchant's claims e.g. we are use the unique code which is embedded in each distributed copy. If that copy is later discovered in the hands of an unauthorized party, the leaker can be identified. Watermarks were initially used in images, video and audio data whose digital representation includes considerable redundancy.

IJERT

Entities and Agents

Firstly we arrange a set T = {t1. . . tm} of valuable data objects. if we wants to share some of the objects with a set of agents (A1, A2, ...An) then the condition is that objects are does not leaked to other third parties. The objects in T could be of any type and size, e.g., they could be tuples in a relation, or relations in a database. An agent Ui receives a subset of objects Ri⊆ T, deter- mined either by a implicit (Sample) request or an explicit request:

1. Implicit (Sample) request Ri = SAMPLE (T, mi): Any subset of mi records from T can be given to Ui.

2.Explicit request Ri = EXPLICIT (T, condi): Agent Ui receives all the T objects that satisfy condi.

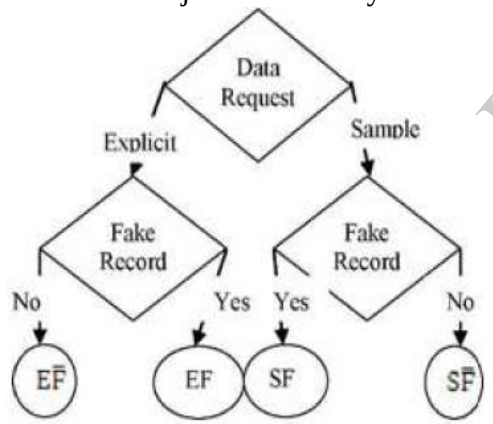

Figure 1: instances of Leakage Problem

The Figure. 1 represents four problem instances with the names EF, E ¯ F, SF and S ¯ F, where E stands for explicit requests, S for sample requests, F for the use of fake objects, and F for the case where fake objects are not allowed

The data are distributed & also for the purpose of improving effectiveness in detecting guilty agents

We are add the fake objects to the distributed data. Sometime, fake objects may impact the correctness of what agents do, so they may not always be allowable.

The distributor creates and adds fake objects to the data that he distributes to agents. In many cases, the the names EF, E ¯ F, SF and S ¯ F, where E stands for explicit requests, S for sample requests, F for the use of fake objects, and F for the case where fake objects are not allowed.

Distributor may be limited in how many fake objects he can create Example: Say T contains customer records for a given company A. Company A hires a marketing agency U1 to do an on-line survey of customers. Since any customers will do for the survey, U1 requests a sample of 1000 customer records. At the same time, company A subcontracts with agent U2 to handle billing for all California customers. Thus, U2 receives all T records that satisfy the condition “state is California.”Although we do not discuss it here, our model can be easily extended to requests for a sample of objects that satisfy a condition (e.g., an agent wants any 100 California customer records).Also note that we do not concern ourselves with the randomness of a sample. (We assume that if a random sample is required, there are enough T records so that the to-be presented object selection schemes can pick random records from T.)

IJERT

Guilty Agents

For the purpose of finding the Guilty data firstly, we are give the data to some agents, then we discovers that a set S T has leaked. This means that some third party called the target has been caught in possession of S. For example, this target may be displaying S on its web site, or perhaps as part of a legal discovery process, the target turned over S to the distributor. Since the agents A1.....An have some of the data, it is reasonable to suspect them leaking the data. However, the agents can argue that they are innocent, and that the S data was obtained by the target through other means.

The main focus of this paper is the data allocation problem: how can the distributor "intelligently" give data to agents in order to improve the chances of detecting a guilty agent? As illustrated in Figure 1, there are four instances of this problem we address, depending on the type of data requests made by agents and whether "fake objects" are allowed. The two types of requests we handle sample and explicit.

Fake objects

Fake objects are objects generated by the distributor that are not in set T. The objects are designed to look like real objects, and are distributed to agents together with the T objects, in order to increase the chances of detecting agents that leak data. We discuss fake objects in more detail in next section.

Creation of Fake tuples

The creation of fake tuples can be done manually by the database owner which is a viable approach, our effort is to make this process as automatic as possible. Therefore, our goal has been to develop an insertion algorithm that, with little supervision, can effectively generate the fake tuples. we can developed the model which can create the fake object for agent Ai as a black-box function CREATEFAKEOBJECT(Ri, Fi, condi) that takes as input the set of all objects Ri, the subset of fake objects Fi that Ui has received so far and condi, and returns a new fake object. This function needs some condi to produce a valid object that satisfies Ai's condition. Set Ri is needed as input so that the created fake object is not only valid but also indistinguishable from other real objects.

If we want to send the same fake object to a set of agents then we can use the function CREATEFAKEOB- JECT () . In this case, the function arguments are the union of the Ri and Fi tables respectively, and the intersection of the conditions condi's. Although we do not deal with the implementation of CREATEFAKEOB- JECT () we note that there are two main design options. Similarly, The function can either produce a fake object on demand every time it is called, or it can return an appropriate object from a pool of objects created in advance.

Adding of fake object

The distributor may be able to add fake objects to the distributed data in order to improve his effectiveness in detecting guilty agents. However, fake objects may impact the correctness of what agents do, so they may not always be allowable. The idea of perturbing data to detect leakage is not new. However, in most cases, individual objects are perturbed, e.g., by adding random noise to sensitive salaries, or adding a watermark to an image. A trace file is maintained to identify the guilty agent. Trace file are a type of fake objects that help to identify improper use of data.

IJERT

4. Programmer's design

The following Figure shows the breakdown structure which is mainly divided into the Four models, basicaly client-server connection data allocation, registration with addition of fake tuple & Agent Guilt identification.Each model specify the control flow of this project & after completion of each model we are getting some desired output which is the input for the further pven therocess.so here we are using the waterfall approach.

the details of these models is as follows

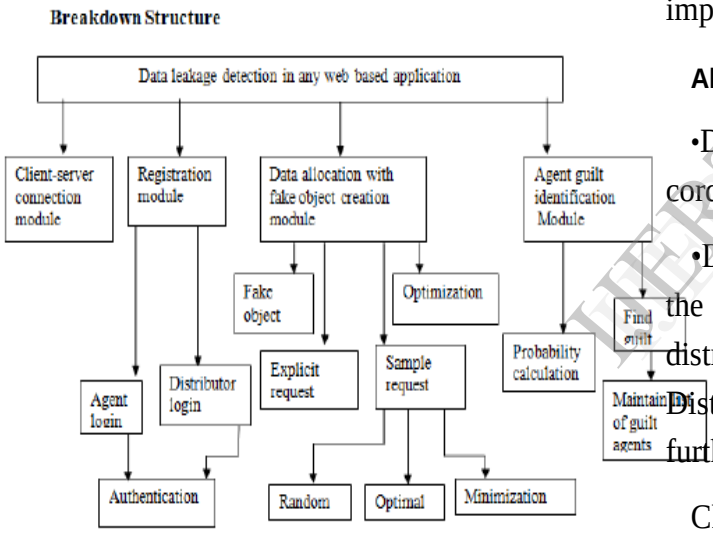

Figure 2: Breakdown Structure

Module 1: Client-Server Connection

We are establishing the connection between the client and server in this module through internet or via LAN. System requires three or more clients for connection. These clients are requesting for connection with server.

Module 2: Registration

The client registers to the database of server in this module. Server provides username and password to the client for authentication. In registration, system registers for both agent and distributor for that purpose system takes following input: Agent id, Agent name, address, username, and password.

Module 3: Data allocation with Fake object creation.

The main focus of our project is the data allocation problem as how can the distributor "intelligently" give data to agents in order to improve the chances of detect- ing a guilty agent.

Algorithm steps for Find Guilt Agent:

•Distributor selects the agents to send the data according to agent request.

•Distributor creates fake data and allocates it to the agent. The distributor can create fake data and distribute with agent data or without fake data. Distributor is able to create more fake data; he could further improve the chance of finding guilt agent.

Check number of agents, who have already received data. Distributor checks the number of agents, who have already received data.

Check for remaining agents. Distributor chooses the remaining agents to send the data.

•we can calculate the probability for guilt agent. To compute this probability, we can required the esti mate probability value that can be "guessed" by the target.
•We can find the Guilt agent after getting this probability.
•Finally, we can maintain the list of Guilt Agent.

IJERT

Input Design

Login form is provided for user. In this, an option for member who has already registered is provided. So that he can directly give the username and password provided by the administrator of this site, other new users who are willing to register can register their details. To register- ing the following details are gathered from the user, First name, Last name, E-Mail id. When they agree for the terms and conditions provided and then submit their de- tails will be updated to the database and they will be considered as valid member. Each administrator checks the request list and if the agent request is valid, he sanctions the request. Else he adds fake objects to the data list and sends those objects to the agent.

Output Design

This module identifies the user and validates the user. An authentication factor is a piece of information and process used to authenticate or verify the identity of a person or other entity requesting access under security constraints. If the user is valid it allows to giving re- quest to distributor. A file is maintained to identify the guilty agents and depending on that the requests are sanctioned. The objects are sent in serialized format. Once after the response from distributor, the agents can see the data.

Conclusion

Whenever valuable or sensitive business information such as customer or patient data, source code or design specifications, intellectual property and trade secrets are handed over to supposedly trusted third parties then there is a possibility of some data leakage. When these are leaked out it leaves the company unprotected and goes outside the jurisdiction. This uncontrollable data leakage put business in a vulnerable position. The main focus of this project is the data allocation problem. It species how the distributor can"intelligently" give data to agents in order to improve the chances of detecting a guilty agent. By adding fake objects to distributed set, the distributor cans find the guilt agent easily.

References

[1]P. Papadimitriou and H. Garcia-Molina, Data leak- age detection. Technical report, Stanford University, 2008.

[2]P. Buneman, S. Khanna and W.C. Tan. Why and where: A characterization of data provenance. ICDT 2001, 8th International Conference, London, UK, January4-6,2001, Proceedings, volume 1973 of Lecture Notes in Computer Science, Springer, 2001.

[3]S. Czerwinski, R. Fromm, and T. Hodes. Digital mu- sic distribution and audio watermarking.

[4] Rakesh Agrawal, Jerry Kiernan. Watermarking Rela- tional Databases IBM Almaden Research

[5] S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subramanian. Flexible support for multiple access control policies. ACM Trans. Dataset Systems,26(2):214-260, 2001.

[6] Papadimitriou P, Garcia-Molina H. A Model For Data Leakage Detection IEEE Transaction On Knowledge And Data Engineering Jan.2011.

IJERT

[7]L. Sweeney. Achieving k- anonymity privacy protection using generalization and suppression. International Journal on Uncertainty, Fuzzyness and Knowledge-based Systems-2002

[8] Ersin Uzun and Bryan Stephenson.Security of Rela- tional Databases in Business Outsourcing HP Labo- ratories, HPL-2008

IJERT