DATA LEAKAGE DETECTION SYSTEM

Manish Tated[1], Shital Patel[2], V.N. Patil[3]

[1]Manish Tated (Student), Bharati Vidyapeeth College of Engineering, Navi Mumbai
[2]Shital Patel: Professor, Bharati Vidyapeeth College of Engineering, Navi Mumbai
[3]V.N. Patil: Professor, Bharati Vidyapeeth College of Engineering, Navi Mumbai

Abstract - *Data Leakage Detection System is one of the most versatile and economical process available for securing & Preventing Data. One of the most important key components during the confidential data prevention of these methods is appropriate selection of technological parameters of processing, as well as initial geometrical features of perform , to provide positive functional characteristics of final Data.*

When any company wants to do any work in minimum time there is one main person who having main server & he passes that data to client server. Client server gets the data & if required then passes to another if required. In this process i/p address & mac address are recorded on the main server due to this data leakage from the particular client is found. This client is called as guilt agent.

Key words - Main server, client server, i/p address, mac address, guilt agent.

1. INTRODUCTION

Data distributors have given sensitive data to a set of supposedly trusted agents. Some of the data are misused and found in an unauthorized place i.e. someone is using that data. The distributors must assess that likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means. We proposed data allocation strategies across the agents that improve the probability of identifying data misuse or leakages when we do business, sometimes sensitive data must be handed over to supposedly trusted third parties. For example, a hospital may give patient records to researchers who will devise new treatments. Similarly, a company may have partnerships with other companies that require sharing customer data. Another enterprise may outsource its data processing, so data must be given to various other companies. We call the owner of the data the distributor and the supposedly trusted third parties the agents. Our 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. We consider applications where the original sensitive data cannot be perturbed.

1.1PERTUBATION:

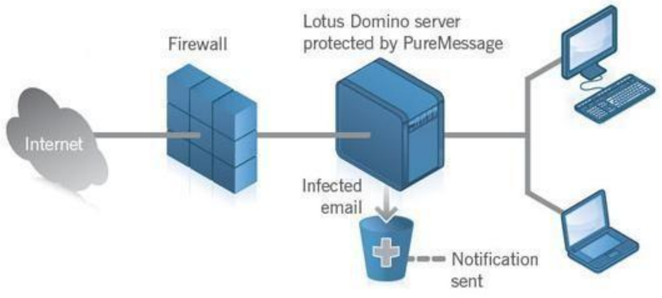
Perturbation is a very useful technique where the data is modified and made "less sensitive" before being handed to agents. For example, one can add random noise to certain attributes, or one can replace exact values by ranges. However, in some cases it is important not to alter the original distributor's data. For example, if an outsourcer is doing our payroll, he must have the exact salary and customer bank account numbers. If medical researcher treating patients they must need accurate data for patients.

1.2 LEAKAGE DETECTION:

Traditionally, leakage detection is handled by watermarking, e.g., a unique code 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 can be very useful in some cases, but again, involve some modification of the original data. Furthermore, watermarks can sometimes be destroyed if the data recipient is malicious.

2. UNOBTUSIVE TECHNIQUE

In this paper we study unobtrusive techniques for detecting leakage of a set of objects or records. Specifically, we study the following scenario: After giving a set of objects to agents, the distributor discovers some of those same objects in an unauthorized place. (For example, the data may be found on a web site, or may be obtained through a legal discovery process.) At this point the distributor can assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means. Using an analogy with cookies stolen from a cookie jar, 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. In this paper we develop a model for assessing the "guilt" of agents. We also present algorithms for distributing objects to agents, in a way that improves our chances of identifying a leaker. Finally, we also consider the option of adding "fake" objects to the distributed set. Such objects do not correspond to real entities but appear realistic to the agents. In a sense, the fake objects acts as a type of watermark for the entire se, without modifying any individual members. If it turns out an agent was given one or more fake objects that were leaked, then the distributor can be more confident that agent was guilty.



Working of Project

The project's main purpose is to know the exact condition of the broadband line's and the associated network's status. The complete IP Monitoring System consists of 3 major areas they are: Importing of IP's, Monitoring and Ticketing.

2.1 USER CLASSES & CHARACTERISTIC:

User

- User registers himself and is given a login name and password.
- User is the person who requests for the monitoring process by feeding in his details.

Administrator

- An administrator may be a dedicated staff whose responsibility is monitoring the IP, managing the user and the system engineer database.
- They may need to create and manage the complete information of the Monitoring IP; Ticket raised IP's and also the resolved IP's.

System Engineer

- System Engineer is a person who looks after the resolving of IP's that have been raised while ticketing.
- He is given a login name and password and comments on the GUI while resolving.
- Once the IP is resolved, he makes a mark of it in the GUI which the user can see.

Operating Environment

- IP Monitoring System is C, C++, java based application. It needs application server for its deployment.

3. CONCLUSION

In doing a business there would be no need to hand over sensitive data to agents that may unknowingly or maliciously leak it. And even if we had to hand over sensitive data, in a perfect world we could watermark each object so that we could trace its origins with absolute certainty. However, in many cases we must indeed work with agents that may not be 100% trusted, and we may not be certain if a leaked object came from an agent or from some other source. In spite of these difficulties, we have shown it is possible to assess the likelihood that an agent is responsible for a leak, based on the overlap of his data with the leaked data and the data of other agents, and based on the probability that objects can be "guessed" by other means. Our model is relatively simple, but we believe it captures the essential trade-offs. The algorithms we have presented implement a variety of data distribution strategies that can improve the distributor's chances of identifying a leaker. We have shown that distributing objects judiciously can make a significant difference in identifying guilty agents, especially in cases where there is large overlap in the data.

ACKNOWLEDGEMENT

The author wishes to place on record his gratitude to **Prof. Mrs. Shital V. Patel** and **Prof. V. N. Patil** – Department of Mechanical Engineering, Bharati Vidyapeeth College of Engineering, Navi Mumbai – for their crucial guidance, corrections and timely suggestions accorded.

REFERENCES

1. User Interfaces in C#: Windows Forms and Custom Controls by Matthew MacDonald.
2. Applied Microsoft® .NET Framework Programming (Pro-Developer) by Jeffrey Richter.
3. R. Agrawal and J. Kiernan. Watermarking relational databases. In VLDB '02: Proceedings of the 28th international conference on Very Large Data Bases, pages 155–166. VLDB Endowment, 2002.
4. Data Communications and Networking, by Behrouz A Forouzan.
5. Computer Networking: A Top-Down Approach, by James F. Kurose.