Data leakage detection project synopsis

ACKNOWLEDGEMENT

For all the efforts behind the paper work, I first & foremost would like to express my sincere appreciation to the staff of Dept. of Computer Sci.& Engg., for their extended help & suggestions at every stage of this paper. It is with a great sense of gratitude that I acknowledge the support, time to time suggestions and highly indebted to my guide Finally, I pay sincere thanks to all those who indirectly and directly helped me towards the successful completion of the paper.

ABSTRACT

We study the following problem: A data distributor has given sensitive data to a set of supposedly trusted agents (third parties). Some of the data are 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.

INTRODUCTION

Data leakage is defined as the accidental or unintentional distribution of private or sensitive data to an unauthorised entity.
Data leakage poses a serious issue for companies as the number of incidents and the cost to those experiencing them continue to increase.
Data leakage is enhanced by the fact that transmitted data including emails, instant messaging, website forms, and file transfers among others, are largely unregulated and unmonitored on their way to their destinations.
The main scope of this module is providing complete information about the data/content that is accessed by the users within the website. Forms Authentication

technique is used to provide security to the website in order to prevent the leakage of the data.
Continuous observation is made automatically and the information is send to the administrator so that he can identify whenever the data is leaked.

OBJECTIVES

Data Leakage Detection Project propose data allocation strategies that improve the probability of identifying leakages. 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.
In the course of doing business, sometimes sensitive data must be handed over to supposedly trusted third parties. Another enterprise may out source its data processing, so data must be given to various other companies. There always remains a risk of data getting leaked from the agent. 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. But again it requires code modification. Watermarks can sometimes be destroyed if the data recipient is malicious. 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. 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 website, 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 five 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 act as a type of watermark for the entire set, without modifying any individual members. If it turns out that an agent was given one or more fake objects that were leaked, then the distributor can be more confident that agent was guilty.

EXISTING SYSTEM

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. E.g. 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.

SCOPE & PURPOSE

Data Leakage Detection: The main scope of this module is provide complete information about the data/content that is accessed by the users within the website. Forms Authentication technique is used to provide security to the website in order to prevent the leakage of the data.

GOAL OF PURPOSED SYSTEM

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. Perturbation is a very useful technique where the data is modified and made "less sensitive" before being handed to agents. we develop unobtrusive techniques for detecting leakage of a set of objects or records. In this section 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 set, 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.

PROJECT REQUIREMENTS

SOFTWARE

Front End technology

- Html5, css3, javascript , bootstrap5.

Backend technology

- PHP, MYSQL

Code editor

- Notepad, notepad++, sublime, visual code, atom

Web browser

- Google chrome
- Mozilla firefox
- Opera

- **HARDWARE**

- Computer System
- 2GB RAM, 500GB Hard disk, Internet

FEASIBILITY STUDY

In the course of doing 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. 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 researchers will be treating patients (as opposed to simply computing statistics), they may need accurate data for the patients. 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.