Data Leakage Detection in PHR (Public Health Record)

Sneha S.Gulawani, Sayli B.Adsure, Ashwini B. Gaikwad, Prof. R.S.Tambe

Abstract— **In every enterprise, data leakage is very serious problem faced by it. An owner of enterprise has given sensitive data to its employee but in most of the situation employee leak the data. That leak data found in unauthorized place such as on the web of comparator enterprise or on laptop of employee of comparator enterprise or the owner of comparators laptop. It is either observed or sometimes not observed by owner. Leak data may be source code or design specifications, price lists, intellectual property and copy rights data, trade secrets, forecasts and budgets. In this case the data leaked out it leaves the company goes in unprotected the influence of the corporation. This uncontrolled data leakage puts business in a backward position. In this paper, we implement methods aimed at improving the odds of detecting such leakages when a distributer's sensitive data has been leaked by trustworthy agents and also to possibly identify the agent(s) that leaked the data.**

Index Terms— **stegnography, L.S.B Algorithm, Watermarking**

I. INTRODUCTION

Data Leakage, put simply, is the unauthorized transmission of private or sensitive data or information from within an organization to a third party, i.e., an unauthorized recipient. Sensitive data in companies and organization include intellectual property (IP), financial information, personal information (like credit card data) and other information depending on the business and the industry.

In the real world scenario, a distributer needs to share sensitive data among various stakeholders such as employees, business partners and customers. This increases the risk that confidential information will fall into unauthorized hands, whether caused by force or by error (maliciously intended or an inadvertent mistake by an employee or a customer).

The problem of data leakage is much more relevant and crucial nowadays as much of our information is available online through social networking sites and third party aggregators. Social networking sites like LinkedIn, Facebook, Twitter etc along with their third party applications all use a part or whole of their users' personal information which, they promise to keep undisclosed and secure. Consider a situation where a user has given permission to three different apps to use his personal data, which he thinks is only a small part. However, somehow the safety of that part of the personal data was compromised, then it would be much help to develop techniques that would enable the parent site to identify the application responsible for this leakage, so as to protect the other users from identity theft, which is why data leakage detection algorithms are crucial in the present scenario. Fig. Detection and cost model architecture

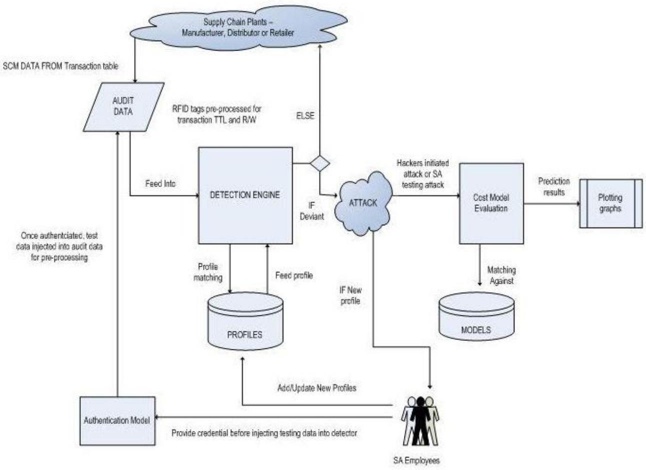
Supply Chain Plants – Manufacturer, Distributor or Retailer
SCM DATA FROM Transaction table
AUDIT DATA
RFID tags pre-processed for transaction TTL and R/W
ELSE
Feed Into
DETECTION ENGINE
IF Deviant
ATTACK
Hackers initiated attack or SA testing attack
Cost Model Evaluation
Prediction results
Plotting graphs
Once authentciated, test data injected into audit data for pre-processing
Profile matching
Feed profile
IF New profile
Matching Against
PROFILES
MODELS
Add/Update New Profiles
Authentication Model
Provide credential before injecting testing data into detector
SA Employees

Fig. Detection and cost model architecture

A. *Data Allocation Problem*

Our proposed work presented here is divided into two parts: the first being implementing and verifying the allocation strategies for sample data requests in a round-robin fashion (proposed by Papadimitriou and Garcia-Molina [3]); and the second part is developing three new techniques for data allocation and comparing the results with the already proposed technique. Our proposed techniques have an additional advantage that an agent is completely satisfied (i.e. it receives all the objects requested) before allocating any object to the next agent. Also, in this case, the data allocation to an agent is independent of agent's requests that come afterwards. Hence, the data allocation of this kind can be possibly extended to handle requests in case when the number of agents is known in advance, as the data allocation to an agent is independent of the agents that are yet to be allocated.

B. *Abbreviations and Acronyms*

L.S.B (Least Significant Bit)

P.H.R (Public Health Record)

II. EQUATIONS

Algorithm :

Algorithm to find the guilty agent:

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

b. 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

Manuscript received April 24, 2015.

Sneha S.Gulawani Computer Department, Pune University, Ahmednagar,India, 9922251217

Sayli B.Adsure, Computer Department,Pune University, Ahmednagar, India, 9028033353

Ashwini B.Gaikwad Computer Department, PuneUniversity, Ahemednagar, India,9766255829

fake data; he could further improve the chance of finding guilt agent.

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

d. Check for remaining agents. Distributor chooses the remaining agents to send the data. Distributor can increase the number of possible allocations by adding fake data.

e. Estimate the probability value for guilt agent. To compute this probability, we need an estimate for the probability that values can be "guessed" by the target.

Analysis of guilt model-

A model for formally defining the concept of guilt for the Data Leakage problem is discussed here. We can say that an agent Ai is guilty if it contributes one or more objects to the leaked data set. Let the event that agent Ai is guilty be Gi and the event that agent Ai is guilty for a given leaked set S be Gi|S. To compute the Pr{Gi|S}, we need an estimate of the probability that values in S can be "guessed" by the target. [3]. Pr{Gi|S} can be calculated as follows –

$Pr\{Gi \mid S\} = 1 - \pi\ d\ \text{€}\ S\ \pi\ A\ \ (1 - (1 - p)/\ |Vd|)$

Where Vd is the count for no. of agents that are requesting every element that is in the leaked set S. Pr(Gj| S = Ai) or simply Pr(Gj|Ai) is the probability that agent Aj is guilty if the distributor discovers a leaked table S that contains all Ri objects.

III. Existing system-

In many cases distributor must indeed work with agents that may not be trusted, and distributor may not be sure that a leaked object came from an agent or from some other source, since sure data cannot admit watermarks. In existing system there is few problem like fixed agents and existing system work comparable with agents whose request known in advance. Also with adding fake object original sensitive data cannot be alter and absences of agent guilt models that capture leakage scenarios and appropriate model for cases where agents can collude and identify fake tuples. Lastly system is not online capture of leak scenario also in existing system more focus on data allocation problem.

IV. Proposed system-

When trying to find appropriate solution we thought of developing a application which will not be trust oriented. All the agents will be tightly mapped and various techniques will be introducedto handle the document sharing. The documents will be marked for later verification. 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.

V. Related Work

The guilt detection approach we present is related to the data provenance problem: tracing the lineage of an subject implies essentially the detection of the guilty agents.[5]It provides a good overview on the research conducted in this field. Suggested solutions are domain specific, such as lineage tracing for data Warehouses, and assume some prior knowledge on the way a data view is created out of data sources. Our problem formulation with objects and sets is more general and simplifies lineage tracing, since we do not consider any data transformation from Ri sets to S.As far as the data allocation strategies are concerned, our work is mostly relevant to watermarking that is used as a means of establishing original ownership of distributed objects. Watermarks were initially used in images, video and audio data whose digital representation includes considerable redundant.

VI. Modules of Data Leakage Detection System

A. Data Allocation Module:

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 detecting a guilty agent, Admin can send the files to the authenticated user, users can edit their account details etc. Agent views the secret key details through mail. In order to increase the chances of detecting agents that leak data.

B. Fake Object Module:

The distributor creates and adds fake objects to the data that he distributes to agents. Fake objects are objects generated by the distributor in order to increase the chances of detecting agents that leak data.[5] The distributor may be able to add fake objects to the distributed data in order toimprove his effectiveness in detecting guilty agents. Our use of fake objects is inspired by the use of "trace" records in mailing lists. In case we give the wrong secret key to download the file, the duplicate file is opened, and that fake details also send the mail. Ex: The fake object details will display.

C. Optimization Module:

The Optimization Module is the distributor's data allocation to agents has one constraint and one objective. The agent's constraint is to satisfy distributor's requests, by providing them with the number of objects they request or with all available objects that satisfy their conditions. His objective is to be able to detect an agent who leaks any portion of his data. User can able to lock and unlock the files for secure.

D. Data Distributor Module:

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)

VII. Conclusion

In this work, we analyzed the likelihood that an agent may be responsible for any data leakage using several techniques. The algorithms were implemented using the four techniques, and in a real world, the distributor can use one of these depending on its needs. We observed that distributing data prudently may improve the chances of detecting the agents effectively especially when there is a large overlap in the data that agents must receive.

ACKNOWLEDGMENT

We would like to express our sincere thanks to our guide, Mr.R.S.Tambe, Assistant Professor at the Department of Computer Engineering for the support and motivation while carrying out the work presented here and also in writing this paper.

References

[1] "Data Leakage: The High Cost of Insider Threats." Worldwide White Paper.,http://www.cisco.com/en/US/solutions/collateral/ns170/ns896/ns895/white_paper_c11-506224.html., accessed January 2013

[2] "Facebook Data Leak: Should You Worry?" By Jeanine Skowronski Available:http://www.mainstreet.com/article/moneyinvesting/news/facebook-data-leak-should-you-worry, accessed January 2013

[3] Panagiotis Papadimitriou, *Member, Ieee, Hector Garcia-Molina, Member, IEEE.*, Data Leakage Detection, IEEE Transactions On Knowledge And Data Engineering, Vol. 23, No. 1, January 2011

[4] R. Agrawal and J. Kiernan, "Watermarking Relational Databases," Proc. 28th Int'l Conf. Very Large Data Bases (VLDB '02),

[5] P. Papadimitriou and H. Garcia-Molina, "Data Leakage Detection," technical report, Stanford Univ., 2008.

[6] N. P. Jagtap, S. J. Patil, A. K. Bhavsar, "Implementation of data watcher in data leakage detection system", International Journal of Computer &Technology Volume 3,No. 1, Aug, 2012.

[7] AnkitAgarwal, MayurGaikwad, KapilGarg, VahidInamdar, "Robust Data leakage and Email Filtering System", International Conference on Computing, Electronics and Electrical Technologies, 2012.

[8] Y. Cui and J. Widom (2001) 'Lineage Tracing For General Data Warehouse Transformations' -In the VLDB Journal,pp. 471– 480.

[9] PreetiPatil, NitinChavan, SrikanthaRao, S B Patil, "Building of a Secure Data Warehouse by Enhancing the ETL Processes for Data Leakage", Intl Conf& Workshop on Recent Trends in Technology,2012

[10] Dangwal et al. International Journal of Advanced Research in Computer Engineering & Technology, Volume 1, Issue 4, June 2012

[11] S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian. "Flexible support for multiple access control policies".