International Journal Of Advanced Research and Innovations Vol.1, Issue .1
ISSN Online: 2319 – 9253
Print: 2319 – 9245

A STUDY ON IMPLEMENTATIONS OF GUILT AGENT MODELS & ANALYSIS

B. KIRAN KUMAR [1] **M.KIRAN KUMAR [2]**

1. Asst.Prof, Vivekananda Institute of Technology &science, Karimnagar
2. Asst.Prof, Vivekananda Institute of Technology &science, Karimnagar

ABSTRACT

Data is nothing but collection of raw facts, those data facts are placed into pockets. Pockets are travelled over the network. During its transit data are damaged /hacked by malicious user's means data are leaked in its travelling. We mainly investigate on identifying these leakages. We propose the data allocation strategies 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.

Key words: data facts, leakage, allocation strategies, data transfers, guilt agents

I. INTRODUCTION

Data leakage is becoming more common throughout industry and government, leading to the development of soft ware and procedural techniques to detect and prevent such occurrences. Research and guidance on data leakage benefit agencies**[2]** .

The Tax Information Security Guidelines for Federal, State and Local Agencies and Entities, (IRS Pub 1075) states:

"The Internal Revenue Service (IRS) **[9]** is acutely aware that in fostering our system of taxation, the public must maintain a high degree of confidence that the personal and financial information furnished to us is protected against unauthorized use, inspection, or disclosure. The concerns of citizens and Congress regarding individual rights to privacy make it important that we continuously assess our disclosure practices and the safeguards we use to protect the confidential information entrusted to us. Those agencies or agents that receive Federal Tax Returns and Return Information (FTI) directly from either the IRS or from secondary sources must have adequate programs in place to protect the data received. The disclosure provisions of the Internal Revenue Code (IRC) make the confidential relationship between the taxpayer and the IRS quite clear. It also stresses the importance of this relationship by making it a crime to violate this confidence."**[9][2]**

Data Leakage is a potential risk to an agency or entity from the exposure and unauthorized disclosure of FTI data. Data leakage of FTI data from an agency puts the IRS at risk. Security industry analysis claims that more than 90% of the data leakages are unintentionally caused by employees. As a part of the Safeguards program, the recipients of FTI data should review their data handling procedures and implement the necessary processes to secure restricted information and eliminate potential data leakages.**[1][2]**

Data leakage is the transmission or exposure of data and information to an unauthorized or unintended recipient. The recipient may be internal to the agency, a known entity external to the agency, or an unknown entity external to the agency. The data leakage may not have been made intentionally or with any malicious intent. Regardless of whom has FTI access (employees, contractors, or vendor partners), or the form of the FTI data (digital media or printed format), protecting FTI data is important to the agency. Data leakage is often confused with data loss. Data loss implies that the data no longer exists or is corrupted beyond

ISSN Online: 2319 – 9253
Print: 2319 – 9245

use. When data leakage occurs, the data still exists and is unaffected.[3][9]

Most of the current security controls that are in place protect an agency's data from unintended intruders, both in the physical and digital environments. The NIST Recommended Security Controls for Federal Information Systems (NIST SP 800-53), the Confidentiality and Disclosure of Returns and Return Information (IRC 6103 (p)), and the Tax Information Security Guidelines for Federal, State and Local Agencies and Entities (Pub 1075), state what protections need to be in place. These documents address how to secure planned data transfers such as back-up tapes to contingency sites, and provide guidance for handling and storing data in printed format to prevent its exposure. However, additional controls must be required to prevent data leakage in digital media.

Each digital media form has a viable method for detecting data leakage. Most of these methods are based on the inspection of the data content within a document as opposed to the context of the data. This is analogous to opening an envelope and inspecting the content of a letter for restricted data. This process is called file cracking.

File cracking retrieves content that may be many layers deep, such as an embedded data table within a Word document that has been compressed or zipped. Data that is encrypted can be retrieved with enterprise recovery keys. For content that cannot be unencrypted and inspected, agencies should establish and enforce rules that can block or quarantine the data for further review.

The following are four common methods**[3][5]** of detecting unauthorized data: rules-based expression analysis, keyword filtering, exact data matching, and partial data matching.

- Rules-based expression analysis examines the content of a digital document for specific rules or patterns (e.g., a nine digit ID pattern (nnn-nn-nnnn) or monetary format ($nn, nnn.cc)) within the document or within a specified proximity of each other. This method is quick and effective at identifying well-defined data structures within the document. In addition, rules-based expression analysis is an effective method to identify potential data leakages for data in motion. As data moves to a gateway, it can be checked for restricted data and if identified, the data can be quarantined or blocked. Data in use can also be analyzed at the endpoints by detecting restricted information and preventing unauthorized copying onto hard drives, flash drives, and other portable media.
- Keyword filtering is similar to rules-based expression except a set of characters or words is used for comparison instead of searching for a data pattern. This is useful when identifying content associated with specific words or unique markers, such as 'Top Secret,' or inappropriate language. Key word filtering quickly reviews all three digital media forms as well as restricted data sets that may be constantly changing.
- A selected set of data can also be identified using exact data matching or database fingerprinting. Fingerprinting compares an existing data set against data from the restricted content to determine if there is an exact match. If restricted data is identified in the content, then that content will be further examined while content with acceptable data will be transmitted or processed. If only a subset of the data is restricted, only that data subset needs to be used for comparison. While this method is accurate and thorough, it can be slow. Fingerprinting works well with imagery such as photos, videos, and PDFs. Because of the latency period, this method is best for data at rest and data in use; however, fingerprinting may also be used with data in motion if a transmission delay is acceptable.
- Another method that compares restricted data to content data is partial data matching. This method can also be used to look for complete data matches in the content. The restricted data is split into small sections and stored. A similar size portion of the content is then

ISSN Online: 2319 – 9253
Print: 2319 – 9245

hashed. The content is then offset by a few characters and again split into small sections. Both hashed results are compared to determine if they match. This method works best when the number of restricted documents is limited or similar content or phrases are found across several restricted documents. This method can be used to assess all three digital media forms, though latency issues may occur with data in motion.

Regardless of the method used to identify restricted data such as FTI, data may not always need to be blocked or quarantined. An Access Control list, containing a white list and a black list, could be created and used for recipients with access privileges, and checked prior to allowing the restricted data to be sent or stored. Encryption of data files and emails can be used to prevent unauthorized recipients from receiving restricted data. If a recipient does not have the authority to view or access restricted data, then they should not have the encryption key. The Access Control list must be maintained and updated by regularly distributing new encryption keys. This prevents the transmission of restricted data to an incorrect or unknowing recipient. Similarly, protective markings can be placed on restricted data so that recipients would have to have the same level of markings as the data before it can be accessed. This also prevents the data from being stored in unauthorized file locations or databases.

As a part of the Safeguards program, agencies must consider procedures for preventing data leakage. Regardless of the approach used to prevent data leakage, all mitigation strategies should be based on identifying a distinctive data element or sequence that is unique from other data, and comparing that element within the data environment. However, when trying to protect FTI data, it is not always the data that is unique but rather the source of the data. For example, the agency could develop procedures to look for Social Security numbers (SSN) in e-mails and other correspondence, or the storing of SSNs and total income in data storage files, but these FTI data elements could be the exact same data elements that the agency has collected. The source of the data is not readily identifiable from the data elements. To identify the source of the data as FTI, agencies could create tags or unique markers that are entered into the data or augments the data. As soon as the data is received, it should be processed to add clearly identifiable unique markers that distinguish the source as FTI. Data labels should be used to implement mandatory access controls, restricting access to authorized users.

For databases and spreadsheets, unique markers can alter data file names, the column headers in the data files, and the data elements in the data files. All files could be renamed to start with the clear distinction of "FTI_". For example, a file named "RETURNS_2008" would be renamed "FTI_RETURNS_2008". Similarly, within the data files, the column headers could be renamed to show the source as FTI, such as "SSN" and "TOTAL_INCOME" being renamed to "FTI_SSN" and "FTI_TOTAL_INCOME". This could even be implemented on the data elements that are non-computational. For example, SSNs and ADDRESSES would be augmented to "FTI_123456789" and "FTI_123 Main ST".

At all times, unique markers should be used to identify any type of FTI data that is received from the IRS. Unique markers could be added to text document filenames identifying the source of the data as FTI. Likewise, the file could be renamed to start with the "FTI_" distinction. Headers and footers of documents could be augmented to include "FTI" notations. This is similar to classification markers such as SECRET, or SENSITIVE BUT UNCLASSIFIED on documentation. This process could also be used for presentation documents such as PowerPoint slides.

I. RELATED WORK

The guilt detection approach **[1]** we present is related to the data provenance problem: tracing the lineage of an S object implies essentially the detection of the guilty agents. It

ISSN Online: 2319 – 9253
Print: 2319 – 9245

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 redundancy. **[4][5]**

II. EXISTING APPROACHES

Leakage detection is handled by watermarking; 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 were initially used in images, video and audio data whose digital representation includes considerable redundancy. Watermarking**[4]** aims to identify a data owner and, hence, is subject to attacks where a pirate claims ownership of the data or weakens a merchant's claims. 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. However, in some cases it is important not to alter the original distributor's data. 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.

III. PROPOSED SCHEME

- After giving a set of objects to agents, the distributor discovers some of those same objects in an unauthorized place.[4]
- 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.
- If the distributor sees "enough evidence" that an agent leaked data, he may stop doing business with him, or may initiate legal proceedings.[6]
- 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.
- 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[2][3].

V. APPROACHES

1) Login / Registration:

This is mainly designed to provide the authority to a user in order to access the other modules of the project. Here a user can have the accessibility authority after the registration.

ISSN Online: 2319 – 9253
Print: 2319 – 9245

2) DATA TRANSFER:

This is mainly designed to transfer data from distributor to agents. The same module can also be used for illegal data transfer from authorized to agents to other agents

3) GUILT MODEL ANALYSIS:

This module is designed using the agent – guilt model. Here a count value(also called as fake objects) are incremented for any transfer of data occurrence when agent transfers data. Fake objects are stored in database.

4)AGENT-GUILT MODEL:

This module is mainly designed for determining fake agents. This module uses fake objects (which is stored in database from guilt model module) and determines the guilt agent along with the probability. A graph is used to plot the probability distribution of data which is leaked by fake agents

VI. DATA ALLOCATION

The two types of requests we handle: sample and explicit [5]. 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. Fake objects are represented using four problem instances with the names EF, EF, SF and SF. Where E stands for explicit requests, S for sample requests, F for the use of fake objects [2] and F for the case where fake objects are not allowed.

Sample request Ri = SAMPLE (T;mi): Any subset of mi records from T can be given to Ui.
Explicit request Ri = EXPLICIT (T; condi): Agent Ui receives all the T objects that satisfy condition.

We represent our four problem instances with the names EF, EF, SF and SF, 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. For simplicity we are assuming that in the E problem instances, all agents make explicit requests, while in the S instances, all agents make sample requests. Our results can be extended to handle mixed cases, with some explicit and some sample requests.

VII. 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[8] . 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. In this case, perturbing the set of distributor objects by adding fake elements is done. In some applications, fake objects may cause fewer problems that perturbing real objects. For example, say the distributed data objects are medical records and the agents are hospitals. In this case, even small modifications to the records of actual patients may be undesirable. However, the addition of some fake medical records may be acceptable, since no patient matches these records, and hence no one will ever be treated based on fake records. 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.

VIII. CONCLUSION AND ENHANCEMENT

The data distribution strategies improve the distributor's chances of identifying a leaker. It has been shown that distributing objects

judiciously can make a significant difference in identifying guilty agents[1], especially in cases where there is large overlap in the data that agents must receive. In some cases "realistic but fake" data records are injected to improve the chances of detecting leakage and identifying the guilty party. In future the extension of our allocation strategies can handle agent requests in an online fashion can be implemented, an agent is responsible for a leak is assessed, 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. The algorithms we have presented implement a variety of data distribution strategies that can improve the distributor's chances of identifying a leaker[2]. 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 that agents must receive.

REFERENCES

[1] Panagiotis Papadimitriou, Hector Garcia-Molina (2010)'Data Leakage Detection', IEEE Transactions on knowledge and data engineering, Vol.22, No.3.

[2] P. Papadimitriou and H. Garcia-Molina, "Data Leakage Detec- tion," technical report, Stanford Univ., 2008.

[3] R. Agrawal and J. Kiernan (2002) 'Watermarking relational databases', In VLDB: Proceedings of the 28th international conference on Very Large Data Bases, pp. 155– 166.

[4] P. Bonatti, S. D. C. di Vimercati, and P.Samarathi (2002)'an Algebra for Compose Access Control Policies'- ACM Trans. Inf.Syst. Secure., Vol.5, No.1, pp. 1– 35

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

[6] RAgrawal and J. Kiernan, "Watermarking RelationalDatabases," Proc. 28th Int'l Conf. Very Large Data Bases (VLDB '02), VLDB Endowment, pp. 155-166, 2002.

[7] RichardWang, Stuart E. Madnick. A polygen model For heterogeneous database systems: the source tagging Perspective// Proceedings of the 16th International Conference on Very Large Data Bases, Brisbane, Queensland, Australia, February 5-9, 1990, 16:519-538.

8] D. P. Lanter. Design of a lineage-based meta-data base For GIS. Cartography and Geographic Information Systems, 1991, 18:255-261.

[9] www.irs.gov/privacy/article/0,,id=201295,00.html Preventing Data Leakage Safeguards Technical