ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2014

Data Leakage Identification and Blocking Fake Agents Using Pattern Discovery Algorithm

Karthik.R[1], Ramkumar.S[2], Sundaram.K[3]

Assistant Professor, Department of computer Science Engineering, Karpagam University, Tamilnadu, India[1]

Research Scholar, Department of computer Science, Karpagam University, Coimbatore, Tamilnadu, India[2]

M.E, Department of computer Science Engineering, Karpagam University, Tamilnadu, India[3]

ABSTRACT: A data distributor has given sensitive data to a set of supposedly trusted agents (third parties). If the data distributed to third parties is found in a public/private domain then finding the guilty party is a nontrivial task to distributor. Traditionally, this leakage of data is handled by water marking technique which requires modification of data. If the watermarked copy is found at some unauthorized site then distributor can claim his ownership. To overcome the disadvantages of using watermark, data allocation strategies are used to improve the probability of identifying guilty third parties. In this project, we implement and analyse a guilt model that detects the agents using a protocol. The guilty agent is one who leaks a portion of distributed data. The idea is to distribute the data intelligently to agents based on data request and explicit data request in order to improve the chance of detecting the guilty agents. The algorithms implemented using fake objects will improve the distributor chance of detecting guilty agents. It is observed that by minimizing the sum objective the chance of detecting guilty agents will increase. We also developed a framework for generating fake objects. Our goal is to detect when the distributor's sensitive data have been leaked by agents, and if possible to identify the agent that leaked the data.

KEYWORDS: Web mining, Cookies, Session**.**

I. INTRODUCTION

Data Leakage is synonymous with the term Information Leakage. The reader is encouraged to be mindful that unauthorized does not automatically mean intentional or malicious. Unintentional or inadvertent data leakage is also unauthorized. The scope for data leakage is very wide, and not limited to just email and web. We are all too familiar with stories of data loss from laptop theft, hacker break-ins, and backup tapes being lost or stolen, and so on. How can we defend ourselves against the growing threat of data leakage attacks via messaging, social engineering, malicious hackers, and more? Many manufacturers have products to help reduce electronic data leakage, but do not address other vectors. This paper aims to provide a holistic discussion on data leakage detection and its prevention, and serve as a starting point for businesses. Motivations are varied, but include reasons such as corporate espionage, financial reward, or a grievance with their employer. The latter appears to be the most likely. Our goal is to detect when the distributor's sensitive data have been leaked by agents, and if possible to identify the agent that leaked the data. The main objective is to detect an agent who leaks any portion of the owner's data. Maximize the chances of detecting a guilty agent that leaks all his data objects.

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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2014

II. LITERATURE REVIEW

Many efficient data leakage detection method had been developed in the last decades, some of the prominent studies are given below. Shreyta Raj discuss a formal method for representing and detecting inconsistencies of combined secrecy models is to detect when the PC distributor's sensitive data has been leaked by their agents, and if possible to identify the agent that leaked the data. Data leakage is a silent type of threat. This sensitive information can be electronically distributed via e-mail, Web sites, FTP, instant messaging, spreadsheets, databases, and any other electronic means available – all without your knowledge. Data allocation strategies (across the agents) are proposed that improve the probability of identifying leakages. These methods do not rely on alterations of the released data (e.g., watermarks). In some cases the distributor can also inject "realistic but fake" data records to further improve our chances of detecting leakage and identifying the guilty party. A model for assessing the "guilt" of agents using C# dot net technologies with MS sql server as backend is proposed to develop. Algorithms for distributing objects to agents, in a way that improves our chances of identifying a leaker is aloes presented. Finally, the option of adding "fake" objects to the distributed set is also considered. Such objects do not correspond to real entities but appear [1]. Sandip A. Kale describes the results of implementation of Data Leakage Detection Model. Currently watermarking technology is being used for the data protection. But this technology doesn't provide the complete security against date leak age. This paper includes the difference between the watermarking & data leakage detection model's technology. This paper leads for the new technique of research for secured data transmission & detection, if it gets leaked [2]. S.Ramkumar et al, investigate and utilized the characteristic of the group movement of objects to explore the group relationship and tracking them. The goal is to efficiently mine the group movement activity using clustering and sequential pattern mining. Clustering was applied to find both groups of similar teams and similar individual members. Sequential pattern mining was used to extract sequences of frequent events. To enable the continuous monitoring the group object movement, the system introduces a special technique called minor clustering and Cluster Ensembling algorithm. Several solutions on route were implemented, but those methods were energy consumed. In order to reduce the energy the proposed system used data mining methods to effectively handle the group movement of objects [3]. Chandni Bhatt et al," study a data distributor has given sensitive data to a set of supposedly trusted agents. Sometimes data is leaked and found in unauthorized place e.g., on the web or on somebody's laptop. 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 might be given to various other companies. The owner of the data is called as distributors and the trusted third parties are called as agents. Data leakage happens every day when confidential business information such as customer or patient data, company secrets, budget information etc are leaked out. When this information is leaked out, then the companies are at serious risk. Most probably data are being leaked from agent's side. So, company has to very careful while distributing such a data to agents. The Goal of Our project is to analyse "how the distributer can allocate the confidential data to the Agents so that the leakage of data would be minimized to a Greater Extent by finding a guilty agent" [4]. Amol O. Gharpande et al, gives review idea about data leakage detection techniques. 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 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 [5].

III. METHODOLOGY

Sensitive data must be handed over to supposedly trusted third parties. We consider applications where the original sensitive data cannot be perturbed. Perturbation is a very useful technique where the data are modified and made "less sensitive" before being handed to agents. 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 [6]-[15] is illustrated in Fig.1.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2014

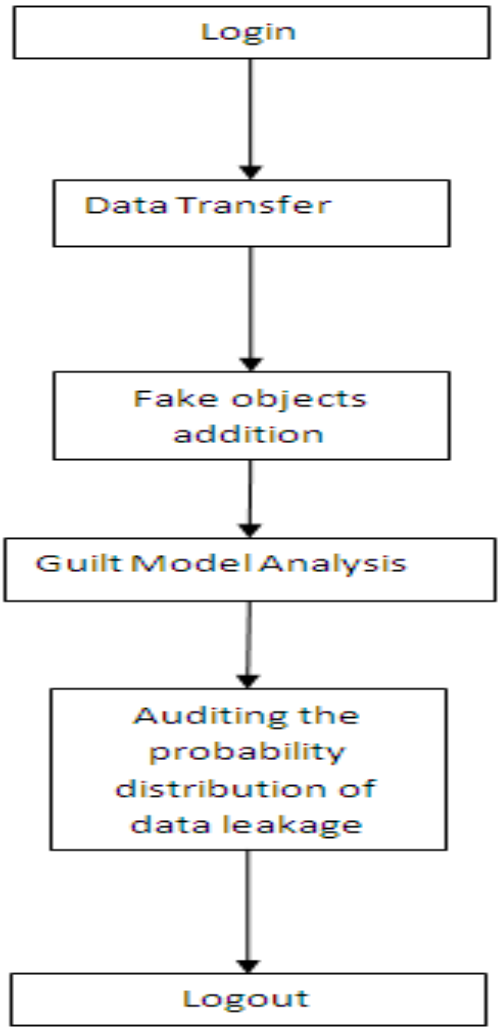

Fig.1 Process Design

Data Leakage detection Protocol- the proposed protocol will be appends on every data transmission, the protocol carries all details about the agent as well as the data. When the data leaks, the protocol identifies the guilty agent immediately.

Pattern Discovery- pattern discovery used to create fake objects. Agents that can receive fake objects.

IV. PROPOSED ALGORITHM

Data leakage detection and pattern discover algorithms used in this study are shown in the Fig.2. Fig.2(a) explains the data request from sender to receiver and agent receiving the data and fake agent detection using proposed algorithms. The main focus of our work 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 is shown in Fig.2 (b) to Fig.2 (e).

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2014

Algorithm Steps

Step: 1Distributor select agent to send data. The distributor selects two agents and gives requested data R1, R2 to both agents.

Step: 2 Distributor creates fake object and allocates it to the agent. The distributor can create one fake object (B = 1)and both agents can receive one fake object (b1 = b2 = 1). If the distributor is able to create more fake objects, he could further improve the objective.

Step: 3 check number of agents, who have already received data Distributor, checks the number of agents, who have already received data.

Step: 4 Check for remaining agents Distributor chooses the remaining agents to send the data. Distributor can increase the number of possible allocations by adding fake object.

Step: 5Select fake object again to allocate for remaining agents. Distributor chooses the random fake object to allocate for the remaining agents.

Step:6 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

Algorithm 1. Allocation for Explicit Data Requests (EF)
Input: $R_1, \ldots, R_n$, $cond_1, \ldots, cond_n$, $b_1, \ldots, b_n$, B
Output: $R_1, \ldots, R_n$, $F_1, \ldots, F_n$

```
1:  R ← ∅                          ▷ Agents that can receive fake objects
2:  for i = 1, ..., n do
3:      if b_i > 0 then
4:          R ← R ∪ {i}
5:      F_i ← ∅
6:  while B > 0 do
7:      i ← SELECTAGENT(R, R_1, ..., R_n)
8:      f ← CREATEFAKEOBJECT(R_i, F_i, cond_i)
9:      R_i ← R_i ∪ {f}
10:     F_i ← F_i ∪ {f}
11:     b_i ← b_i − 1
12:     if b_i = 0 then
13:         R ← R\{R_i}
14:     B ← B − 1
```

(a)

Algorithm 2. Agent Selection for e-random

```
1: function SELECTAGENT (R, R_1, ..., R_n)
2:     i ← select at random an agent from R
3: return i
```

(b)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2014

Algorithm 3. Agent Selection for e-optimal

1: **function** SELECTAGENT $(\mathbf{R}, R_1, \ldots, R_n)$

2: $i \leftarrow \underset{i':R_{i'} \in \mathbf{R}}{\operatorname{argmax}} \left(\frac{1}{|R_{i'}|} - \frac{1}{|R_{i'}|+1} \right) \sum_j |R_{i'} \cap R_j|$

3: **return** i

(c)

Algorithm 4. Allocation for Sample Data Requests $(\overline{SF})$

Input: $m_1, \ldots, m_{n'}$ $|T|$ ▷ Assuming $m_i \leq |T|$

Output: $R_1, \ldots, R_n$

1: $a \leftarrow \mathbf{0}_{|T|}$ ▷ $a[k]$:number of agents who have received object t_k

2: $R_1 \leftarrow \emptyset, \ldots, R_n \leftarrow \emptyset$

3: $remaining \leftarrow \sum_{i=1}^{n} m_i$

4: **while** $remaining > 0$ **do**

5: **for all** $i = 1, \ldots, n : |R_i| < m_i$ **do**

6: $k \leftarrow$ SELECTOBJECT(i, R_i) ▷ May also use additional parameters

7: $R_i \leftarrow R_i \cup \{t_k\}$

8: $a[k] \leftarrow a[k] + 1$

9: $remaining \leftarrow remaining - 1$

(d)

Algorithm 5. Object Selection for s-random

1: **function** SELECTOBJECT(i, R_i)

2: $k \leftarrow$ select at random an element from set $\{k' \mid t_{k'} \notin R_i\}$

3: **return** k

(e)

Fig.2. (a), (b), (c), (d), (e) Data leakage detection and pattern discover algorithm

V. RESULT

Previous system concentrates on the watermarking technique, which identifies data leakage. To overcome the drawbacks the proposed paper represents invisible watermarking technique. It may not be certain if a leaked objects came from an agent or from some other source, since certain data cannot admit watermarks. The proposed work will give an effective identification of data leakage, and the agent who leaked the data. Where the paper concentrates on invisible watermarked notations to identify the guilty agent, but the proposed techniques concentrates on identifying and blocking the agents by gathering all histories of data transfer through the special protocol. Various modules created for identifying the guilt agent and probability of guilty agent accessing is shown in Fig.3 to Fig.6. This study has been

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2014

implemented and evaluated in .Net framework by creating a client server application on the network. The flow of the methodology implementation has been described in the following diagrams. 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. The distributor may be able to add fake objects to the distributed data in order to improve 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 is shown in Fig.3.

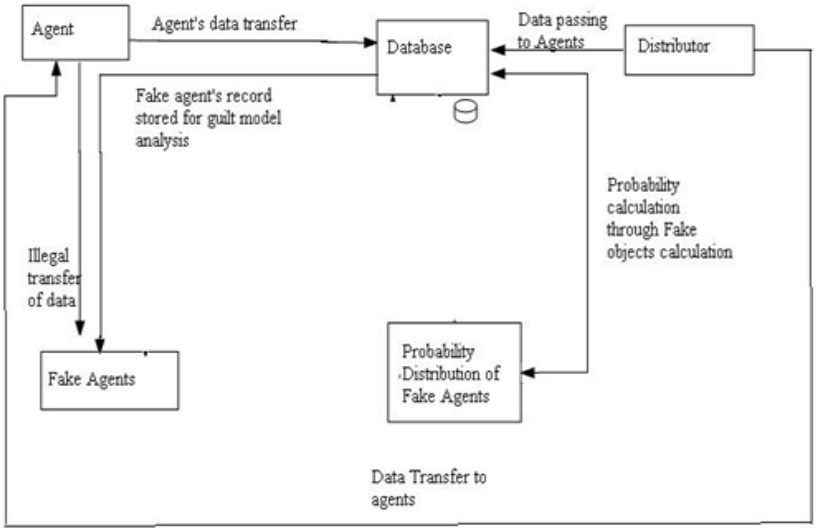

Fig.3 Design Architecture

Fig.4 and Fig.5 is mainly designed for determining fake agents. This design 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.

Fig.4 Agent Authentication

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2014

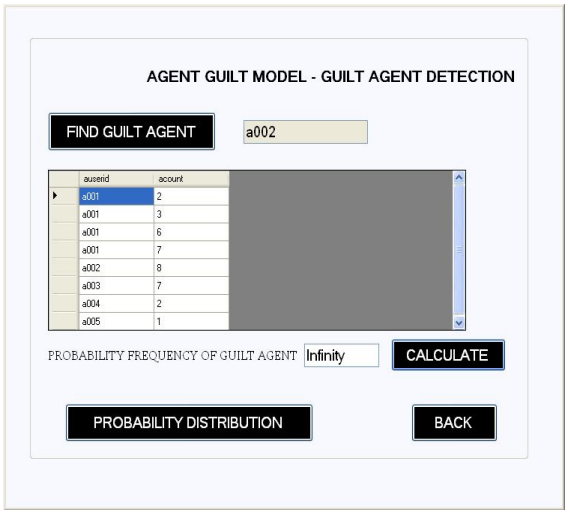

Fig.5 Agent Guilty Detection Module

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. 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 Admin can able to view the which file is leaking and fake user's details also is explained in Fig.6.

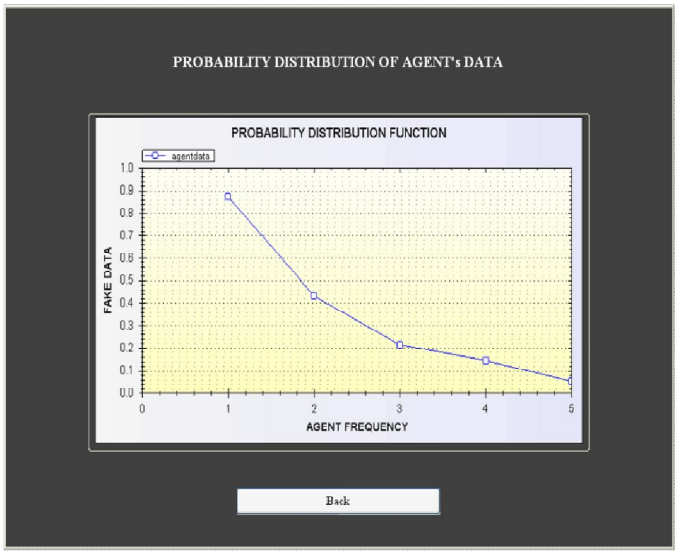

Fig.6 Probability Distribution of Agent Data

VI. CONCLUSION AND FUTURE WORK

In a perfect world 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, since

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2014

certain data cannot admit watermarks 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 that agents must receive. Our future work includes the investigation of agent guilt models that capture leakage scenarios that are not studied in this paper. For example, what is the appropriate model for cases where agents can collude and identify fake tuples? A preliminary discussion of such a model is available. Another open problem is the extension of our allocation strategies so that they can handle agent requests in an online fashion (the presented strategies assume that there is a fixed set of agents with requests known in advance.

REFERENCES

1. Shreyta Raj, Dr. Ravinder Purwar, Ashutosh Dangwal," A Model for identifying Guilty Agents in Data Transmission", International Journal of Advanced Research in Computer Engineering & Technology, Vol.1,pp.709-713, Jun2012.
2. Sandip A. Kale, S.V.Kulkarni," Data Leakage Detection", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 1, pp.668-678, Nov 2012.
3. Ramkumar.S, Elakkiya.A, Emayavaramban.G," Data Transfer Model - Tracking and Identification of Data Files Using Clustering Algorithms", IJLTEMAS, Volume III, pp.13-21, Aug2014.
4. Chandni Bhatt, Richa Sharma," Data Leakage Detection, International Journal of Computer Science and Information Technologies, Vol. 5, pp. 2556-2558, 2014.
5. Amol O. Gharpande ,V. M. Deshmukh, "Data Leakage Detection", International Journal of Computer Science and Applications Vol. 6, pp.216-219, Apr 2013.
6. R. Agrawal and J. Kiernan.,"Watermarking relational databases", International conference on Very Large Data Bases, pp. 155–166, 2002.
7. P. Bonatti, S. D. C. di Vimercati, and P. Samarati, "An algebra for composing access control policies", ACM Trans. Inf. Syst. Secur.,5, pp.1–35, 2002.
8. P. Buneman, S. Khanna, and W. C. Tan, "Why and where: A characterization of data provenance", pp.316–330, Springer, 2001.P. Buneman and W.-C. Tan, "Provenance in databases", international conference on Management of data, pp. 1171–1173, 2007
9. Y. Cui and J. Widom, "Lineage tracing for general data warehouse transformations", In The VLDB Journal, pp. 471–480, 2001.
10. S. Czerwinski, R. Fromm, and T. Hodes," Digital music distribution and audio watermarking".
11. F. Guo, J. Wang, Z. Zhang, X. Ye, and D. Li, "Information Security Applications", An Improved Algorithm to Watermark Numeric Relational Data, pp.138–149, 2006.
12. F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video", . Journal of Signal Processing",, Vol.66, pp.283–301, 1998.
13. S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian, " Flexible support for multiple access control policies", ACM Trans. Database Syst., vol.26, pp.214–260, 2001.
14. Y. Li, V. Swarup, and S. Jajodia., "Fingerprinting relational databases: Schemes and specialties", IEEE Transactions on Dependable and Secure Computing, vol.2, pp.34–45, 2005.