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

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 7, July 2019

Dynamic Data Leakage Using Guilty Agent Detection over Cloud

Kaveshwari R Wadile[1], Nilesh Choudhary[2]

M. Tech Student, Dept. of Computer Engineering, GF's Godavari of Engineering, Jalgaon, Maharashtra, India[1]

Assistant Professor, Dept. of Computer Engineering, GF's Godavari of Engineering, Jalgaon, Maharashtra, India[2]

ABSTRACT -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.

KEYWORDS: Distributor, Guilty Agent, Third Party, Sensitive Data, Alert.

I. INTRODUCTION

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 [18]. 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. 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 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. 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 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..

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

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 7, July 2019

II. LITERATURE SURVEY

2. 1.1 AGENT GUILT MODEL:

Suppose an agent Ui is guilty if it contributes one or more objects to the target. The event that agent Ui is guilty for a given leaked set S dies noted by G i| S. The next step is to estimate Pr {Gi| S }, i.e., the probability that agent Gi is guilty given evidence S. To compute the Pr {Gi| S}, estimate the probability that values in Sbcean "guessed" by the target. For instance, say some of the objects in t are emails of individuals. Conduct an experiment and ask a person to find the email of say 100 individuals, the person may only discover say 20, leading to an estimate of 0.2. Call this estimate as pt, the probability that object t can be guessed by the target. The two assumptions regarding the relationship among the various leakage events.

Assumption 1: For all t, t ∈ S such that ≠t T the
provenance of t is independent of the provenance of T.

The term provenance in this assumption statement refers to the source of a value t that appears in the leaked set. The source can be any of the agents who have t in their sets or the target itself.

Assumption 2: An object t ∈ S can only be obtained by the target in one of two ways.

- A single agent Ui leaked t from its own Ri set, or
- The target guessed (or obtained through other means) without the help of any of the n agents.

To find the probability that an agent Ui is guilty given a set S, consider the target guessed t1 with
probability p and that agent leaks t1 to S the with
probability 1-p. First compute the probability that he leaks a single object t to S. To compute this, define the set of agents Vt= {Ui| t<-Rt} that have t in their data sets. Then using Assumption 2 and known probability p,
We have,
Pr {some agent leaked t to S} = 1- p (1.1) Assuming that all agents that belong to Vt can leak t to S with equal probability and using Assumption 2 obtain,
Pr {Ui leaked t to S} = (1.2)
Given that agent Ui is guilty if he leaks at least one value to S, with Assumption 1 and Equation 1.2 compute the probability Pr { Gr| S}, agent Ui is guilty,
Pr {Gi| S}

2.1.2 DATA ALLOCATION PROBLEM:

The distributor "intelligently" gives
data to agents in order to improve the chances of detecting a guilty agent. There are four instances of this problem, depending on the type of data requests made by agents and whether "fake objects" [4] are allowed.
Agent makes two types of requests, called sample and explicit. Based on the requests the fakes objects are added to data list.
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.

2.1.3OPTIMIZATION PROBLEM:

The distributor's data allocation to agents has one constraint and one objective. The distributor's constraint is to satisfy agents' 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.
We consider the constraint as strict. The distributor may not deny serving an agent request
and may not provide agents with different perturbed versions of the same objects. The fake object distribution as the only possible constraint relaxation.
The objective is to maximize the chances of detecting a guilty agent that leaks all his data objects.
probability that agent Ui is guilty if the distributor the discovers a leaked table S that contains all Ri objects.

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

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 7, July 2019

The difference functions Δ (i, j) is defined as:
Δ (The Pr {Gi |S =Ri } or simply Pr {Gi |Ri } is i, j) = P r {Gi |Ri } – Pr {G |Ri } (1.4)
i. *Problem Definition*
Let the distributor have data requests from n
agents. The distributor wants to give tables
R1, .Rn. to agents, U1 . . . , Un
respectively, so that
• Distribution satisfies agents' requests; and
• Maximizes the guilt probability differences Δ (i, j) for
all i, j = 1. . . n and i= j.
Assuming that the sets satisfy the agents'
requests, we can express the problem as a multi criterion
ii. Optimization Problem
Maximize (. . . , Δ (i, j), . . .) i! = j
(Over R1,….., Rn,)
The approximation [3] of objective of the above equation does not depend on agent's probabilities and therefore minimize the relative overlap among the agents as
Minimize (. . . ,(|Ri∩Rj|) / Ri , . . .) i != j (1 .6)
(over R1 , . . . ,Rn)
This approximation is valid if minimizing the
relative overlap, (|Ri∩Rj|) / Ri maximizes Δ (i, j).

III. ALLOCATION STRATEGIES ALGORITHM

There are two types of strategies algorithms

3.1.1ALLOCATION FOR EXPLICIT DATA REQUEST(EF) WITH FAKE OBJECTS:

Algorithm 1. Allocation for Explicit Data Requests (EF)
Input: R1; . . .;Rn, cond1; . . . ; condn, b1; . . . ; bn, B
Output: R1; . . .;Rn, F1; . . . ; Fn
1: R□Ø Agents that can receive fake objects
2: for i= 1; . . . ; n do
3: if bi > 0 then
4: R⟵ R ∪ {i}
5: Fi ⟵Ø
6: while B > 0 do
7: i⟵SELECTAGENT(R;R1; . . .;Rn)
8: f ⟵CREATEFAKEOBJECT(Ri,Fi,condi)
9: Ri ⟵Ri ∪ {f}
10: Fi ⟵Fi ∪{f}
11: bi ⟵bi - 1
12: if bi = 0then
13: R⟵ R\{Ri}
14: B⟵ B - 1
Input : Employee request an article within a particular category i.e., with an Explicit condition.
Output: Employee receive the requested articled with the fake object water marked in the document.

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

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 7, July 2019

3.1.2ALLOCATION FOR SAMPLE DATA REQUEST(EF) WITHOUT ANY FAKE OBJECTS:

Algorithm 4. Allocation for Sample Data Requests (SF) ss

Input: m1; . . .;mn, |T| Assuming mi ≤|T|

Output: R1, . . .,Rn

1: a ←0|T| a[k]:number of agents who have
received object tk
2: R1 ;←Ø, . . .,Rn←Ø ;
3: remaining←Σni=1 mi
4: while remaining > 0 do
5: for all i=1,. . . ,n : [Ri] <mi do
6: k ←SELECTOBJECT(I,Ri) May also use
additional parameters
7: Ri ←Ri ∪ {tk}
8: a[k]←a[k]+1
9: remaining← remaining - 1

Input : Employee can request any kind of article independent of category ie., a sample request.

Output: Employee receive the requested articled without any fake object water marked in the document.

IV. EXISTING SYSTEM

There are conventional techniques being used and include technical and fundamental analysis. The main issue with these techniques is that they are manual and need laborious work along with experience. 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 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[4].

We call the owner of the data the distributor and the supposedly trusted third parties the agents. The distributor gives the data to the agents. These data will be watermarked. Watermarking is the process of embedding the name or information regarding the company. The examples include the pictures we have seen in the internet. The authors of the pictures are watermarked within it. If anyone tries to copy the picture or data the watermark will be present. And thus the data may be unusable by the leakers.

4.1.1 DISADVANTAGE:

This data is vulnerable to attacks. There are several techniques by which the watermark can be removed Thus the data will be vulnerable to attacks.

V. PROPOSED SYSTEM

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. We also present algorithm for distributing object to agent. 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.

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

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 7, July 2019

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. Today the advancement in technology made the watermarking system a simple technique of data authorization. There are various software which can remove the watermark from the data and makes the data as original.

VI. IMPLEMENTING THE PROJECT

ANALYSIS: The implemented website provide content to the users, and also allows users to contribute to the site in several ways, such as Discussion Board. Also the site provides different content to different users. All of these features rely on our site being able identify its users in some way, to prevent another person from using a particular user's account. For all this work, user accounts will need to be created and maintained and users will need to be correctly identified by their accounts.

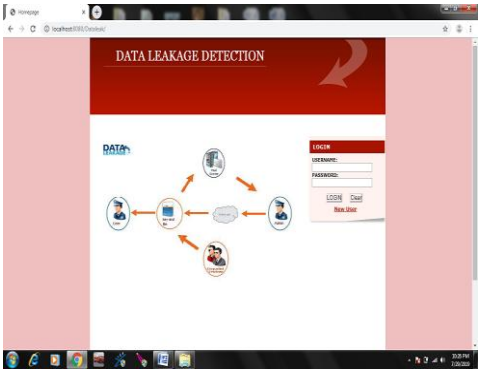

Figure1: Login Page

Figure2: Login for Distributor

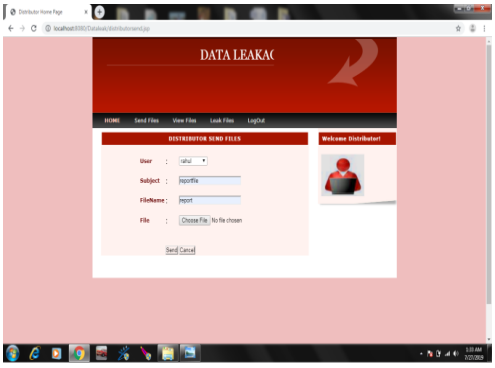

Figure5: Distributor send file to the Agent

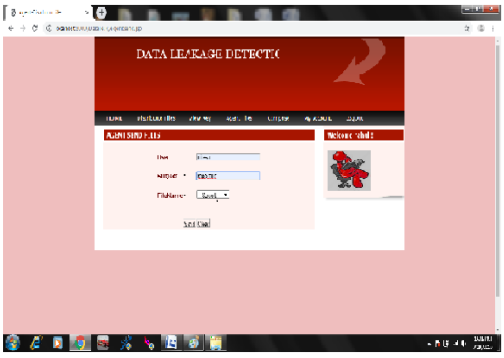

Figure 6 : Agent to Agent file send

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

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 7, July 2019

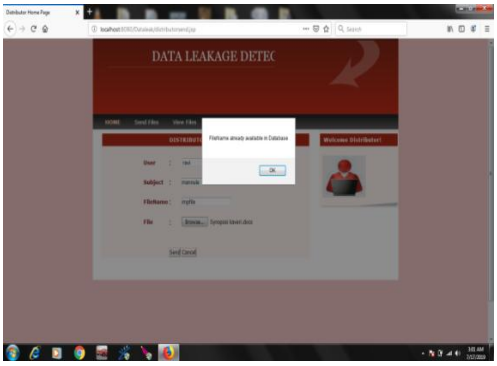

Figure7 : Already file send generate alert message

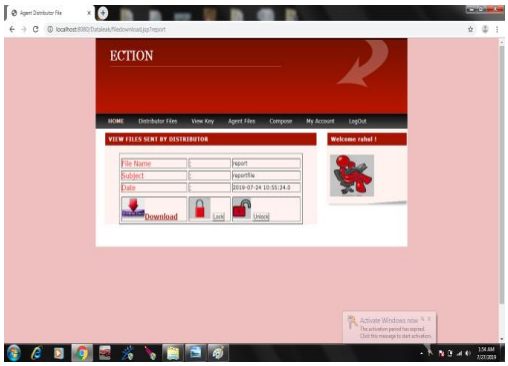

Figure7 :File lock and unlock

VII. CONCLUSIONS

Thus, these modules successfully work according to IEEE paper. It can successfully login distributor to the system and register the new agent request and show confirmation message for registration. In our work the distributor can check the list of registration request for new agent and the agent and distributor also updates its information successfully. In this project in next modules we can implement following idea: User ID and password send to the agent for login to system. The agent should send data request to the distributor and distributor check the request and send data to the agent by adding fake object in data allocation module. In agent guilt module we can check send alert message to the distributor when the agent has share any confidential data. This is main goal of our project.

REFERENCES

[1] R. Agrawal and J. Kiernan, "Watermarking Relational Databases,"Proc. 28th Int'l Conf. Very Large Data Bases (VLDB '02), VLDBEndowment, pp. 155-166, 2002. IEEE Transaction and knowledge and data engineering, Vol.23, No.1, January 2011.
[2] P. Bonatti, S.D.C. di Vimercati, and P. Samarati, "An Algebra for Composing Access Control Policies," ACM Trans. Information and System Security, vol. 5, no. 1, pp. 1-35, 2002.
[3] P. Buneman, S. Khanna, and W.C. Tan, "Why and Where: Characterization of Data Provenance," Proc. Eighth Int'l Conf. Database Theory (ICDT '01), J.V. den Bussche and V. Vianu, eds., pp. 316-330, Jan. 2001.
[4] P. Buneman and W.-C.Tan, "Provenance in Databases," Proc.ACM SIGMOD, pp. 1171-1173, 2007.
[5] Ms. Aishwarya Potdar1, Ms. Rutuja Phalke2, Ms. Monica Adsul3, Ms.Prachi Gholap4
B.E, Department of Computer Engineering, KJCOEMR, Pune University, Pune, India, International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 4, April 2013.
[6] 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.
[7] P. Bonatti, S. D. C. di Vimercati, and P. Samarati. An algebra for composing access control policies. ACM Trans. Inf. Syst. Secur., 5(1):1–35, 2002.
[8] P. Buneman, S. Khanna, and W. C. Tan. Why and where: A characterization of data provenance. In J. V. den Bussche and V. Vianu, editors, Database Theory - ICDT 2001, 8th International Conference, London, UK, January 4-6, 2001, Proceedings, volume 1973
of Lecture Notes in Computer Science, pages 316–330. Springer, 2001.
[9] P. Buneman and W.-C. Tan. Provenance in databases. In SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data, pages 1171–1173, New York, NY, USA, 2007. ACM.
[10] Y. Cui and J. Widom. Lineage tracing for general data warehouse transformations. In The VLDB Journal, pages 471–480, 2001.
[11] 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.
[12] P. Bonatti, S. D. C. di Vimercati, and P. Samarati. An algebra for composing access control policies. ACM Trans. Inf. Syst. Secur., 5(1):1–35, 2002.
[13] P. Buneman, S. Khanna, and W. C. Tan. Why and where: A characterization of data provenance. In J. V. den Bussche and V. Vianu, editors, Database Theory - ICDT 2001, 8th International Conference, London, UK, January 4-6, 2001, Proceedings, volume 1973
of Lecture Notes in Computer Science, pages 316–330. Springer, 2001.
[14] P. Buneman and W.-C. Tan. Provenance in databases. In SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data, pages 1171–1173, New York, NY, USA, 2007. ACM.
[15] Y. Cui and J. Widom. Lineage tracing for general data warehouse transformations. In The VLDB Journal, pages 471–480, 2001.