Volume 7, Issue 4, April 2017 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Data Leakage Detection and Data Prevention Using Algorithm

Dr. A R. Pon Periyasamy
Associate Professor,
PG and Research Department
Nehru Memorial College (Autonomous)
Puthanampatti, Tamilnadu, India

E. Thenmozhi
Research Scholar,
PG and Research Department
Nehru Memorial College (Autonomous)
Puthanampatti, Tamilnadu, India

Abstract— Data leakage is defined as the accidental or unintentional distribution of private or sensitive data to an unauthorized entity. Sensitive data in companies and organizations include intellectual property (IP), financial information, patient information, personal credit-card data, and other information depending on the business and the industry. 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 (both inbound and out- bound), including emails, instant messaging, website forms, and file transfers among others, are largely unregulated and unmonitored on their way to their destinations. The potential damage and adverse consequences of a data leakage incident can be classified into two categories: Direct and Indirect Losses. Direct losses refer to tangible damage that is easy to measure or to estimate quantitatively. Indirect losses, on the other hand, aremuch harder to quantify and have a much broader impact in terms of cost, place, and time. Direct losses include violations of regulations (such as those protecting customer privacy) resulting in fines, settlements or customer compensation fees litigation involving lawsuits loss of future sales costs of investigation and remedial or restoration fees. Indirect losses include reduced share price as a result of negative publicity damage to a company's goodwill and reputation customer abandonment and exposure of intellectual property (business plans, code, financial reports, and meeting agendas) to competitors.

Keywords— Data Mining, classification ,clustering association, bi clustering and Data leakage

I. INTRODUCTION

Data leakage is an error condition in information systems in which information is destroyed by failures or neglect in storage, transmission, or processing. Information systems implement backup and disaster recovery equipment and processes to prevent data loss or restore lost data. Data leakage is distinguished from data unavailability, such as may arise from a network outage. Although the two have substantially similar effects, data unavailability is temporary, while data loss may be permanent. Data leakage is also distinct from data spill, although the term data loss has been sometimes used in those incidents. Data leakage incidents can, however, be also data spill incidents, in case media containing sensitive information is lost and subsequently acquired by another party. However, data spills are possible without the data being lost in the originating side. There are 10 common causes of data loss.

1. **Accidental Deletion of Data:** There are times when you accidental delete a file or a program from your hard drive. This is an unintentional deletion which may go unnoticed for a long time. Administrative errors also fall under this category. The best thing is to think carefully before you delete any data or program.
2. **Accidental drive format**: Users accidental format their drives and this result in instant loss of data. However, it is possible to recover your data in a situation like this. Get help from experts.
3. **Accidental Damage:** If a drive or disk is mishandled or accidentally dropped, this may cause trauma and loss of data. Data recovery is also possible in this case
4. **Natural Disaster:** Your hard drive can be damaged due fire, flood or some other unforeseen disasters. The good news is that data can still he retrieved in such situations.
5. **Purposeful Deletion of Data:** You may have deleted a file intentionally from your system and later decided you wanted the file back. You can still recover your data from the recycle bin. If you have emptied your recycle bin, you can use software recover deleted recycle bin files.
6. **Power Failure:** If you experience power failure before you have the opportunity to save your work, you may lose valuable data. The advice is to keep saving as your work.
7. **Corrupt Data:** If your file system or database is corrupt, then you are bound to loss data. Again it is possible to recover data from a corrupt file system with the right software tool.
8. **Software Failure:** When your application software suddenly crashes or freezes while working, this may result in severe damage to your hard drive. This causes the program close suddenly and all unsaved work is lost.
9. **Virus Attack:** If a machine is deeply infected by viruses and worms, spyware, adware and some deadly computer parasites, this can be very deadly and it may result to total corruption and loss of data. Installing a very good anti-virus program will reduce the possibility of having a fatal virus attack.

10. **Malicious Attack:** Professional hackers or competitors can invade your machine and destroy your system. This will obviously lead to loss of data.

II. RELATED WORK

The guilt detection approach presented is related to the data provenance problem [4] tracing the lineage of objects Available at: www.researchpublications.org implies essentially the detection of the guilty agents Tutorial [5] provides a good overview on the research conducted in this field. Suggested solutions are domain specific, such as lineage tracing for data warehouses [6], 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, As far as the data allocation strategies are concerned, work is mostly relevant to watermarking that is used as a means of establishing original ownership of distributed objects. [9], and audio data [7] whose digital representation includes considerable redundancy. Recently, [2], [11], [8], and other works have also studied marks insertion to relational data.

Data Leakage Detection:

With the fast growth of database business on the net, the data may be unsafe after passing through the unsecure network. The data purchasers may hesitate to buy the data service for the following suspicion. First, the data receiver may suspect that the data are tampered with by unauthorized person. Second, they may suspect the data received are not produced and provided by the authorized suppliers. Third, the suppliers and purchasers actually with different interest should have different roles of rights in the database management or using. So how to protect and verify the data becomes very important here.

The recent surge in the growth of the internet results in offering of a wide range of web-based services, such as database as a service, digital repositories and libraries, e-commerce, online decision support system etc. 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. 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 have 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 are 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 [14].

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. In this paper, section I provides the study of techniques for detecting leakage of a set of objects or records.

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 obtainedthrough 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 section II a guilty agent is introduce which is develop for assessing the "guilt" of agents and also present algorithms for distributing objects to agents, Sections III and IV, present a model for calculating "guilt" probabilities in cases of data leakage. Finally, in Section V, evaluating the strategies in different data leakage scenarios, and check whether they indeed help to identify a leaker.

III. ACCESS TO THE GAINED DATA

"Who caused the leak?" attribute. These attributes are not interchangeable, but rather complementary and the various ways to gain access to sensitive data can be clustered into the following groups. Physical leakage channel means that physical media (e.g., HDD, laptops, workstations, CD/DVD, USB devices) containing sensitive information or the document itself was moved outside the organization. This more often means that the control over data was lost even before it leaved the organizations **CHALLENGES are**

a) **Encryption:** and preventing data leaks in transit are hampered due to encryption and the high volume of electronic communications. While encryption provides meansto ensure the confidentiality, authenticity and integrity of the data, it also makes it difficult to identify the data leaks occurring over encrypted channels. Encrypted emails and file transfer protocols such as SFTP imply that complementary DLP mechanisms should be employed for greater coverage of leak channels. Employing data leak prevention at the endpoint – outside the encrypted channel has the potential to detect the leaks before the communication is encrypted**.**

b) **Access Control:** It provides the first line of defense in DLP. However, it does not have the proper level of granularity and may be outdated. While access control is suitable for data at rest, it is difficult to implement for data in transit and in use is not involved in.

c) **Semantic Gap in DLP:** DLP is a multifaceted problem. The definition of a data leak is likely to vary between organizations depending on the sensitive data to be protected, the degree of interaction between the users and

the available communication channels. The current state-of-the-art mainly focuses on the use of misuse detection (signatures) and post-mortem analysis (forensics). The common shortcoming of such approaches is that they lack the semantics of the events being monitored. When a data leak is defined by the communicating parties as well as the data exchanged during the communication, a simple pattern matching or access control scheme cannot infer the nature of the communication. Therefore, data leak prevention mechanisms need to keep track of who, what and where to be able to defend against complex data leak scenarios. The classification by leakage channel is Available at: important in order to know how the incidents may be prevented in the future and can be classified as physical or logical. 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

IV. GUILTY AGENT

To detect when the distributor's sensitive data has been leaked by agents, and if possible to identify the agent hat leaked the data. Perturbation is a very useful technique where the data is modified and made "less sensitive" before being handed to agents. An unobtrusive technique is developed for detecting leakage of a set of objects or records. Suppose that after giving objects to agents, the distributor discovers that a set S _ T has leaked. This means that some third party, called the target, has been caught in possession of S. For example, this target may be displaying S on its website, or perhaps as part of a legal discovery process, the target turned over S to the distributor. Since the agents U1; . .. ; Un have some of the data, it is reasonable to suspect them leaking the data. However, the agents can argue that they are innocent, and that the S data were obtained by the target through other means. For example, say that one of the objects in S represents a customer X. Perhaps X is also a customer of some other company, and that company provided the data to the target. Or perhaps X can be reconstructed from various publicly available sources on the web. Our goal is to estimate the likelihood that the leaked data came from the agents as opposed to other sources. Intuitively, the more data in S, the harder it is for the agents to argue they did not leak anything. Similarly, the "rarer" the objects, the harder it is to argue that the target obtained them through other means. Not only do we want to estimate the likelihood the agents leaked data, but we would also like to find out if one of them, in particular, was more likely to be the leaker. For instance, if one of the S objects was only given to agent U1, while the other objects were given to all agents, we may suspect U1 more. The model we present next captures this intuition. We say an agent Ui is guilty and if it contributes one or more objects to the target. We denote the event that agent Ui is guilty by Gi and the event that agent Ui is guilty for agiven leaked set S by Gi/jS.

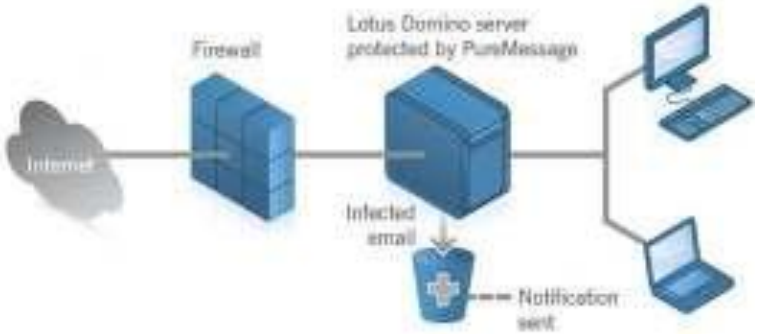

Figure 1.Data Leakage Detection

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. However, fake objects may impact the correctness of what agents do, so they may not always be allowable.

Algorithm: 1 Allocation for Explicit Data Requests (EF)

Input: R1; . . .Rn, cond1, ..., condn, b1; . . . bn, B

Output: R1; . . .; Rn, F1; . . . ; Fn

```
1:  R_ 0 Agents that can receive fake objects
2:  for i_1, . . . ; n do
3:  if bi > 0 then
4:  R_ R U{i}
5:  Fi_0 ;
6:   while B > 0 do
7:  i_ SELECTAGENT(R,R1; . . .;Rn)
8:  f _CREATEFAKEOBJECT(Ri; Fi; condi)
9:  Ri _Ri U{f}
10: Fi_ Fi U{f}
11: bi_ bi-1
```


12: if bi= 0 then
13: R_R\{Ri}
14: B _B – 1

Algorithm 2. Agent Selection for e-random
1: function SELECTAGENT (R, R1; . . . Rn)
2: i_ select at random an agent from R
3: return i

In lines 1-5, Algorithm 1 finds agents that are eligible to receiving fake objects in O (n) time. Then, in the main loop in lines 6-14, the algorithm creates one fake object in every iteration and allocates it to random agent. The main loop takes O(B) time. Hence, the running time of the algorithm is O (n+B).

If B ≥Σni-1 bi, the algorithm minimizes every term of the objective summation by adding the maximum number bi of fake objects to every set Ri, yielding the optimal solution. Otherwise, if B ≤ Σn i-1 bi, the algorithm just selects at random the agents that are provided with fake objects. Algorithm 3 It denote the combination of

Algorithm 3. Agent Selection for e-optimal
1: function SELECTAGENT (R;R1; . . .;Rn)
2: i← argma$(1 \div |Ri| - 1 \div |Ri| + 1) \sum \quad |Ri \cap Rj|)$
3: return i
Algorithm 3 makes a greedy choice by selecting the agent that will yield the greatest improvement in the sum objective.
The cost of this greedy choice is Oðn2Þ in every iteration [1]. The overall running time of e-optimal is $O(n+ n^2B) = O(n^2B)$.

V. DATA LEAKAGE PREVENTION

Data leak prevention (DLP) is a set of information security tools that is intended to stop users from sending sensitive or critical information outside of the corporate network. Adoption of DLP, variously called *data* loss prevention, information loss prevention or extrusion prevention, is being driven by significant insider threats and by more rigorous state privacy laws, many of which have stringent data protection or access components. DLP products use business rules to examine file content and tag confidential and critical information so that users cannot disclose it. Tagging is the process of classifying which data on a system is confidential and marking it appropriately. A user who accidentally or maliciously attempts to disclose confidential information that's been tagged will be denied. For example, tagging might even prevent a sensitive financial spreadsheet from being emailed by one employee to another within the same corporation. DLP productsgenerally have the following components:
Endpoint: Monitor and control activities
Network: Filter data streams
Storage: Protect data at rest. Implementing an enterprise DLP product can be complicated.

Most large organizations have hundreds of servers with thousands of directories and files stored on them and specific types of data that needs to be tagged. The software can be useful for identifying well-defined content like Social Security or credit cards numbers) but tends to fall short when an administrator is trying to identify other sensitive data, like intellectual property that might include graphic components, formulas or schematics. To implement enterprise DLP successfully, personnel from all levels of management need to be actively involved in creating the business rules for tags.. Data leak prevention (DLP) is a suite of technologies aimed at stemming the loss of sensitive information that occurs in enterprises across the globe. By focusing on the location, classification and monitoring of information at rest, in use and in motion, this solution can go far in helping an enterprise get a handle on what information it has, and in stopping the numerous leaks of information that occur each day. DLP is not a plug-andplay solution. The successful implementation of this technology requires significant preparation and diligent ongoing maintenance. Enterprises seeking to integrate and implement DLP should be prepared for a significant effort that, if done correctly, can greatly reduce risk to the organization. Those implementing the solution must take a strategic approach that addresses risks, impacts and mitigation steps, along with appropriate governance and assurance measures. New small and midsize enterprises can absorb both the financial and PR damage inflicted by serious breaches targeting sensitive data. And yet, they're often under protected because data leak prevention, or DLP, products are, overall, simply too expensive. Meanwhile, there's been a significant upswing in cybercrime after a steady five-year decline, according to the 2007 CSI Computer Crime and Security Survey. Insider abuse of network assets is the most prevalent attack, ahead even of viruses, with average losses of around $350,000.Code Green Networks, which was launched by the founders of Sonic Wall, aims to tackle this problem. Code Green's newest offering, the CI-750 Content Inspection Appliance, is geared specifically for networks with 250 or fewer users and offers the same features and functionality as its higher-end products, starting at $10,000. The CI-750 uses "fingerprints" to identify both structured data such as Social Security or credit card numbers, and unstructured data such as documents, files, source code, and so on. Where many DLP products for smaller businesses

rely on filtering for certain file types or provide only basic keyword or pattern matching, Code Green's technology creates hash values of the actual data to be protected and scans outgoing traffic for matches. We found Code Green's fingerprinting technology accurate, and a built-in mail transfer agent. However, without the help of third-party proxies, the appliance is blind to encrypted data, and it can't stop movement of internetwork and web-based traffic. This means the appliance represents only part of a robust . DLP system.

FINGERPRINT TRAIL: The CI-750 can be deployed in a variety of ways. Included a kit it was a network tap device, which let us passively monitor traffic flowing through our WAN connection, and a mail transfer agent. Customers can route outgoing messages from their mail servers through the mail transfer agent for additional mail-filtering abilities; questionable e-mail can be held until approved by an administrator. Admin also can create policies to encrypt e-mail carrying sensitive information. This functionality is provided via Code Green's partnership with the Voltage Security Network, which offers e-mail encryption as a service. After connecting the device to network, A selected sources of data that the appliance should protect. It has built-in functionality to fingerprint both structured and unstructured data such as that in CIFS. Setup for CIFS was simply a matter of providing the server and share name, along with appropriate access credentials. The device then scans the share at user-defined intervals. CIFS scanning was trouble-free and didn't cause performance issues on our Windows file server. However, it's incumbent on IT to ensure that content to be fingerprinted gets placed into the appropriate CIFS share. This can be problematic. For example, our company relies heavily on private wiki pages and not shared volumes for most of our internal information. Code Green's suggested workaround is to have a script that dumps the contents of our wikis to a CIFS share on a regular basis. Given the uptick in collaborative workspaces such as wikis in the business community, we'd like to see a fully automated way to get such data fingerprinted Available at: www.researchpublications.org also would make more sense if the device could use Web pages as sources directly; support for other data stores also would increase the out-of-the-box functionality of this appliance and eliminate the need for extra scripting. It should be noted, however, that many competing offerings, some substantially more expensive, don't even offer database integration. After selecting data sources for fingerprinting, IT then defines traffic to monitor and what actions should be taken in the event a leak is detected. We configured some very widely scoped rules and found that the CI-750 did an outstanding job alerting us to data leaks occurring within e-mail, Web, IM, and even compresed archive transmissions. We included a two sentence excerpt from a contract in an e-mail to a client. A moment later, we had an e-mail stating that there had been a violation. The administrator interface on the appliance showed that an e-mail had been sent to our customer and had the full context of the e-mail to show the violation. The interface can also display past violations that may have been related.

PARTIAL PREVENTION: While we were impressed with the accuracy of the fingerprinting, the appliance wasn't able to actually quarantine the message because it was sent via Web mail. Companies that want robust blocking of Web and network traffic will have to invest in a proxy device. The Code Green appliance can be configured as an Internet Content Adaptation Protocol server when connected to an ICAP proxy, such as those from Blue Coat Systems or Squid. When so connected, Code Green can block HTTP, HTTPS, and FTP traffic. It also can decrypt traffic for inspection. Laptops also will pose a problem for Code Green customers. The company offers an endpoint agent that controls the use of removable media such as flash drives and CDs. It also can enforce encryption of data saved to removable media, and the agent tracks the file names and types that are read from or stored on this media. However, laptops that are off the corporate network also are outside the policy controls of the Code Green appliance, meaning sensitive data can be sent via the Web or network protocols. How does DLP work: Following are the various methods how data leakage protection helps your organization to protect your valuable or sensitive information which is in transit, at rest or in use. 1) DLP provides a robust solution to protect data in transit [network actions] by sniffing network traffic of emails, chat messages, etc to discover content being sent across the communication channel. 2) It also provides a solution to protect data at rest by scanning storage area content like USB drives, hard drives, etc and discover content from it. It is also termed as Content Discovery.3) It also provides a solution to protect data in use [endpoint actions] i.e., it protects the data which is in use by the user for example if a user has connected USB drives to the computer. Most DLP solutions do this in combinations of the following: Rule-based Regular Expressions, Database Fingerprinting, Exact File Matching, Partial Document Matching, Statistical Analysis, Conceptual/Lexicon, Categories

VI. CONCLUSION

From the study of the data leakage, we can detect and prevent the data from the leak by using some algorithms and techniques. 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.

REFERENCES

[1] Panagiotis Papadimitriou and Hector Garcia-Molina, “Data Leakage Detection,” IEEE Trans, Knowledge and Data Engineering, vol. 23, no. 1, January 2013.

[2] R. Agrawal and J. Kiernan, “Watermarking Relational Databases,”Proc. 28th Int’l Conf. Very Large Data Bases (VLDB ’02), VLDB Endowment, pp. 155-166, 2002.

[3] 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, 2011.

[4] P. Buneman, S. Khanna, and W.C. Tan, “Why and Where: A 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.

[5] P. Buneman and W.-C. Tan,"Provenance in Databases,"Proc. ACM SIGMOD, pp. 1171-1173, 2013.
[6] Y. Cui and J. Widom, "Lineage Tracing for General Data Warehouse Transformations," The VLDB J., vol. 12, pp. 41-58, 2014.
[7] S. Czerwinski, R. Fromm, and T. Hodes, "Digital Music Distribution and Audio Watermarking,"http://www.scientificcommons.org/430256 58, 2007. Available at: www.researchpublications.org NCAICN-2013, PRMITR,Badnera 399
[8] F. Guo, J. Wang, Z. Zhang, X. Ye, and D. Li, "An Improved Algorithm to Watermark Numeric Relational Data," Information Security Applications, pp. 138-149, Springer, 2013.
[9] F. Hartung and B. Girod, "Watermarking of Uncompressed and Compressed Video," Signal Processing, vol. 66, no. 3, pp. 283-301,1998.
[10] S. Jajodia, P. Samarati, M.L. Sapino, and V.S. Subrahmanian, "Flexible Support for Multiple Access Control Policies," ACM Trans. Database Systems, vol. 26, no. 2, pp. 214-260, 2001.
[11] Y. Li, V. Swarup, and S. Jajodia, "Fingerprinting Relational Databases: Schemes and Specialties," IEEE Trans. Dependable and Secure Computing, vol. 2, no. 1, pp. 34-45, Jan.-Mar. 2015.
[12] B. Mungamuru and H. Garcia-Molina, "Privacy, Preservation and Performance: The 3 P's of Distributed Data Management," technical report, Stanford Univ., 2008.
[13] R. Sion, M. Atallah, and S. Prabhakar, "Rights Protection for Relational Data," Proc. ACM SIGMOD, pp. 98-109, 2013. [14] L. Sweeney, "Achieving K-Anonymity Privacy Protection UsingGeneralization and Suppression," http://en.scientificcommons. org/43196131, 2002.

