ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Prevention and Data Leakage Detection using Trace Log

Parvathi Maheswari T
Assistant Professor, Department of Computer Science & Engineering, Anna University
R.M.D Engineering College, Kavaraipettai – 601206, India
parvathi.cse@rmd.ac.in

Abstract

In the world of modern computer era, need for handling sensitive data to third party agents has become inevitable. Data loss and its prevention has become a vital problem that needs to be solved in many organizations. There are lots of possible identified routes to increase security and data loss which resulted to increase complexity. Therefore, agents need to provide confidentiality for users, maintain integrity and assure availability of all data. In general Data Leakage Prevention is done by transforming highly sensitive data to least sensitive or encrypting using cryptic algorithm or watermarking an image. Probable solution for the Data Leakage Prevention is by encrypting Data and Fake Records using Secure Hash Algorithm (SHA) and Detection by using Explicit Data allocation strategies (across the agents), which are used to identify the leakages. To further increase the chance of detecting the agent Trace log file is created. The Solution is achieved by sending the same data over many agents and restricting the Distributor to collect data only through agent.

Keywords: Agent, Distributor, Encryption, Fake Records, Trace Records.

Introduction

A web is outlined as collection of many sites and it includes information, data supplier and collector. Data supplier is the distributor and agent is the one who collects the data. Internet is been accessed by each illegitimate and legit user and it is terribly necessary to secure the information that are unit sent in the network. The web browser is used for accessing the content. Security is to safeguard the data or information that's passed on.

One of the biggest concerns companies have around cloud technologies is the security of their digital content. Companies manage sensitive data, ranging from product plans to employee personnel files, all of which need to be secure. Thus, it's important for companies to map out a specific cloud security strategy before storing and sharing their business data via cloud solutions. In Tradition, leakage detection is been handled by Watermarking and Fingerprinting.

In this research work we study associate degree unobtrusive technique to prevent and detect the leakage of data. This prevents unauthorized person to access the data. In case if the distributor discovers some of those related information in an unauthorized place the agent who were been responsible for the leakage is also found.

The fake records act as a kind of watermark for the complete set, while not modifying any individual members but appear realistic to the agents. If it seems agent was given one or lot of fake objects. If the content is been leaked or found somewhere, by using Fake Records distributor can be more confident that agent was guilty. The Trace Log File acts like a mail box. It stores id of agent and his continuous mail chain with doubtful third person. Hackers Report is to see who tried to view the content in a skeptical manner.

Background and related work

The main focus of the researchers in the recent past is the safer distribution of data. The first approach [2] focuses on how the datawarehousing system collects data from multiple distributed sources and stores the collected information as materialized views in a local datawarehouse. The second approach [3] focuses on conceptual framework for relational database based on secret sharing algorithm based on Shamir’s secret sharing scheme to address this problem. [4] Focuses on sound synthesized process in digital instruments and propose a real-time watermark method. [5] it focuses on Fingerprinting, is a class of information hiding techniques that insert digital marks in to a data purpose of identifying the recipients who have been provided data. [6] Watermarking is another class of information hiding techniques whose purpose is to identify the sources of data, focuses on the challenges of watermarking is to insert an indelible mark in the document.

The third approach [7] [8] focuses on algebra for compositing attribute-based access control policies, using ABAC and the special policy composition requirements. Here the access control policies will allow only authorized users to access sensitive data through access control policies. These approaches prevent leakage by sharing information only with trusted parties. In the web the data cannot be passed only to the restrictive users and hence this approach is impossible.

The other drawback it Uses Probabilistic based approaches for allocating different data to different agents and hence it is easy to detect who has leaked thedata. It is not used in the case where the same data are transferred to different agents as this case requires more secure transaction.

Proposed architecture

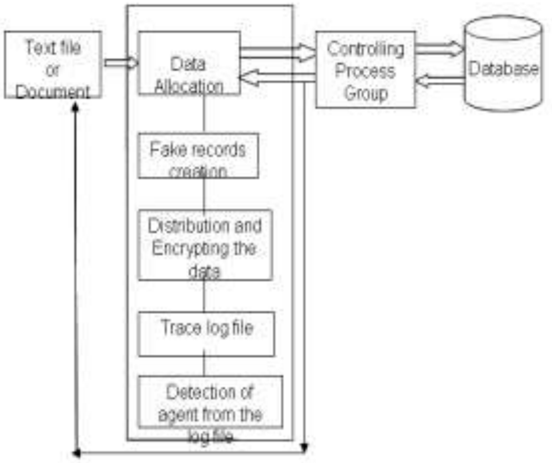

Text file or Document
Data Allocation
Controlling Process Group
Database
Fake records creation
Distribution and Encrypting the data
Trace log file
Detection of agent from the log file

Figure 1. Block diagram

The secured Data can be either Text (or) Document. The Data are sent between Distributor and the agents. It is to be hided from the illegitimate user. Therefore, the data are sent very intelligently using fake records and Explicit Data Allocation Requests approaches. Fake records are created using the function **CreateFakeObject (Ri,Fi,condi).** Then the data and Fake Records are **encrypted** using **Secure Hash Algorithm (SHA)** and transferred to respective agents. Along with the data a log file called Trace Log is created and sent to the owner. This Inbox will contain the mail id of those persons for whom the data transfer were made without the knowledge of distributor. This approach is to detect the trusted party who leaks data. The unauthorized attacker's ID who is trying to view the content, will be loaded in the database and from that information Hackers Report, Graph, Pie-Chart can be created.

Agents notation

A *data supplier is responsible for data and has* Data objects. The distributor wants to share some of the objects with a set of agents but does not wish the objects be *leaked* to other third parties. The data could be of any type and size, or a database. In this scenario a set of agents been created.

Data Leakage Detection

Figure 2. Agent Creation

Fake Records

The distributor adds 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. The distributor creates and adds fake objects to the data that he distributes to agents. Fake Records will be looking like real data. Each agent is provided with unique Fake Records,from this the distributor can easily found out who has leaked.Fake Object Creation: The fake objects are created using the model Ui as a black-box function CREATEFAKEOBJECT(Ri, Fi, condi) that takes as input the set of all objects Ri, the subset of fake objects Fi that Ui has received so far and condi, and returns a new fake object. This function needs condi to produce a valid object that satisfies Ui's condition. Set Ri is needed as input so that the created fake object is not only valid but also indistinguishable from other real objects.

The distributor can also use function CREATEFAKEOBJECT () when it wants to send the same fake object to a set of agents. In this case, the function arguments are the union of the Ri and Fi tables respectively, and the intersection of the conditions condi's.

Trace Records

Trace Record is used with "mailing list". This Record is to identify the agents who leaked the data. Once the data is sent by the distributor to agents a mailing list is created. From there on if a particular agent wishes to send the data to entrusted third parties. Using this trace log file the distributor will get a copy of mail, which will include details like who leaked and to whom the data is been sent. Fig 2 These records are a type of fake objects that help identify improper use of data.

Figure 3. Trace Records

Proposed system

With the improvement of information level, more and more enterprises are conscious of the importance of the intranet security. Usually, a variety of means such as IDS, IPS, firewalls and VPN are utilized to ensure that intranet can work correctly. However, they only defend attacks from external network; because all these measures are based on an assumption that internal network is reliable. However, the reality is often not like this. According to the survey of FBI/CSI, more than 80% attacks come from internal staffs. The internal staff once after receiving the data from the owner he may send it to some unauthorized party that losses the company's confidentiality.

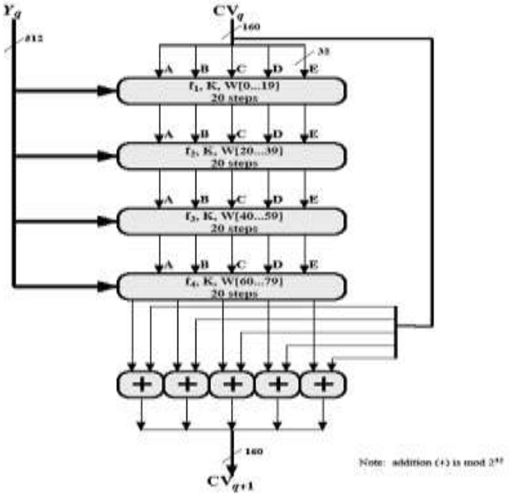
It guarantees the security of confidential data, while the Based on Windows file system filter driver, we proposed a novel model for Data Leakage Prevention (DLP), which works at internal node and encrypts data automatically. All confidential data are protected transparently without the consciousness of users.

ENCRYPTION (Prevention)

In order to ensure that data are not read by any unauthorized party both objects and fake records are encrypted using cryptographic algorithm **Secure Hash Algorith[SHA].** Only those agents who have correct key can decrypt on the other end. Using this fake record, agent who has leaked data can be identified.

Secure Hash Algorithm (SHA-1)

SHA was designed by NIST & NSA in 1993, revised 1995 as SHA-1US standard for use with DSA signature scheme standard is FIPS 180-1 1995, also Internet RFC3174and the algorithm is SHA, the standard is SHS It produces 160-bit hash values It is the generally preferred hash algorithm It is based on design of MD4 with key differences



When transparent file encryption technique is applied in data leakage prevention, it will be convenient not only for user operations, but also protecting data from leaking out effectively with its two remarkable characteristics:

Data Leakage Detection

Figure 4. Encrypted Data

DECRYPTION

Data Leakage Detection

Figure 5. Decrypted Data

DETECTION

Distribution of Data among Agents:

Done via 2 ways:

- Sample request Ri = SAMPLE(T,mi): Any data can be transferred among agents.
- Explicit request Ri = EXPLICIT(T, $condi$): Agent *receives data based on condition.*

Explicit Data Requests

In this case only few objects are shared among multiple agents. In problems of class EF the distributor is not allowed to add fake objects to the distributed data. So, the data allocation is fully defined by the agents' data requests. In EF problems, objective values are initialized by agents' data requests. Say, for example, that $T = \{t1, t2\ \}$ and there are two agents with explicit data requests such that $R1 = \{t1,t2\}$ and $R2 = \{t1\}$.

Assumption1: *The data are leaked only by the agent and not by any other means.*

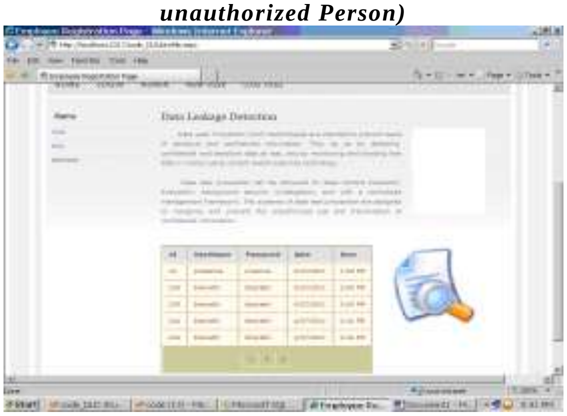
The leaked data came from the agents as opposed to other sources. We say an agent Ui is *guilty* and if it contributes one or more objects to the target. We denote the event that agent Ui is guilty as Gi and the event that agent Ui is guilty for a given leaked set S as $Gi|S$.

Assumption2: *Only few objects are shared among multiple agents*

It focuses on scenarios where same objects are shared among agents. These are the most interesting scenarios, since object sharing makes it difficult to distinguish a guilty from non-guilty agents. Scenarios with more objects to distribute or scenarios with objects shared among fewer agents are obviously easier to handle. As far as scenarios with many objects to distribute and many overlapping agent requests are concerned, they are similar to the scenarios we study, since we can map them to the distribution of many small subsets. This requires more security as it is very hard to find who the agent who leaks the data. Fig 6:shows the Hackers List, the count of unauthorized persons trying to view the content. Fig 7 Shows the Report which gives the status like which user has viewed and time so on.

Data Leakage Detection

Figure 6: Hackers List (To Trace the unauthorized Person)



Data Leakage Detection

Figure 7: Hackers Report

Experimental results & performance issues

The result from the algorithm shows that the data are not leaked, in case even if it is leaked by agents those agents who have leaked can be found by adding the fake records using explicit data requests approach. The fake records are added in random and in optimal way for the agents using the algorithm. The increase in number of Fake Records yields higher chances of detecting the agents. As the Hackers count increases the probability of detecting the guilty agent yields good result. Fig 8 : Blue color curve deals with the length as how many times that particular data has been viewed by the unauthorized person. The Red color denotes the user id and the Green color denotes the Username.

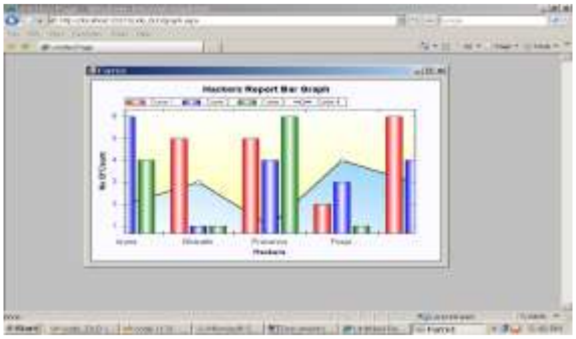


Figure 8: Performance Measures

From the PIE CHART we can get to know The unauthorized person who has tried more times to view the content.

Fig 9: Graphical Representation of Hackers Report

Conclusions & future work

In Today's world it is highly required to transfer data across users. Hence there are high chances of leaking. In order to avoid this we can watermark the content,as we could not watermark all the content,it is possible that we are going for prevention and also detection of agent who is responsible for 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. The algorithms we have presented implement a variety of data distribution strategies that can improve the distributor's chances of identifying a leaker.

The future work includes the investigation of agent guilt models. Another open problem is the extension of our allocation strategies so that they can handle agent requests in an online fashion.

References

1. *Panagiotis Papadimitriou, Member, IEEE, Hector Garcia-Molina, Member, IEEE "Data Leakage Detection" IEEE Transactions on Knowledge and Data Engineering, VOL. 22, NO. 3, 2010.*
2. *Y. Cui and J. Widom. "Lineage tracing for general data warehouse transformations",In The VLDB Journal, 2001 .*
3. *Cong Jin, Yu Fu and Feng Tao , Wuhan 430079, P.R.China "The Watermarking Model for Relational Database Based on Watermarking Sharing " in the Department of Computer Science, Central China Normal University, International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Proceedings of the 2006.*
4. *Kotaro Yamamoto Munetoshi Iwakiri Department of Computer Science National Defense Academy in Hashirimizu, Yokosuka, Kanagawa, "Real-Time Audio Watermarking with Wavetable Alternation in Digital Instrument " IEEE computer society, 2009.*
5. *Y. Li, V. Swarup, and S. Jajodia. "Fingerprinting relational databases: Schemes and specialties". IEEE Transactions on Dependableand Secure Computing, 2005.*
6. *R. Sion, M. Atallah, and S. Prabhakar "Rights protection for relational data". New York, NY, USA,. ACM, pages 1509 – 1524 , 2004.*
7. *Cheng Xiangran#, Chen Xingyuan, Zhang Bin, Yang Yan Zhengzhou "An Algebra for Composing Access Control*

Policies in Grid "Inf. Sci. & Technol., Inst. China. IEEE 2009.

8. *S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian "Flexible support for multiple access control policies" ACM Trans.Database Syst., pages 214–260, 2001.*
9. *"A Model for Data Leakage Detection" Panagiotis Papadimitriou , Hector Garcia-Molina(2009).*
10. *"Research and Application of the Transparent Data Encryption In Intranet Data Leakage Prevention": Zhang Xiaosong ,etal.2009.*