A SURVEY OF DATA LEAKAGE DETECTION IN CLOUD COMPUTING PLATFORM

Anunay Ghosh[1]
[1]JIS College of Engineering, Kalyani
anunay.ghosh@jiscollege.ac.in

Priyangshu Dhar[2]
[2]JIS College of Engineering, Kalyani
priyangshudhar700@gmail.com

Annwesha Banerjee[3]
[3]JIS College of Engineering, Kalyani
annwesha.banerjee@jiscollege.ac.in

Monali Sanyal[4]
[4]JIS College of Engineering, Kalyani
4monali.sanyal@jiscollege.ac.in

Abstract:

Cloud computing is the foundation of the contemporary IT era, where both small and large businesses use its services directly or indirectly. Cloud computing is currently a necessary talent that any IT professional should possess. It is no longer a specialized expertise.. Cloud computing technology is now a standard for IT deployment in the business, educational, and governmental sectors as a result of its extensive adoption. Yet, there are security flaws in cloud technologies like Web-based dashboards and hypervisors that could allow data to leak. The consequences of data leakage are severe; for a corporation like Exactis, one such event resulted in the exposure of 340 million client details. This paper provides an overview of different cloud computing techniques for identifying data leakage. We'll start by discussing the possibility of data leaking while using the proposed method. The Audit Trail/Transaction, Watermarking Method, Data Allocation Techniques, and VM Migration Process will be covered in more detail later.

Keywords: Data Leakage, Cloud Computing, Cloud Security, Information Security

\

I. INTRODUCTION

Data loss refers to the loss of data from any data-storing device. Anyone who uses a computer will have this issue. Data loss occurs when information is mistakenly or purposely physically or logically removed from the organization. Today, data loss has taken the top spot among organizational issues, and it is up to the organizations to provide solutions. Data leakage is an instance where information's confidentiality has been violated. Unauthorized data transmission from within a company to an outside location is what it alludes to. Data loss is the loss of data due to deletion, system failure, etc., whereas data leakage is the disclosure of private or confidential information. One of the major concerns that organizations now have is a data breach, which can be used to refer to both terms collectively.

There are several modules on this research era.

I. METHODOLOGY

A. Data Allocation Module:

This module's major goal is to address the issue of data distribution, specifically how to "intelligently" distribute data to agents to increase the likelihood that a guilty agent will be found. Admin may send files to authenticated users; users can modify their account information, etc. Agent receives correspondence containing the secret key details to improve the likelihood of finding data leakage agents.

B. Fake Object Module:

The data that the distributor sends to the agents includes phony objects that he builds and adds. In order to maximize the likelihood of finding agents who leak data, the distributor creates fake objects. In order to increase his ability to identify guilty agents, the distributor may be able to include phony objects in the data that is disseminated. The use of "trace" records in mailing lists served as inspiration for our use of fake objects. If we use the incorrect secret key to download the file, a duplicate file is opened and phony information is also included in the letter. Ex: Details of the bogus object will be shown.

C. Optimization Module:

The distributor's data allocation to agents is done by the Optimization Module, which has one aim and one constraint. The agent's restriction is to fulfill the distributor's requests, either by giving them the exact amount of items they ask for or by giving them all of the objects that are available and meet their requirements. His goal is to be able to identify any agent who discloses even a small amount of his data. The files can be locked and unlocked by the user for security.

D. Data Distributor Module:

Sensitive information was provided by a data distributor to a group of purportedly reliable agents (third parties). Some of the information has been compromised and is now in an unauthorized location (such on someone's laptop or the internet). The distributor is required to determine if it is more likely that one or more agents were responsible for the data leak than that it was independently acquired. Admin can view which file is leaking and fake user's information as well.

E. Agent Guilt Module:

We require an estimate of the chance that values in S can be "guessed" by the target in order to calculate this. Let's imagine, for example, that some of the objects in T are people's emails. We can test this by asking someone with roughly the target's knowledge and resources to locate the emails of, say, 100 people. We might safely infer that the likelihood of discovering one email is 0.9 if this person can find, let's say, 90 emails. If the objects in the questionnaire were bank account numbers, on the other hand, the person might only find, say, 20, yielding an estimate of 0.2. The likelihood that the target can guess object t is what we refer to as the estimate, or pt. We assume that all T objects have the same pt, which we refer to as p, in order to make the formulas we present in the next sections of the paper simpler. Although they are difficult to depict, our equations may be simply adapted to different points of view. Next, we assume two things about how the various leakage events are related. The first presumption merely asserts that other items have no bearing on an agent's choice to leak a particular object.

Review of Work on Cloud-Data Leakage

[1] In this work, a proposed system is used to identify the agent who leaked the distributor's sensitive data and to identify when that data has been leaked by agents. Data is altered and made "less sensitive" before being given to agents, which is a highly helpful technique called perturbation. In order to increase the security of user data, the model employs an encrypted method that enables secure data transmission between the user and TPA to the cloud. Additionally, the current techniques for allocating items to agents increase the likelihood that a leaker will be found. When false data objects are added to a distributed collection, incorrect information is provided in the event that a third party gained access to the data. Although these items don't correspond to actual things, they seem plausible to the agents. In a way, the phoney items serve as a kind of watermark for the overall group without changing any of the individual components. The distributor can be more certain that an agent was guilty if it turns out they were supplied one or more fraudulent items that were disclosed. The original data will be uploaded on the cloud by the TPA in encrypted form, along with the fictitious data (predefined). The administrator in this case, acting as TPA, is the company's authorized data owner and has the authority to give employees access to sensitive data. The administrator has the authority to add new employees, upload data to the server, or communicate data to the appropriate corporate personnel. The data is also hashed in this system. This security feature is included to improve the security of sensitive data transfer over the internet. The SHA secure algorithm provides the hash function at the time of data upload with references to the third party agent and to the actual cloud. Because it uses message digest to ensure data confidentiality and integrity, the SHA algorithm is utilized. In this method, the data is secured using SHA-2, a subset of the Keccak family of cryptographic primitives. The 256-128-security bit SHA-2 has been designed using an internal 1024 block size and will work using the operations And, Xor, Rot, Add (mod 264), Or, and Shr.

In the next work, they employed the water marking model in this study. An example of a security strategy is water marking, which involves embedding a specific code or encryption on the information that will be transferred. The data may be presented as an image, a video, or any official file. This encryption enables the business to assert control over any specific data. A bit pattern is added to the data at a specific location on the tuples and subset of the data in the water marking technique. The tuple, subset, and attributes are algorithmically constructed to be under the control of a key that can only be used by the owner. To find the watermark, we don't require access to the original data or knowledge of the watermark's pattern. A limited subset of data that only contains a small portion of the watermark can be used to detect the watermark. Using watermarking software that infuses the data with little defects, the watermarking is added to the file. A watermark is created when all of these tiny errors come together. This watermark has no real meaning and cannot be removed by an outside source. The ease with which digital material, such as pictures, movies, and documents, can be compromised and spread widely online can be effectively combated by adding a watermark that serves as a signature-based evidence of ownership.

Additionally, this system has done data allocation along with it. Information system is the process that enables one company to access the data of another company. It is very important for businesses and must be maintained with high security. The monitoring of institute-provided software and materials to ensure that they are only used within the institute, the provision of data privacy, and the maintenance of the relationship between the gathering and disclosure of data with all legal and political issues are the main components of security issues. Only when an organization sends secure information to an unreliable third party while unintentionally mistaking it for a reliable third party does the data become public. The goal is now to find the agent and prevent the agent from accessing the data when the distributor finds that some group of objects were transported to a new place and the data was being leaked. Not only must we block access to the data, but we also need to determine whether the agent is a data leaker or not. By include bogus objects in the information sent; the distributor increases the likelihood of the agent being found by sending the data to the agent utilizing data allocation algorithms. If someone who receives the data leaks it, the distributor will track down the agent with the aid of numerous released false items, wait until he has sufficient proof, and then eventually recognize the agent and cancel the business with him or pursue legal action against the agent.

[2]In the next work, the system used for the subsequent task includes an audit trail/transaction log system that verifies and keeps track of user activity and transactions related to computing resources at each stage of the procedure. Each user is registered by the system administrator, who also assigns them duties. An administrator, a distributor, and a user are among the roles. New files are uploaded into the system by the distributor. In this system, information provided by the user at registration is utilized to produce login information as well as the user's secret key. To receive user input into the system, the interface uses graphic user interface elements such text boxes and radio buttons. The system administrator signs in and has the ability to register users, access a list of all users who have registered, assign roles to members, and review the audit trail to find a leaker. The admin can assign roles to users as well as view the list of all registered users. The audit trail or log that records system user activity. It displays all of the actions taken by the user (admin), along with the date, time, and actual actions taken at each step of the application process. This audit trail table is protected against manipulation by other tables since it does not depend on or link to any other tables in the program. It is designed to be engaged at each level of the login process in order to identify and validate the user. It has been connected to application accessibility, so before access is provided to any software application, a user must be successfully confirmed. By doing this, it will be ensured that users who have not been validated biometrically by this system cannot access any software. It is merely an additional level of user identification and authorization inside the framework of cloud computing. The security is supplied via dynamic key generation, which is an automatically produced random unique number for every file when user or an employee attempts to see the content of file. The system safeguards the data spilled from guilty agent who act as a third party. The administrator may take action or block the user if they discover the leaker.

[3]In this work, the authors concentrated on data leakage that occurs during replication or VM transfer. Replication and migration are two frequent actions carried out on a virtual machine (VM) running on a cloud platform. Data leakage is a possibility during those procedures as a result of

incorrect configuration. When cloud users access the self-care portal or dashboard, there is also a chance that data will be compromised during the authentication phase of a communication session. The risk of data leaking during the VM migration procedure and web dashboard authentication on the cloud computing platform are both covered in this article. They employed a technique that involved packet collection during the dashboard login and VM migration processes. Different management and communication packet kinds will be examined and evaluated while the cloud process was running in order to find data leaks. They ran an experiment to show how the technique works. They created a cloud testbed environment for the demonstration based on VMware VCenter and OpenStack technologies. The outcome of the trial will confirm the method's efficacy in identifying data leaking incidents. Data leaking during cloud dashboard authentication was another area of emphasis. Many of the web dashboard components offered by various cloud management tools are susceptible to web-based attacks including man-in-the-middle and cross-site scripting attacks (XSS). In addition, a report that examines the security of OpenStack Horizon notes that it has flaws, such as not encrypting web traffic with SSL/TLS and having a lax password policy. As a result, it is necessary to do further research into the possibility of data leakage occurring during the authentication procedure to access the dashboard. They employed a method to detect the data leakage in order to determine or confirm whether data leakage occurred in some processes or features of the cloud platform. When comparing data leakage models, a few factors must be used as data leakage detection measurement, such as traffic shape, regularity, distribution, data context, and inter arrival time. The method for detecting data leakage on cloud platforms proposed in this paper involves executing packet capture during VM movement and dashboard authentication. Different sorts of management and communication packets that are sent between cloud nodes and components when the cloud process is running will be examined and analyzed in order to find data leaks. This study conducts two experiments that imitate VM replication and migration and cloud use authentication processes in order to look into and confirm a data leakage on the cloud. While the experiment is underway, packets will be recorded using the Wireshark program and packet analysis will be done on the packet dump. The packet analysis enables the detection and confirmation of data leakage.

More information about the experiments is provided in the section that follows.

In the second experiment, the online dashboard login page and a cloud user are made to simulate an authentication process. The cloud platform used in this project is based on OpenStack, a popular and open-source cloud management system. A web-based dashboard on the OpenStack controller node offers cloud users and administrators a simple interface for managing virtual machines, keeping track of performance, and carrying out other administrative tasks. The web dashboard facilitates client-server communication by using the HTTP protocol. This experiment uses a Dell Optiplex 990 workstation with a 3.40GHz processor, 16GB of RAM, and a 1TB hard drive as the OpenStack controller host. There were also two hosts that would serve as a client and a packet capturing device. An unmanaged Ethernet switch was used to connect each computer in this experiment; the testbed network topology is shown in Figure 5. They start the packet capturing operation before starting the user authentication process on the online dashboard. The user will use a web browser on the client host to visit the dashboard and begin the authentication procedure by inputting the controller node IP on the URL. To access the control panel and the OpenStack Web Dashboard login page, the user must first provide their username and password. We will halt packet capturing after the authentication procedure is over and start packet dump analysis.

Sandip A. Kale and S.V. Kulkarni [4] concentrate on watermarking. A robust watermarking technique can be quite useful in some instances, although it does require some change of the original data. Furthermore, if the data recipient is hostile, watermarks can be removed. To boost his effectiveness in finding guilty agents, the distributor may be able to add phoney objects to the dispersed data.

Data leakage is a major concern for enterprises and other institutes, according to Sushilkumar N. Holambe, Dr.Ulhas B. Shinde, and Archana U. Bhosale [5]. They talked about how the distributor produces and inserts false objects into the data that he sends to agents. To boost his effectiveness in finding guilty agents, the distributor may be able to add phoney objects to the dispersed data. Although leakage detection is handled by algorithms, there is a significant concern about the integrity of the systems' users.

Priyanka Barge, Pratibha Dhawale, and Namrata Kolashetti [6] present data capture and distribution options that improve the distributor's odds of detecting a leak. In the situation of data agent overlap, the suggested technique finds guilty agents. Based on the overlap of his data with the leaked data and the data of other agents, as well as the possibility that objects can be "guessed" by other means, it is also feasible to assess the likelihood that an agent is responsible for a leak.

Every day, confidential corporate information such as customer or patient data, source code or design specifications, pricing lists, intellectual property and trade secrets, and spreadsheet predictions and budgets are leaked. When these are disclosed, the company is no longer protected and falls outside of the corporation's jurisdiction. This unregulated data leaking [7][11] exposes businesses to risk. Once this material is no longer in the domain, the company is in grave danger.

The authors of paper [8] built a domain-specific concurrency model that supports a broad class of IDS analysis without relying on a specific detection approach. The implemented technique splits the stream of network events into subsets that the IDS will analyze separately, while ensuring that each subset contains each event relevant to a detection instance. The proposed partitioning approach is based on the concept of detection scope, i.e., the smaller the "slice" of traffic that a detector needs study in order to execute its duty. Because this concept has some general applicability, the designed model will enable simple, per-flow detection techniques and more complex, high-level detectors.

According to the author's findings [9], the introduction of important data is not necessary due to information changes in the content. Transformations (such as insertion and deletion) produce highly unexpected leakage patterns. Because of their intimidating complexity, current automata-based string coordinating methods are illogical for detecting altered data leakage while presenting the required consistent expressions. They provide two unique techniques for detecting both long and incorrect data leaks. In comparison to the best in class inspection techniques, their framework provides high detection precision in identifying changing breaks. They parallelize our concept on graphics processing units and have demonstrated the strong scalability of the data leakage detecting setup when evaluating massive amounts of data.

The authors of paper [10] state that a number of apparent distance measures used to compute behavioral similarity between network hosts fail to capture the semantic value instilled by network protocols. Furthermore, they frequently overlook the long-term temporal structure of the objects being tallied. To investigate the role of these semantic and temporal variables, they develop a new behavioral distance metric for network hosts and compare its performance to that of a metric that ignores such information. They provide semantically essential metrics for common data types present in network data, show how these metrics may be consolidated to consider network data as a uniform metric space, and show a temporal sequencing technique that captures long-term causal links.

II. CONCLUSION

The aforementioned discussion leads to the conclusion that the data leakage detection market is quite diverse because it developed from the ripe product lines of top IT security companies. To provide defenses against different aspects of the data leakage issue, a wide range of enabling technologies, including firewalls, encryption, access control, identity management, machine learning content/context-based detectors, and others, have already been implemented.

REFERENCES

1. Data Leakage Detection and Security in Cloud Computing ,Chandu Vaidya, Prashant Khobragade, Ashis Golghate, ISSN: 2455-5703
2. An improved data leakage detection system in a cloud computing environment, Prisca I. Okochi, Stanley A. Okolie, Juliet N. Odii, 11(02), 321–328
3. Data Leakage Detection in Cloud Computing Platform,Muhammad Azizi Mohd Ariffin1, Khadijah Ab Rahman2, Mohamed Yusof Darus3 , Norkhushaini Awang4, Zolidah Kasiran5, ISSN 2278-3091
4. Sandip A. Kale, Prof. S.V.Kulkarni," Data Leakage Detection", International Journal of Advanced Research in Computer and Communication Engineering,Vol. 1, Issue 9, November 2012
5. Prof. Sushilkumar N. Holambe, Dr.Ulhas B.Shinde, Archana U. Bhosale,"data Leakage Detection Using Cloud Computing ", International Journal Of Scientific & Engineering Research, Volume 6, Issue 4,(April-2015)
6. Priyanka Barge, Pratibha Dhawale, Namrata Kolashetti," A Novel Data Leakage Detection", International Journal of Modern Engineering Research (IJMER) ISSN: 2249-6645 ,Vol.3, Issue.1, Jan-Feb. 2013
7. Archana Vaidya, Prakash Lahange, Kiran More, Shefali Kachroo and Nivedita Pandey," Data Leakage Detection", International Journal of Advances in Engineering & Technology, ISSN: 2231-1963, March 2012.
8. L. D. Carli, et al., "Beyond pattern matching: A concurrency model for stateful deep packet

inspection," in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., pp. 1378-1390, 2014.

9. X. Shu, et al., "Rapid and parallel content screening for detecting transformed data exposure," in Proc. 3rd Int. Workshop Secur. Privacy Big Data (BigSecurity), pp. 191-196, 2015.
10. S. E. Coull, et al., "On measuring the similarity of network hosts: Pitfalls, new metrics, and empirical analyses," in Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS), pp. 1-16, 2011.
11. Chandu Vaidya etl. & BE scholars "Data leakage Detection and Dependable Storage Service in cloud Computing" IJSTE volume 2 issues 10 April 2016 ISSN online 2349-784X
12. https://scholar.google.com/
13. https://www.researchgate.net/