International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/IJSRCSEIT

Detection of Data Leakage Using Cloud Computing

S. Fazloon[1], Mrs P. Poornima[2]
MCA Student[1], Assistant. Professor[2]
Mother Theresa Institute of Computer Applications, Palamaner, S. V. University, Andhra Pradesh, India

Article Info

Publication Issue :
Volume 8, Issue 6
November-December-2022

Page Number : 76-81

Article History
Accepted: 01 Nov 2022
Published: 05 Nov 2022

ABSTRACT

Enterprise resource planning has helped the life sciences and the healthcare industries of today produce enormous amounts of data. Since managing such a large volume of data is challenging and internal employee intimidation around data leakage is on the rise, businesses are implementing security measures like Data Loss Prevention and Digital Rights Management to stop data leakage. Cloud Service Provider will include the doctors in this project and mail their credentials. Patients can ask doctors to upload reports on their behalf. Doctors are able to view patient requests and respond to them. Here, the patient can upload his or her files, and the cloud service provider may ask that specific user (the patient) to upload any missing files if necessary. On the other hand, the mechanism for preventing data leaking likewise becomes complex and difficult. In order to handle large amounts of data with increasing efficiency and a set of rules that give workers the outcomes they need, cloud computing approaches are deployed. It lessens the need for employees to actively choose components to address issues with supervised, unsupervised, and semi-supervised healthcare data.

Keywords: Data Leakage, Cloud Computing, Healthcare, Detection.

I. INTRODUCTION

There are numerous development models available now for identifying the culpable parties. We also provide algorithms for allocating items to agents in a way that increases the likelihood that we will spot a leaker. Finally, we take into account the possibility of including fictitious objects in the distributed collection. Although these items don't correspond to actual things, they seem plausible to the agents. The bogus items function something like a watermark for the entire group, without changing any particular members. The distributor can be more certain that an agent was guilty if it turns out that they were provided one or more fraudulent items that were leaked. Additionally, we'll employ the approximation technique to locate guilty agents. There is no cap on the number of consumers for the model we provided, and it can fulfil all customer requirements. The model provides data allocation techniques to increase the likelihood of finding leaks. Additionally, there is software that functions as a distributor, sending users' requests for files containing sensitive information and handling such files. For each request, a log is kept, which is later used to assess the likelihood of guilt and to determine whether any requests overlap with the

collection of files that were leaked. Every day, confidential corporate information including patient or client information, source code or design specifications, pricing lists, trade secrets and intellectual property, as well as predictions and budgets in spreadsheets, leak out. When these are revealed, the business is no longer shielded and is no longer under the corporate control. Businesses are exposed as a result of this unchecked data leaking. When this information leaves the domain, the organisation faces a major risk from fraudsters.

• Cash out costs our company money, erodes our brand, reputation, and competitive edge, and erodes customer trust. To solve this issue, we create a model for determining an agent's level of guilt.

• The distributor will intelligently provide information to agents to increase the likelihood that a guilty agent will be found, such as adding the phoney objects to dispersed sets.

The distributor can now determine whether it is more likely that one or more agents were responsible for the data breach than that it was independently obtained. If the distributor has sufficient proof that an agent has leaked data, they may stop doing business with him or file a lawsuit. It primarily has one aim and one set of restrictions. The Distributor's constraint satisfies the agent, by providing number of object they request that satisfy their conditions.

II. RELATED WORKS

The procedure of passing sensitive data from the distributor to the trusted third parties in a virtual and widely dispersed network always happens frequently in the modern world. Data distributor has provided a group of purportedly trustworthy agents access to sensitive data in order to protect the security and dependability of the service in response to customer demand (third parties). A few of the files had been compromised and were discovered elsewhere.

B. K. Adhikari, W. Zuo, R. Maharjan, X. Han and S. Liang, "Detection of sensitive data to counter global terrorism", Applied Sciences (Switzerland), vol. 10, no. 1, 2020, [online] Available: https://doi.org/10.3390/app10010182: Due to its irregular activities, which result in financial loss, cyberwar, and cybercrime, global terrorism has posed difficulties for the criminal justice system. The correct mining of criminal information from big data for the estimate of possible risk at the national and international levels is therefore a global challenge in the monitoring of terrorist organisation activities. There is little to no literature that addresses these problems by using big data analytical tools and methodologies, despite the widespread success of many conventional methods of computation. This study aims to close the gap in the literature by extracting correct criminal data from the vast amount of different types of data using Hadoop clusters, which will aid social justice organisations in their global fight against terrorism. Many algorithmic strategies were successfully used to accomplish this purpose, including parallelization, annotators and annotations, lemmatization, stop word Remover, term frequency and inverse document frequency, and singular value decomposition. Using the same hardware, software, and system configuration, the effectiveness of this work is empirically compared. Additionally, criminal data was used to test the experiment's effectiveness in terms of concepts and matching results. . Ultimately, the testing findings demonstrated that the suggested method could uncover criminal material with 100% accuracy, while matching numerous criminal terms with documents could only achieve 80% accuracy. The effectiveness of this strategy was also demonstrated in many node clusters. Finally, the research opens security authorities' minds to fresh perspectives on fighting terrorism on a worldwide scale.

Alzahrani, A. Alqazzaz, N. Almashfi, H. Fu and Y. Zhu, "Web Application Security Tools Analysis", Studies in Media and Communication, vol. 5, no. 2, pp. 118, 2017, [online] Available: https://doi.org/10.11114/smc.v5i2.2663: Effective web

application security is essential for the success of your online presence. The necessity of security has significantly increased, particularly for web applications. Due to the abundance of tools at hand as well as the infancy of the sector, dealing with web application or website security issues involves in-depth analysis and strategy. Finding the right tools thus necessitates a thorough understanding and a number of processes, including a consideration of the complexity of the web applications as well as the development environment and business requirements. In this article, we first present the architecture of web apps before listing and analysing the common security flaws. These flaws are: HTTP Splitting, Cross-Site Scripting, SQL Injection, Information Leakage, and Insufficient Transport Layer Protection. The methods that are used to check for these pervasive vulnerabilities in web applications are also examined in this research. Finally, it assesses tools based on security flaws and makes suggestions to users and managers of online applications in an effort to inform them.

N. F. Awang and A. A. Manaf, "Detecting Vulnerabilities in Web Applications Using Automated Black Box and Manual Penetration Testing", Communications in Computer and Information Science, vol. 381 CCIS, pp. 230-239, 2013, [online] Available: https://doi.org/10.1007/978-3-642-40597-6_20**:** Web applications are currently the most widely used technology for providing consumers with a variety of services. However, prior investigation and study shown that a significant number of deployed web apps have serious security flaws. One of the well-known methods that is regularly used to find security flaws in web applications is penetration testing. This method can be carried out manually or with the aid of automated tools. However, a prior study found that automated black box techniques had a higher false positive rate and had found more vulnerabilities. In order to improve accuracy in vulnerability detection in online applications, this study developed a methodology that includes both automated black box testing and manual penetration testing.

J. Bozic and F. Wotawa, "Planning-based security testing of web applications with attack grammars", Software Quality Journal, vol. 28, no. 1, pp. 307-334, 2020, [online] Available: https://doi.org/10.1007/s11219-019-09469-y: Web apps are used on computers all around the world and provide nearly universal accessibility. These programmes provide continuous, functional interconnection between various parts. Data confidentiality and secure authentication rank among the top needs. However, implementation errors and unmet requirements frequently lead to security breaches that nefarious individuals finally took advantage of. Applying various testing techniques in this context is crucial for identifying software flaws early in the development process and averting unwanted access. We make a contribution to test automation for web applications in this study. We put a lot of emphasis on preparation for testing, where we offer supporting models that cover attacks and how to use them to test web apps. The planning approach surpasses the limitations of conventional graphical representations and offers a high degree of extendibility and configurability. As a result of improved vulnerability identification brought on by new testing opportunities, web services and apps are made to be more secure.

D. J. I. Z. Chen, S and D. S, "Social Multimedia Security and Suspicious Activity Detection in SDN using Hybrid Deep Learning Technique", Journal of Information Technology and Digital World, vol. 2, no. 2, pp. 108-115, 2020, [online] Available: https://doi.org/10.36548/jitdw.2020.2.004: With increased usage and ongoing development of multimedia-based services and applications, social media traffic is expanding tremendously. Secure data transmission allows for the realisation of criteria such as quality of service (QoS), quality of information (QoI), scalability, reliability, and other elements

crucial for social multimedia networks. Multimedia analytics is carried out using a trust-based paradigm to give timely and actionable information to meet the user's increasing demands. Limiting certain features, such as energy-aware networking and runtime security in Software Defined Networks, makes it easier to administer and govern the network effectively. In order to increase the SDN dependability, a hybrid deep learning based anomaly detection system is used to detect suspicious flows in social multimedia contexts. The entire process is divided into two modules: one that facilitates anomaly detection using support vector machines based on gradient descent and improved restricted Boltzmann machines, and the other that uses end-to-end data delivery to meet the strict QoS requirements of low latency and high bandwidth in SDN. Data distribution and anomaly detection services are crucial in social multimedia to increase the system's efficacy and efficiency. For this, we empirically test the suggested scheme using benchmark datasets and real-time evaluation. Using the CMU-based insider threat dataset for large-scale analysis, detection of hostile events such sensitive data collection, profile cloning, and identity theft are done to assess the system's performance.

III. Methodology

We provide data distribution plans (across the agents) that increase the likelihood of finding leaks. These techniques don't rely on changing the data that has been made public (e.g., watermarks). To increase our chances of detecting leakage and locating the offender, we can occasionally add "realistic but phoney" data records.

Dijkstra's algorithm: Dijkstra's algorithm is an iterative algorithm that provides us with the shortest path from one particular starting node (a in our case) to all other nodes in the graph.To keep track of the total cost from the start node to each destination we will make use of the distance instance variable in the Vertex class. The distance instance variable will contain the current total weight of the smallest weight path from the start to the vertex in question. The algorithm iterates once for every vertex in the graph; however, the order that we iterate over the vertices is controlled by a priority queue (actually, in the code, I used heapq). The value that is used to determine the order of the objects in the priority queue is distance. When a vertex is first created distance is set to a very large number. When the algorithm finishes the distances are set correctly as are the predecessor (previous in the code) links for each vertex in the graph.

The Advanced Encryption Standard (AES): The encryption algorithms that we have examined so far all have certain flaws. On contemporary computer platforms, the older cyphers are easily cracked. In 1998, a machine that cost roughly $250,000 was used to crack the DES algorithm. As it was created for mid-1970s hardware and did not produce effective software code, it was also much too slow in terms of software. Contrarily, Triple DES is three times slower and has three times as many rounds as DES. In addition, the efficiency and security of the 64 bit block size used by triple DES and DES are questionable. A completely new encryption algorithm was needed. one that would be impervious to all recognised assaults. A new standard needed to be created, and the National Institute of Standards and Technology (NIST) wanted to assist. However, this was likely to arouse strong scepticism due to the controversy surrounding the DES algorithm and the years in which some branches of the U.S. government did everything they could to prevent the deployment of secure cryptography. The issue was that NIST wanted to directly contribute to the development of a new, great encryption standard but was unable to do so. Sadly, they were the only ones with the technical standing and resources to take the initiative.

Instead of creating or aiding in the creation of a cypher, they decided to organise a competition in which anybody could participate. The goal of the competition, which was launched on January 2nd, 1997, was to create a new encryption algorithm that would be used to safeguard sensitive, non-classified, U.S. government data.

There were several requirements for the cyphers, and the entire design had to be well documented (unlike the DES cipher). Following the submission of the proposed algorithms, several years of review in the form of cryptographic conferences occurred. 15 algorithms were accepted in the competition's first round before being reduced to 5 in the second. They are the fifteen algorithms. Table 7 displays the fifteen algorithms, with the five chosen ones highlighted in bold. Some of the most accomplished and well-known cryptographers in the world, as well as NIST itself, tested the algorithms for effectiveness and security

The AES cipher: AES is a symmetric block cypher, just like DES. This indicates that the encryption and decryption processes use the same key. AES, however, differs significantly from DES in a number of respects. Other block and key sizes besides the 64 and 56 bits of DES' block and key size are supported by the Rijndael algorithm. In reality, the block and key don't have to match and can be chosen from a range of 128, 160, 192, 224, or 256 bits. The AES standard specifies that the method can only handle keys with a choice of three different key lengths: 128, 192, or 256 bits, and a block size of 128 bits. The standard's name is changed to AES-128, AES-192, or other variations depending on which version is utilised. . The name of the standard is changed to AES-128, AES-192, or AES256 depending on which version is being utilised. AES differs from DES in addition to these aspects by not being a feistel structure. Remember that a feistel structure uses one half of the data block to change the other half of the data block before swapping the two parts. In this example, permutations and replacements are used to process the full data block concurrently during each round.

A number of AES parameters depend on the key length. For example, if the key size used is 128 then the number of rounds is 10 whereas it is 12 and 14 for 192 and 256 bits respectively. At present the most common key size likely to be used is the 128 bit key. This description of the AES algorithm therefore describes this particular 59 Chapter 7 The AES Algorithm implementation. Rijndael was designed to have the following characteristics:

- Resistance against all known attacks.
- Speed and code compactness on a wide range of platforms.
- Design Simplicity.

The overall structure of AES can be seen in 7.1. The input is a single 128 bit block both for decryption and encryption and is known as the in matrix. This block is copied into a state array which is modified at each stage of the algorithm and then copied to an output matrix (see figure 7.2). Both the plaintext and key are depicted as a 128 bit square matrix of bytes. This key is then expanded into an array of key schedule words (the w matrix). It must be noted that the ordering of bytes within the in matrix is by column. The same applies to the w matrix.

IV. Conclusion

Enterprise resource planning has helped the life sciences and the healthcare industries of today produce enormoa large volume of data is challenging and internal employee intimidation around data leakage is on the rise, businesses are adopting security measures like Data Loss Prevention. In order to handle large amounts of data with increasing efficiency and a set of rules that give workers the

outcomes they need, cloud computing approaches are deployed. It lessens the issue of workers having to choose specific components to address issues for supervised, unsupervised, and semi-supervised healthcare data's.

V. REFERENCES

[1]. Chen Xiaolin, Feng Junwen. A Study on Information Security Management of Agricultural Supply Chain. Science of Science and Management of Science and Technology. 2007(11) :38-42.

[2]. Chi T H, Zhou X. Web GIS Resolution for China's Sustainable Development. Resources Science [J] 2001, 23 (1): 34-39.

[3]. Fearne A, Hughes D.Success factors in the fresh produce supply chain [J].British Food Journal, 2000, 102(10):760-772.

[4]. Han Y,Geng H.VINCA-A visual and personalized business-level composition language for chaining web-based services[C].Proceedings of the 1st International Conference on Service-Oriented Computing.Springer-Verlag,2003:165-177.

[5]. Hau L. Lee, Seungjin Whang. Information sharing in a supply chain [J]. International Journal of Technology Management, 2000, 20, (3):373-387.

[6]. Lambert D M, Cooper M C.Issues in supply chain management [J].Industrial Marketing Management. 2000, (29):65-83

[7]. Li Yan-xia, Zhao Qing-zhei. The Design of Management Information System of Agricultural Products. Computer Technology and Application. [J].2005(4):93-95.

[8]. [Qi Yuan. Research on Information Sharing in Supply Chain [D]. Shang Hai University. 2002, P49-50.

[9]. Song ke, Shi Shenghui. The Implementation of Information System Based on Browser/Server Structure. North China Electric Power [J].2001(3):12-14.

Cite this article as :

S. Fazloon, Mrs P. Poornima, "Detection of Data Leakage Using Cloud Computing", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 8 Issue 6, pp. 76-81, November-December 2022.

Journal URL : https://ijsrcseit.com/CSEIT22863