IJARSCT

ISSN (Online) 2581-9429

IJARSCT

Impact Factor: 7.301

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Volume 3, Issue 12, May 2023

Data Leakage Detection using Cloud Computing

Isha Gujarathi, Unnati Komal, Prachi Chandra, Kumar Shivam, Prof. N. H. Deshpande

Department of Information Technology

Sinhgad College of Engineering, Pune, Maharashtra, India

nhdeshpande.scoe@sinhgad.edu

Abstract*: Data leakage has become a critical concern for organizations, as sensitive information can be unintentionally or maliciously disclosed, leading to severe consequences such as financial losses, reputation damage, and legal implications. Cloud computing offers a scalable and cost-effective solution for data storage and processing, but it also introduces new challenges in terms of data security and privacy. The proposed approach leverages the capabilities of cloud computing to enhance data leakage detection.*

Keywords: Cloud Computing, Data security, Machine Learning, Risk Mitigation, Anomaly Detection.

I. INTRODUCTION

1.1 BACKGROUND

Data leakage refers to the unauthorized or accidental disclosure of sensitive information, such as personal data, financial records, trade secrets, or intellectual property. It is a significant concern for organizations, as data breaches can lead to severe consequences, including financial losses, damage to reputation, legal implications, and loss of customer trust. With the rapid growth of cloud computing, where data is stored and processed on remote servers, the risks associated with data leakage have become even more pronounced.

Traditional methods of data leakage prevention, such as firewalls and intrusion detection systems, are not always effective in the cloud environment. Therefore, there is a need for specialized techniques and approaches that can specifically address data leakage in cloud computing. Detecting data leakage in cloud computing requires proactive monitoring, analysis of data access patterns, user behaviors, and network traffic, and the ability to identify anomalous activities that may indicate potential data leakage incidents. Machine learning algorithms play a crucial role in this process by learning from historical data leakage incidents and identifying patterns that signify potential risks.

1.2 RELEVANCE / MOTIVATION

Data breaches and data leakage incidents have become a common occurrence, leading to significant financial losses and reputational damage for organizations. The exposure of sensitive data can result in legal consequences and loss of customer trust. The motivation to detect and prevent data leakage in the cloud stems from the need to protect sensitive information and maintain data security and privacy.

The advancements in machine learning algorithms and cloud computing technologies provide an opportunity to develop more sophisticated and efficient data leakage detection approaches. Machine learning algorithms can analyze vast amounts of data, detect patterns, and identify anomalies that may indicate potential data leakage. Leveraging the scalability and resources of cloud computing, these algorithms can be deployed effectively to monitor and analyze data access patterns, user behavior, and network traffic.

II. SPECIFICATIONS

2.1 PROBLEM STATEMENT

The focus of our project is to develop a robust system for detecting and preventing data breaches in a distributor's network, with a particular emphasis on safeguarding sensitive information. Additionally, we aim to implement mechanisms that can help trace and identify the potential sources or agents responsible for the data leakages.

DOI: 10.48175/IJARSCT-10683

HARDWARE REQUIREMENTS

Processor	- Intel i3/i5/i7
Speed	- 3.1 GHz
RAM	- 4 GB(min)
Hard Disk	- 40 GB

SOFTWARE REQUIREMENTS

Operating System	- Windows 7/8/10
Application Server	- Apache Tomcat 7/8/9
Front End	- HTML, JDK 1.8, JSP
Scripts	- JavaScript.
Server side Script	- Java Server Pages.
Database	- My SQL
IDE	- Eclipse

III. DESIGN

3.1 SYSTEM ARCHITECTURE

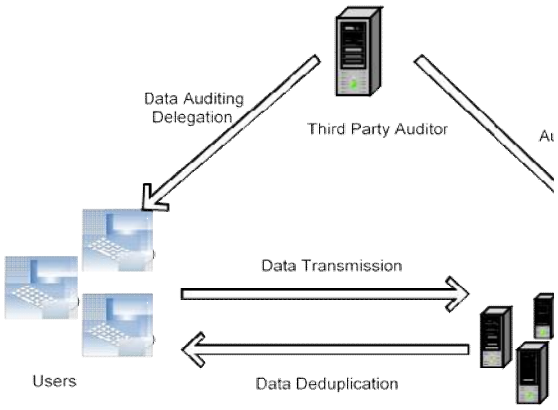


Figure 3.1. System Architecture

System Architecture is shown in figure 1. The entire system works in stages :

Cloud Servers

The cloud computing framework is an information-technology (IT) concept that enables all kinds of popular classes of configurable tools to be available easily and often via the Internet (such as computer networks, databases, processing, applications, and services). Cloud computing provides consumers and businesses with varying computational capabilities, the option of storing data in private servers, or on a third party database in a data center, thereby increasing the efficiency and consistency of data access systems.

Data Deduplication

Duplication of data occurs when the data owner attempts to archive the same information already contained in the CSP. The CSP should check it by comparing tokens. In the case of a successful match, the CSP should contact the deduplication system supplying the data holder with the token and public key.

Storage Management

Some storage management techniques, including virtualized storage, deduplication along with compression, allow businesses to use current data storage efficiently. The advantages of these approaches include reduced cost, the specific spending on resources for processing facilities as well as continuing operational costs for the repair of these systems. Network and equipment management is also simplified through the majority of storage management techniques. In order to save time and money, the number of IT employees required for maintaining their storage systems can even be reduced by businesses and the total cost of storage in turn can be reduced. The quality of the data center can also be enhanced by data management. The compression and engineering, for instance, will increase the distribution and automation of storage assets to various applications.

3.2 SYSTEM USE- CASE DIAGRAM

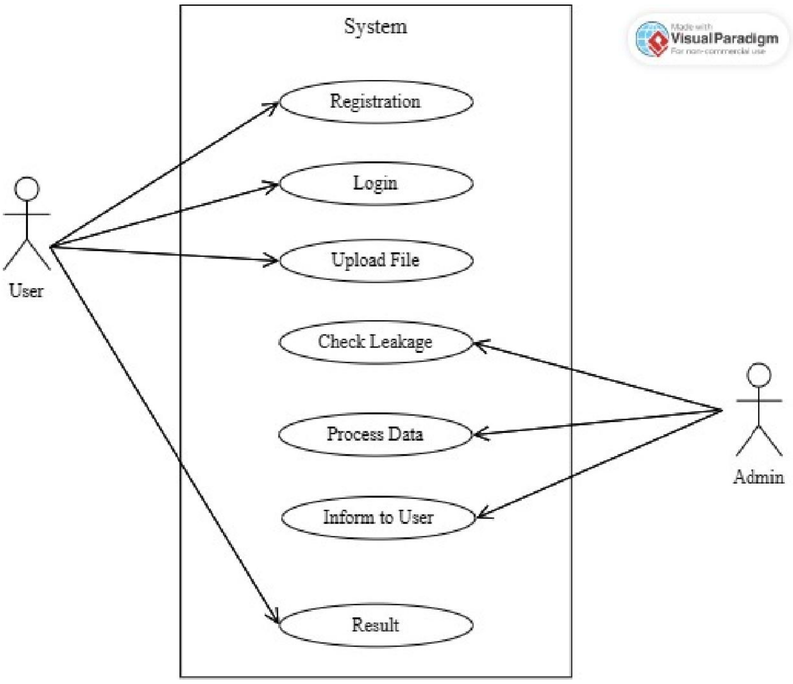


Figure 3.2. Use- case Diagram

3.3 SYSTEM COMPONENT DIAGRAM

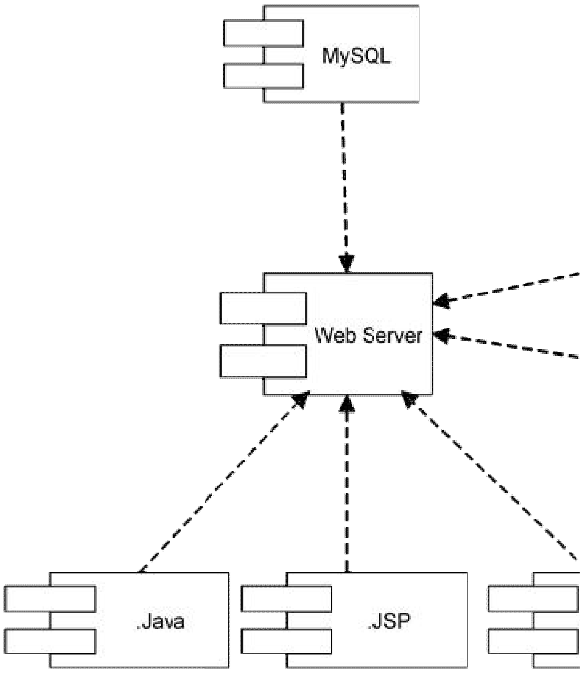


Figure 3.3. Component Diagram

IV. IMPLEMENTATION

1] Choose a Cloud Platform

We have Selected a Cloud Platform that suits our requirements i.e Amazon Web Services (AWS). AWS (Amazon Web Services) is a popular cloud service provider, and it offers a wide range of services and features that make it well-suited for data leakage detection using cloud computing. Here are some reasons why AWS is often chosen for this purpose: Scalability, Data Storage and Security , Data Processing, Monitoring and Alerting , Integration with Other Services , Compliance and Privacy.

2] Setup Cloud Environment

Set up your cloud environment by creating virtual machines or containers to deploy your system components. Ensure you have the necessary networking and security configurations in place.

3] Data Collection

Define the data sources you want to monitor for potential data leakage. This could include databases, file systems, APIs, or network traffic. Use appropriate mechanisms to collect and store the data securely in the cloud environment.

4] Data Preprocessing

Perform preprocessing tasks on the collected data, such as cleaning, filtering, and transforming it into a suitable format for analysis. This step ensures that the data is in a standardized and usable form.

5] Algorithm Selection

According to the K Anonymity definition, it is used for detecting and mitigating data leakage in the context of topic data within a cloud computing environment. Specifically, we seek to investigate how the K-anonymity algorithm can be utilized to anonymize sensitive topic data while preserving its utility for analysis, thereby minimizing the risk of data leakage and ensuring the privacy of individuals' information. Furthermore, we aim to evaluate the effectiveness and efficiency of the K-anonymity algorithm in the cloud computing setting and propose any necessary enhancements or modifications to optimize its performance in data leakage detection.

6] Algorithm steps

Step 1 - Data Generalization: Identify sensitive attributes in the dataset.

Apply generalization techniques to transform the values of sensitive attributes into broader categories or ranges.For example, generalize an age attribute into age groups or generalize a zip code attribute into regions.

Step 2 - Attribute Suppression:

Remove or suppress any personally identifiable information (PII) or sensitive attributes that could lead to the identification of individuals. This can include removing fields like names, addresses, or any other direct identifiers.

Step 3 - Partitioning:

Divide the dataset into partitions or groups based on common characteristics or quasi-identifiers.Quasi-identifiers are non-sensitive attributes that can potentially be used in combination to re-identify individuals.Ensure that each partition has a minimum size of K to achieve K-anonymity.

Step 4 - Generalization Hierarchy: Create a generalization hierarchy for each quasi-identifier attribute. A generalization hierarchy defines the levels of generalization for an attribute, allowing for different levels of anonymity. Determine the appropriate level of generalization for each attribute to achieve the desired K-anonymity level.

Step 5 - Data Perturbation: Apply data perturbation techniques to add random noise or perturbations to the values of certain attributes Perturbed values should preserve the statistical properties of the original data while making it more difficult to identify individuals.

Step 6 - Evaluation and Validation:

Assess the effectiveness of the K-anonymity algorithm by measuring the degree of anonymity achieved. Conduct privacy and utility evaluations to ensure that the anonymized dataset still maintains useful information while protecting individual privacy.

Step 7 - Testing:

Perform testing of the K-anonymity algorithm using generated test data to validate its effectiveness and evaluate its impact on data utility.

DOI: 10.48175/IJARSCT-10683

7] Testing

Testing of project problem statement using generated test data (using mathematical models, GUI, Function testing principles, if any) selection and appropriate use of testing tools, testing of UML diagram's reliability.

Test Case_ID	Description	Test case I/P	Actual Result	Expected result	Test case criteria (P/F)
101	Enter the case insensitive Username click on Submit button.	Username	Error comes	Error Should come	P
102	Enter the case sensitive Username click on Submit button.	Username	Accept	Accept Username	P
201	Enter the case insensitive Password click on Submit button.	Password	Error comes	Error Should come	P
202	Enter the case sensitive Password click on Submit button	Password	Accept	Accept	P
301	Enter the case insensitive Mobile Number click on Submit button	Mobile Number	Error comes	Error Should come	P
302	Enter the case sensitive Mobile Number click on Submit button.	Mobile Number	Accept	Accept	P

Table 1 : Test case

Module-ID:-01

Modules to be tested:-Registration

1. Enter the case insensitive Username click on Submit button.

Expected: It should display error.

2. Enter the case sensitive Username click on Submit button.

Expected: It should accept.

3. Enter the case insensitive Password click on Submit button.

Expected: It should display error.

4. Enter the case sensitive Password click on Submit button.

Expected: It should accept.

5. Enter the case insensitive Mobile Number click on Submit button.

Expected: It should display error.

6. Enter the case sensitive Mobile Number click on Submit button.

Expected: It should accept.

7. Enter the wrong address and click on Submit button.

Expected: It should display error.

8. Enter the correct address and click on Submit button.

Expected: It should accept.

DOI: 10.48175/IJARSCT-10683

Test Case_ID	Description	Test case I/P	Actual Result	Expected result	Test case criteria (P/F)
001	Enter the correct username and wrong password click on Login button.	Username Password	Error comes	Error Should come	P
002	Enter the wrong username and correct password click onLogin button,	Username Password	Error comes	Error Should come	P
003	Enter the correct username and password and click on Login button.	Username Password	Accept	Accept	P

Table 2: Test Cases

Module-ID:-2

Modules to be tested:- Login

1. Enter the correct username and wrong password click on Submit button.

Expected: It should display error.

2. Enter the wrong username and correct password and click on Submit button.

Expected: It should display error.

3. Enter the correct username and password and click on Login button.

Expected: It should display welcome page.

4. After login with valid credentials click on back button.

Expected: The page should be expired.

5. After login with valid credentials copy the URL and paste in another browser.

Expected: It should not display the user's welcome page.

6. Check the password with Lower case and upper case.

Expected: Password should be case sensitive.

8] Deployment

Once you are satisfied with the performance of your trained models, deploy them in your cloud environment. This deployment can be in the form of cloud functions, APIs, or microservices to facilitate real-time or batch processing of incoming data.

9] Monitoring and Alerts

Continuously monitor the deployed models to identify potential data leakage incidents. Set up alerts or notifications to notify relevant stakeholders whenever a data leakage event is detected.

10] Maintenance and Updates

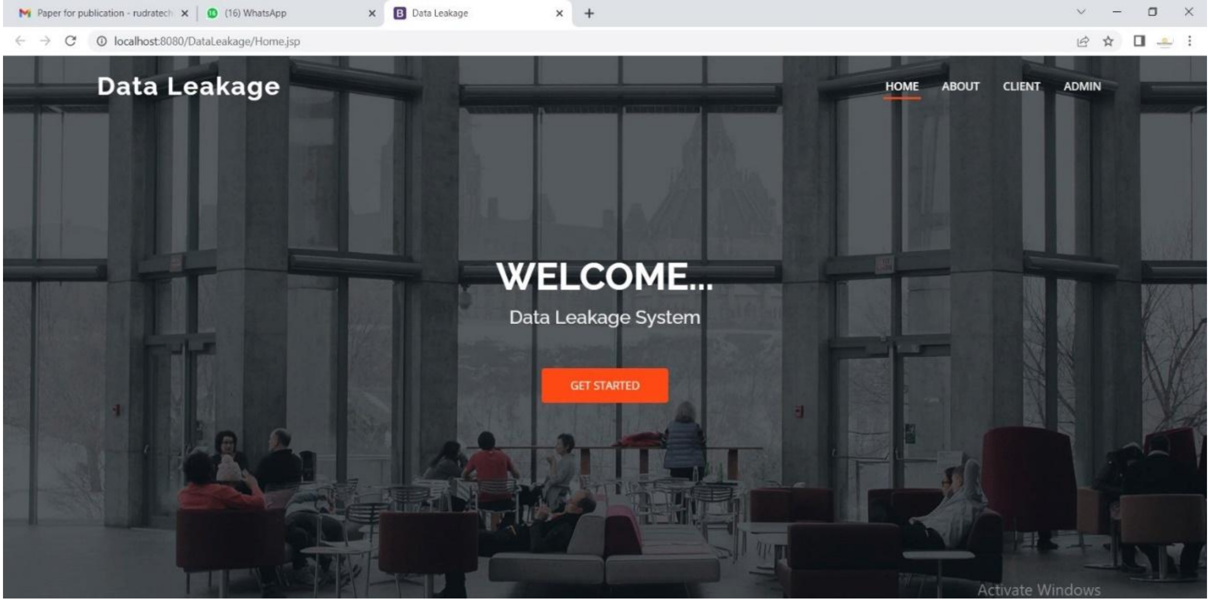
Regularly update your system with the latest security patches, algorithm improvements, or changes in the data sources. Monitor the system's performance and make necessary adjustments to ensure optimal results.

DOI: 10.48175/IJARSCT-10683

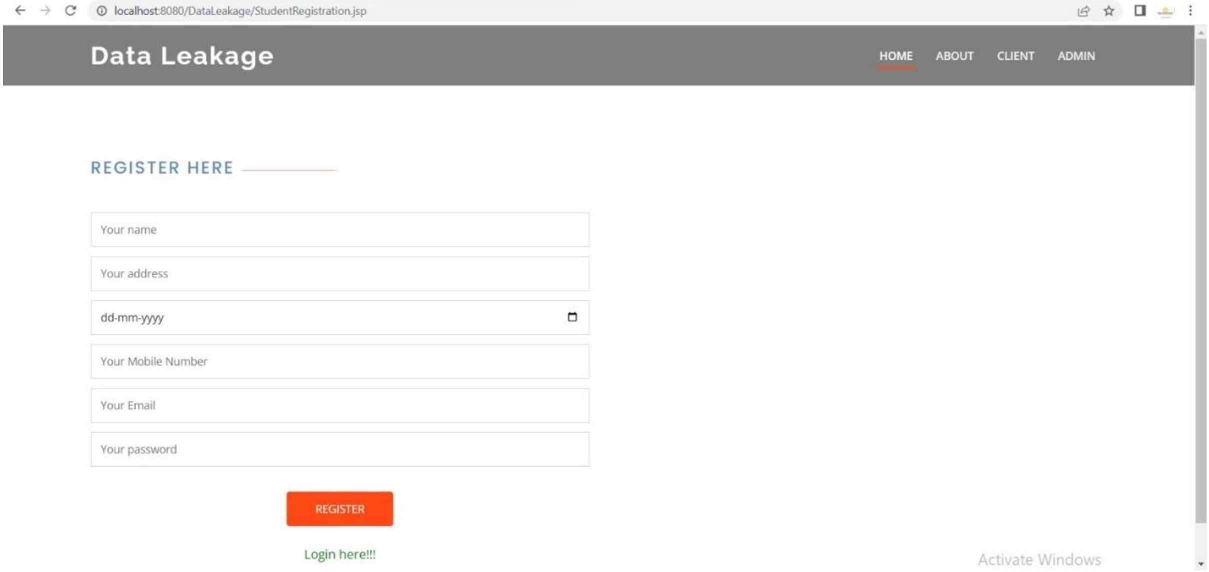
V. RESULTS AND EVALUATION

5.1 Screenshots of Results

Welcome Page :

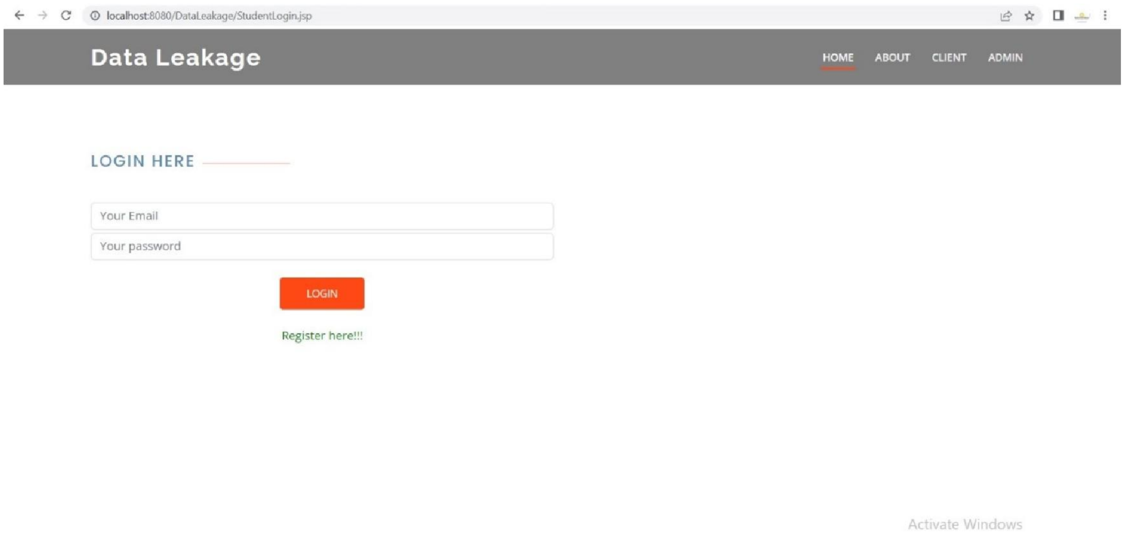


Registration Page :



DOI: 10.48175/IJARSCT-10683

Login Page :

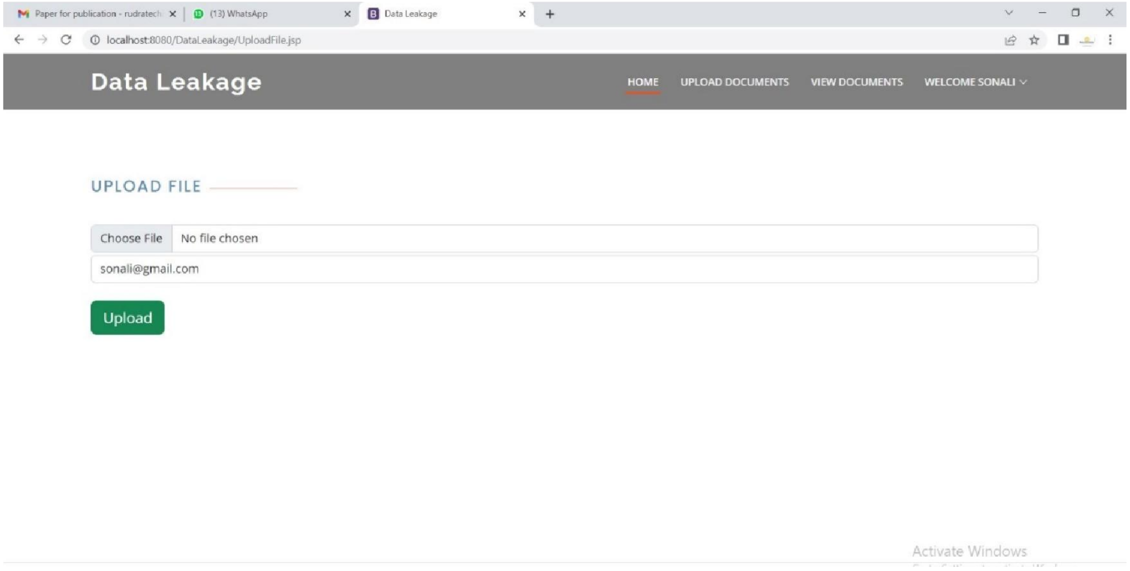


LOGIN HERE

Your Email

Your password

File Uploading Page

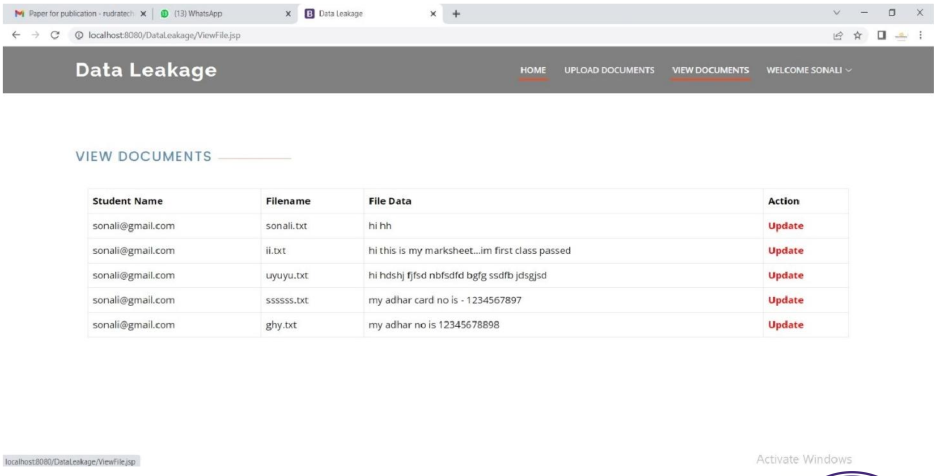


UPLOAD FILE

No file chosen

sonali@gmail.com

View Documents :

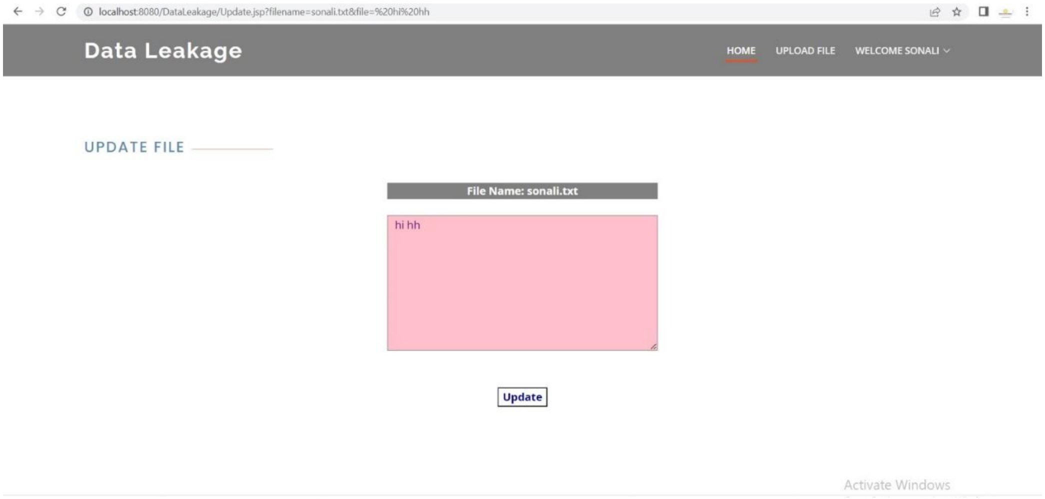

Paper for publication - rudratech
(13) WhatsApp
Data Leakage
localhost:8080/DataLeakage/ViewFile.jsp
Data Leakage
HOME
UPLOAD DOCUMENTS
VIEW DOCUMENTS
WELCOME SONALI
localhost:8080/DataLeakage/ViewFile.jsp
Activate Windows

VIEW DOCUMENTS

Student Name	Filename	File Data	Action
sonali@gmail.com	sonali.txt	hi hh	Update
sonali@gmail.com	ii.txt	hi this is my marksheet...im first class passed	Update
sonali@gmail.com	uyuyu.txt	hi hdshj fjfsd nbfsdfd bgfg ssdfb jdsgjsd	Update
sonali@gmail.com	ssssss.txt	my adhar card no is - 1234567897	Update
sonali@gmail.com	ghy.txt	my adhar no is 12345678898	Update

DOI: 10.48175/IJARSCT-10683

Updating File :



localhost:8080/DataLeakage/Update.jsp?filename=sonali.txt&file=%20hi%20hh

File Name: sonali.txt

hi hh

VI. CONCLUSION

In conclusion, data leakage detection using cloud technology provides organizations with advanced capabilities to protect sensitive information and prevent unauthorized access. By leveraging the scalability, real-time monitoring, and centralized management offered by cloud-based solutions, organizations can proactively identify and mitigate data leaks. This approach enhances data security, maintains customer trust, and ensures compliance with regulatory requirements. However, it is important to complement cloud-based detection with other security measures, such as encryption, access controls, and user training, for a comprehensive data protection strategy. Overall, data leakage detection using the cloud is a critical component of a robust data security framework in today's digital landscape. Aditionally, we have provided more Scalability and Security for user.

REFERENCES

[1]. Ryan, D. Mark, ―Cloud computing privacy concerns on our doorstep.‖ Communications of the ACM 54.1 (2011): 36-38.

[2]. Srijanya K and N. Kasiviswanath, ―Data Integrity Verification by Third Party Auditor in Remote Data Cloud,‖ International Journal of Soft Computing and Engineering, 3(5), 2013.

[3]. K. He, C. Huang, J. Shi and J. Wang, ―Public Integrity Auditing for Dynamic Regenerating Code Based Cloud Storage,‖ IEEE Symposium on Computers and Communication (ISCC), 2016.

[4]. P. Mell and T. Grance, ―The NIST definition of cloud computing,‖ Communications of the ACM, vol. 53, no. 6, 2010.

[5]. K. Yang and X. Jia, ―An Efficient and Secure Protocol for Ensuring Data Storage Security in Cloud Computing,‖ IEEE Transactions on Parallel and Distributed Systems, 2012.

[6]. D. Meister, J. Kaiser, and A. Brinkmann, "Block locality caching for data deduplication," in Proc. 6th Int. Syst. Storage Conf., 2013, pp. 1–12.

[7]. M. Lillibridge, K. Eshghi, and D. Bhagwat, "Improving restore speed for backup systems that use inline chunk-based deduplication," in Proc. 11th USENIX Conf. File Storage Technol, Feb. 2013, pp. 183–197.

[8]. V. Tarasov, A. Mudrankit, W. Buik, P. Shilane, G. Kuenning, and E. Zadok, "Generating realistic datasets for deduplication analysis," in Proc. USENIX Conf. Annu. Tech. Conf., Jun. 2012, pp. 261–272.

[9]. D. T. Meyer and W. J. Bolosky, "A study of practical deduplication," ACM Trans. Storage, vol. 7, no. 4, p. 14, 2012.

[10]. G. Wallace, F. Douglis, H. Qian, P. Shilane, S. Smaldone, M. Chamness, and W. Hsu, "Characteristics of backup workloads in production systems," in Proc. 10th USENIX Conf. File Storage Technol., Feb.2012,pp.33–48.

DOI: 10.48175/IJARSCT-10683