

Datashadow – Privacy First Cloud Data Exposure & Leakage Detection System

Sabitha K¹, Sabaripriya S², Suganthi R³

Sri Shakthi Institute Of Engineering And Technology ,Coimbatore Tamil Nadu India

*Corresponding Author

E-Mail Id: Sabithacys@Siet.Ac.In

ABSTRACT: Why have so many teams migrated their digital records to cloud setups? Mostly convenience, since moving information around off-site fundamentally altered daily administrative work. Yet these remote spaces still invite trouble. They leave confidential records exposed, let files slip out through unapproved sharing, leak company data, and host fishy account behavior. Older defenses mainly checked credentials, granted permissions, and handled basic gatekeeping, which leaves a noticeable blind spot. Teams still need active eyes on the actual files moving through the pipeline. To tackle this gap, we built DataShadow: a Privacy-First Cloud Data Exposure and Leakage Detection System with Risk-Based Analysis and Automated Incident Response. It tracks file movements, inspects uploads, catches sensitive text, and weighs potential threats using one unified interface. The codebase relies on Python alongside Flask, keeps structured tables inside MySQL, and hooks directly into Amazon Web Services (AWS) for hosting. In practice, operators get standard login tools, active session tracking, and a primary console where they inspect document histories and security alerts. Its tracking pipeline logs core metadata right away, noting the owner, upload timestamp, name, format, and byte size. Once a file lands, the system generates a SHA-256 hash, checks for dubious file extensions, sifts through content for private records, and grades overall danger. Each document receives a score from zero to one hundred, landing in Safe, Medium, High, or Critical tiers. Predefined rules then tag the record as Public, Internal, Confidential, or Restricted. DataShadow also runs User Behavior Analytics (UBA) against raw activity streams. It flags odd routines, such as persistent failed logins or bursts of uploading, downloading, and external sharing. Once an account crosses designated alert thresholds, the system flags possible insider threats and logs formal incidents. The system can lock down accounts on its own. When high-risk or critical triggers trip, the automated incident response tool logs the measure and restricts access, which prevents further damage. You also get a clean paper trail: audit logs, download records, sharing tracking, checksum integrity checks, origin data, and unbroken chain-of-custody details all map out what happens to each file. How does the AWS side handle this? DataShadow relies on Amazon Simple Storage Service (Amazon S3) for object storage, wired in via the Python Boto3 library. When an event crosses into the high-risk bracket, Amazon Simple Notification Service (Amazon SNS) pushes out email notices. This bridge tethers the core analysis to cloud storage and outward paging. From there, everything surfaces in a single interface: alerts, observed user habits, open incidents, risk ratings, labels, and audit

Keywords: Cloud Security, Data Leakage Detection, Sensitive Data Detection, User Behavior Analytics, Risk Assessment, Data Classification, SHA-256, File Integrity Monitoring, Insider Threat Detection, Automated Incident Response, Amazon S3, Amazon SNS.

I. INTRODUCTION:

Modern work runs on cloud platforms to hold, compute, and handle digital records. It scales well, adapts easily, and keeps storage expenses reasonable for single users or whole companies. Yet leaning so

heavily on these remote systems invites trouble: sensitive files drift into the open, people share assets without permission, odd account behaviors crop up, and leaks happen. The usual defenses tend to stop at who gets through the door, relying on login checks, user permissions, basic encryption, and server-level armor. In practice, you need a closer eye on what

people do with documents after fetching them. To plug these gaps, I built DataShadow: A Privacy-First Cloud Data Exposure and Leakage Detection System with Risk-Based Analysis and Automated Incident Response. The software tracks file movements, inspects incoming uploads, flags private information, and spots hazards from one consolidated console. It relies on Python with Flask on the backend, alongside MySQL to retain user profiles, document logs, security notices, system events, and case files. Users log in, retain active sessions, and review findings on a clean dashboard. During inspection, the pipeline looks for delicate content, spots questionable extensions, and generates SHA-256 hashes so we can verify file integrity. A risk scoring engine marks items on a scale from zero to one hundred, assigning tiers of Safe, Medium, High, or Critical. Beside that, an indexing step labels records as Public, Internal, Confidential, or Restricted. How do we catch internal misuse? DataShadow folds in User Behavior Analytics, scanning for patterns like strung-together bad passwords, rapid-fire uploads, quick downloads, or reckless file distributions. Once an account crosses fixed safety limits, the tool flags a plausible inside risk and logs an incident report. When high-risk or critical triggers trip, the automated incident response module logs the event and shuts down offending user accounts on the spot. It also handles routine security auditing, watches downloads, tracks sharing history, checks file integrity, establishes file provenance, and maintains chain of custody. Investigators rely on these records, which lay out an orderly timeline of everything that happened to a file. How does it tie into the cloud? DataShadow stores assets in Amazon Simple Storage Service (Amazon S3) and relies on Python's Boto3 library to handle AWS calls,

while Amazon Simple Notification Service (Amazon SNS) shoots off email warnings whenever severe file anomalies occur. Everything surfaces in one central console: file metadata, risk assessments, warning flags, user sessions, breaches, classification tallies, and audit trails. Many tools fixate strictly on access permissions. DataShadow, by contrast, factors in actual file traits, hidden sensitive data, and user habits. To be fair, how well this works in practice rests entirely on detection rules, risk thresholds, granted cloud permissions, and baseline response setups. We still need proper testing

to measure true accuracy, throughput, and system scalability. The platform pulls together cloud data exposure tracking and incident management into a single architecture. We assembled Python, Flask, MySQL, Amazon S3, and Amazon SNS to clarify visibility across stored data, prompt fast interventions, and tighten oversight on sensitive material. Down the road, we might introduce deeper content inspection, sharper anomaly models, tighter cloud auditing, and rigorous security verification. It keeps security logs ready for anyone probing fishy activity. Through that shared console, administrators spot incoming threats and take action. What gives this setup a clearer view of cloud data security is the pairing of file checks, risk scoring, and automated alerts. Tying in AWS tools broadens what the system can handle, adding cloud storage alongside email notices. In practice, DataShadow tries to give teams a dependable, orderly way to spot potential data leaks before handling them. DataShadow keeps watching file actions around the clock, mostly to flag odd patterns. It logs significant user steps and security incidents, which keeps people accountable. Admins can sort through dangerous files and alerts first because of that risk-centered view.

II. LITERATURE SURWAY:

Cloud platforms handle our digital records, running computations and storing files at low cost whether for solo operators or big firms. They adapt quickly. Still, relying so much on offsite servers brings trouble: private materials slip out, unauthorized transfers happen, strange account patterns surface, and breaches occur. Standard defenses focus almost entirely on perimeter barriers like sign-in verifications, basic encryption, user permissions, and server shielding. In practice, you have to watch what users do with documents once they pull them down. To patch these blind spots, I put together DataShadow: A Privacy-First Cloud Data Exposure and Leakage Detection System with Risk-Based Analysis and Automated Incident Response. Can a central console make monitoring manageable? The application tracks document transfers, sifts through uploads, marks private material, and flags exposure risks from a unified screen. The backend uses Python paired with Flask, relying on MySQL to hold onto system events, document logs,

security warnings, user profiles, and incident histories. After logging in, operators keep persistent sessions active while surveying reports across the interface. The processing pipeline searches for tender details, flags suspect extensions, and produces SHA-256 hashes to verify file integrity. A risk engine grades items between zero and one hundred, slotting entries into four specific tiers: Safe, Medium, High, or Critical. Alongside this scoring, an indexing phase tags every record as Public, Internal, Confidential, or Restricted. How do we catch internal misuse? DataShadow incorporates User Behavior Analytics to watch for odd habits, like repeated failed passwords, hasty uploads, brisk downloads, or careless file sharing. Break a set security ceiling, and the system marks the user as a likely insider threat before drafting an incident record. When triggers hit high-risk or critical tiers, automated incident response kicks in, logging the mishap and suspending the guilty account right away. It attends to standard auditing too. The platform tracks download traffic, keeps tabs on sharing logs, verifies file integrity, notes original provenance, and protects the custody chain. That leaves investigators with a clear chronicle of an asset's entire life cycle. How does the

cloud side function? DataShadow keeps objects inside Amazon Simple Storage Service (Amazon S3), using Python's Boto3 library for its AWS requests, while Amazon Simple Notification Service (Amazon SNS) dispatches email alerts on critical file abnormalities. A main dashboard displays everything: metadata, risk scores, alert warnings, active sessions, incidents, classification tallies, and audit logs. Most utilities focus merely on permissions. DataShadow looks closer at raw file traits, masked sensitive material, and worker routines. To be fair, field utility depends on detection rules, risk bounds, assigned cloud privileges, and initial response configurations. We have yet to run thorough tests evaluating precise accuracy, processing volume, and system scaling. The platform unites cloud exposure observation with breach remediation under one roof. We linked Python, Flask, MySQL, Amazon S3, and Amazon SNS together so we could see stored assets plainly, act quickly, and mind sensitive records with fewer blind spots. Later on, we might look into more thorough content inspection, better models for irregular behavior, stricter cloud audits, and formal security verification. Does it help investigators? It keeps

security logs on hand whenever someone needs to inspect suspicious events. Admins work from that single interface to catch hazards and respond. The combination of risk scores, file scans, and automated notifications accounts for why this design gives a clearer picture of cloud data safety. We folded in AWS components to expand capacity, pairing cloud repositories with outbound email alerts. In practice, DataShadow attempts to offer teams a steady, systematic routine for catching leaks before resolving them. DataShadow monitors file activity day and night, primarily to pick out aberrant behavior. By recording notable user moves alongside safety breaches, it keeps everyone accountable. This focus on threat severity lets operators triage high-risk records and warnings before touching smaller matters.

III. PROPOSED FRAMEWORK:

The proposed DataShadow framework provides a structured approach to monitoring cloud data exposure and managing security risks. It combines multiple security modules into a single application to improve visibility into file-related activities. The framework begins with user authentication, which verifies user credentials before granting access to the application. Session management helps control authenticated user access. After successful authentication, users can upload files through the application interface. The file monitoring module records relevant file activities and maintains information about uploaded files. These records help administrators track file-related operations. The system then analyzes uploaded files to identify potentially sensitive information and suspicious file extensions. The sensitive data detection module examines file contents for selected patterns that may indicate confidential information. The suspicious extension detection module identifies file types that require additional attention. These analysis results contribute to the overall risk assessment process. The system generates a SHA-256 hash for file identification and integrity verification. Hash values can be compared to determine whether file contents have changed. The risk analysis module evaluates file-related findings using predefined rules and assigns a score between zero and one hundred. Files are categorized as Safe, Medium, High, or Critical based on the assigned score.

This classification helps administrators identify files that may require further investigation. The data classification module separately categorizes files as Public, Internal, Confidential, or Restricted according to the implemented rules. The User Behavior Analytics module monitors selected user activities, including repeated failed login attempts, frequent uploads, downloads, and excessive sharing. The system records these activities and checks them against configured conditions. When suspicious patterns are detected, the application can generate security events and create incidents. These records help administrators review unusual activities and investigate potential insider threats. The automated incident response module supports selected actions based on predefined security conditions. Depending on the configured rules, the system can record response actions and block user accounts. The alert management module records security alerts and presents relevant information through the dashboard. For qualifying high-risk file events, Amazon Simple Notification Service (Amazon SNS) sends email notifications to the configured subscribers.

The framework also integrates Amazon Simple Storage Service (Amazon S3) for cloud file storage. The Python Boto3 library enables communication between the application and AWS services. This integration extends the framework by supporting cloud storage alongside local file processing. Cloud operations depend on appropriate permissions, service availability, and correct configuration. Security auditing provides records of important application activities and security events. Download monitoring and sharing logs help maintain visibility into file access and sharing operations. File provenance records provide information about file-related history, while chain-of-custody records support the tracking of evidence-related information. File integrity monitoring provides additional support for identifying changes in file contents. The centralized dashboard brings together information from the different modules. It displays file details, analysis results, risk scores, data classification, alerts, user activities, and security incidents. This helps administrators review security information without examining each record separately. The dashboard also supports the review of recorded events and response actions. Overall, the proposed DataShadow framework integrates authentication, file monitoring, sensitive data detection, risk assessment, data classification, behavior

monitoring, auditing, cloud storage, and email notifications. These modules work together to identify potential data exposure risks and support security incident management. The framework aims to improve monitoring, accountability, and response within a unified application. Its effectiveness depends on the accuracy of detection rules, the suitability of risk thresholds, and the reliability of the cloud configuration. Further testing can be used to evaluate detection accuracy, system performance, and scalability.



1. Fig:1 Proposed Framework Diagram

The DataShadow framework illustrates the architecture of a cloud data exposure and leakage detection system. Users can register, log in, upload files, and access the dashboard through the web interface. The application is developed using Python and Flask, with authentication and session management. File monitoring records activities, while file analysis detects sensitive information, suspicious extensions, and generates SHA-256 hashes. The risk assessment module assigns scores from zero to one hundred and categorizes files as Safe, Medium, High, or Critical. Data classification organizes files as Public, Internal, Confidential, or Restricted. User Behavior Analytics monitors suspicious activities and potential insider threats. Automated incident response records actions and can block user accounts when required. Security auditing, download monitoring, sharing records, file provenance, chain-of-custody tracking, and integrity monitoring support security investigations. Amazon S3 provides cloud storage, and Amazon SNS sends high-

risk email alerts. MySQL stores user information, file records, alerts, incidents, and security logs. The centralized dashboard displays security findings and user activities. Finally, the administrator reviews alerts, investigates incidents, and takes appropriate security actions.



2. Fig:2 DataShadow Workflow Diagram

The DataShadow workflow begins when a user securely logs in or registers with the system. The user uploads files, and the system monitors file activities, including uploads, downloads, and sharing. The file analysis module checks for sensitive information, suspicious file extensions, and file integrity using the SHA-256 hashing algorithm. Based on these findings, the risk assessment module calculates a risk score and classifies the file as Safe, Medium, High, or Critical. User Behavior Analytics monitors activities and identifies unusual behavior. The automated incident response module helps manage potential security threats. AWS S3 is used to store uploaded files, while AWS SNS sends security alerts and email notifications. MySQL maintains user information, file records, alerts, and security logs. Finally, the centralized dashboard displays risk findings, classification results, user activities, and audit records, allowing the administrator to review events, investigate incidents, and take appropriate action.

IV. RESULTS AND DISCUSSION:

The DataShadow system provides a centralized approach to cloud data exposure and leakage detection. The system monitors uploaded files, checks for sensitive information and suspicious file extensions, and verifies file integrity using SHA-256 hashing. The risk assessment module calculates a risk score and

classifies files into different risk levels, helping users identify potentially unsafe files. User Behavior Analytics and security auditing support the detection of unusual activities, while automated incident response helps manage potential security incidents. AWS S3 supports cloud file storage, and AWS SNS enables security alerts and email notifications. The centralized dashboard displays file details, risk findings, classification results, alerts, and audit records for easy monitoring. The results demonstrate how the integration of file analysis, risk assessment, cloud services, and security monitoring can improve visibility into potential data exposure. However, the accuracy of risk classification depends on the detection rules and scoring logic, which must be tested with different file types and security scenarios. Overall, DataShadow provides a foundation for improving cloud data protection, monitoring suspicious activities, and supporting timely security decisions.



Fig 3: Authentication Dashboard

The DataShadow system successfully monitors uploaded files and analyzes potential security risks. It identifies sensitive data, checks suspicious file extensions, and calculates risk scores for file classification. The dashboard displays file details, risk levels, alerts, and security activities. AWS S3 and SNS support cloud storage and security notifications. Overall, the system helps improve data exposure monitoring and supports timely security decisions.



Fig 4: Result Dashboard

The DataShadow dashboard provides a centralized interface for monitoring cloud files and identifying potential data security threats. It displays file statistics, sensitive data detection results, risk levels, and security alerts. Users can access file monitoring, risk assessment, security auditing, and incident response features through the dashboard. The interface helps users understand the security status of their data and supports the investigation of suspicious activities.

Version	File Size	File Type	File Name	Notes	File Date
202308-2045a6976-28 900000701_281_28123	3885K	apk	28123.apk	File: Traced File (checksum is uncorrupted)	2023-08-16 16:32Z
202308-2045a6976-28 900000701_281_28123	3885K	apk	28123.apk	File: Traced File (checksum is uncorrupted)	2023-08-16 16:35Z
202308-2045a6976-28 900000701_281_28123	3885K	apk	28123.apk	File: Traced File (checksum is uncorrupted)	2023-08-16 16:37Z
202308-2045a6976-28 900000701_281_28123	3885K	apk	28123.apk	File: Traced File (checksum is uncorrupted)	2023-08-16 16:47Z
202308-2045a6976-28 900000701_281_28123	3885K	apk	28123.apk	File: Traced File (checksum is uncorrupted)	2023-08-16 16:53Z
202308-2045a6976-28 900000701_281_28123	3885K	apk	28123.apk	File: Traced File (checksum is uncorrupted)	2023-08-16 16:55Z
202308-2045a6976-28 900000701_281_28123	3885K	apk	28123.apk	File: Traced File (checksum is uncorrupted)	2023-08-16 16:57Z
202308-2045a6976-28 900000701_281_28123	3885K	apk	28123.apk	File: Traced File (checksum is uncorrupted)	2023-08-16 16:57Z

Fig: 5 Datashadow Secuirty Report

The DataShadow security report provides a detailed overview of file activities and potential security threats. It displays risk scores, risk levels, and reasons for identifying potentially unsafe files. The report includes sensitive data detection, suspicious file extension checks, and file integrity monitoring results. Security

alerts and audit logs help administrators investigate suspicious activities. Overall, the report supports security analysis, incident investigation, and improved cloud data protection.

V. CONCLUSION AND ENHANCEMENT:

DataShadow can also be enhanced with a comprehensive security analytics module to provide deeper insights into file exposure and user activities. The system can maintain a detailed history of file uploads, downloads, sharing operations, and access attempts. Advanced filtering options can help administrators quickly locate suspicious events. A centralized incident management module can organize detected threats according to their severity and status. The application can generate downloadable security reports in PDF and Excel formats. Scheduled reports can provide regular summaries of security activities and risk trends. A notification management feature can allow administrators to configure alert preferences. The system can also introduce customizable risk thresholds based on organizational security policies. Improved file classification rules can help distinguish public, internal, confidential, and restricted information. Automated integrity verification can periodically compare file hashes to identify unauthorized modifications. File provenance tracking can record information about file origins and related activities. A secure chain-of-custody feature can preserve evidence details for forensic investigations. Threat timeline visualization can help administrators understand the sequence of suspicious events. Additional user behavior analytics can identify unusual access patterns and repeated download activities. Cloud audit log integration can provide broader visibility into storage operations. A secure API can allow DataShadow to communicate with other security applications. Containerization using Docker can simplify deployment and improve environment consistency. CI/CD integration can automate application testing and deployment. Database optimization can improve query performance as records increase. Database backup and recovery mechanisms can protect important security records. Improved session management can help reduce unauthorized account access.

Rate limiting can protect login and upload endpoints from excessive requests. Input validation can reduce the risk of malicious file names and unsafe requests. Secure error handling can prevent sensitive technical details from being exposed. A scalable architecture can support more users and larger file collections. Cloud monitoring tools can help track application health and resource usage. Access review features can help administrators identify unnecessary permissions. Security policy configuration can support different organizational requirements. Regular vulnerability assessments can help identify and address security weaknesses.

These enhancements can help DataShadow evolve into a more comprehensive, reliable, and scalable platform for cloud data protection and leakage monitoring.

DataShadow can further improve its security capabilities by introducing intelligent threat prioritization based on the severity and impact of detected events. The system can maintain separate monitoring policies for different categories of files. It can provide administrators with recommendations for resolving identified security issues. A secure recovery mechanism can help restore files affected by unauthorized modifications. The application can support configurable retention periods for security logs and reports. Privacy impact analysis can help assess the potential consequences of data exposure.

Improved alert correlation can combine related events to reduce duplicate notifications. The system can introduce a dashboard view for comparing risk levels across different time periods. Additional validation can ensure that risk scores and displayed security reasons remain consistent. Automated regression tests can verify that normal files are not incorrectly classified as high risk. The file analysis module can support additional document formats and classification rules. A secure administrator approval process can be introduced for critical response actions. Cloud storage permissions can be reviewed periodically to identify excessive access. The application can provide detailed explanations for each risk score to improve transparency. Improved configuration management can simplify deployment across development and production environments. Documentation can be expanded to include installation instructions, troubleshooting procedures, and security guidelines.

User feedback can help improve the usability of the dashboard and reports. Performance tests can evaluate the system under different workloads. Regular updates can help maintain compatibility with cloud services and security tools. These improvements can strengthen DataShadow's reliability, usability, and ability to support privacy-focused cloud data security.

The DataShadow system can be improved further by introducing advanced security features and intelligent monitoring capabilities. Machine learning techniques could be integrated to identify unusual user behaviour and detect previously unknown data leakage patterns. Real-time monitoring could be extended to multiple cloud platforms, enabling centralized security management across different environments. Multi-factor authentication and role-based access control could strengthen user authentication and restrict access to sensitive information. Automated incident response could also be enhanced to quarantine suspicious files, restrict risky sharing activities, and notify administrators immediately. In addition, advanced analytics, customizable reports, and threat visualization could help security teams understand emerging risks and make better security decisions.

REFERENCES

1. S. Markande et al., "SaaS API Monitoring for Data Leakage and Access Anomalies," IEEE Access, 2026, doi: 10.1109/ACCESS.2026.3683585.
2. "Cloud Data Leakage Detection in WeChat Mini-Programs Based on Static and Dynamic Analysis," Proceedings of the 28th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2025. DOI: 10.1109/CSCWD64889.2025.11033378.
3. "Privacy-Aware Secure Data Auditing for Cloud-Based Intelligence of Things Environment," IEEE Internet of Things Journal, 2025. DOI: 10.1109/JIOT.2025.3528117.
4. "An Integrated Framework for Data Security in Cloud Computing Using Deep Learning Techniques," Communications on Applied Nonlinear Analysis, 2025. DOI: 10.52783/cana.v32.3817.

5. "Advanced Cloud Intrusion Detection Framework Using Graph-Based Features, Transformers and Contrastive Learning," *Scientific Reports*, vol. 15, 2025. DOI: 10.1038/s41598-025-07956-w.
6. "Detecting Attacks in Fog and Cloud Computing Environments Using Deep Learning: A Systematic Literature Review," *Computer Networks*, vol. 264, article 111269, 2025. DOI: 10.1016/j.comnet.2025.111269.
7. S. Markande et al., "SaaS API Monitoring for Data Leakage and Access Anomalies: A Privacy-Aware and Adaptive Behavioral Analytics Framework for Enterprise Cloud Systems," *IEEE Access*, vol. 14, pp. 59706–59720, 2026. DOI: 10.1109/ACCESS.2026.3683585.
8. U. B. Lokhande and K. Sonawane, "Adaptive Intrusion Detection System for Cloud Security Using Deep Learning," *Journal of Cloud Computing*, vol. 15, article 103, 2026. DOI: 10.1186/s13677-026-00955-w.
9. N. Mathivanan and B. Lanitha, "An Integrated Methodology for Intrusion Detection and Mitigation to Optimize Cloud Security," *Journal of Cloud Computing*, vol. 15, article 18, 2026. DOI: 10.1186/s13677-026-00858-w.
10. "A Two-Stage Generative-AI Fusion Intrusion Detection System for Calibrated and Reliable Cloud Security," *Scientific Reports*, vol. 16, 2026. DOI: 10.1038/s41598-026-51527-6.
11. "SecureRiskNet: An Advanced AI-Driven Framework for Intelligent Security Risk Detection in Heterogeneous Cloud-Fog Computing Networks," *International Journal of Computational Intelligence Systems*, 2026. DOI: 10.1007/s44196-025-01117-7.
12. "An Integrated Methodology for Intrusion Detection and Mitigation to Optimize Cloud Security," *Journal of Cloud Computing*, vol. 15, article 18, 2026. DOI: 10.1186/s13677-026-00858-w