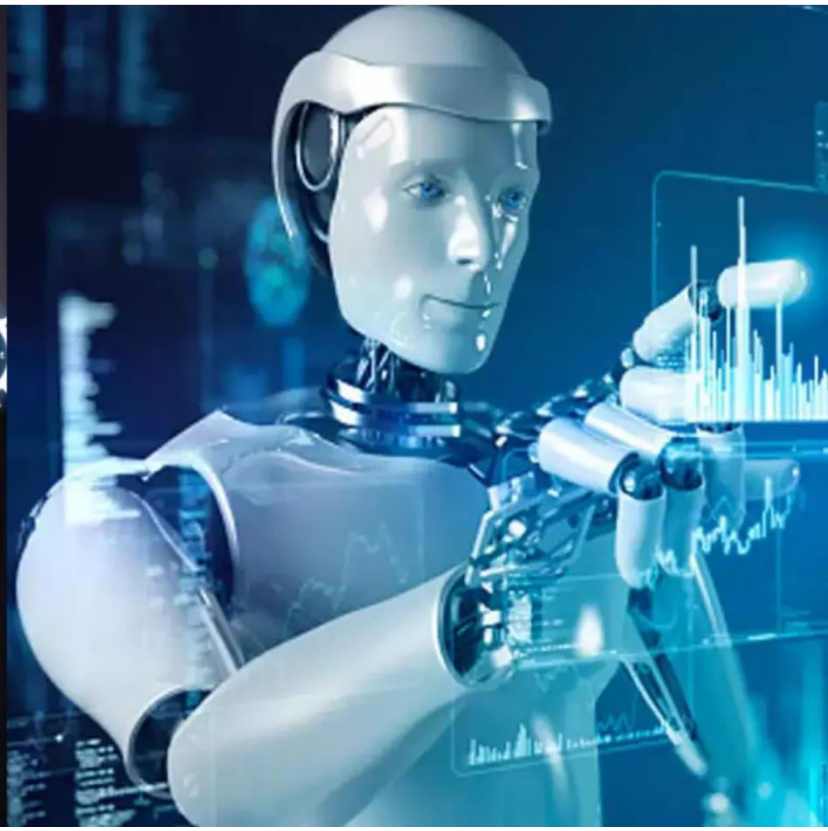
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.935

Volume 15, Issue 6, June 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506127|

Analyzing Security Risks in Cloud File Upload Systems

Ms. V. Varshini, **Mrs. A. Mary Joycy**

PG Scholar, Department of Master of Computer Applications, R.V.S. College of Engineering, Dindigul, Tamil Nadu, India

Assistant Professor, Department of Master of Computer Applications, R.V.S. College of Engineering, Dindigul, Tamil Nadu, India

ABSTRACT: With the increasing adoption of cloud computing, web applications rely heavily on cloud storage services to handle large volumes of user-generated content. A modern file upload approach allows users to upload files directly to cloud storage using pre-signed URLs generated by application servers. While this approach significantly improves performance and scalability, it introduces substantial security risks arising from complex interactions among users, web servers, and cloud storage services. This paper systematically analyzes six key vulnerability classes in direct cloud file upload systems, including insecure direct object access, weak credential exposure, insufficient file validation, improper access control, cloud storage misconfigurations, and lack of continuous monitoring. A secure framework is proposed that integrates robust authentication mechanisms, controlled credential management, comprehensive file validation, and real-time monitoring. Experimental results demonstrate that the proposed system effectively mitigates identified vulnerabilities while maintaining scalability and operational efficiency, making it suitable for modern web applications handling sensitive user data.

I. INTRODUCTION

The rapid growth of cloud computing and web-based applications has significantly increased the demand for scalable and efficient data storage solutions. Cloud storage platforms offered by major providers such as Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure enable applications to store and manage large volumes of user-generated data including images, videos, and documents with high availability and low latency.

This study conducts a systematic analysis of the major security vulnerabilities associated with direct cloud file upload systems. Six distinct vulnerability categories are identified and examined, with a focus on their root causes, exploitation vectors, and real-world impact. The proposed secure framework addresses these vulnerabilities through layered security controls encompassing strong authentication, scope-limited credential generation, deep file validation, storage configuration enforcement, and continuous behavioral monitoring.

II. EXISTING SYSTEM

In the traditional file upload paradigm, web applications manage all aspects of file submission through centralized web servers. When a user initiates a file upload, the request is transmitted to the application server, which performs validation, authentication, and authorization checks before forwarding the file to either local storage or a backend cloud storage service.While this architecture provides strong security guarantees, it introduces significant operational drawbacks. As user traffic scales and file sizes increase, the application server becomes a performance bottleneck, leading to elevated latency, increased infrastructure costs, and reduced system responsiveness. The inability to efficiently distribute upload workloads has prompted widespread adoption of direct cloud upload mechanisms; however, this architectural transition has revealed a range of new security vulnerabilities that were not present in the server-mediated model.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506127|

DISADVANTAGES:

- High server load resulting in processing bottlenecks and degraded application performance.
- Limited scalability due to centralized server dependency for all upload operations.
- Increased latency for end users, particularly for large file uploads over high-traffic periods.
- Absence of real-time monitoring and comprehensive auditing for upload activities.

III. PROPOSED SYSTEM

The proposed system introduces a comprehensive security framework designed to address the vulnerabilities inherent in direct user-to-cloud file upload architectures. Rather than permitting unrestricted access or relying on static, long-lived credentials, the proposed model enforces strict authentication and access control at every stage of the upload pipeline.Users upload files to cloud storage exclusively through securely generated, time-limited pre-signed URLs issued by the application server following successful authentication. The scope and duration of these URLs are strictly constrained to prevent unauthorized reuse or modification. Additionally, the system integrates a multi-layer file validation pipeline that performs file type verification, size constraint enforcement, and deep content inspection including malware scanning, prior to final storage commitment.A continuous monitoring module operates in real time, tracking upload activity patterns, logging security events, and triggering automated alerts upon detection of anomalous behavior. Together, these mechanisms provide a holistic security posture that significantly reduces the attack surface associated with direct cloud file uploads.

ADVANTAGES:

- Enhanced security through multi-factor authentication and role-based access control (RBAC).
- Secure, time-limited pre-signed URL generation that prevents credential reuse and unauthorized uploads.
- Comprehensive file validation and malware scanning to detect and block malicious file uploads.
- Automated cloud storage security configuration enforcement following least-privilege principles.
- Real-time vulnerability detection and continuous monitoring with automated incident response.
- Improved system scalability through cloud-native direct upload mechanisms with reduced server load.

IV. SYSTEM ARCHITECTURE

The system architecture of the proposed Secure Cloud File Upload System is organized around three principal entities: the web user, the application server, and the cloud storage service. The application server functions as the central security orchestrator, responsible for user authentication, pre-signed URL generation, and policy enforcement prior to authorizing any direct cloud interaction.Three parallel security layers operate between the web interface and cloud storage: the Authentication and Validation Layer, which generates secure timelimited credentials; the Security Gateway, which performs file scanning and content validation; and the Policy Enforcement module, which governs access control and monitors system activity. Files are stored in the Secure Cloud Storage backend only after successfully traversing all three layers.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506127|

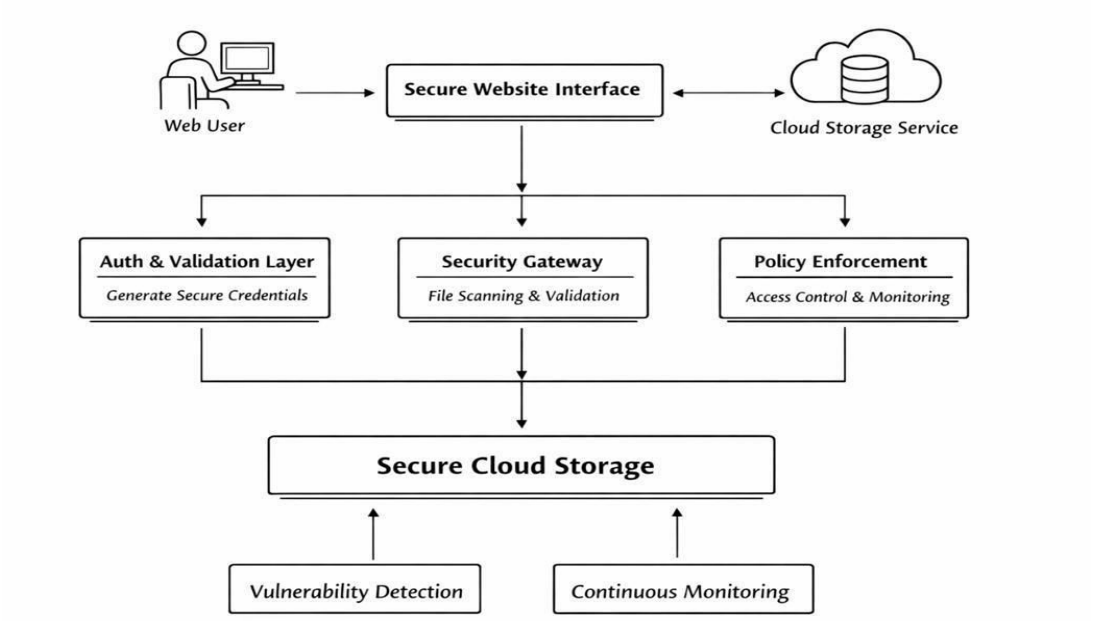


Figure 4.1.1

MODULES:

- User Authentication and Authorization
- Admin dashboard and user management
- Secure Credential & Pre-Signed URL Management
- File Validation and Security Scanning
- Cloud Storage Security Configuration
- Vulnerability Detection and Continuous Monitoring

V. MODULE DESCRIPTIONS

- **USER AUTHENTICATION AND AUTHORIZATION**

This module constitutes the primary security boundary of the proposed system, responsible for verifying user identity and establishing authorized sessions prior to granting access to any upload functionality. The module implements credential-based authentication supported by secure session management using server-side session tokens with configurable expiration policies.Role-Based Access Control (RBAC) is enforced at the module level, assigning each user a designated role that determines their permitted operations within the system. Distinct roles such as standard user and administrator are defined .Ensuring that users can only perform actions commensurate with their authorization level.

- **ADMIN DASHBOARD AND USER MANAGEMENT**

The user management component enables administrators to maintain visibility over registered users, assess usage patterns and identify potential misuse or suspicious account behavior. ensuring accountability and enabling timely administrative intervention when security anomalies are detected. The module enforces a clear privilege boundary that limits the blast radius of potential insider threats.

- **SECURE CREDENTIAL AND PRE-SIGNED URL MANAGEMENT**

This module addresses one of the primary attack vectors in direct cloud upload architectures: the exposure or misuse of cloud storage credentials. Rather than issuing persistent credentials to users, the module generates pre-signed URLs that grant temporary access to a specific cloud storage path for a defined duration. The generated URLs enforce strict temporal constraints, expiring within a configurable short window to limit the window of exploitation in the event of

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506127|

interception. URL integrity is enforced through cryptographic signing, preventing modification of embedded parameters.

- **FILE VALIDATION AND SECURITY SCANNING**

Prior to finalizing any file in cloud storage, this module executes a comprehensive multi-stage validation pipeline to ensure that only safe, well-formed files are accepted by the system. The initial validation stage enforces structural constraints including MIME type verification against an allowlist, file extension validation, and file size enforcement .Beyond structural validation, the module performs deep content inspection to detect embedded malicious payloads, executable scripts.

- **CLOUD STORAGE SECURITY CONFIGURATION**

This module governs the security posture of the cloud storage service by enforcing access control policies, encryption standards, and bucket configuration requirements . It implements least-privilege access policies to ensure that storage buckets are never publicly accessible and that all access is mediated through authenticated, authorized requests. The module incorporates automated misconfiguration detection that periodically audits bucket permissions, access control lists, and policy configurations against defined security baselines.

- **VULNERABILITY DETECTION AND CONTINUOUS MONITORING**

This module provides the system with ongoing visibility into its operational security posture through real-time monitoring of all user activities, upload events, and system interactions. Automated alerting mechanisms notify administrators of high-severity events in real time, enabling rapid incident response. This continuous monitoring capability ensures that the system's defenses remain effective against evolving threats throughout its operational lifecycle.

VI. RESULTS AND DISCUSSION

The proposed Secure Cloud File Upload System was implemented using Java as the primary backend language, MySQL as the relational database management system, and Apache Tomcat as the application server. The frontend was developed using HTML, CSS, and JavaScript to provide a responsive and intuitive user interface.

Figure 6.1.1

Figure 6.1.1 Illustrates the Home interface of the proposed secure cloud file upload system.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506127|

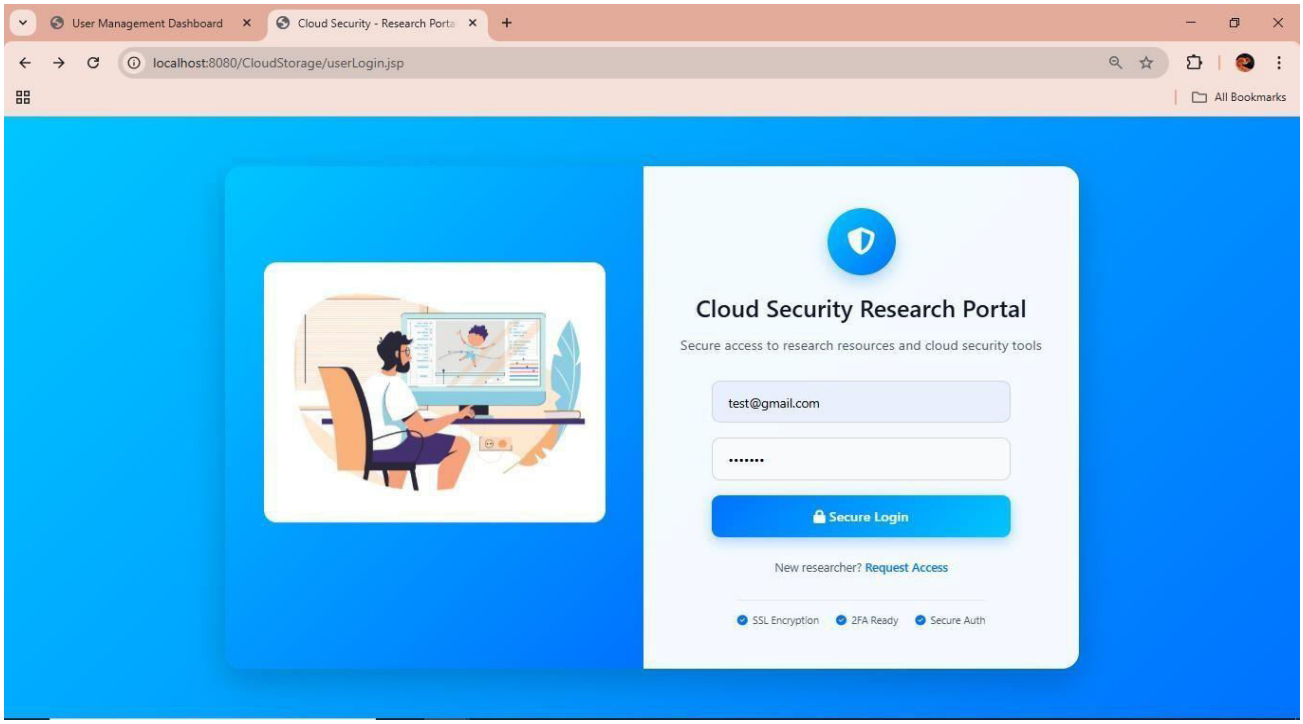


Figure 6.1.2

Figure 6.1.2 Depicts the user login interface of the Cloud Security Research Portal.

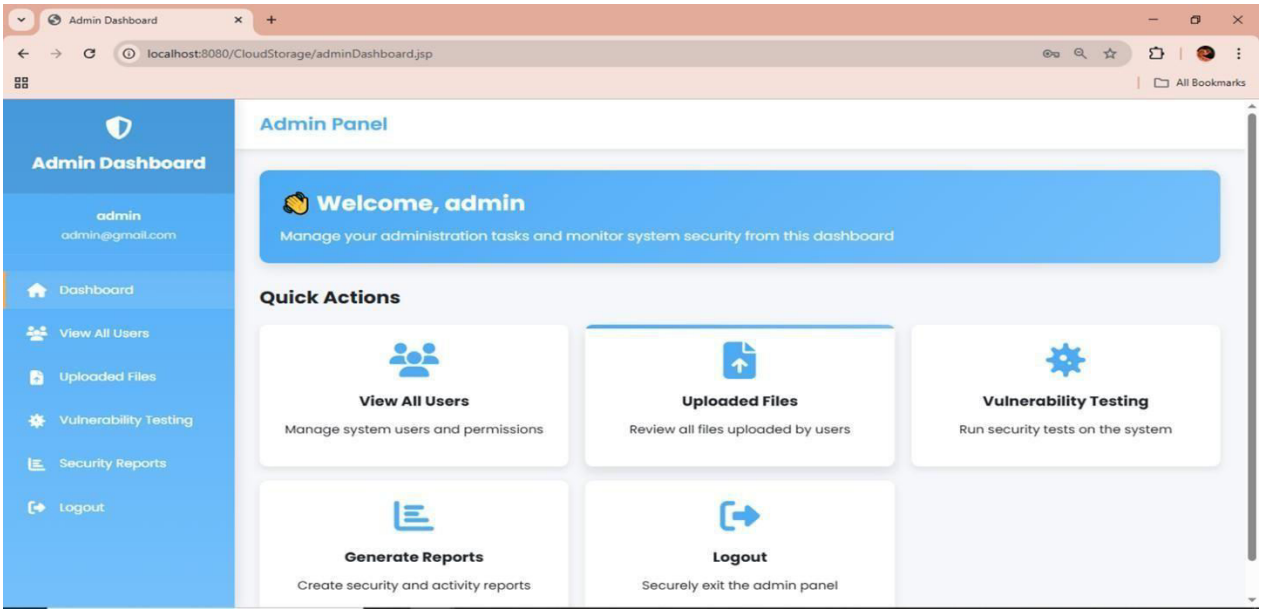


Figure 6.1.3

Figure 6.1.3 Illustrates the Administrative dashboard used for monitoring system activities and user operations.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506127|

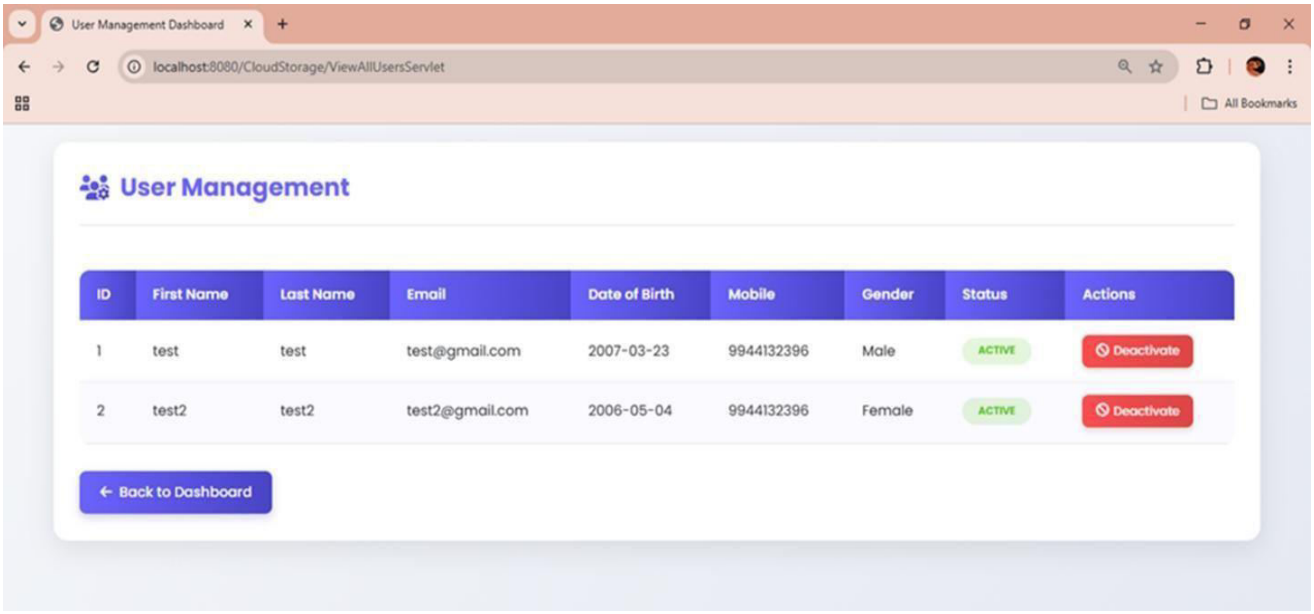


Figure 6.1.4

Figure 6.1.4 Presents the User management panel used for maintaining registered user information.

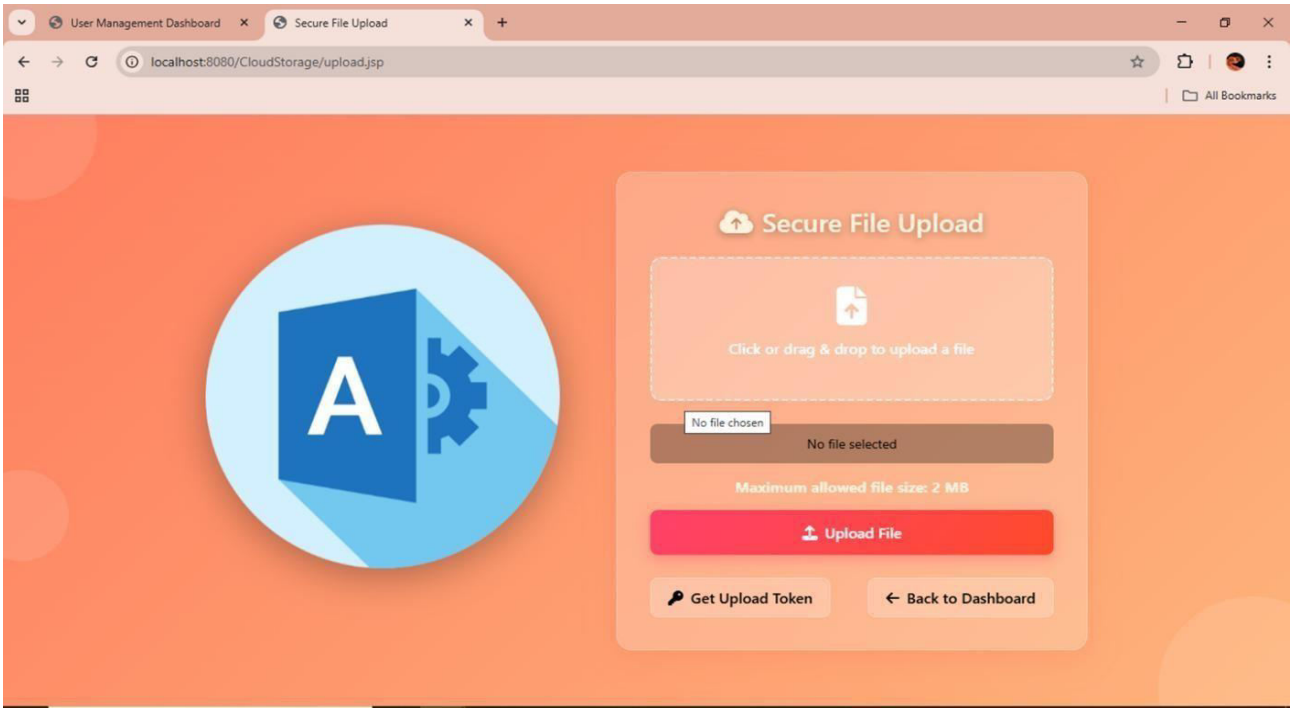


Figure 6.1.5

Figure 6.1.5 Illustrates the Secure cloud file upload interface integrated with validation mechanisms.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506127|

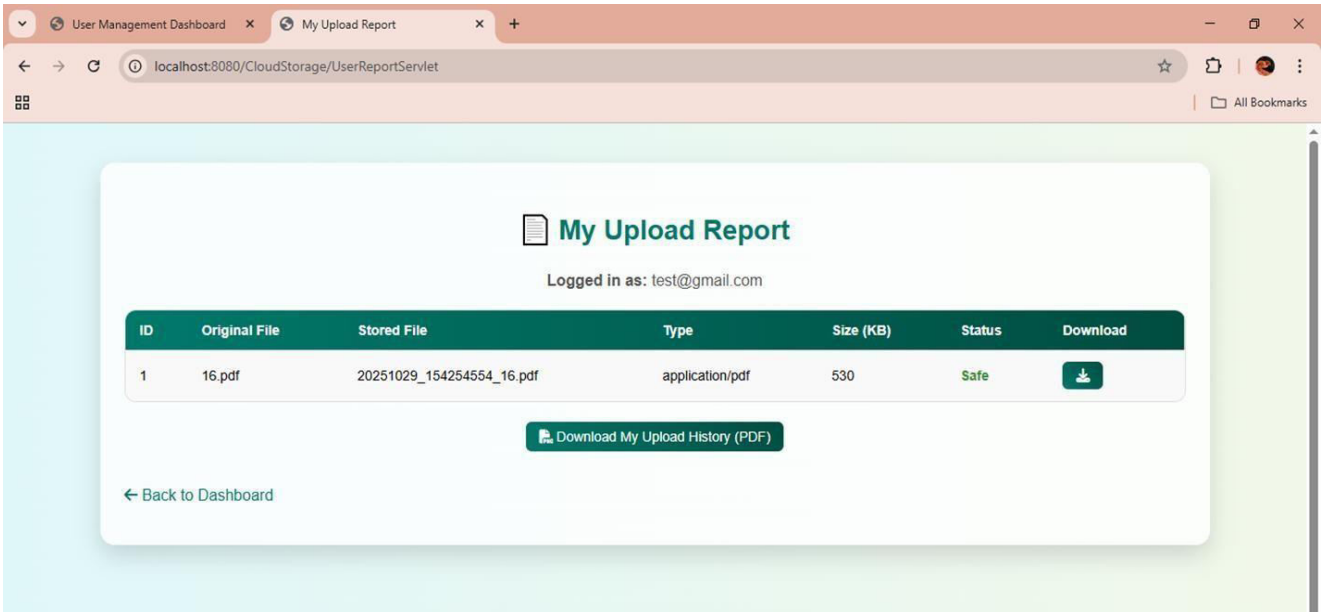


Figure 6.1.6

Figure 6.1.6 Presents the Upload report generated after successful file validation and storage.

VII. CONCLUSION

This paper has presented a systematical analysis of the security vulnerabilities inherent in modern direct cloud file upload architectures and proposed a comprehensive security framework to address the identified threat classes. The proposed system mitigates these risks through a layered security architecture encompassing time-limited pre-signed URL management, comprehensive file validation with malware scanning and continuous real-time monitoring. Experimental implementation and evaluation demonstrate that the proposed framework effectively reduces the attack surface of direct cloud upload systems .

VIII. FUTURE ENHANCEMENTS

Future research directions for this work include the integration of artificial intelligence and machine learning techniques for behavioral anomaly detection, enabling the monitoring module to identify novel attack patterns. Additionally, the system could be extended with automated vulnerability remediation capabilities to apply corrective actions autonomously. Finally, cross-provider portability enhancements would extend the applicability of the framework to multi-cloud deployment scenarios.

REFERENCES

[1] Amazon Web Services, "Amazon S3 Security Best Practices," AWS Documentation, Amazon Web Services, Inc., Seattle, WA, USA. [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/security-best-practices.html

[2] Google Cloud, "Cloud Storage Security Overview," Google Cloud Documentation, Google LLC, Mountain View, CA, USA. [Online]. Available: https://cloud.google.com/storage/docs/security

[3] Microsoft Azure, "Secure File Uploads to Azure Blob Storage," Azure Documentation, Microsoft Corporation, Redmond, WA, USA. [Online]. Available: https://docs.microsoft.com/enus/azure/storage/blobs/security-recommendations

[4] OWASP Foundation, "OWASP Top 10 Web Application Security Risks," OWASP Foundation, Wilmington, DE, USA, 2021. [Online]. Available: https://owasp.org/www-projecttop-ten/

[5] Alexa Internet, "Top Websites Ranking Methodology," Alexa Reports, Amazon Web Services, Inc., 2023.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506127|

[6] P. Yang, N. Xiong, and J. Ren, "Data security and privacy protection for cloud storage: A survey," *IEEE Access*, vol. 8, pp. 131723–131740, 2020.
[7] T. Lee, S. Wi, S. Lee, and S. Son, "FUSE: Finding file upload bugs via penetration testing," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2020, pp. 1–11.
[8] C. Zuo, Z. Lin, and Y. Zhang, "Why does your data leak? Uncovering the data leakage in cloud from mobile apps," in *Proc. IEEE Symp. Secur. Privacy (SP)*, 2019, pp. 1296–1310.
[9] H. Wen, J. Li, Y. Zhang, and D. Gu, "An empirical study of SDK credential misuse in iOS apps," in *Proc. 25th Asia–Pacific Softw. Eng. Conf. (APSEC)*, 2018, pp. 258–267.
[10] X. Wang, Y. Sun, S. Nanda, and X. F. Wang, "Credit karma: Understanding security implications of exposed cloud services through automated capability inference," in *Proc. 32nd USENIX Secur. Symp. (USENIX Secur.)*, 2023, pp. 6007–6024.
[11] J. Cable, D. Gregory, L. Izhikevich, and Z. Durumeric, "Stratosphere: Finding vulnerable cloud storage buckets," in *Proc. 24th Int. Symp. Res. Attacks, Intrusions Defenses*, 2021, pp. 399– 411.
[12] M. Chatterjee, P. Datta, F. Abri, A. Siami Namin, and K. S. Jones, "Abuse of the cloud as an attack platform," in *Proc. IEEE 44th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, 2020, pp. 1091–1092.
[13] M. Wachs, L. Xu, A. Kanevsky, and G. R. Ganger, "Exertion-based billing for cloud storage access," in *Proc. 3rd USENIX Workshop Hot Topics Cloud Comput. (HotCloud)*, 2011, p. 7.
[14] E. Trickel et al., "Toss a fault to your witcher: Applying grey-box coverage-guided mutational fuzzing to detect SQL and command injection vulnerabilities," in *Proc. IEEE Symp. Secur. Privacy (SP)*, 2023, pp. 2658–2675.
[15] Z. Yu Ding, B. Khakshoor, J. Paglierani, and M. Rajpal, "Sniffing for codebase secret leaks with known production secrets in industry," 2020, arXiv:2008.05997.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details