Web Application Security Audit Tool

Mohammed Aadhil Syed A[1], Madhu Sree P S[2], DR. N. Susila[3],
Dr. N. Suresh Kumar[4], Mr. G. Ram Sundar[5]
[1,2,3,4,5]*Department of Information Technology, Sri Ramakrishna Engineering College*

***Abstract*—The rapid growth of cloud computing has increased the need for secure data sharing mechanisms. This paper proposes a secure cloud-based file sharing system that ensures data confidentiality, controlled access, and user accountability. The system uses Advanced Encryption Standard (AES) for securing files before storage. Access to files is controlled through an approval-based mechanism, where file owners grant permission and share secret keys securely via email. Additionally, the system includes intrusion detection by monitoring login attempts and blocking unauthorized users. A machine learning-based malicious URL detection module using Random Forest classifier enhances cybersecurity by identifying phishing and harmful links. The proposed system provides a secure and efficient solution for cloud data sharing.**

I. INTRODUCTION

Cloud computing provides scalable and flexible storage solutions but introduces several security challenges such as unauthorized access, data breaches, and malicious attacks. Ensuring secure data sharing and user authentication is essential. This paper presents a secure cloud data sharing system that uses encryption, controlled access, and threat detection mechanisms to enhance security in cloud environments.

II. LITERATURE REVIEW

Several researchers have contributed to web application security and cloud protection.
Alazmi et al. (2022) analyzed vulnerability scanners and their effectiveness. Mustacoglu et al.
(2020) proposed password-based encryption techniques. Mohana Prabha et al. (2019) introduced multi-factor authentication models. Kumar et al.
(2018) focused on hybrid encryption and authentication. Odeh et al. (2023) discussed detection of web vulnerabilities. These studies highlight the importance of encryption, authentication, and intrusion detection.

III. PROPOSED SYSTEM

The proposed system consists of four modules:

- Authentication Module: Ensures secure login and blocks accounts after failed attempts
- Encryption Module: Uses AES algorithm to encrypt files
- Access Control Module: Owner approves file access requests
- Malicious URL Detection: Uses Random Forest classifier

This architecture ensures data confidentiality and system security.

IV. IMPLEMENTATION & RESULTS

The system is implemented using Python Flask, MySQL, and machine learning techniques. Files are encrypted before storage, ensuring confidentiality. Users can securely share files through an approval-based mechanism. Unauthorized users are blocked after multiple failed login attempts. The malicious URL detection module successfully identifies phishing and harmful links. The system demonstrates improved security and reliability.

V. PLAN OF EXPERIMENTS

Environment Setup and Deployment Validation: This experiment verifies that the application runs reliably in a local environment. The backend is executed using a Python virtual environment (or Docker), and the frontend is served correctly. The objective is to ensure the server starts without errors, the correct port is accessible, and the user interface loads properly in the browser.

User Registration and Authentication Testing: This experiment tests the user registration and login functionalities by creating multiple user accounts with valid and invalid credentials. The system is observed for proper input validation, successful authentication, and correct handling of login failures.

Login Failure and Intrusion Monitoring Testing: This experiment evaluates the intrusion detection mechanism by attempting repeated incorrect logins. After three failures, the system should block the account and add it to the intrusion list for administrator review.

Secure File Upload and Encryption Testing: This experiment verifies that uploaded files are encrypted using the AES algorithm before being stored in the cloud. The system is checked to ensure that original files cannot be accessed without decryption.

Controlled File Access and Key Distribution Testing: This experiment tests file sharing by requesting access to shared files and validating that secret keys are sent only after owner approval.

File Decryption and Download Validation: This experiment ensures that only users with valid secret keys can successfully decrypt and download shared files.

Malicious URL Detection Testing: This experiment evaluates the Random Forest classifier by submitting various URLs and verifying correct classification as phishing, benign, defaced, or malicious.

Server Management and Intrusion List Testing: This experiment checks administrator controls by reviewing blocked users, unblocking accounts, and verifying proper system logging and accountability.

VI. DESIGN / MODELLING AND SIMULATION

The proposed Secure Cloud Data Sharing System is designed as a privacy- focused and security-enhanced platform that ensures protected storage, controlled accessibility, and user accountability in cloud environments. The system integrates secure authentication, cryptographic encryption, controlled data sharing, and malicious URL detection mechanisms to safeguard sensitive information. The architecture is organized into four major components: the Identity and Access Management Module, Secure File Encryption and Storage Module, Controlled Data Sharing Module, and Malicious URL Detection Module. Each component contributes to maintaining confidentiality, integrity, and secure user interaction within the cloud system.

6.1 Identity and Access Management Module:

- Enables secure user registration and authentication using validated credentials.
- Monitors repeated login attempts and blocks accounts after three consecutive failures to prevent unauthorized access.
- Maintains an intrusion list where administrators can view and unblock restricted accounts when required.

6.2 Secure File Encryption and Storage Module:

- Allows authenticated users to upload files securely to the cloud platform.
- Applies Advanced Encryption Standard (AES) encryption with a unique secret key before storing files.
- Ensures confidentiality and data integrity by storing only encrypted versions of user files.

6.3 Controlled Data Sharing Module:

- Provides an approval-based file access mechanism managed by the file owner.
- Allows owners to approve or deny file access requests from other users.
- Sends the secret decryption key securely to the requester's registered email
- upon approval, enabling safe file download and decryption.

6.4 Malicious URL Detection Module:

- Offers a dedicated interface for users to submit URLs for security verification.
- Uses a Random Forest classifier to categorize URLs as phishing, benign, defaced, or malicious.
- Communicates classification results to users, adding an additional cybersecurity protection layer and reducing exposure to harmful links.

VII. PUBLICATIONPRINCIPLES

The journal publishes peer-reviewed research articles that contribute to the advancement of knowledge in the field. Authors are expected to maintain originality, clarity, and technical quality in their work while following proper research and ethical standards.

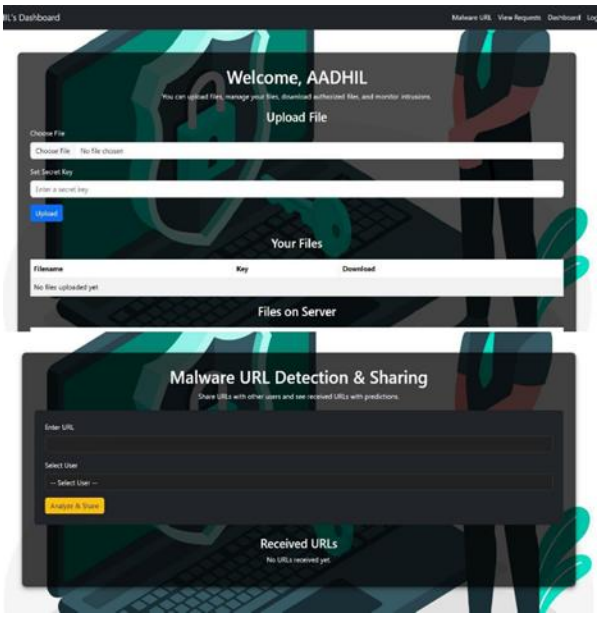
1) Technical papers submitted for publication must advance the state of knowledge and must cite relevant prior work.
2) The length of a submitted paper should be commensurate with the importance or appropriate to the complexity of the work.
3) Authors must convince both peer reviewers and editors of the scientific and technical merit of the paper.
4) Results and conclusions should be supported with sufficient data and proper analysis, especially when reporting extraordinary outcomes.
5) Papers submitted for publication must provide sufficient information to allow readers to perform similar experiments or calculations.
6) The work should contain new, usable, and clearly described information.
7) Authors should follow ethical standards, avoid plagiarism, and ensure proper referencing throughout the paper.

VIII. CONCLUSION

The proposed system provides a secure and efficient approach for cloud-based file sharing by integrating encryption, controlled access, and intrusion detection mechanisms. By using AES encryption, the system ensures data confidentiality, while the approval-based access control prevents unauthorized file usage and enhances user accountability.

The inclusion of malicious URL detection using a machine learning model strengthens the overall cybersecurity of the system. The combination of secure authentication, encryption, and threat detection makes the system reliable, user-friendly, and suitable for protecting sensitive data in cloud environments.

IX. APPENDIX



ACKNOWLEDGMENT

The authors express their sincere gratitude to the management and faculty of Sri Ramakrishna Engineering College, Coimbatore, for providing the necessary facilities and support to carry out this work. We would like to thank our project supervisor, Mr. G. Ram Sundar, for his valuable guidance, encouragement, and continuous support throughout the development of this project. We also extend our thanks to the Department of Information Technology

for their assistance and motivation. Finally, we are grateful to all those who directly or indirectly contributed to the successful completion of this work.

REFERENCES

[1] S. S. Alazmi and D. C. De Leon, "A systematic literature review on the characteristics and effectiveness of web application vulnerability scanners," IEEE Access, vol. 10, pp. 33200–33219, 2022.
[2] M. Mustacoglu, F. O. Catak, and G. Fox, "Password-based encryption approach for securing sensitive data," Security and Privacy, vol. 3, no. 2, 2020.
[3] K. Mohana Prabha et al., "Suppressed K-anonymity multi-factor authentication-based Schmidt-Samoa cryptography for privacy preserved data access in cloud computing," Int. J. Eng. Adv. Technol., vol. 8, no. 6, 2019.
[4] R. S. Kumar et al., "Evaluation of multifactor user security through multi authentication verifiable hybrid revert encryption for cloud computing environment," Int. J. Adv. Res. Comput. Sci., vol. 9, no. 2, 2018.
[5] N. Odeh and S. Hijazi, "Detecting and preventing common web application vulnerabilities: A comprehensive approach," Int. J. Inf. Technol. Comput. Sci., vol. 15, no. 3, pp. 26–41, 2023.
[6] A. Jain, S. Gupta, and V. Kumar, "Phishing detection using Random Forest, SVM and neural networks," in Proc. IEEE Int. Conf. Computing, Communication and Automation, 2020.
[7] R. Verma and A. Das, "Intelligent phishing website detection using Random Forest classifier," in Proc. IEEE Int. Conf. Information Technology, 2018.
[8] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," J. Netw. Comput. Appl., vol. 34, no. 1, pp. 1–11, 2011.
[9] M. Armbrust et al., "A view of cloud computing," Commun. ACM, vol. 53, no. 4, pp. 50–58, 2010.
[10] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," IEEE Internet Comput., vol. 16, no. 1, pp. 69–73, 2012.
[11] C. Tankard, "Advanced persistent threats and how to monitor and deter them," Netw. Security, vol. 2011, no. 8, pp. 16–19, 2011.
[12] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," in Proc. IEEE Int. Workshop Quality of Service, 2009.
[13] D. Boneh and X. Boyen, "Short signatures without random oracles," in Advances in Cryptology, Berlin, Germany: Springer, 2004.
[14] W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed. Upper Saddle River, NJ, USA: Pearson, 2017.
[15] P. Mell and T. Grance, "The NIST definition of cloud computing," NIST Special Publication 800-145, 2011.
[16] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
[17] T. Mitchell, Machine Learning. New York, NY, USA: McGraw-Hill, 1997.
[18] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2011.
[19] M. Bishop, Computer Security: Art and Science. Boston, MA, USA: Addison-Wesley, 2003.
[20] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd ed. New York, NY, USA: Wiley, 1996.
[21] N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography," IEEE Security Privacy, vol. 1, no. 3, pp. 32–44, 2003.
[22] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," IEEE Trans. Circuits Syst. Video Technol., vol. 14, no. 1, pp. 4–20, 2004.
[23] G. Ateniese et al., "Provable data possession at untrusted stores," in Proc. ACM Conf. Comput. Commun. Security, 2007.
[24] H. Takabi, J. B. Joshi, and G. Ahn, "Security and privacy challenges in cloud computing environments," IEEE Security Privacy, vol. 8, no. 6, pp. 24–31, 2010.
[25] Y. Yu et al., "Achieving secure, scalable, and fine-grained data access control in cloud computing," in Proc. IEEE INFOCOM, 2010.