International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March – 2026 SJIF Rating: 8.659 ISSN: 2582-3930

A Security Analysis of Website-Enabled Direct File Uploads to Cloud Storage Services

D. Srihitha Rao
Student, Computer Science and Engineering
Guru Nanak Institutions Technical Campus (Atunomous)
Hyderabad,Telangana,India-501506
srihithadondula@gmail.com

D. Praneeth Reddy
Student, Computer Science and Engineering
Guru Nanak Institutions Technical Campus (Atunomous)
Hyderabad,Telangana,India-501506
praneethreddydonthireddy@gmail.com

D. Arun Kumar
Student, Computer Science and Engineering
Guru Nanak Institutions Technical Campus (Atunomous)
Hyderabad,Telangana,India-501506
arunreddy050204@gmail.com

Dr. Geeta Tripathi
Professor, Computer Science and Engineering
Guru Nanak Institutions Technical Campus (Atunomous)
Hyderabad,Telangana,India-501506
hodcse1.gnitc@gniindia.org

ABSTRACT

With the increasing reliance on cloud storage services for handling large volumes of user data, websites have begun enabling direct file uploads from users to cloud platforms. While this approach offers greater convenience and scalability, it also introduces new security challenges due to the involvement of multiple entities, including web users, web servers, and cloud storage providers. In this study, we present the first comprehensive security evaluation of this direct upload model. Through an in-depth investigation, we identify six distinct categories of vulnerabilities and perform large-scale testing across the top 500 websites ranked by Alexa. Our findings reveal that 182 websites (36.4%) utilize cloud storage services, and a focused analysis of 28 popular websites with upload functionality shows that all exhibit at least one of the identified vulnerabilities. In total, we uncover 79 previously unreported vulnerabilities, which we responsibly disclosed to the respective platforms, including major services like Google, Reddit, and CSDN. The positive responses highlight the practical impact of our findings. We further examine the root causes of these issues and suggest effective mitigation strategies. This work contributes valuable insights into the security implications of cloud-based file uploads and aims to guide both developers and researchers in building more secure web applications.

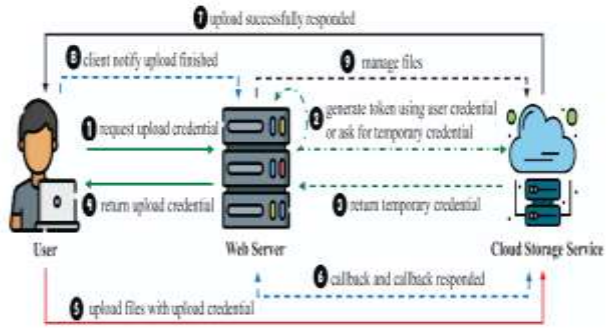
1.INTRODUCTION

The rapid growth of web applications and user-generated content has increased the demand for scalable data storage solutions. Cloud storage services such as Amazon S3, Google Cloud Storage, and Microsoft Azure provide efficient methods for managing large volumes of data. In modern architectures, users can upload files directly from their browsers to cloud storage without passing through the web server. Although this method improves performance and reduces server workload, it introduces several security risks. Attackers may exploit weaknesses in credential management or access control mechanisms to gain unauthorized access to stored files. Direct file uploads involve three key entities: the user, the web server, and the cloud storage service. The interaction between these entities requires proper authentication, authorization, and data validation. If any stage is improperly configured, attackers may manipulate upload credentials, modify stored files, or intercept data transmissions. Therefore, a comprehensive security analysis is necessary to ensure that direct-to-cloud upload mechanisms remain secure and reliable.

2. LITERATURE REVIEW

Several researchers have explored vulnerabilities associated with cloud storage systems and web applications. Chen et al. proposed URadar, a dynamic testing framework that detects unrestricted file upload vulnerabilities in cloud-integrated applications. Wang et al. introduced the Credit Karma framework for identifying exposed cloud services through automated capability inference. Hong et al. analyzed malicious abuse of image hosting services and developed detection models to identify suspicious uploads. Another study by Lv et al. examined the security risks associated with misconfigured Content Security Policies in cloud storage environments. These studies highlight the importance of secure configuration, policy enforcement, and vulnerability detection in cloud-based systems. However, few studies specifically focus on the direct-to-cloud upload model used in modern websites. This research addresses this gap by analyzing vulnerabilities across all stages of the upload process and proposing a secure architecture.

3. SYSTEM ARCHITECTURE



The system architecture of this project is designed to provide a secure and efficient framework for direct file uploads from web users to cloud storage while preventing vulnerabilities in the upload process. The architecture mainly consists of two core modules — the User Module and the Server Module — interconnected with the Cloud Storage Service. The User Module acts as the client interface, allowing users to register, log in, request upload credentials, encrypt files using the RSA algorithm, and upload them securely to the cloud. It also enables users to request decryption keys and download decrypted files after server approval. The Server Module is the central control unit responsible for verifying user identities, generating upload credentials, managing decryption keys, and validating authorization for all requests. It communicates with the cloud storage to confirm file uploads and ensures all operations meet security standards. The Cloud Storage component stores encrypted files, validates upload credentials, and sends callback notifications about upload status to the server. All communication between the modules is performed over secure HTTPS protocols to prevent interception or tampering. The system also addresses six major vulnerabilities—credential misuse, file overwriting, file stealing, unrestricted upload access, credential validity flaws, and callback spoofing—through strict authentication, encryption, and access control mechanisms. By integrating cryptographic security with layered authorization, the architecture ensures data confidentiality, integrity, and trustworthiness throughout the entire file lifecycle, providing a strong foundation for secure cloud-based storage operations.

4. PROPOSED METHODOLOGY

A systematic study of security risks was conducted to examine a modern cloud-based file upload scenario, where users upload files directly from websites to cloud storage services. Unlike traditional methods that route files through web servers, this direct interaction introduces multiple roles—users, web servers, and cloud services—each requiring secure coordination. The study identifies six major categories of vulnerabilities, including improper handling of upload credentials, unrestricted file types and sizes, file overwriting risks, unauthorized file access, and spoofed callback notifications. Through an extensive evaluation of the top 500 Alexa-ranked websites, it was found that 36.4% utilize cloud storage, and among 28 popular sites that allow uploads, all contained at least one vulnerability. In total, 79 new vulnerabilities were uncovered. This research is significant in highlighting the emerging threats in direct-to-cloud uploads and offers guidance on preventing these issues through improved authentication, access control, and secure communication mechanisms.

5. RESULTS AND DISCUSSION

Experiments were conducted using a simulated cloud upload environment. The system was tested against various attack scenarios, including credential misuse, unauthorized file uploads, and callback spoofing. Results indicate that implementing strict credential validation significantly reduces unauthorized upload attempts. Furthermore, RSA-based encryption protects sensitive data from interception or unauthorized access. The performance evaluation also shows that direct-to-cloud

uploads reduce server processing time compared to traditional upload architectures. This improvement demonstrates that strong security measures can be implemented without compromising system efficiency. Secure direct-to-cloud upload systems have applications in multiple domains, including social media platforms, cloud-based file sharing services, enterprise document management systems, and healthcare data storage. These systems enable efficient data transfer while maintaining strong privacy protection for sensitive information. Organizations handling confidential data can benefit from the proposed architecture because it ensures data confidentiality, integrity, and secure access control.

6.PERFORMANCE ANALYSIS

Direct-to-cloud upload systems are vulnerable to several security threats. The first vulnerability involves unrestricted credential acquisition, where attackers obtain upload credentials without proper authentication. The second vulnerability is credential validity flaws, where expired or manipulated credentials remain usable. The third vulnerability relates to unrestricted file types and sizes, which may allow malicious files to be uploaded. Additional risks include file overwriting, where attackers replace existing files, and file stealing, where unauthorized users download sensitive data. Another critical vulnerability is callback spoofing, in which attackers send fake notifications to the server to manipulate system responses. Addressing these vulnerabilities requires strong authentication, credential expiration policies, and encrypted data transfer mechanisms.

7.CONCLUSION

In conclusion, this project provides a secure and efficient framework for directly uploading files from web users to cloud storage while addressing critical security vulnerabilities. By implementing RSA-based encryption and a robust credential management system, it ensures data confidentiality, integrity, and authorized access throughout the file lifecycle. The User and Server modules work in coordination to manage authentication, upload credential issuance, decryption key distribution, and callback verification, effectively mitigating vulnerabilities such as credential misuse, file stealing, file overwriting, and callback spoofing. The systematic study of the three-stage upload process allows the identification and prevention of security risks that traditional methods may overlook. The architecture is designed to be scalable, reliable, and user-friendly, supporting secure file uploads, downloads, and key management. Overall, this project demonstrates a comprehensive approach to securing cloud storage interactions, providing a strong foundation for future enhancements and safe cloud-based data management for modern web applications.

8. FUTURE SCOPE

In the future, the project can be enhanced to provide more advanced security, usability, and scalability features. One enhancement could be the integration of multi-factor authentication for users, adding an extra layer of security before issuing upload credentials or decryption keys. Another improvement could be implementing role-based access control, where different types of users have fine-grained permissions to upload, download, or manage files. The system can also support real-time monitoring and anomaly detection to identify suspicious activities, such as unauthorized file access or credential misuse. Integrating advanced encryption algorithms alongside RSA, like ECC (Elliptic Curve Cryptography), can improve encryption efficiency for large-scale file uploads. The cloud storage module can be extended to include redundancy and fault-tolerance mechanisms to prevent data loss in case of server failures. Future versions may allow cross-platform file access, enabling mobile and desktop users to securely upload and download files. End-to-end encryption with user-managed keys could give users full control over their file security. Enhancing the user interface with intuitive dashboards can improve user experience and simplify file management. The system can support batch uploads and downloads for better efficiency. Finally, performance optimization and load balancing can ensure the system remains fast and reliable even with millions of users. These enhancements will make the system more secure, scalable, and user-friendly, aligning it with modern cloud storage requirements.

REFERENCES

[1] 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.

[2] (2021). Gmail Statistics, Users, Growth and Facts for 2021. [Online].Available: https://saasscout.com/statistics/gmailstatistics/

[3] (2019). Reddit’s 2019 Year in Review. [Online]. Available: https://www.redditinc.com/blog/reddits-2019-year-in-review/

[4] R. Nachiappan, B. Javadi, R. N. Calheiros, and K. M. Matawie,"Cloud storage reliability for big data applications: A state of the art survey," J. Netw. Comput. Appl., vol. 97, pp. 35–47,Nov.2017.
[5] 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.
[6] Y. Chen, Y. Li, Z. Pan, Y. Lu, J. Chen, and S. Ji, "URadar: Discovering unrestricted file upload vulnerabilities via adaptive dynamic testing," IEEE Trans. Inf. Forensics Security, vol. 19, pp. 1251–1266,2024.
[7] 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), May 2019, pp. 1296–1310.
[8] Y. Zhou, L. Wu, Z. Wang, and X. Jiang, "Harvesting developer credentials in Android apps," in Proc. 8th ACM Conf. Secur. Privacy Wireless Mobile Netw., Jun. 2015, pp. 1–12.
[9] M. Meli, M. R. McNiece, and B. Reaves, "How bad can it git? Characterizing secret leakage in public GitHub repositories," in Proc.Netw. Distrib. Syst. Secur. Symp., 2019, pp. 1–18.
[10] 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), Dec. 2018, pp. 258–267.
[11] 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.
[12] T. Bhatia and A. K. Verma, "Data security in mobile cloud computing paradigm: A survey, taxonomy and open research issues," J. Supercomput., vol. 73, no. 6, pp. 2558–2631, Jun. 2017.
[13] Z. Xia, N. N. Xiong, A. V. Vasilakos, and X. Sun, "EPCBIR: An effcient and privacy-preserving content-based image retrieval scheme in cloud computing," Inf. Sci., vol. 387, pp. 195–204, May 2017.
[14] D. He, N. Kumar, S. Zeadally, and H. Wang, "Certificateless provable data possession scheme for cloud-based smart grid data management systems," IEEE Trans. Ind. Informat., vol. 14, no. 3, pp. 1232–1241, Mar. 2018.
[15] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An effcient public auditing protocol with novel dynamic structure for cloud data,"IEEE Trans. Inf. Forensics Security, vol. 12, no. 10, pp. 2402–2415, Oct.2017.
[16] J. Cable, D. Gregory, L. Izhikevich, and Z. Durumeric, "Stratosphere: Finding vulnerable cloud storage buckets," in Proc. 24th Int. Symp. Res. Attacks, Intrusions Defenses, Oct. 2021, pp. 399–411.
[17] Academia.edu. Accessed: 2022. [Online]. Available: https://www.academia.edu/
[18] Uploading to Amazon S3 Directly From a Web or Mobile Application. Accessed: 2022. [Online]. Available: https://aws.amazon.com/blogs/compute/uploading-to-amazon-s3-directly-from-a-web-or-mobile-application/
[19] Uploading Objects to OSS Directly From Clients. Accessed: 2022.[Online]. Available: https://help.aliyun.com/zh/oss/use-cases/uploadingobjects- to-oss-directly-from-clients/
[20] Comparing Aws Account Root User Credentials and IAM User Credentials.Accessed: 2022. [Online]. Available: https://docs.aws.amazon.com/accounts/latest/reference/root-user-vs-iam.html
[21] Introduction to Resource Access Management—Alibaba Cloud Document Center. Accessed: 2022. [Online]. Available: https://www.alibabacloud.com/help/en/resource-access-management
[22] Use STS Temporary Access Credentials to Access OSS. Accessed:2022. [Online]. Available:https://help.aliyun.com/document detail/100624.html
[23] Y. Tang, P. P. C. Lee, J. C. S. Lui, and R. Perlman, "Secure overlay cloud storage with access control and assured deletion,"IEEE Trans. Dependable Secure Comput., vol. 9, no. 6, pp. 903–916,Nov. 2012.
[24] Y. Tang, P. P. C. Lee, J. C. S. Lui, and R. Perlman, "FADE: Secure overlay cloud storage with file assured deletion," in Proc. Int. Conf.Secur. Privacy Commun. Syst. Cham, Switzerland: Springer, Jan. 2010,pp. 380–397.
[25] 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), Jul. 2020,pp. 1091–1092.
[26] (2023). Amazon S3 Simple Storage Service Pricing—Amazon Web Services. [Online]. Available:https://aws.amazon.com/s3/pricing/?nc1=h ls
[27] (2023). Azure Blob Storage Pricing — Microsoft Azure. [Online]. Available:https://azure.microsoft.com/enus/pricing/details/storage/blobs/
[28] (2023). OSS (Object Storage Service) Pricing & Purchasing Methods—Alibaba Cloud. [Online].

Available:
https://www.alibabacloud.com/product/oss/pricing
[29] 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 11), Jun. 2011, p. 7.
[30] G. Hong, M. Wu, P. Chen, X. Liao, G. Ye, and M. Yang, "Understanding and detecting abused image hosting modules as malicious services,"in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., Nov. 2023,pp. 3213–3227.
[31] Y. Lv, W. Shi, W. Zhang, H. Lu, and Z. Tian, "Don't trust the clouds easily: The insecurity of content security policy based on object storage," IEEE Internet Things J., vol. 10, no. 12, pp. 10462–10470,Jun. 2023.
[32] L. Weichselbaum, M. Spagnuolo, S. Lekies, and A. Janc, "CSP is dead,long live CSP! On the insecurity of whitelists and the future of content security policy," in Proc. ACM SIGSAC Conf. Comput. Commun. Secur.,Oct. 2016, pp. 1376–1387.
[33] S. Calzavara, A. Rabitti, and M. Bugliesi, "Content security problems: Evaluating the e ectiveness of content security policy in the wild,"in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., Oct. 2016,pp. 1365–1375.
[34] G. E. Rodr´ıguez, J. G. Torres, P. Flores, and D. E. Benavides, "Crosssite scripting (XSS) attacks and mitigation: A survey," Comput. Netw.,vol. 166, Jan. 2020, Art. no. 106960.
[35] G. Pellegrino, M. Johns, S. Koch, M. Backes, and C. Rossow,"Deemon: Detecting CSRF with dynamic analysis and property graphs,"in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., Oct. 2017,pp. 1757–1771.
[36] D. Kim and H. Kim, "Performing clickjacking attacks in the wild: 99% are still vulnerable!," in Proc. 1st Int. Conf. Softw. Secur. Assurance (ICSSA), Jul. 2015, pp. 25–29.
[37] D. Bates, A. Barth, and C. Jackson, "Regular expressions considered harmful in client-side XSS filters," in Proc. 19th Int. Conf. World Wide Web, Apr. 2010, pp. 91–100.
[38] M. Mohammadi, B. Chu, H. R. Lipford, and E. Murphy-Hill, "Automatic Web security unit testing: XSS vulnerability detection," in Proc.IEEE/ACM 11th Int. Workshop Autom. Softw. Test (AST), May 2016,pp. 78–84.
[39] Amazon.(2022). Cloud Object Storage—amazon S3. [Online]. Available:https://aws.amazon.com/s3/
[40] Cloud Computing Services — Microsoft Azure. Accessed: 2022.[Online]. Available: https://azure.microsoft.com/
[41] Object Storage Data Processing-Cos Data Processing-Data Processing Solution-Tencent Cloud. Accessed: 2022. [Online]. Available: https://cloud.tencent.com/product/cos
[42] Object Storage BOS. Accessed: 2022. [Online]. Available: https://cloud.baidu.com/product/bos.html
[43] About Object Operations. Accessed: 2022. [Online]. Available: https://help.aliyun.com/document detail/31977.html
[44] Amazon S3 Rest API Introduction—Amazon Simple Storage Service.Accessed: 2022. [Online]. Available:https://docs.aws.amazon.com/zh cn/AmazonS3/latest/API/Welcome.html
[45] J. Qian, S. Hinrichs, and K. Nahrstedt, "ACLA: A framework for access control list (ACL) analysis and optimization," in Communications and Multimedia Security Issues of the New Century: IFIP TC6 / TC11 Fifth Joint Working Conference on Communications and Multimedia Security (CMS'01) May 21–22, 2001, Darmstadt, Germany. Cham, Switzerland:Springer, 2001, pp. 197–211.
[46]GitHub.Accessed: 2022. [Online]. Available:https://github.com/qingzhenyun/qingzhenyuna pi gateway/blob/5cd0eb45e82d6276b400aee6dc871e3a07b b4e03/src/app/web/controllers/v1/store.jsn#L64
[47] 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), May2023, pp. 2658–2675.
[48] Post Policy—Amazon Simple Storage Service. Accessed: 2022.[Online]. Available:https://docs.aws.amazon.com/AmazonS3/lates t/API/sigv4HTTPPOSTConstructPolicy.html
[49] Response Header. Accessed: 2022. [Online]. Available: https://developer.mozilla.org/zh-CN/docs/Glossary/Response header
[50] Authenticating Requests: Browser-Based Uploads Using Post (AWS Signature Version 4). Accessed: 2022. [Online].Available: https://docs.aws.amazon.com/AmazonS3/latest/API/sigv 4-authentication-HTTPPOST.html
[51] Authenticating Requests: Using Query Parameters (AWS Signature Version 4). Accessed: 2022. [Online]. Available:https://docs.aws.amazon.com/AmazonS3/lates t/API/sigv4-query-stringauth.html

[52] CVE-2017-11882. Accessed: 2022. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-11882
[53] CVE-2021-40444. Accessed: 2022. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-40444
[54] Object Storage Service (OBS)—Huawei Cloud. Accessed: 2022.[Online]. Available: https://www.huaweicloud.com/intl/en-us/product/obs.html
[55] V. S. Sinha, D. Saha, P. Dhoolia, R. Padhye, and S. Mani, "Detecting and mitigating secret-key leaks in source code repositories," in Proc.IEEE/ACM 12th Work. Conf. Mining Softw. Repositories, May 2015, pp. 396–400.
[56] Z. Yu Ding, B. Khakshoor, J. Paglierani, and M. Rajpal, "Sning for codebase secret leaks with known production secrets in industry," 2020,arXiv:2008.05997.
[57] A. D. Diego, "Automatic extraction of API keys from Android applications," Ph.D. dissertation, Dept. Eng., Universit`a degli Studi di Roma Tor Vergata, Rome, Italy, 2017.
[58] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get o of my cloud: Exploring information leakage in third-party compute clouds," in Proc. 16th ACM Conf. Comput. Commun. Secur., Nov. 2009,pp. 199–212.
[59] K. Borgolte, T. Fiebig, S. Hao, C. Kruegel, and G. Vigna, "Cloud strife: Mitigating the security risks of domain-validated certificates," in Proc. Appl. Netw. Res. Workshop (ANRW). New York, NY, USA: ACM, 2018, p. 4, doi: 10.1145/3232755.3232859. [Online]. Available: https://dl.acm.org/doi/10.1145/3232755.3232859.
[60] E. Pauley, R. Sheatsley, B. Hoak, Q. Burke, Y. Beugin, and P. McDaniel, "Measuring and mitigating the risk of IP reuse on public clouds," in Proc. IEEE Symp. Secur. Privacy (SP), May 2022,pp. 558–575.
[61] X. Zhang, H.-T. Du, J.-Q. Chen, Y. Lin, and L.-J. Zeng, "Ensure data security in cloud storage," in Proc. Int. Conf. Netw. Comput. Inf. Secur.,vol. 1, May 2011, pp. 284–287.
[62] A. Markandey, P. Dhamdhere, and Y. Gajmal, "Data access security in cloud computing: A review," in Proc. Int. Conf. Comput., Power Commun. Technol. (GUCON), Sep. 2018, pp. 633–636.
[63] D. Zhe, W. Qinghong, S. Naizheng, and Z. Yuhan, "Study on data security policy based on cloud storage," in Proc. IEEE 3rd Int. Conf.Big Data Secur. Cloud (Bigdatasecurity) Int. Conf. High Perform. Smart Comput. (HPSC), IEEE Int. Conf. Intell. Data Secur. (IDS), May 2017,pp. 145–149.
[64] Y. Zhang, C. Xu, and X. S. Shen, Data Security in Cloud Storage. Cham, Switzerland: Springer, 2020.
[65] Z. Xia, T. Shi, N. N. Xiong, X. Sun, and B. Jeon, "A privacy-preserving handwritten signature verification method using combinational features and secure KNN," IEEE Access, vol. 6, pp. 46695–46705,2018.