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

G Srujana Bharathi[1], Vemula Nikhitha[2], Yara Bharath Kumar[3], Tungala Rohith Varma[4]

[1]Assistant Professor, Department of CSE, Guru Nanak Institutions Technical Campus, Hyderabad.
[2,3,4]Department of CSE, Guru Nanak Institutions Technical Campus, Hyderabad.

Abstract: Websites have started allowing users to upload files directly to cloud platforms due to the growing reliance on cloud storage services for managing massive amounts of user data. Due to the participation of several organizations, such as online users, web servers, and cloud storage providers, this strategy presents additional security issues even if it offers increased convenience and scalability. We provide the first thorough security analysis of this direct upload approach in this work. After a thorough study, we find six different types of vulnerabilities and conduct extensive testing on the top 500 Alexa-ranked websites. 182 websites (36.4%) employ cloud storage services, according to our research, and a targeted examination of 28 well-known websites with upload capabilities reveals that all of them have at least one of the vulnerabilities found. We found 79 previously undiscovered vulnerabilities in total, which we appropriately notified to the relevant platforms, including well-known sites like Google, Reddit, and CSDN. The favorable reactions demonstrate the usefulness of our conclusions. We further investigate the core causes of these difficulties and recommend appropriate mitigation techniques. The goal of this effort is to help academics and developers create better secure online applications by offering insightful information about the security implications of cloud-based file uploads.

Keywords: Cloud Storage Security, Direct File Upload, Web Security, Credential Management, Cloud Computing, Data Integrity.

I.INTRODUCTION

The user base and data quantities of websites grow significantly as they get more and more popular. As a result, handling and storing large amounts of data created by users becomes extremely difficult. For instance, hundreds of millions of people visit well-known websites like Google and Reddit, according to statistical data. Among Google services, Gmail alone has more than 1.8 billion users. There are over 430 million active users on Reddit each month. As a result, it is critical to solve the data storage and management concerns of the aforementioned large users. Because of its flexibility, scalability, and pay-as-you-go storage options, cloud storage services have become the go-to option for managing and storing website data. These features meet the requirements of websites in a range of sizes and sectors.A new file upload scenario has been added to the cloud storage service, in which online users submit files directly to the cloud. This new file upload scenario typically involves three crucial steps: (1) obtaining and dispatching upload credentials; (2) uploading and verifying files; and (3) callback notice and response. In stage (1), the client1 sends an HTTP request to the web server to request an upload credential when a user wishes to upload a file to a website. After verifying the user's identity, the web server should provide the upload credentials if it is successful. In step (2), the client uploads the file straight to the cloud storage provider using the credentials. The cloud storage provider should confirm the signature details of the credential and determine if the uploaded files meet the policy criteria of the credential. To let the web server know the outcome of the file upload, either the client or the cloud storage service should send a callback message in stage (3). This novel scenario removes the requirement for the server to handle data storage or the transit of submitted files, in contrast to conventional techniques where the web server saves user files locally or acts as a middleman to send files to cloud services. For web servers, this lowers transmission overhead and provides ease. But it also creates the following new attack surfaces. First, there are three roles involved in the process, and each job should be in charge of its own security authentication. For example, if a user's identity is not properly authenticated by the web server or cloud storage service, attackers may be able to fake the user's identity. Second, a new authorization mechanism—upload credentials—is introduced by the new scenario. Privilege escalation vulnerabilities may result from improperly configured upload credentials or policies. Adversaries could leverage these flaws, for example, to steal or alter other users' files in cloud storage systems. Lastly, information synchronization across the three roles is a feature of the new scenario. The

legitimacy of the callback notification information cannot be ensured if the web server or cloud storage provider does not include a secure synchronization information verification method. Attackers can then deceive the web server by forging or manipulating callback notifications. Prior research on cloud storage security has concentrated on mobile applications that have problems with data management and user credentials abuse.

To the best of our knowledge, however, no research has been done on the security concerns of this novel situation. Analyzing this scenario's security threats presents two major issues.1) It is difficult to comprehend the intricate mechanisms of interaction between various jobs and identify any weaknesses. A typical online user cannot see many of these exchanges. In reality, for example, a user cannot access the communication process between a website and cloud storage. 2) Determining whether an interaction mechanism's actual deployments or implementations are insecure is a difficult task. The diversity of cloud service providers and websites, each of which often has unique implementations or deployments, contributes to the complexity. We provide the first comprehensive analysis of the security vulnerabilities associated with online users directly uploading data to cloud storage services in this research. Based on the public development guides offered by cloud service providers, we do a thorough study of the three crucial steps in this situation in order to handle the aforementioned issues. We classify the newly discovered vulnerabilities into six categories based on access control, credential validity, file restrictions, data integrity, data confidentiality, and consistency: unrestricted upload credential acquisition (V1), upload credentials validity flaw (V2), unrestricted file types and file size (V3), file overwriting (V4), file stealing (V5), and callback notification spoofing (V6). These flaws, which include resource-consumption assaults, privacy leakage, sensitive data overwriting, etc., have the potential to seriously impact online users, web servers, and cloud storage services.

II.RELATED WORK

The analysis and security of the new cloud-based file upload scenario—in which online users upload files directly to cloud storage services—are the main objectives of this research. In the three crucial phases—upload credential management, file uploading and verification, and callback notification handling—it seeks to detect and reduce any security threats. The project tackles vulnerabilities including data manipulation, file overwriting, illegal access, and credential abuse. It guarantees safe user, web server, and cloud service synchronization, authorization, and authentication. In the end, the project improves large-scale cloud-based web applications' data integrity, confidentiality, and dependability.

Ensuring safe, dependable, and effective file uploads from online users straight to cloud storage providers is the primary goal of this project. In order to find any security flaws, it focuses on examining the three crucial phases of the upload process: credential generation, file verification, and callback response. Through the implementation of robust authentication and policy enforcement methods, the project seeks to avoid risks including unauthorized access, credential abuse, file overwriting, and data theft. Through safe communication and synchronization, it also aims to increase confidence between users, web servers, and cloud platforms. Building a strong foundation that enhances data security, integrity, and confidentiality in contemporary cloud-based online environments is the ultimate objective.

In order to meet the increasing need for extensive user data management on websites, the current system makes use of cloud storage services. Instead of using conventional server-side storage, users upload data straight to the cloud from the client side in this configuration. Due to the inclusion of several roles—online users, web servers, and cloud service providers—this architecture presents new security issues despite being effective and scalable. Credential issuance, file upload and verification, and callback notification are the three primary phases of the upload procedure. Cloud storage services offer a scalable, adaptable, and affordable substitute for conventional local storage systems as websites struggle to manage enormous volumes of user data. A new scenario in the present architecture eliminates the necessity for the website server to function as a middleman by enabling users to upload files straight from their browsers to cloud storage. Due to the participation of several parties—online users, web servers, and cloud storage providers—this approach improves efficiency and lowers server strain, but it also creates new security threats. Nevertheless, the current system's synchronization, authorization, and authentication processes are insufficiently secure.

III.LITERATURE SURVEY

Y. Chen, Y. Li, Z. Pan, Y. Lu, J. Chen, and S. Ji, URadar is an automated security testing framework designed to discover unrestricted file upload vulnerabilities in cloud-backed web applications. It models the three-stage direct-to-cloud upload workflow—credential issuance, client-to-cloud upload, and callback synchronization—to expose gaps attackers can exploit. Using adaptive dynamic testing, URadar generates and refines targeted upload credentials and file payloads to probe real-world policy and validation weaknesses. The system simulates malicious actors who attempt privilege escalation, file overwrites, unauthorized downloads, and callback spoofing to reveal practical attack paths. URadar correlates observable behaviors across the web server, client, and cloud storage service to infer invisible interactions and detect inconsistencies. It includes modules for credential abuse detection, file-type/size bypass testing, integrity checks, and callback authenticity validation. When a potential vulnerability is found, URadar performs confirmation steps (safe, non-destructive where possible) to reduce false positives. The framework automatically produces human-readable reports that describe the vulnerability, the exploited vector, PoC steps, and recommended fixes.

X. Wang, Y. Sun, S. Nanda, and X. F. Wang, 2023, credit Karma is a security analysis framework designed to understand and assess the risks associated with exposed cloud services by performing automated capability inference. The project focuses on identifying misconfigured or publicly accessible cloud components that may unintentionally expose sensitive data or functionalities. By analyzing access policies, API endpoints, and resource permissions, Credit Karma automatically infers what an attacker could potentially do if the exposed service is exploited. It simulates real-world attack scenarios to evaluate privilege escalation, unauthorized data access, and service manipulation risks. The system employs automated scanning and dynamic testing to map out exposure surfaces across multiple cloud platforms. It categorizes detected misconfigurations based on severity and potential business impact, helping organizations prioritize remediation. Credit Karma also correlates findings across different services to detect chained vulnerabilities that may arise due to weak cross-service authentication. The framework generates comprehensive reports detailing the identified exposures, inferred attacker capabilities, and recommended mitigation strategies. Ultimately, the project aims to improve cloud security visibility, prevent data breaches, and help developers deploy secure, well-configured cloud environments.

G. Hong, M. Wu, P. Chen, X. Liao, G. Ye, and M. Yang, 2023, this project, "Understanding and Detecting Abused Image Hosting Modules as Malicious Services," focuses on identifying and analyzing how legitimate image hosting platforms and modules are misused to carry out malicious activities. The study explores how attackers exploit these services to host phishing pages, malware payloads, or command-and-control content under the guise of harmless images. It investigates common abuse patterns, such as steganographic embedding of malicious code and the misuse of API-based upload functions to bypass content moderation. The system employs automated crawling, static and dynamic content inspection, and behavior-based analysis to detect suspicious uploads. By comparing normal and malicious hosting behaviors, the project builds classification models capable of recognizing abuse with high accuracy. It also examines weak authentication, lack of content validation, and insufficient URL filtering as root causes of such exploitation. Detected cases are further analyzed to understand attacker strategies, distribution techniques, and the propagation of malicious links. Finally, the project proposes a defense framework that integrates automated abuse detection with enhanced verification.

Security Threat	Vulnerability Source	Impact	Severity
Credential Leakage	Exposed presigned URLs	Unauthorized uploads	High
File Type Spoofing	Client-side validation only	Malware upload	High
Object Overwrite	Predictable file paths	Data tampering	Medium
Storage Abuse	No file size limit	Increased storage cost	Medium
Callback Spoofing	Weak verification	Unauthorized processing	Medium
Misconfigured Storage	Public bucket access	Data leakage	Critical

Y. Lv, W. Shi, W. Zhang, H. Lu, and Z. Tian, 2023, the project "Don't Trust the Clouds Easily: The Insecurity of Content Security Policy Based on Object Storage" investigates the hidden security risks that arise when web applications rely on cloud object storage services to host and deliver web content. It focuses on how developers commonly configure Content

Security Policies (CSP) to trust entire cloud domains, unknowingly introducing new attack surfaces. Attackers can exploit misconfigured or shared storage buckets to inject malicious scripts, host fake resources, or perform cross-site scripting (XSS) attacks under trusted domains. The project systematically analyzes how weak or overly broad CSP rules undermine the intended protection mechanisms of modern browsers. Using large-scale empirical testing, it examines real-world websites that depend on Amazon S3, Google Cloud Storage, and other object storage platforms. The system automates the discovery of insecure CSP configurations, correlates them with exposed storage resources, and evaluates their exploitability. It further demonstrates practical attack scenarios to show how trust abuse in CSP policies can compromise user data and web integrity. Finally, the project proposes mitigation strategies—such as granular domain whitelisting, signed URL enforcement, and integrity verification—to strengthen CSP deployment in cloud-integrated environments.

E. Trickel et al, 2023, toss a Fault to Your Witcher applies grey-box, coverage-guided mutational fuzzing to find SQL injection and command injection flaws in real-world applications. The framework combines lightweight instrumentation (to collect execution feedback) with smart mutation operators that craft input perturbations targeted at database- and shell-interacting code paths. By operating in a grey-box mode it balances efficiency and depth—using runtime coverage signals to prioritize mutations that exercise novel program behavior while avoiding blind blind fuzzing. The mutator set includes SQL-aware token swaps, quote/escape manipulations, command-chaining payloads, and context-sensitive encodings to increase the chance of hitting injection sinks. A harnessing layer automatically identifies input entry points (HTTP parameters, file uploads, CLI arguments) and adapts fuzzing strategies per target language and framework. When suspicious behavior or crashes occur, the system collects execution traces, extracts minimal reproducing inputs, and performs taint-style checks to reduce false positives. The tool also integrates lightweight oracle checks—database error patterns, unexpected shell exits, and resource-access anomalies—to confirm exploitability. Evaluation on diverse web and backend services shows the approach discovers complex, context-dependent injection bugs that static analysis and naive fuzzers miss. The project produces actionable reports with PoC requests, vulnerable code locations, and mitigation guidance like parameterized queries, input sanitization, and least-privilege execution.

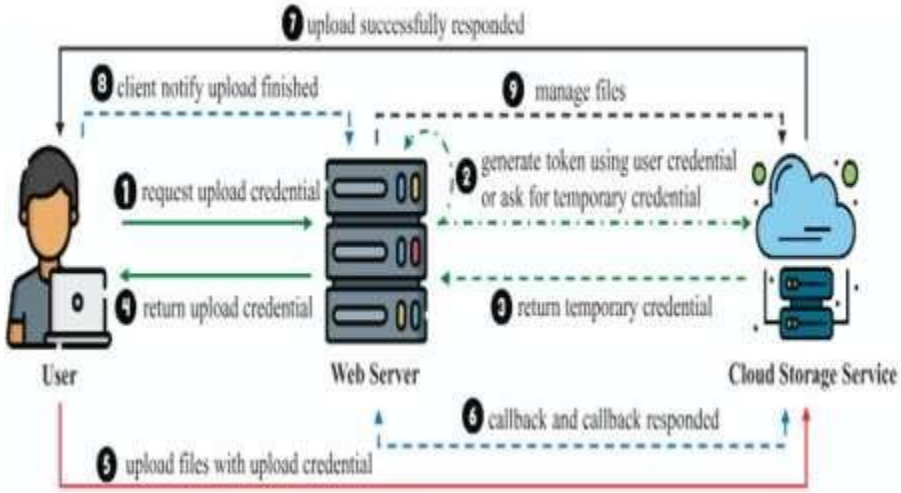
E. Pauley, R. Sheatsley, B. Hoak, Q. Burke, Y. Beugin, and P. McDaniel, 2022, the project “Measuring and Mitigating the Risk of IP Reuse on Public Clouds” investigates the security risks caused by the frequent reuse of public IP addresses in cloud environments. When cloud providers reassign IPs to new tenants, residual trust or cached DNS entries can expose new users to data leakage or hijacking attacks. The project systematically measures how often IP reuse occurs across major cloud platforms and analyzes its real-world impact on network security. It identifies attack scenarios such as misdirected traffic, SSL certificate mismatches, and unintended data exposure. Using large-scale scanning and traffic monitoring, it quantifies the persistence of stale DNS and TLS associations. The framework also explores how different providers handle IP recycling and what security gaps remain. Based on empirical results, the project proposes defense mechanisms like IP lease delay, DNS cache invalidation, and certificate revocation strategies. Finally, it offers guidelines for cloud tenants and providers to reduce risks linked to IP address reuse and improve overall cloud infrastructure security.

IV.PROPOSED SYSTEM

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.

When a web user needs to upload files, the web server will be requested to ask for the upload credentials. Note that different from user credentials, the upload credentials are typically generated temporarily from the user credentials according to the predefined security policies. In other words, the upload credentials obtained by the different web users

are typically generated from the same cloud service account. From the perspective of the cloud storage service, users are granted identical permissions. The policies may include the limitations of the allowed file types, size, storage paths, credential expiration times, etc. The web server checks whether the user is logged in, and has the upload privileges. If the verifications are passed, the web server will dispatch the upload credentials to the user in two common ways. (i) The upload credential is signed according to the preshared user credential when the website developer configures the cloud storage service. In this case, the web server does not need to interact with the cloud storage service. (ii) The web service calls the cloud storage service temporary key generation API according to the information of the user uploaded file to apply for a temporary upload credential. The credential encodes the authorization policy configured by the web server (e.g., Alibaba Cloud OSS Temporary Identity Authorization STS service). During this process, the cloud storage service needs to authenticate the web server, usually through the cloud storage service's access key (such as the 'Access Key ID' and 'Access Key Secret' of the IAM user and RAM user). After obtaining the upload credential, the web server returns it to the user. During this process, the security of the user's identity, permissions, and uploaded files is of paramount importance. With the received upload credential, the client will upload the file to the cloud storage service. In this process, since the website user is not a direct user of the cloud storage service, the cloud storage service usually does not authenticate the website user. The cloud storage service should perform security checks on the uploaded file and the upload credential. For the uploaded file, the security checks could include 1) whether it can overwrite the existing file; 2) the file's properties, e.g., whether the size or type of the file satisfies the requirements of the upload policy. For the upload credential, the cloud storage service should check its validity. For instance, if the user's upload credential is expired, the upload request should not be allowed. Overall, the cloud storage service does not pay attention.



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.

Experiment / Evaluation Parameter	Dataset / Environment	Observation / Result	Impact on Security
Website Adoption of Cloud Storage	Analysis of Top 500 Alexa-ranked websites	182 websites (36.4%) used cloud storage services for file management	Shows widespread adoption of direct cloud uploads in modern web systems
Websites Allowing Direct File Uploads	Detailed evaluation of 28 popular websites	All 28 websites had at least one security vulnerability	Indicates poor implementation of secure upload mechanisms
Total Vulnerabilities Discovered	Real-world vulnerability analysis	79 new vulnerabilities identified across evaluated websites	Demonstrates significant security risks in direct upload architectures

Experiment / Evaluation Parameter	Dataset / Environment	Observation / Result	Impact on Security
Upload Credential Acquisition (V1)	Access control analysis	Many sites allowed unrestricted acquisition of upload credentials	Attackers may obtain valid credentials and upload malicious files
Upload Credential Validity (V2)	Token lifetime evaluation	Long-lived or reusable upload tokens observed	Enables replay attacks and unauthorized uploads
File Type and Size Validation (V3)	File validation tests	Several websites allowed unrestricted file types and large files	Malware or malicious scripts can be uploaded to cloud storage
File Overwriting Attack (V4)	Object key manipulation tests	Some implementations allowed overwriting existing files	Leads to data tampering or loss of legitimate data
File Stealing Vulnerability (V5)	Access control checks	Weak permission checks enabled unauthorized download of files	Causes data leakage and privacy breaches
Callback Notification Spoofing (V6)	Callback verification testing	Missing verification allowed spoofed upload notifications	Server may process fake uploads or malicious data
Overall Security Risk	Combined vulnerability analysis	100% of analyzed upload-enabled websites were vulnerable	Indicates the need for stronger access control and credential management

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.

Systematic study of the Security risks, RSA

The proposed algorithm for the project is designed to systematically secure the direct-to-cloud file upload workflow while testing for six major vulnerabilities (V1–V6) and ensuring data confidentiality using RSA encryption. It starts with the User module requesting an upload credential from the Server, which authenticates the user and issues a scoped, time-limited credential. The algorithm first tests for unrestricted credential acquisition (V1) and credential validity flaws (V2) by attempting controlled requests with invalid, expired, or role-switched credentials to see if the Server incorrectly issues or accepts them. Once a valid credential is obtained, the User module prepares the file for upload. Here, the file is encrypted to ensure security: an RSA-based encryption approach is used, often in combination with RSA for efficiency. In this hybrid scheme, the file content is encrypted using a randomly generated RSA session key, and then this RSA key is encrypted using the Server's or per-file RSA public key. This ensures that only the authorized Server or user possessing the corresponding RSA private key can decrypt the RSA key and access the original file. During the upload stage, the algorithm checks for unrestricted file types and sizes (V3), file overwriting (V4), and file stealing (V5) by attempting safe test uploads with boundary cases, colliding filenames, or manipulated object keys. Finally, in the callback stage, the algorithm observes and simulates notifications from the cloud to the Server and tests for callback spoofing (V6) by sending crafted or tampered notifications, ensuring that the Server validates signatures, nonces, and consistency with stored metadata. Throughout this process, the system adaptively refines tests based on observed behaviors, logs evidence for each vulnerability check, and confirms findings safely to minimize false positives. This algorithm ensures that all six vulnerabilities are systematically tested while preserving the confidentiality and integrity of uploaded data.

V. CONCLUSION

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. Adding automated logging and auditing can help administrators track all operations and detect security breaches quickly. Integration with AI-based threat detection could proactively prevent attacks such as file stealing or callback spoofing. The system can also incorporate file versioning to prevent overwriting sensitive data accidentally. 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. Future development may include API-based integration with third-party applications for seamless cloud services. 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.

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.

REFERENCES

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.

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

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

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.

Annapurna Gummadi, “Reliable Optical Wireless Communication in Underwater Sensor Networks Using GRO Based DCO-OFDM”, Wireless Personal Communications, ISSN: 09296212Volume: 141Issue: 1Pages: 249 – 276.

Annapurna Gummadi, “A Machine Learning Approach in Communication 5G-6G Network”, Journal of Theoretical and Applied Information Technology, ISSN: 1992-8645, 31st May 2024. Vol.102. No. 10.

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.

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.

Annapurna Gummadi, "Deep Learning Techniques to Analysis Facial Expression and Gender Detection", IEEE International Conference on New Frontiers in Communication, Automation, Management And Security(Iccma-2023),Presidency University , Bangalore, ISSN: 979-8-3503-1706-0/23, DOI: 10.1109/ICCAMS60113.2023.10525942.

Annapurna Gummadi, "Monte Carlo Tree Search Algorithms for Strategic Planning in Humanoid Robotics", 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC - ROBINS), ISBN:979-8-3503-7274-8, DOI: 10.1109/ICC-ROBINS60238.2024.10533937, May 2024, IEEE Xplore

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.

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.

Ravindra Changala,"Proactive Market Crash Prediction: Investigating GNN-LSTM Networks for Early Detection in Stock Markets", 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), ISBN:979-8-3503-7024-9, DOI: 10.1109/ICCCNT61001.2024.10726065, November 2024, IEEE Xplore

Ravindra Changala, "Implementing Cross-Lingual Information Retrieval Systems to Enhance Resource Accessibility in English Language Learning", 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), ISBN:979-8-3503-7024-9, DOI: 10.1109/ICCCNT61001.2024.10725465, IEEE Xplore

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.

Annapurna Gummadi, "Swarm Intelligence for Multi-Robot Coordination in Agricultural Automation", 2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS), ISSN: 2575-7288, DOI: 10.1109/ICACCS60874.2024.10717088, October 2024, IEEE Xplore

Annapurna Gummadi, "A Hybrid Algorithm for Adopting the WSM System to Park the Massive Number of Vehicles in Linear and Manage the Energy Consumption", 2023 3rd International Conference on Advancement in Electronics & Communication Engineering (AECE), 979-8-3503-3072-4©2023 IEEE, DOI: 10.1109/AECE59614.2023.10428651.

Ravindra Changala, "Integration of Adaptive Neuro-Fuzzy Systems in Mobile Commerce Strategy: Enhancing Customer Relationship Management through Personalized Recommendations",2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), ISBN:979-8-3503-7024-9, DOI: 10.1109/ICCCNT61001.2024.10725950, IEEE Xplore.

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.

Ravindra Changala, "Optimization of BERT Algorithms for Deep Contextual Analysis and Automation in Legal Document Processing", 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), ISBN:979-8-3503-7024-9, DOI: 10.1109/ICCCNT61001.2024.10723962, IEEE Xplore

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.

Annapurna Gummadi, "Creating A Resilient Blockchain Framework To Enhance The Efficiency And Security Of Data Management Within Internet Of Things Networks", 2024 Third International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN), ISSN:979-8-3503-9156-5, DOI: 10.1109/ICSTSN61422.2024.10671062.

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.

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.

Annapurna Gummadi, "Human Centric Explainable AI for Personalized Educational Chatbots, 2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS), ISSN: 979-8-3503-8436-9, DOI: 10.1109/ICACCS60874.2024.10716907.

Annapurna Gummadi, "Neuromorphic Computing Architectures for Energy-Efficient Edge Devices in Autonomous Vehicles", 2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS).

Ravindra Changala, "Real-Time Multilingual Communication Enhancement Using Transformer Model for Social Media Platform", 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), ISBN:979-8-3503-7024-9, DOI: 10.1109/ICCCNT61001.2024.10725522, IEEE Xplore

Annapurna Gummadi, "A Deep CNN Self-Attention Model for Multidimensional Speech Quality Prediction Using Crowdsourced Datasets", 2025 Fifth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), EID: 2-s2.0-105004548860, DOI: 10.1109/ICAECT63952.2025.10958196.

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.

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.

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.

Academia.edu. Accessed: 2022. [Online]. Available: https://www.academia.edu/.

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/.

Ravindra Changala, "Advanced Integration of Graph Neural Networks for Collaborative Interfaces in Immersive Virtual Reality Environments", 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), ISBN:979-8-3503-7024-9, DOI: 10.1109/ICCCNT61001.2024.10724828, IEEE Xplore.

Ravindra Changala, "Sustainable Manufacturing through Predictive Maintenance: A Hybrid Jaya Algorithm and Sea Lion Optimization and RNN Model for Industry 4.0", 2024 8th International Conference on I-SMAC (IoT in Social,

Mobile, Analytics and Cloud) (I-SMAC), ISSN: 2768-0673, DOI: 10.1109/I-SMAC61858.2024.10714701, October 2024, IEEE Xplore.

Ravindra Changala, "Enhancing Robotic Surgery Precision and Safety Using a Hybrid Autoencoder and Deep Belief Network Approach: Real-Time Feedback and Adaptive Control from Image Data",2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), ISSN: 2768-0673, DOI: 10.1109/I-SMAC61858.2024.10714701, October 2024, IEEE Xplore.

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/.

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.

Ravindra Changala, "Swarm Intelligence for Multi-Robot Coordination in Agricultural Automation", 2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS), ISSN: 2575-7288, DOI: 10.1109/ICACCS60874.2024.10717088, October 2024, IEEE Xplore.

Ravindra Changala, "Hybrid AI Approach Combining Decision Trees and SVM for Intelligent Tutoring Systems in STEM Education", 2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS), ISSN: 2575-7288, DOI: 10.1109/ICACCS60874.2024.10717088, October 2024, IEEE Xplore.

Introduction to Resource Access Management—Alibaba Cloud Document Center. Accessed: 2022. [Online]. Available: https://www.alibabacloud.com/help/en/resource-access-management.

Use STS Temporary Access Credentials to Access OSS. Accessed:2022. [Online]. Available:https://help.aliyun.com/document detail/100624.html.

Ravindra Changala, "Next-Gen Human-Computer Interaction: A Hybrid LSTM-CNN Model for Superior Adaptive User Experience", 2024 Third International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), ISBN:979-8-3503-6908-3, DOI: 10.1109/ICEEICT61591.2024.10718496, October 2024, IEEE Xplore.

Ravindra Changala, "Enhancing Early Heart Disease Prediction through Optimized CNN-GRU Algorithms: Advanced Techniques and Applications", 2024 Third International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), ISBN:979-8-3503-6908-3, DOI: 10.1109/ICEEICT61591.2024.10718395, October 2024, IEEE Xplore

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.

Ravindra Changala, "Sentiment Analysis in Mobile Language Learning Apps Utilizing LSTM-GRU for Enhanced User Engagement and Personalized Feedback", 2024 Third International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), ISBN:979-8-3503-6908-3, DOI: 10.1109/ICEEICT61591.2024.10718406, October 2024, IEEE Xplore.

Ravindra Changala, "Image Classification Using Optimized Convolution Neural Network", 2024 Parul International Conference on Engineering and Technology (PICET), ISBN:979-8-3503-6974-8, DOI: 10.1109/PICET60765.2024.10716049, October 2024, IEEE Xplore.

Ravindra Changala, "Sentiment Analysis Optimization Using Hybrid Machine Learning Techniques", 2024 Parul International Conference on Engineering and Technology (PICET), ISBN:979-8-3503-6974-8, DOI: 10.1109/PICET60765.2024.10716049, October 2024, IEEE Xplore.

Ravindra Changala, "Using Generative Adversarial Networks for Anomaly Detection in Network Traffic: Advancements in AI Cybersecurity", 2024 International Conference on Data Science and Network Security (ICDSNS), ISBN:979-8-3503-7311-0, DOI: 10.1109/ICDSNS62112.2024.10690857, October 2024, IEEE Xplore.

Ravindra Changala, "Advancing Surveillance Systems: Leveraging Sparse Auto Encoder for Enhanced Anomaly Detection in Image Data Security", 2024 International Conference on Data Science and Network Security (ICDSNS), ISBN:979-8-3503-7311-0, DOI: 10.1109/ICDSNS62112.2024.10690857, October 2024, IEEE Xplore.