



A Hybrid Monitoring Framework for Detecting and Blocking Cross-VM Network Attacks in Cloud Infrastructure

Syeda Zeba Qureshi¹, Shaik Sabeel², Rehan Dawood Ghori³, Zaid Mohammed Siddiqui⁴,
Mohd Khaja Misbah Uddin⁵

¹Assistant Professor, Department of CSE (Data Science), Lords Institute of Engineering and Technology, Hyderabad, Telangana, India.

^{2,3,4,5}UG Students, Department of CSE (Data Science), Lords Institute of Engineering and Technology, Hyderabad, Telangana, India.

Abstract— This project presents a cloud-based file management system integrated with a virtual machine (VM) monitoring mechanism to identify and handle abnormal user activities. The system allows users to upload and download files through a simulation interface while a monitoring component continuously analyses incoming requests. Normal requests are processed and stored successfully, whereas suspicious or unusually large requests are detected as potential attacks and are blocked to protect system resources. The implementation demonstrates how real-time monitoring can prevent system crashes caused by malicious or excessive data uploads. Additionally, the system provides visual insights through graphical representation of normal and attack requests, helping users understand system behaviour. This approach enhances cloud security by combining user interaction, monitoring intelligence, and preventive control measures in a simple and effective manner.

Keywords— Deep Fake Tweet Detection, Machine Learning, Deep Learning, FastText Embedding, Convolutional Neural Network (CNN), Hybrid CNN, Social Media Analysis, Bot Detection, Natural Language Processing, Text Classification.

I. INTRODUCTION

Cloud computing has become an essential technology for storing and managing large volumes of data in modern applications [17]. Many organizations rely on cloud platforms to handle user requests efficiently and provide seamless access to resources. However, with the increasing dependency on cloud systems, security concerns have also grown significantly [14]. Unauthorized

access, excessive data uploads, and malicious activities can lead to system failures or data loss [3], [16]. Therefore, it is important to develop systems that not only provide functionality but also ensure protection against abnormal behaviours. This project focuses on creating a controlled cloud environment where user interactions are monitored and managed effectively. By combining file operations with a monitoring mechanism, the system ensures that only valid requests are processed, improving reliability and security in cloud-based operations.

In this system, a cloud server is designed to handle file upload and download operations from multiple users. The process begins with starting the server, followed by activating a monitoring component that observes all incoming requests. A simulation node acts as the user interface where files can be selected and sent to the cloud. When a file is uploaded, the server processes the request and stores the data if it meets normal conditions. The monitor continuously checks parameters such as file size and request patterns to determine whether the activity is safe or suspicious [6], [7]. This structured workflow ensures that every operation passes through a verification stage before execution, thereby reducing the chances of system misuse and improving overall performance.

A key feature of this project is the detection of attack-like behaviour based on unusual request characteristics. For example, when a very large file is uploaded with the intention of overloading the system, the monitor identifies it as a potential threat. Instead of allowing the request to proceed, the system blocks it to prevent damage or service interruption [4], [8]. This proactive approach helps maintain system stability even under stress conditions. The monitoring component keeps track



of all requests and categorizes them as normal or attack based on predefined criteria. This classification not only enhances security but also provides valuable insights into system usage patterns, enabling better decision-making for future improvements.

Another important aspect of the system is its ability to provide feedback and visualization of monitored data. Users can view responses from the cloud server indicating whether their files have been successfully uploaded or rejected. Additionally, the system generates a graphical representation showing the number of normal and attack requests received over time. This visual output helps in understanding how effectively the monitoring mechanism is functioning. It also makes it easier to analyse trends and identify potential risks. By integrating monitoring, detection, and visualization, the project demonstrates a practical solution for managing cloud-based systems securely while maintaining ease of use and efficiency [9], [15].

II. RELATED WORK

Seshadri et al., [2007] [1] Seshadri et al. introduced SecVisor, a lightweight hypervisor designed to ensure the integrity of kernel-level code in operating systems. Their approach focuses on preventing unauthorized code execution by restricting kernel memory access. The system enforces strict policies that allow only approved code to run in privileged mode. This work highlights the importance of protecting the core of the operating system from malicious modifications. It also demonstrates how virtualization can be used to enhance system security without significant performance overhead. The proposed solution is efficient and suitable for commodity hardware environments. However, it mainly focuses on kernel integrity and does not address higher-level application threats. The study laid a strong foundation for later research in hypervisor-based security mechanisms. It is particularly useful in environments where system-level attacks are common.

Ristenpart et al., [2009] [3] Ristenpart et al. explored the risk of information leakage in shared cloud computing environments. Their research demonstrates how attackers can identify and co-locate with target virtual machines in the cloud. By achieving co-residency, attackers can exploit side-channel attacks to extract sensitive data. The study provides practical experiments showing the feasibility of such attacks in real-world cloud platforms. It highlights vulnerabilities arising from

resource sharing in multi-tenant environments. The authors emphasize the need for stronger isolation mechanisms in cloud infrastructures. Their findings raised awareness about privacy risks in cloud computing. Although the work focuses on attack strategies, it also indirectly suggests the importance of monitoring systems. This research has influenced many later studies on cloud security and data protection. It remains a significant contribution to understanding cloud-based threats.

Wu et al., [2012] [4] Wu et al. investigated covert channel attacks within cloud environments, particularly focusing on high-speed data transmission between virtual machines. Their study shows how attackers can secretly communicate by exploiting shared resources such as CPU cache or memory. The proposed methods demonstrate that such channels can achieve significant data transfer rates without detection. This research highlights a hidden security threat that is difficult to identify using traditional monitoring techniques. It emphasizes the need for advanced detection mechanisms in cloud systems. The authors also discuss potential countermeasures to reduce the risk of covert communication. Their work contributes to understanding how internal system features can be misused. It is particularly relevant for multi-tenant cloud environments. The study encourages further research into secure virtualization techniques. Overall, it exposes critical gaps in cloud security.

Modi et al., [2013] [14] Modi et al. presented a comprehensive survey of security challenges across different layers of cloud computing. The paper categorizes threats into network, host, and application-level vulnerabilities. It also discusses various attack types, including data breaches, denial-of-service attacks, and insider threats. The authors provide an overview of existing security solutions and their limitations. Their work emphasizes the importance of adopting a multi-layered security approach. It highlights the need for continuous monitoring and intrusion detection systems. The study is useful for understanding the overall security landscape of cloud computing. It serves as a reference point for researchers and practitioners. The paper also identifies gaps where further improvements are needed. Overall, it provides a structured view of cloud security issues and solutions.

Wu et al., [2010] [16] Wu et al. examined security concerns related to virtual machines in cloud computing environments. Their work focuses on the vulnerabilities that arise due to virtualization technology. The authors discuss issues such as VM isolation, data leakage, and unauthorized access. They also analyse how attackers can exploit weaknesses in virtual machine management



systems. The study highlights the importance of securing both the hypervisor and guest operating systems. It suggests implementing strict access controls and monitoring mechanisms. The research provides insights into protecting virtual infrastructure from internal and external threats. It also emphasizes the role of network security in virtual environments. The findings are relevant for designing secure cloud architectures. Overall, the study contributes to improving virtual machine security practices.

III. DATASET DETAILS

The dataset used in this project is generated dynamically during the execution of the system rather than being collected from an external source. It consists of user-uploaded files along with their associated request information captured at different stages of the process. Each request includes attributes such as file name, file size, upload time, request type, and system response. When a user uploads a file through the simulation interface, the system records whether the request is processed successfully or flagged as abnormal. Normal requests typically involve smaller file sizes and standard usage patterns, while attack requests are identified based on unusual characteristics such as excessively large file uploads intended to stress the server. The dataset is continuously updated as multiple interactions occur, allowing the system to maintain a real-time log of all activities handled by the cloud server and monitored by the VM component.

In addition to request-level information, the dataset also includes monitoring outcomes and classification labels assigned by the VM monitor. Each entry is categorized as either “normal” or “attack” based on predefined conditions, such as threshold limits for file size or frequency of requests. The monitor keeps track of counts for both categories, which are later used for generating graphical representations of system activity. The dataset also stores information related to file retrieval operations, including file selection and download status. These details help in evaluating the system’s performance in handling both secure and potentially harmful operations. By combining user actions, system responses, and monitoring results, the dataset provides a comprehensive view of how the cloud environment behaves under different conditions, supporting both analysis and visualization of request patterns.

IV. PROPOSED METHODOLOGY

The proposed system is designed to ensure secure communication between users and the cloud server through a controlled execution process. Initially,

three main components are activated: the cloud server, the VM monitor, and the simulation node. The cloud server is responsible for storing and managing files, while the simulation node provides an interface for users to upload and download data. Every request generated from the user side is first routed through the VM monitor before reaching the server. The monitor evaluates each request based on predefined rules such as file size limits, request frequency, and data patterns. If the request satisfies all conditions, it is forwarded to the cloud server for execution. This layered approach ensures that all incoming data is verified before processing, reducing the risk of unauthorized or harmful operations.

The system also incorporates a detection mechanism to identify abnormal activities that may affect server performance. When a request exceeds the defined threshold, such as uploading a very large file, the VM monitor classifies it as a potential attack. Instead of allowing the request to proceed, the system blocks it immediately and prevents any impact on the server. All requests are logged and categorized into normal and attack types, which helps in tracking system behaviour over time. The collected data is then used to generate graphical representations, making it easier to analyse trends and system performance. Additionally, users can securely retrieve stored files from the cloud without affecting system stability. This methodology improves reliability, enhances security, and ensures smooth operation even under challenging conditions.

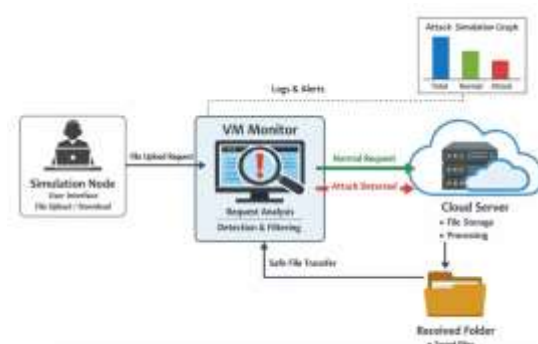


Figure [1]: System Architecture of Cloud-Based VM Monitoring and Attack Detection System

Figure [1] illustrates the overall architecture of the cloud-based file management system integrated with a virtual machine (VM) monitoring mechanism. The process begins with the Simulation Node, which acts as the user interface where users upload and download files. When a user sends a file upload request, it is first directed to the VM Monitor instead of directly reaching the



cloud server. The VM Monitor analyses each request based on predefined conditions such as file size and request behaviour. If the request is identified as normal, it is forwarded to the Cloud Server for storage and processing. However, if the system detects abnormal patterns indicating a potential attack, the request is immediately blocked to prevent system failure. The cloud server stores valid files, which can later be accessed and downloaded by the user into the received folder. Additionally, the monitoring component maintains logs of all requests and generates an attack simulation graph, showing the number of normal and malicious requests handled by the system.

V. RESULT AND DISCUSSION

The implementation of the proposed system demonstrates effective handling of both normal and abnormal file upload requests in a controlled cloud environment. During execution, the cloud server successfully processed standard file uploads without any delay or failure, confirming stable system performance under normal conditions. The VM monitor accurately identified regular requests and allowed them to proceed, ensuring smooth data storage and retrieval. When a large file was intentionally uploaded to simulate an attack, the monitoring component detected the unusual behaviour based on predefined thresholds. The request was immediately blocked, preventing potential server overload or crash. The system maintained a record of all incoming requests and categorized them correctly as normal or attack. The graphical output clearly represented the total number of requests along with their classification, providing a clear understanding of system activity. Overall, the results confirm that the system can efficiently distinguish between valid and harmful requests while maintaining reliability and responsiveness.



Figure [2]: Cloud User Simulation Interface for VM Monitoring System

Figure [2] The interface represents the user-side simulation module of the cloud-based VM monitoring system, designed to facilitate file operations and system interaction. It provides various options such as sending file requests to the

cloud, retrieving available files, downloading selected files, and viewing the attack simulation graph. The “Available Files” section allows users to view and select files stored in the cloud. Through the “Send File Request to Cloud” option, users can upload files, which are then analysed by the VM monitor before processing. The interface also includes controls for downloading files securely after validation. Additionally, the “Attack Simulation Graph” option enables users to visualize the number of normal and attack requests handled by the system. Overall, the interface is designed to be simple and interactive, allowing users to perform operations efficiently while ensuring system monitoring and security.

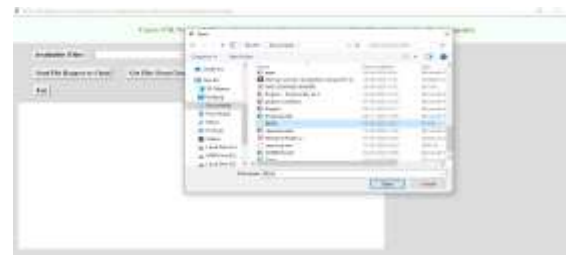


Figure [3]: File Selection and Upload Interface

Figure [3] This interface shows the file selection window that appears when the user initiates a file upload request to the cloud system. It allows the user to browse local directories and choose a specific file to be uploaded. The window displays file details such as name, type, and modified date, enabling the user to select the appropriate file easily. Once a file is selected, the user can proceed by clicking the “Open” button, which sends the file to the system. After selection, the file request is forwarded to the VM monitor for analysis before reaching the cloud server. This step plays an important role in ensuring that only valid and safe files are processed. The interface provides a simple and familiar browsing experience, making it easy for users to interact with the system while maintaining security through monitoring.

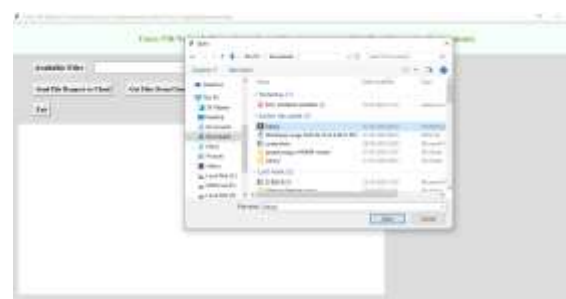


Figure [4]: File Selection from Downloads Directory



Figure [4] This screen displays the file browsing interface where the user navigates through the system directories, specifically the Downloads folder, to select a file for uploading to the cloud. The window provides a structured view of folders and files along with details such as file names, types, and modification dates, helping the user identify the required file easily. After choosing the desired file, the user can click the “Open” button to proceed with the upload process. Once selected, the file is sent to the VM monitoring system for validation before being forwarded to the cloud server. This step ensures that all user inputs are checked for security and compliance with system rules. The interface enhances usability by offering a familiar file selection environment while supporting secure and controlled data transfer within the system.



Figure [5]: Attack Simulation Graph Representation

Figure [5] This screen presents the graphical representation of system activity in the form of an attack simulation graph. The graph illustrates the total number of requests received by the system along with their classification into normal and attack requests. The x-axis represents different categories such as total requests, attack requests, and normal requests, while the y-axis indicates the corresponding count. This visualization helps in understanding how the VM monitor processes incoming requests and differentiates between safe and harmful activities. It clearly shows that normal requests are accepted and processed, whereas attack requests are identified and restricted. The graph provides a quick overview of system performance and monitoring effectiveness. It also assists in analysing patterns of user behaviour and detecting unusual activities. Overall, this visual output enhances system transparency and supports better evaluation of security measures implemented in the cloud environment.

DISCUSSION

The results highlight the importance of integrating monitoring mechanisms within cloud-based systems to enhance security and performance. The VM monitor plays a critical role in analysing incoming requests before they reach the cloud server, acting as a protective layer against potential

threats. By using simple yet effective criteria such as file size and request patterns, the system is able to detect abnormal behaviour in real time. This approach reduces the risk of system failure caused by excessive or malicious data uploads. The ability to block harmful requests ensures that server resources are preserved and legitimate users are not affected. Additionally, the graphical representation of request activity provides valuable insights into system behaviour, making it easier to identify trends and potential risks. While the current system performs well under defined conditions, further improvements can be made by incorporating advanced detection techniques and adaptive thresholds to handle more complex attack scenarios in dynamic cloud environments.

VI. CONCLUSION

The developed system demonstrates an effective approach for securing cloud-based file operations through continuous monitoring and controlled request handling. By integrating a VM monitor between the user interface and the cloud server, the system ensures that every request is analysed before execution. This helps in identifying abnormal activities such as large or suspicious file uploads that may affect system performance. Normal requests are processed efficiently, allowing users to store and retrieve files without interruption, while potentially harmful requests are blocked to maintain system stability. The implementation highlights the importance of combining monitoring techniques with cloud services to improve security and reliability. The ability to classify requests and generate visual representations provides better understanding of system behaviour and performance. Overall, the system offers a simple yet practical solution for preventing misuse of cloud resources, ensuring safe data handling, and maintaining consistent service availability.

REFERENCES

1. Seshadri, M. Luk, N. Qu, and A. Perrig, “SecVisor: A lightweight hypervisor for ensuring kernel code integrity in commodity operating systems,” *ACM SIGOPS Operating Systems Review*, vol. 41, pp. 335–350, 2007.
2. MITRE Corporation, “Common Vulnerabilities and Exposures: CVE-2008-0923,” Available: <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0923>
3. .



4. T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: Exploring data leakage risks in shared cloud environments," in Proceedings of the 16th ACM Conference on Computer and Communications Security, pp. 199–212, 2009.
5. Z. Wu, Z. Xu, and H. Wang, "Whispers in the hyper-space: High-speed covert channel attacks in cloud systems," in USENIX Security Symposium, pp. 159–173, 2012.
6. J. Rutkowska, "Subverting the Windows Vista kernel for exploitation purposes," Black Hat Briefings, 2006.
7. D. Hyde, "A comprehensive study on virtual machine security," Technical Report, Department of Computer Science, Washington University in St. Louis, 2009.
8. S. R. Kumari and V. Kathiresan, "Security practices and considerations in virtual environments," International Journal of Networking and Communication Engineering, vol. 3, no. 2, pp. 87–92, 2011.
9. S. Zhang, "An in-depth analysis of inter-virtual machine attacks in cloud systems," Technical Report, Kansas State University, Manhattan, 2012.
- A. Bates et al., "Detecting co-resident cloud instances using network flow watermarking," International Journal of Information Security, vol. 13, no. 2, pp. 171–189, 2014.
10. V. Varadarajan et al., "Analysis of placement vulnerabilities in multi-tenant public cloud environments," (publication details incomplete).
11. Adobe Systems, "Security advisory addressing vulnerabilities in Flash Player, Adobe Reader, and Acrobat (CVE-2010-1297)," 2010. Available: <http://www.adobe.com/support/security/advisories/apsa10-01.html>
12. S. Ragan, "Adobe confirms zero-day vulnerability used to bypass Windows security protections," Available: <http://www.thetechherald.com/articles/Adobe-confirms-Zero-Day-ROP-used-to-bypass-Windows-defenses/11273/>
13. R. Hund, T. Holz, and F. C. Freiling, "Return-oriented rootkits: Techniques to bypass kernel integrity protection," in USENIX Security Symposium, pp. 383–398, 2009.
14. Modi, D. Patel, B. Borisaniya, A. Patel, and M. Rajarajan, "A review of security challenges and solutions across different layers in cloud computing," Journal of Supercomputing, vol. 63, no. 2, pp. 561–592, 2013.
- A. Saeed et al., "Cross-virtual machine network channel attacks using mirroring and impersonation techniques," in IEEE International Conference on Cloud Computing (CLOUD), pp. 606–613, 2018.
15. Babburi, S. Lightweight Distributed Provenance Framework for Edge and IoT Data Systems.
16. Gaddam, S. From Fixed Specifications to Self-Adapting Systems: A Machine Learning Perspective on Software Engineering.
17. Immadi, S. K. (2025). Optimizing ERP for Human Capital Management. Applied Research for Growth, Innovation and Sustainable Impact, 377–384. <https://doi.org/10.1201/9781003684657-63>
18. Reddy, S. K. R. Developing a Modular AI Framework to Enhance Scalability and Personalization in Next-Generation Reward Platforms.
19. Poojari, R. Frameworks for Data Management and Lineage in Large-Scale Healthcare Data Systems.
20. Mahimalur, R. K., Vasgam, M., & Manoharan, D. Devops Lifecycle



- Management And Cloud Migration Assessments: A Security-Driven CICD Perspective.
21. Viswanathan, V. Generative AI for Smarter Workforce Planning and Enterprise Resource Decisions.
 22. Gajula, S. (2026, March). Two Pillars of Banking Intelligence: A Comparative Analysis of AI Techniques for Fraud Prevention and Churn Mitigation. In 2026 14th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-6). IEEE.
 23. Maturi, S. Y. (2023). Crowdsourced frontier: Unveiling autonomous adversarial cybercapabilities via open AI competition. *International Journal of Intelligent Systems and Applications in Engineering*, 11(1s), 275–284.
 24. Venkata Ramana, P. (2024). AI-driven predictive analytics in ERP systems for proactive supply chain optimization. *International Journal of Research in Information Technology and Computing*, 8(4).
 25. Srikanth Kavuri. (2025). AI-DRIVEN TEST AUTOMATION FRAMEWORKS: ENHANCING EFFICIENCY AND ACCURACY IN SOFTWARE QUALITY ASSURANCE. *International Journal of Applied Mathematics*, 38(10s), 699–710. <https://doi.org/10.12732/ijam.v38i10s.990>
 26. Venkata Pavan Kumar Gummadi. (2026). Infrastructure Optimization Techniques for Enterprise Integration Platforms: A Comprehensive Analysis. *Computer Fraud and Security*, 37–44. <https://doi.org/10.52710/cfs.875>
 27. Shashank, A. (2025). AI-Enhanced ETL Processes: Leveraging Artificial Intelligence for Optimized Data Integration Systems. *Journal Of Multidisciplinary*, 5(8), 219-225.
 28. Kandula, S. T. R., Susarla, R. S., & Boyapati, P. K. (2025, July). Enhanced Cyber Security Using Global Local Artificial Neural Network Based Intrusion Detection in Big Data Environment. In 2025 IEEE 4th World Conference on Applied Intelligence and Computing (AIC) (pp. 426-431). IEEE.
 29. Boyapati, P. K. Building a centralized data operations hub for healthcare enterprise integration. *IJSAT-Int. J. Sci. Technol.* 16 (2). <https://doi.org/10.71097/IJSAT.v16.i2.3219>