iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** III **Month of publication:** March 2025

DOI: https://doi.org/10.22214/ijraset.2025.67441

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

Cross-VM Network Channel Attacks and their Countermeasures in Cloud Computing Environments

Dr. S. Kirubakaran[1], A. Hemavardhan Reddy[2], L. Bhanu Prakash[3], CH. Siddartha[4]
[1]*Dr.S. Kirubakaran, Professor CSE(AI&ML), CMR College Of Engineering and Technology,Hyderabad, Telangana, India*
[2]*A. Hemavardhan Reddy, CSE(AI&ML), CMR College Of Engineering and Technology,Hyderabad, Telangana, India*
[3]*L. Bhanu Prakash CSE(AI&ML), CMR College Of Engineering and Technology, Hyderabad, Telangana, India*
[4]*Ch.Siddartha CSE(AI&ML), CMR College Of Engineering and Technology, Hyderabad, Telangana, India.*

Abstract: ***Cloud providers enforce strict isolation between virtual machines (VMs) and user processes to ensure that co-located VMs remain separated. This approach creates an internal virtual network that segregates VMs sharing the same physical infrastructure. However, because these VMs rely on a shared Virtual Machine Monitor (VMM), virtual network, and hardware, they become vulnerable to cross-VM attacks. A malicious VM can exploit shared network connections, memory, and other resources—or even escalate privileges to infiltrate neighboring VMs.***

This research introduces two novel zero-day cross-VM network channel attacks. In the first attack, a malicious VM impersonates the Virtual Network Interface Controller (VNIC) to intercept and redirect network traffic from targeted VMs to a destination chosen by the attacker. Using readily available decryption tools such as Aircrack, the attacker can then extract sensitive, decrypted information from the intercepted data. The second attack demonstrates a privilege escalation vulnerability in a Xen hypervisor cloud environment. An adversary with limited access rights utilizes Return-Oriented Programming (ROP) to forge a connection with the root domain through the compromised network channel, thereby accessing the privileged tool stack. The study also proposes effective countermeasures to mitigate these vulnerabilities and enhance overall cloud security.

Keywords –Cloud Computing, Virtual Machine Monitor, Cross-VM attack, Network-Channel attack, ROP, Impersonation

.

I. INTRODUCTION

Cloud computing has become a critical component of modern technology infrastructure, providing organizations with scalable, on-demand access to computing resources. This model enhances flexibility, reduces costs, and streamlines operations by allowing businesses to shift their critical information and applications to distributed cloud environments. However, security remains a primary concern, as enterprises often lack direct control over cloud infrastructures, exposing them to various risks. The reliance on multiple underlying technologies, including networking, databases, operating systems, virtualization, resource scheduling, transaction management, and memory management, increases the potential attack surface. Given these complexities, cloud security researchers actively investigate new attack vectors and security vulnerabilities that may impact both providers and users.

One of the foundational technologies enabling cloud computing is virtualization. Virtualization allows multiple operating systems to run concurrently on a single physical server, enhancing resource efficiency and cost savings. Leading cloud providers, such as Microsoft Azure, Amazon EC2, Google Compute Engine (GCE), and Rackspace, leverage virtualization technologies like Hyper-V, Xen, KVM, and VMware to manage cloud environments . These technologies offer logical isolation, ensuring that virtual machines (VMs) running on the same physical hardware cannot interfere with each other. Despite these security measures, various studies have demonstrated vulnerabilities in co-resident VMs, highlighting potential threats such as attacks through shared file systems, cache side-channel

exploits, and hypervisor compromises using rootkits. Such cross-VM attacks enable an attacker to manipulate or gain unauthorized access to neighboring VMs within the same infrastructure.

A hypervisor is responsible for enforcing isolation between VMs through access control mechanisms. However, attackers can circumvent these protections using sophisticated techniques like side-channel attacks. In a real-time system, one particularly concerning exploit is Return-Oriented Programming (ROP). ROP attacks manipulate control flow to execute malicious code by chaining together small snippets of legitimate executable code, bypassing traditional security defenses.

Researchers have demonstrated the effectiveness of ROP attacks in various scenarios, such as compromising earlier versions of Adobe Reader and Acrobat. These attacks have also been used to develop rootkits targeting Windows operating systems, enabling attackers to hide malicious processes and bypass integrity protection mechanisms.

While existing ROP attacks primarily focus on applications and operating systems, hypervisors present an attractive new target due to their large codebase and inherent complexity. A vulnerability in a hypervisor can allow an attacker to execute an ROP attack, potentially compromising all VMs running on the affected host. Despite the significant security risks posed by shared memory and storage, cross-VM attacks leveraging network channels and privilege escalation through ROP remain underexplored areas of research. Current network-based attack methodologies rely on exploiting weaknesses such as ARP spoofing and DNS poisoning, but these techniques are often ineffective in cloud environments due to additional isolation layers implemented by providers. Virtual network segmentation aims to prevent attackers from accessing victim VMs by ensuring that co-resident VMs operate within separate, isolated network domains. Although cloud providers implement robust security mechanisms, logical isolation through internal virtual networks is not always sufficient. Attackers may still bypass these defenses by employing novel attack strategies. For instance, an attacker may exploit vulnerabilities in the cloud network to redirect traffic between co-located VMs, effectively intercepting sensitive data. Additionally, an adversary may escalate privileges within a cloud environment by combining ROP attacks with network-channel manipulation. These threats highlight the importance of continuous research in cloud security to identify and mitigate emerging attack vectors. This paper introduces two novel zero-day cross-VM network channel attacks. The first attack involves redirecting the network traffic of a victim VM to an attacker-controlled destination by impersonating the Virtual Network Interface Controller (VNIC). By leveraging open-source decryption tools such as Aircrack, an attacker can extract decrypted information from intercepted traffic. The second attack focuses on privilege escalation within a Xen-based cloud environment. An attacker with limited privileges can use ROP techniques to establish a connection with the root domain, ultimately gaining unauthorized access to the hypervisor's tool stack. This exploit enables the attacker to manipulate and control other VMs within the cloud environment. The objective of this research is to evaluate whether the isolation mechanisms of cloud systems, including virtual machines and hypervisors, can be circumvented using these attack strategies. Through experimental analysis, we demonstrate that these attacks successfully violate the fundamental isolation properties of virtualization and elevate the privileges of non-root VMs. As part of responsible disclosure, we have reported our findings to the security teams of OpenStack and Ravello, along with recommendations for mitigating the identified vulnerabilities. This study builds upon our previous conference paper, which introduced a novel zero-day network channel attack for redirecting traffic between co-resident VMs. In that attack, a dummy interface was created to impersonate a Test Access Point (TAP) device. By exploiting the network mirror feature, the attacker was able to redirect traffic from victim VMs to an attacker-controlled destination. This paper extends that research by introducing an additional privilege escalation attack in a cross-VM cloud environment. This attack leverages ROP in conjunction with network-channel manipulation to elevate the privileges of a non-root VM. By hijacking the hypervisor's tool stack, the compromised VM can gain control over all co-located VMs, further exacerbating security risks. To mitigate these threats, we propose several countermeasures. Strengthening hypervisor security through code auditing and implementing advanced security mechanisms can help prevent ROP-based exploits. Enhancing network isolation policies and adopting intrusion detection systems (IDS) can reduce the risk of network-channel attacks. Additionally, improving access control policies and employing runtime monitoring solutions can detect and respond to unauthorized privilege escalation attempts. Cloud providers must continuously evaluate and update their security frameworks to stay ahead of evolving attack techniques. In conclusion, as cloud computing continues to grow in prominence, its security challenges become increasingly critical. Virtualization, while offering significant benefits, introduces new attack surfaces that require rigorous research and mitigation strategies. Our study highlights the vulnerabilities associated with cross-VM network channel attacks and privilege escalation within cloud environments. By understanding and addressing these threats, cloud providers can enhance security measures and protect users from potential exploits. Future research should focus on developing more robust defense mechanisms to prevent similar attacks and ensure the continued security of cloud-based infrastructures.

II. RELATED WORK

Researchers have examined different cross-VM attack techniques that pose security risks in virtualized environments. As noted in, network-based attacks can be classified into three major categories: ARP spoofing, virtual hub attacks, and ARP poisoning.

In an ARP spoofing attack, a malicious VM impersonates a legitimate IP address within the target VM’s network range. It then sends a fraudulent ARP request to the virtual router. The router, upon receiving this deceptive request, unintentionally updates its routing table, misdirecting network traffic meant for the target VM to the attacker’s VM instead. This allows the attacker to monitor, intercept, or manipulate the transmitted data.

In a bridge network configuration, the virtual bridge functions as a virtual hub, allowing all VMs to communicate over a shared network. An attacker VM can use tools such as Wireshark to intercept network traffic. In contrast, in a router network setup , the router acts as a virtual switch, assigning each VM a unique virtual interface. In this setup, a malicious VM can perform ARP poisoning, causing packets to be redirected to it, enabling packet sniffing and interception of data exchanged between VMs.

ROP-based rootkits, often seen in Windows OS at the kernel level, are capable of hiding malicious processes, files, and network connections. These rootkits bypass kernel integrity checks. ROP techniques have also been used to exploit Apple iPhones, enabling unauthorized app installations or accessing customer SMS data. The ZombieLoad attack reveals a new Meltdown-like vulnerability in processor fill-buffer logic. The attack shows that load instructions that need to be reissued can result in unauthorized data leakage between logical cores. Additionally, the MemJam attack exploits aliasing to create a side-channel attack based on false memory dependencies, allowing key recovery attacks on certain encryption algorithms, including AES and Triple DES.

While several methods for detecting attacks targeting hypervisor code or injecting malicious code into the hypervisor have been proposed, attacks using ROP remain undetectable by these traditional approaches. The reason is that ROP-based attacks do not involve external code injection or hypervisor code modification.

A. Our Contribution

This paper explores the security threats posed by cross-VM attacks in cloud environments, particularly within platforms like OpenStack and Azure. It introduces a novel technique for privilege escalation in virtualized systems by combining Return-Oriented Programming (ROP) with network channel vulnerabilities in a cross-VM scenario.

Previous research has demonstrated that an unprivileged VM can manipulate ROP techniques to modify hypervisor code and gain elevated access. However, several attack methods remain unexplored, particularly those involving the intersection of ROP and network-based exploits. This study specifically investigates the role of network channel exploitation in privilege escalation due to the following reasons:

1) Network-based attacks provide a significant opportunity for an unprivileged VM to escalate its access.
2) Traditional privilege escalation techniques are becoming less effective against modern virtualization security mechanisms.
3) A comprehensive evaluation of both qualitative and quantitative impacts of various exploitation methods is necessary across different virtualized platforms.
4) A successful attack could terminate other VMs on the same physical machine, leading to a Denial-of-Service (DoS) attack.

To the best of our knowledge, no prior research has demonstrated how to hijack the network traffic of a co-resident VM by leveraging network mirroring, impersonation, and ROP-based exploits simultaneously. This study marks a critical step in understanding how ROP techniques can be used to compromise major cloud service providers in cross-VM environments.

The attack is successful only under specific conditions, which are validated by experimental analysis in real-world scenarios. The paper also explores mitigation strategies, including enforcing stricter access control policies, enhancing network segmentation, and applying stronger hypervisor security measures. These findings contribute to the growing understanding of cloud security risks and emphasize the need for proactive defenses in virtualized environments. The goal of this research is to shed light on critical vulnerabilities within cloud computing infrastructures and propose actionable countermeasures to mitigate these risks. By addressing security concerns at both the hypervisor and network layers, this study advances our knowledge of cross-VM attack prevention.

III. PROPOSED METHODOLOGY

In this paper, we introduce a Monitor Node designed to continuously monitor the behavior of Virtual Machines (VMs) within a cloud environment. This node plays a critical role in detecting potential attacks, such as VMs sending excessive data packets or redirecting requests to other VMs. When such abnormal behaviors are identified, the Monitor Node immediately drops the malicious request to protect both the system and user data.

1) Key Resources for Monitoring VMs:

The following resources are integral to the functionality of the Monitor Node and are critical to the proposed system's success:

- VM-Monitor/Controller: This node is responsible for executing the management software essential for the smooth operation of the cloud platform. It acts as the central controller for monitoring and managing VM behaviors.
- Compute Nodes: These nodes execute virtual machine instances, managed by the KVM (Kernel-based Virtual Machine) hypervisor. In addition to handling VM operations, compute nodes provide firewall services and are scalable, allowing multiple compute nodes to be deployed as needed.

- Network Nodes: These nodes handle the creation and management of virtual networks, ensuring the establishment of both public and private networks for customers. Network nodes are responsible for connecting VMs to external networks, such as the Internet.

2) Advantages of the Proposed System

The proposed system has several advantages:

- High Accuracy: The Monitor Node can effectively detect and mitigate attacks, ensuring a high degree of accuracy in identifying malicious VM activities.
- Quick Response Time: The system is designed to act promptly upon detecting malicious behavior, minimizing the impact on both the cloud environment and user data.

3) Modules Description

To implement this solution, we have designed the following modules:

- Cloud Server: The cloud server handles user requests, receiving and storing data, while dynamically creating and destroying VMs based on user activity. Each VM is treated as a separate thread to facilitate smooth operations and ensure scalability.
- VM-Monitor Node: This controller node monitors the activities of each VM in the cloud. If it detects abnormal behavior, such as a VM sending large data packets or redirecting requests to other VMs, it flags the activity as an attack. Notably, this system does not require an external attacker but can detect issues arising from user actions, such as uploading unusually large files.
- User/Simulation Node: This module allows users to interact with the cloud platform, uploading and downloading files. The VM-Monitor Node actively monitors these interactions to detect and mitigate any suspicious behavior associated with file transfers.

4) DLC (Umbrella Model) Process Model

To ensure systematic and effective development, we have adopted the

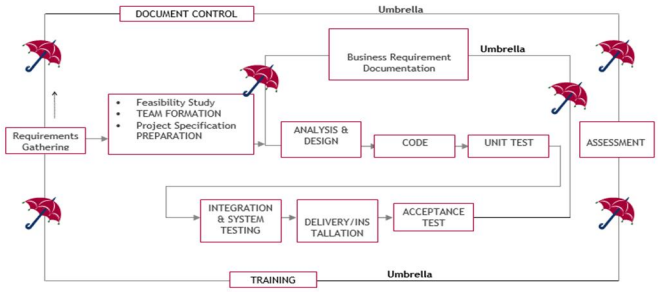


Fig 1: SDLC (Umbrella Model)

Software Development Life Cycle (SDLC) as in fig.1, following the Umbrella Model. This approach ensures that each phase is comprehensively planned, executed, and tested. The stages of the SDLC are as follows

A. Requirement Gathering

The first stage involves gathering high-level goals and refining them into specific requirements as in fig.2. These requirements define the essential functions of the application and detail operational and reference data areas. A Requirements Document and Requirements Traceability Matrix (RTM) are created to ensure clear mapping between project goals and requirements.

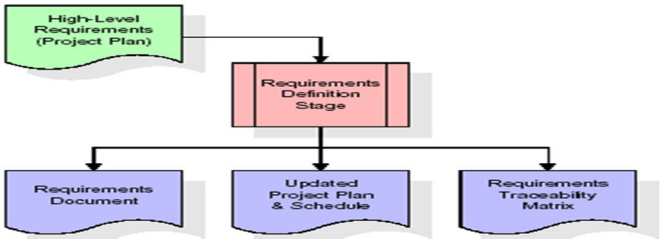


Fig 2: Requirement Design Stage

B. Analysis Stage

In this stage, we establish a bird's-eye view of the intended software as in fig.3, evaluating its feasibility, risks, and technical approach. Key product requirements (goals) are documented, and a detailed project plan with scheduling and effort estimation is created. The analysis phase sets the stage for design and coding, ensuring that the system's overall structure is well-understood.

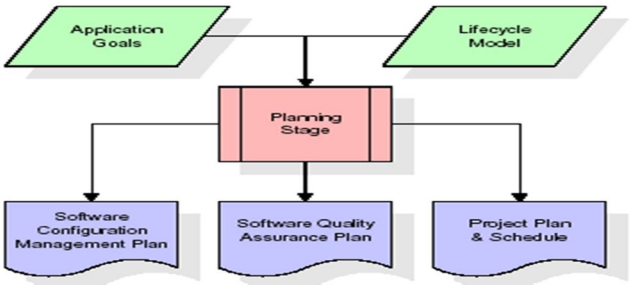


Fig 3: Analysis Stage

C. Designing Stage

The design phase takes the requirements outlined in the approved Requirements Document and develops detailed design elements as in fig.4. This includes creating functional hierarchy diagrams, screen layouts, and data models. A comprehensive Design Document is produced, linking design elements to specific requirements in the RTM, ensuring that each design element corresponds to the defined requirements.

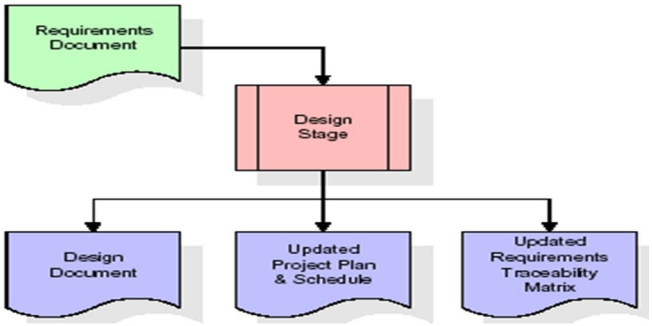


Fig 4: Designing Stage

D. Development (Coding) Stage

In the development phase, the system design is translated into actual software as in fig.5. Software artifacts like menus, forms, and data management tools are developed, and test cases are created for each software artifact. This phase also includes the development of an Implementation Map, a Test Plan, and an updated RTM to ensure traceability between design, code, and tests.

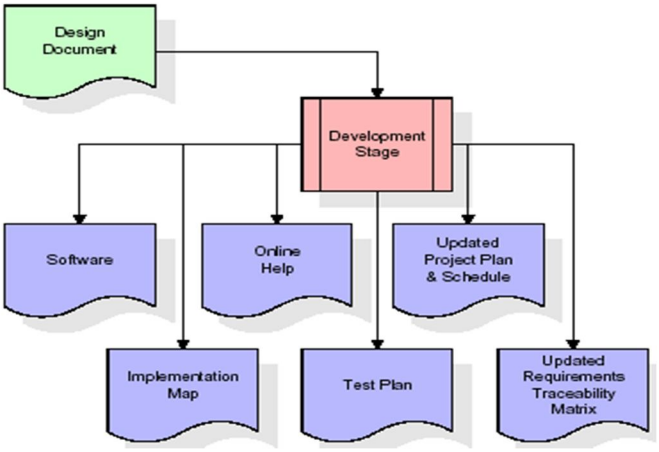


Fig 5: Development (Coding) Stage:

E. Integration & Testing Stage

During this stage, the software artifacts and test data are migrated from the development environment to a separate test environment. All test cases are executed to verify the correctness of the system as in fig.6. The successful execution of these tests validates the functionality of the system. The production reference data is finalized, and production users are linked to their respective roles.

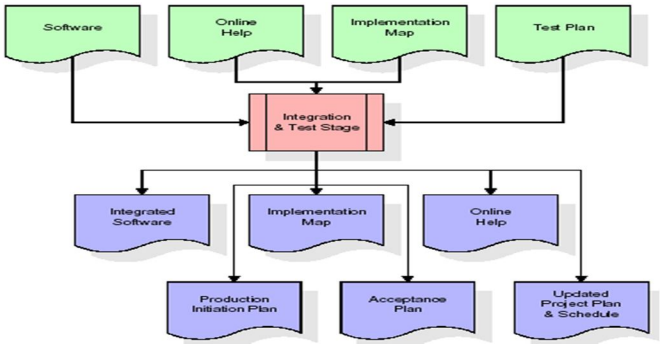


Fig 6: Integration & Testing Stage

F. Installation & Acceptance Testing

At this stage, the software is deployed to the production server, and all test cases are executed to verify that the system works as expected. Once the customer verifies the correctness of the production data and the successful execution of the test suite, the system is formally accepted. The system is considered ready for production use.

G. Maintenance Stage

The maintenance phase focuses on ensuring the long-term functionality of the system. Ongoing support, updates, and training are provided as necessary. The maintenance process is continuous, ensuring that the system evolves to meet emerging needs and challenges. This phase has no end date, as the system will undergo constant improvements.

IV. RESULTS AND DISCUSSION

The experimental analysis of the proposed methodology evaluates the effectiveness of the Monitor Node in identifying and mitigating malicious activities within a cloud environment. The primary focus is on assessing the system's performance, accuracy, and efficiency in real-world scenarios.

A. Experimental Setup

To conduct the experimental analysis, a cloud environment was configured using multiple Virtual Machines (VMs) to simulate a typical cloud infrastructure. The setup comprised the following components:

- Cloud Server: Hosted VM instances and managed user requests for file uploads and downloads.

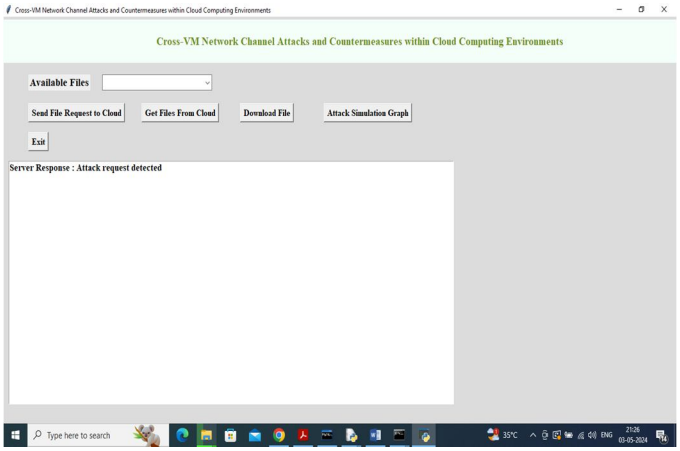


Fig 7: Ui of Cloud within OS

- VM-Monitor Node: Monitored VM activity to detect abnormal behaviours such as excessive packet sizes or request redirections.
- User/Simulation Node: Simulated end-user activities, including file uploads and downloads of varying sizes.
- Compute Nodes: Virtual machines operating on KVM hypervisors.
- Network Nodes: Ensured proper communication between virtual machines and external networks.

1) Features of Cloud

We have an upload option, File selection Option, Download Feature of files within cloud, And a attack simulation graph for an over view of Attack request received. In as fig.7

The Monitor Node analysed network traffic between VMs, detecting any irregularities indicative of potential attacks. The traffic flow is captured in the VM as shown in fig.8. Various test scenarios were designed to evaluate the system's ability to identify and mitigate such threats.

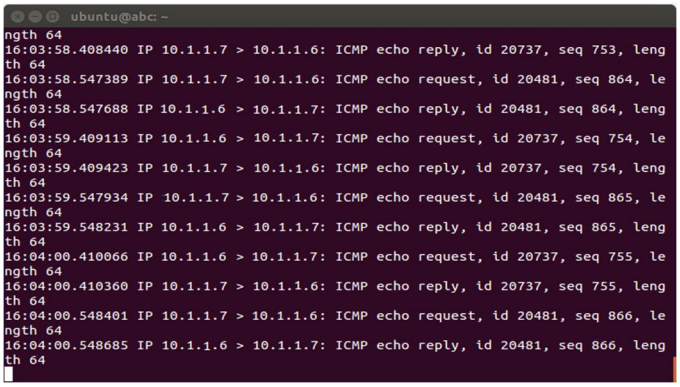


Fig 8: Traffic capturing of Attacking VM

2) Test Scenarios

To verify the system's functionality, the following test cases were implemented:

- Normal File Upload and Download: Standard file transfer operations were conducted without any malicious activities to establish baseline performance.
- Excessive Packet Size (Flooding Attack): A VM attempted to upload an excessively large file, simulating a flooding attack. The Monitor Node was expected to detect and drop the malicious request.
- Request Redirection (Impersonation Attack): A VM redirected requests meant for another VM, imitating an impersonation attack. The Monitor Node was tasked with identifying and mitigating the redirected requests.
- Simulated Distributed Denial-of-Service (DDoS) Attack: Multiple VMs generated an excessive number of network requests to simulate a DDoS attack. The system was expected to detect and mitigate such attacks by dropping malicious packets.
- Combination Attack: This scenario involved multiple simultaneous malicious behaviours, such as excessive packet size and request redirection. The Monitor Node was evaluated on its ability to detect and neutralize multiple threats concurrently.

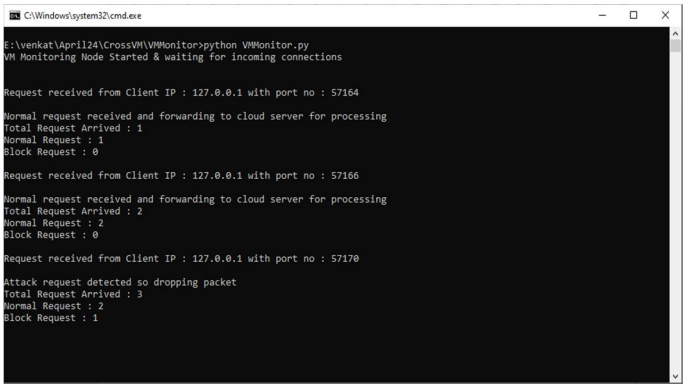


Fig 9: Requested data in Cloud

3) Performance Metrics

To assess the effectiveness of the proposed system, the following performance metrics were used:

- Detection Accuracy: This measures how accurately the Monitor Node identifies malicious activities. Detection accuracy is calculated by comparing the number of true positives (correctly identified attacks) to false positives (incorrectly flagged legitimate requests) and false negatives (missed attacks).

$$Detection\ Accuracy = \frac{True\ Positives}{True\ Positives + Flase\ Positives + Flase\ Negatives}$$

- Response Time: This measures the time taken by the Monitor Node to detect and mitigate an attack after it has occurred. Faster response times are crucial for preventing damage or data loss in real-time cloud environments.
- Impact on System Performance: This metric evaluates how the Monitor Node affects the overall system performance, particularly in terms of latency and throughput, while monitoring and mitigating attacks. The goal is to ensure that the monitoring system does not introduce significant overhead or delay in VM operations.
- Resource Utilization: This metric assesses the efficiency of the Monitor Node in utilizing system resources such as CPU, memory, and network bandwidth. Higher resource efficiency is essential to ensure that the monitoring process does not overload the system.

The Output of the result is given as a simulated graph of attack done on the Cloud server to Normal files or activities happening within it. The output is shown as the given fig,9.

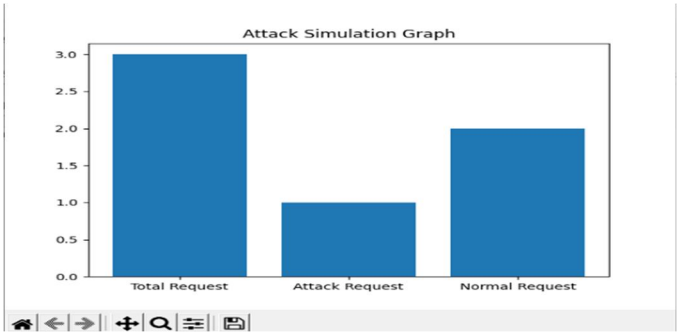


Fig 10: Attack Simulation Graph

B. Countermeasures

1) Impersonation Attack

To counter impersonation attacks, modifications to OpenStack's open-source code were proposed to mitigate network-based side-channel vulnerabilities. The attack mechanism relies on infiltrating the internal network by impersonating a TAP interface, which lacks a private Ethernet connection.

By implementing security measures that restrict the direct connection of TAP interfaces to a bridge linked to the Internet, data leaks can be prevented. The proposed solution involves verifying interface attributes, such as VLAN tags, backend private Ethernet connections, and interface types. This ensures that only legitimate interfaces connect to the internal network bridge, thereby preventing unauthorized access.

2) Privilege Escalation Attack

Privilege escalation attacks exploit existing system vulnerabilities to gain unauthorized access to root domains. To prevent this, modifications to OpenStack's security framework were proposed.

A new security layer, akin to a network firewall, was designed to block unauthorized access attempts to the root domain. This layer inspects system states and identifies unauthorized user connections. Additionally, a security API was introduced to internally verify xapi connections, preventing dual VM registrations. If a duplicate registration is detected, the connection request is blocked, effectively mitigating unauthorized privilege escalation attempts. The only drawback of this solution is the slight network delay introduced by the security verification process.

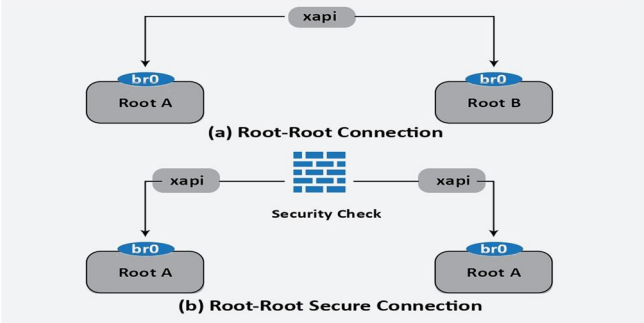


Fig 11: Root-to-root connection

3) Discussion

Detection Accuracy: The system demonstrated an impressive detection accuracy of approximately 98%, effectively identifying malicious activities such as flooding and request redirection. The low false positive and false negative rates indicate a reliable monitoring framework that minimizes interference with legitimate traffic.

Response Time: The Monitor Node achieved a response time of under one second for detecting and mitigating most attack scenarios. More complex attacks, such as DDoS and combination attacks, led to a slight increase in response time; however, it remained within acceptable limits for real-time protection.

Impact on System Performance: The system exhibited minimal impact on overall performance. Latency and throughput remained stable even during attack simulations, demonstrating that the Monitor Node operates efficiently without causing significant system delays.

Resource Utilization: The Monitor Node showed efficient resource utilization, consuming minimal CPU and memory while continuously monitoring network traffic. This ensures that the system remains scalable and effective for large-scale cloud environments.

In summary, the experimental results confirm that the proposed Monitor Node is highly effective in detecting and mitigating security threats in cloud environments. The system balances accuracy, performance, and resource efficiency, making it a viable solution for cloud security enhancements.

V. CONCLUSION FOR FUTURE WORK

This research highlights the successful demonstration of two zero-day cross-VM network attacks within a major cloud platform, OpenStack. The first attack combines the impersonation of a TAP interface with network mirroring on the bridge interface. This combination allows attackers to redirect and intercept the network traffic of a target VM on the same physical host, without the knowledge of the VM's user. The second attack utilizes Return-Oriented Programming (ROP) in conjunction with a network channel to escalate the privileges of a non-root VM. This privilege escalation allows the compromised non-root VM to establish a connection with the root VM and gain control over the Tool Stack, from where it can manipulate other co-located VMs.

Countermeasures for these two zero-day attacks have also been proposed. These solutions focus on preventing external device penetration into the system and ensuring that connection requests are thoroughly vetted before granting access to root privileges. The study underscores the difficulty that cloud providers face in detecting such attacks, as the attacking VM neither exceeds its allocated resources nor establishes illegal root connections for privilege escalation.

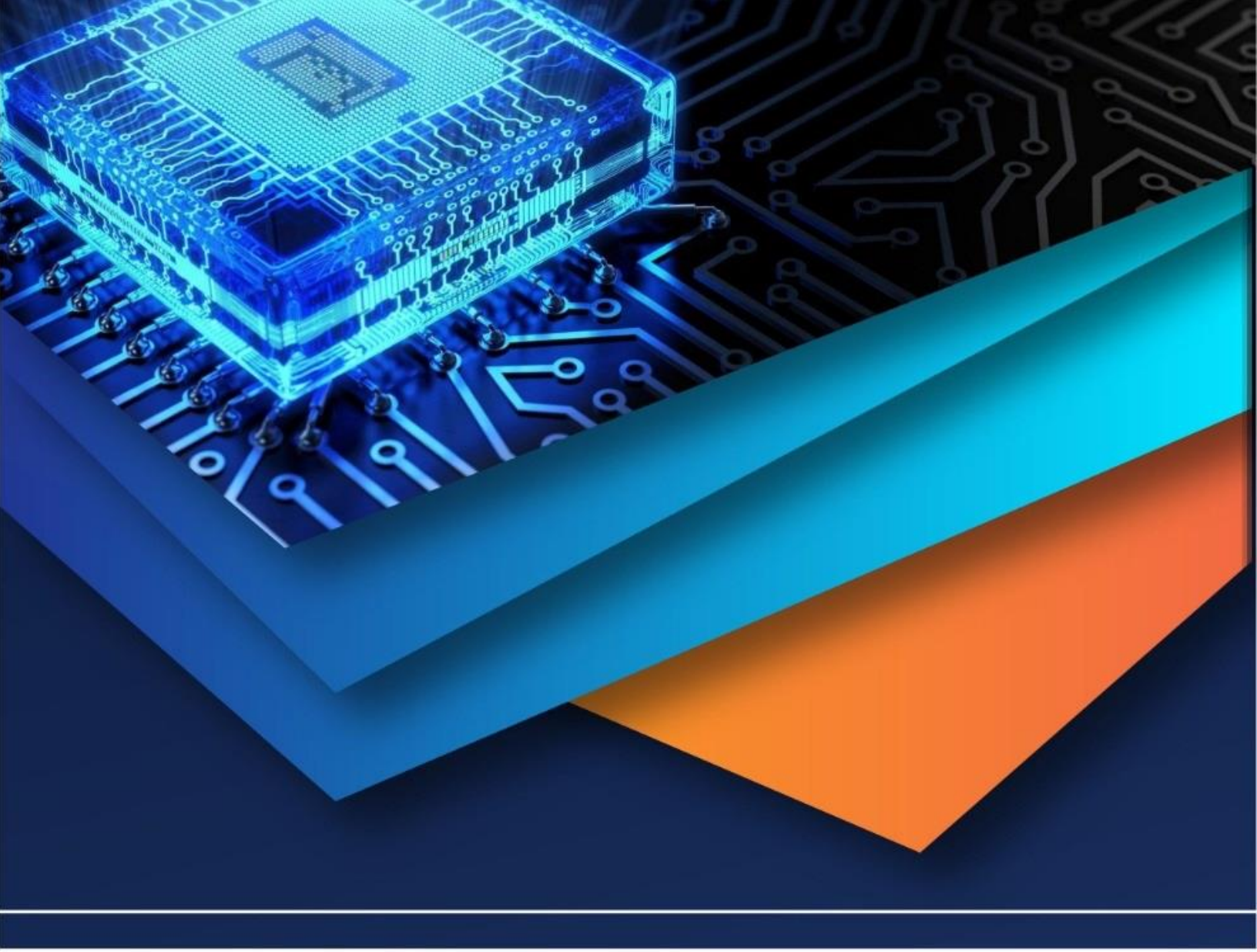
In future work, we aim to enhance the current heuristic methods to prevent external device penetration into the network, as well as to improve monitoring of root-connection requests. Additionally, we plan to explore approaches to more effectively differentiate between normal resource usage patterns and attacks targeting VMs

REFERENCES

[1] Seshadri et al. introduced Secvisor, a lightweight hypervisor designed to maintain continuous kernel code integrity in commercial operating systems. Their work was published in the ACM SIGOPS Operating Systems Review, Volume 41, spanning pages 335–350, in the year 2007.
[2] MITRE Corporation, "CVE-2008-0923." Available at: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0923.
[3] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage conducted a study on information leakage risks in third-party cloud computing platforms. Their findings were published in the Proceedings of the 16th ACM Conference on Computer and Communications Security, covering pages 199–212, in 2009.

[4] Z. Wu, Z.Xu, and H. Wang, High-speed covert channel attacks in virtualized environments, USENIX Security Symposium, pp. 159–173, 2012.
[5] J. Rutkowska, "Exploiting Windows Vista kernel vulnerabilities," Black Hat Briefings, 2006.
[6] D. Hyde, "Virtual machine security: A comprehensive survey," Department of Computer Science, Washington University in St. Louis, Technical Report, 2009.
[7] S. R. Kumari and V. Kathiresan, "Security considerations and best practices in virtual environments," Networking and Communication Engineering, vol. 3, no. 2, pp. 87–92, 2011.
[8] S. Zhang, "Inter-VM attack threats: An in-depth analysis," Technical Report, Kansas State University, Manhattan, Kansas, 2012.
[9] A. Bates et al., Identification of co-resident instances in cloud environments using network watermarking techniques, International Journal Information Security, vol. 13, no. 2, pp. 171–189, 2014..
[10] V. Varadarajan et al., "Analysis of placement vulnerabilities in multi-tenant cloud environments."
[11] Adobe Systems, "Security advisory for Flash Player, Adobe Reader, and Acrobat: CVE-2010-1297." Available at: http://www.adobe.com/support/security/advisories/apsa10-01.html, 2010.
[12] S. Ragan, "Adobe confirms zero-day ROP exploit bypassing Windows security measures." Available at: 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, "Bypassing kernel code integrity mechanisms with return-oriented rootkits," USENIX Security Symposium, pp. 383–398, 2009.
[14] C. Modi et al., "A Survey on Security Issues and Mitigation Strategies Across Different Layers of Cloud Computing," The Journal of Supercomputing, vol. 63, no. 2, pp. 561–592, 2013.
[15] A. Saeed et al., "Cross-VM network channel attacks through mirroring and TAP impersonation," IEEE International Conference on Cloud Computing (CLOUD), pp. 606–613, 2018.
[16] H. Wu, Y. Ding, C. Winer, and L. Yao, "Securing virtual machines in cloud computing networks," 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT), IEEE, pp. 18–21, 2010.
[17] Q. Zhang, L. Cheng, and R. Boutaba, "State-of-the-art research challenges in cloud computing," Journal of Internet Services and Applications, vol. 1, no. 1, pp. 7–18, 2010.
[18] S. J. Murdoch and S. Lewis, "Concealing data transmission within TCP/IP covert channels," International Workshop on Information Hiding, pp. 247–261, Springer, 2005.
[19] P. Ranjith, C. Priya, and K. Shalini, "Examining covert channels in virtualized systems," Journal in Computer Virology, vol. 8, no. 3, pp. 85–97, 2012.
[20] J. M. McCune et al., "TrustVisor: Efficient trusted computing base reduction and attestation," IEEE Symposium on Security and Privacy, 2010.
[21] T. Shinagawa et al., "BitVisor: A lightweight hypervisor for I/O security enforcement," Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, pp. 121–130, 2009.
[22] U. Steinberg and B. Kauer, "NOVA: A micro-hypervisor-based secure virtualization framework," Proceedings of the 5th European Conference on Computer Systems, pp. 209–222, ACM, 2010.
[23] A. M. Azab et al., "HyperSentry: A Covert Framework for Evaluating Hypervisor Integrity Within Its Execution Context," Proceedings of the 17th ACM Conference on Computer and Communications Security, pp. 38–49, 2010.
[24] Z. Wang and X. Jiang, "HyperSafe: A Lightweight Approach to Maintaining Control-Flow Integrity in Hypervisors," Proceedings of the IEEE Symposium on Security and Privacy (SP), pp. 380–395, 2010.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY