Journal of Theoretical and Applied Information Technology
15th February 2024. Vol.102. No 3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SECURE AND ISOLATED COMPUTING IN VIRTUALIZATION AND CLOUD ENVIRONMENTS: A SYSTEMATIC REVIEW OF EMERGING TRENDS AND TECHNIQUES

[1] MOHAMMAD SHKOUKANI [2] SAMER MURRAR [3]RAWAN ABU LAIL [4]KHALIL YAGHI

[1]Department of Computer Science, Applied Science Private University, Amman, Jordan

[1]MEU Research Unit, Middle East University, Amman, Jordan

[2] Department of Computer Science, Applied Science Private University, Amman, Jordan

[3]Department of Computer Science, Philadelphia, Amman, Jordan

[4]Faculty of Information Technology, King Abdulaziz University, Jeddah, Sudia Arabia

E-mail: [1] m.shkokani@asu.edu.jo, [2] 202215019@students.asu.edu.jo, [3]rabulail@philadelphia.edu.jo
[4]kaahamad1@kau.edu.sa

ABSTRACT

In an era defined by the omnipresence of virtualization and the increasing dependence on cloud computing, ensuring efficient and secure virtual machine management is of paramount importance. This paper presents a detailed review of some innovative studies, each addressing distinct challenges in the realm of cybersecurity and virtual machine performance. The first study examines the use of Lightweight Kernels and Trusted Execution Environments to optimize security isolation capabilities, demonstrating promising results for high-performance computing platforms. The second study explores the application of machine learning techniques for detecting anomalies in cloud based virtual machine resource usage, contributing to a proactive security approach. The third study presents SecFortress, an approach that enhances hypervisor security using cross-layer isolation techniques. Together, these studies underscore the significance of continual research in secure, efficient computing and offer promising avenues for future development.
Keywords: *Virtual Machine, High-Performance Computing, Cybersecurity, Machine Learning, Hypervisor Security*

1. INTRODUCTION

The continuous evolution of the digital landscape necessitates rigorous investigation and enhancement of computing security, particularly within virtualization and cloud environments. Given the rapid advancements and unique challenges in these areas, this systematic literature review is essential to synthesize and analyze current research trends, methods, and strategies. Unlike previous studies that may focus on narrower aspects, this SLR aims to provide a more comprehensive perspective. As trusted execution and secure resource isolation become more prevalent features in modern hardware architectures, their focus has been largely targeted at end-user devices and commodity systems platforms. However, there remains a significant gap in research that specifically focuses on the security requirements of high-performance computing (HPC) systems and cloud computing environments. The gap becomes increasingly apparent when considering the increasing complexity and sophistication of cyber threats in these domains. The present studies, while valuable, frequently fail to comprehensively address the complex difficulties associated with protecting high-performance computing (HPC) and cloud environments. Moreover, the ever-changing nature of these fields requires constant evaluation and incorporation of the most recent advancements and methodologies.

The purpose of this study is to address this deficiency by gathering and analyzing pertinent research in these underexplored domains. The distinguishing feature of our systematic literature review is its comprehensive coverage of a diverse range of emerging technologies and strategies.

Additionally, it offers valuable insights into the effective implementation and integration of these technologies into existing systems. An all-encompassing approach is essential for gaining a thorough comprehension of the present condition of security in virtualized environments [1].

A crucial shift is needed to address the security needs of high-performance computing (HPC) class systems and cloud computing environments, which are yet to be fully explored. This systematic review attempts to explore and synthesize findings from publications on developing trends and strategies in safe and isolated computing [1].

Cloud Computing, as defined by The National Institute of Standards and Technology (NIST), is a model that provides ubiquitous and on-demand access to a pool of shared and configurable computing resources [2]. Its key characteristics such as on-demand self-service, broad network access, resource pooling, rapid elasticity, and scalability [2] have contributed to its growth and adoption. However, as the usage of cloud computing increases, it brings along various security vulnerabilities including data breaches, privacy concerns, and unauthorized access [3], [4]. These issues underscore the urgent need for improved security measures in cloud environments.

Virtualization plays a pivotal role in cloud computing, with the hypervisor being its most critical software component. Hypervisors face substantial security threats primarily from untrusted VM states and a more susceptible but privileged OS that assists in virtualization and manages resources [5]. Existing strategies attempt to 'harden' or 'isolate' the hypervisors to tackle these vulnerabilities [6], [7], but these methods often fall short or induce additional overhead.

An essential feature of modern secure computing is the Hafnium hypervisor, offering isolated resource partitions which allow for the secure execution of private workloads [8]. However, the necessity of a full-weight Linux kernel instance to manage VM scheduling on each CPU core indicates room for enhancement.

In this systematic review, we aim to synthesize the information available in distinct studies concerning secure and isolated computing within virtualization and cloud environments. Our focus will be on identifying and analyzing emerging trends, techniques, and methodologies while discussing the associated challenges and potential strategies to address these issues. By providing a comprehensive overview of the current state of research in the field, we aim to contribute to the ongoing discourse and further advancements in secure computing.

This paper will begin by providing an overview of HPC environments and their future scope, followed by a discussion on secure and trusted computing capabilities for these environments. We will then shift our focus to cloud computing and the security challenges it faces, before concluding with an examination of hypervisor security and the innovative SecFortress design.

2. BACKGROUND

The burgeoning demand for secure computing platforms has led to the development of several technologies designed to meet this need. Hafnium, ARM TrustZone, and nested kernel are three such technologies that have been developed with the intent of enhancing security, particularly within the realm of virtualization and cloud computing environments. Each of these technologies comes with unique characteristics and benefits that contribute to the development of secure systems [8].

The presented text encompasses a comprehensive examination of a number of important technologies and mechanisms pertinent to system security, isolation, and virtualization. Specifically, this includes the Hafnium project, ARM Trust-Zone technology, hypervisors, and the nested kernel approach.

a) Hafnium, a part of the Trusted Firmware Initiative, is a valuable project that provides reference implementations of secure system software for ARM based platforms [9]. By focusing on virtualization, Hafnium offers an isolation layer for virtual machines, enabling entire VM contexts to be insulated from other software components on the system. This unique design lends itself to deploying full HPC applications in secure computing environments. However, Hafnium's architecture imposes certain restrictions on dynamic resource partitioning and VM management, necessitating unique solutions to these challenges [9].

b) ARM TrustZone is an implementation of a Trusted Execution Environment (TEE) that provides hardware-enforced memory isolation at the system firmware level [1]. This technology is integral to the augmentation of Hafnium's memory protection capabilities. TrustZone creates a system bifurcation into secure and non-secure worlds, enforcing a strict division at boot time which prevents non-secure partitions from accessing secure memory contents

[1]. While Hafnium and TrustZone can work in unison, the implementation involves managing the complexity of handling both secure and non-secure VM instances [1].

c) Hypervisors are paramount to the virtualization software stack, especially in the management of physical resources. They operate as a vital layer of abstraction between hardware and the virtual machines running atop it, ensuring the isolation of each VM [10]. The hosted hypervisor, as typified by the KVM, delegates resource management to a host OS, like Linux, thereby implementing CPU and memory virtualization [10].

d) The nested kernel approach provides an additional layer of security by situating a small, isolated kernel within a larger monolithic one [11]. By effectively controlling all updates to virtual address translation, this approach safeguards physical memory and significantly diminishes the Trusted Computing Base (TCB) in complex systems [11].

These technologies, in conjunction with each other, form a layered defense strategy to secure various components of a computing environment. They underscore the increasing significance of layered security approaches in complex systems, aiming to ensure maximum protection in our digital age.

3. RELATED WORKS

Cloud security and privacy are paramount concerns in the era of digital data, with a variety of challenges identified and measures proposed to address these issues. One of the major issues includes abuse of cloud computational resources, leading to efficiency loss and service degradation [12], [13]. Additionally, breaches of data confidentiality and integrity pose significant risks to both service providers and clients [12].

Malicious insiders, cyber theft, and malware injections present additional security threats. These potential risks highlight the necessity for comprehensive and robust security strategies. Chou [12] has emphasized the role of access management, suggesting the implementation of firewalls and intrusion detection systems to prevent unauthorized access. Other measures proposed include user behavior profiling to detect and respond to anomalous activities, as well as the deployment of advanced authorization and authentication technologies [12].

In public cloud services, particularly Infrastructure as a Service (IaaS) models, these security challenges can be even more pronounced due to shared resources. In such settings, the vulnerabilities of virtualization, hypervisor, and shared resources become more significant [14], [15]. Prashanthi [15] highlighted the need for novel security measures that are explicitly designed with virtualization in mind, providing a more proactive and preventive security solution for cloud systems.

The role of Intrusion Detection Systems (IDS) as a key defense mechanism in the security architecture of cloud computing has been underscored by various researchers [16], [17]. These systems function as a security layer for detecting potential threats, including intruders and attackers. Despite their importance, IDS are primarily used for the detection of network-level attacks, suggesting a need for expansion into other areas of security.

Statistical and machine learning approaches have been suggested as promising avenues for improving the efficacy of IDS [18], [19]. These techniques have the potential to enhance detection capabilities and response times. However, much of the existing work on statistical and machine learning frameworks has been focused on network security, specifically against attacks such as denial of service [20]. This points to the opportunity for broader application of these techniques in other aspects of cloud security.

Several approaches have been taken to harden the security of the virtualization layer, addressing the inherent vulnerabilities present in the hypervisor and virtual machines. HyperLock and DeHype, for instance, attempt to isolate each virtual machine with a dedicated hypervisor instance, thus reducing the threat surface [21]. SecFortress, in comparison, applies a similar concept but extends the protection to the host OS as well [22]. In essence, it aims to safeguard the hypervisor against a compromised host OS, which is not addressed by HyperLock and DeHype.

Security solutions like SecVisor adopt a unique approach by creating a tiny hypervisor that operates at a higher privilege level than the host OS, thereby ensuring kernel integrity [22]. However, such a model introduces additional context switches and overhead. Contrarily, SecFortress focuses on maintaining isolation among different guest VMs, thereby not only providing kernel integrity protection but also building comprehensive security for cloud platforms [22].

In recent years, hardware-based defense technologies have also seen significant

advancements. AMD's Secure Encrypted Virtualization (SEV) and Intel's TDX are some notable examples that protect guest memory and isolate VMs from the hypervisor, respectively [23], [24]. Although they provide enhanced security, these systems primarily focus on safeguarding VMs from an untrusted hypervisor, and do not necessarily protect the hypervisor against malicious VMs. In contrast, SecFortress offers bidirectional isolation protection between VMs and the hypervisor [22]. Such comprehensive security measures are critical for ensuring the integrity of the entire cloud infrastructure.

Cloudvisor [22] and Cloudvisor-D [25], two other innovative solutions, leverage nested virtualization design to protect virtual machines from a compromised hypervisor. However, unlike SecFortress, their primary focus is to isolate the impact of the hypervisor on VMs and do not necessarily aim to harden the hypervisor itself.

NOVA, HypSec [26], SecKVM [27], and SKEE [28] are other noteworthy efforts in the domain of hypervisor security. These systems adopt a microkernel design to reconstruct the hypervisor, effectively reducing the hypervisor's Trusted Computing Base (TCB). However, unlike SecFortress, which takes additional steps to ensure cross-VM control protection, these systems do not provide such extensive security measures. The approaches by vTZ [29], HA-VMSI [30], and HyperCoffer [31] also deserve mention. While vTZ protects the guest-TEE from an untrusted hypervisor by virtualizing ARM TrustZone, it does not support protection for normal world VMs. HA-VMSI, on the other hand, provides protection for guest VMs from an untrusted hypervisor but supports only a limited set of virtualization features. HyperCoffer uses a secure processor with encryption to protect VM's data, offering another layer of security.

The use of hardware-based defense technologies, including AMD SEV [23] and Intel TDX [24], has been prevalent in recent years. These technologies are designed to isolate VMs from the hypervisor and non-Trusted Domain (TD) software by creating a secure arbitration mode. However, their main aim is to prevent the virtualization platform from attacking VMs, and they do not protect the hypervisor from attacks by malicious VMs.

In contrast to these, AWS Nitro Enclaves [32] creates isolated VM execution environments or 'enclaves' on the Nitro hypervisor, providing another perspective to VM and hypervisor security. The Arm CCA [33] architecture, meanwhile, builds on TrustZone and introduces the Realm Management Extension to protect all data and code, although its focus is more on application protection rather than the entire VM.

4. METHODOLOGY

In this section, we delve into the methodology of innovative studies that have each contributed significant advancements in the realm of cybersecurity and virtual machine management. The studies we will explore are:

" Low Overhead Security Isolation using Lightweight Kernels and TEEs" [34] which investigates how to optimize security isolation capabilities with minimal overhead using Lightweight Kernels and Trusted Execution Environments (TEEs) [37].

"Machine Learning-Based Anomalies Detection in Cloud Virtual Machine Resource Usage" [35] presents an exploration of machine learning techniques for detecting anomalies in the resource usage of cloud-based virtual machines.

"SecFortress: Securing Hypervisor using Cross-layer Isolation" [36] discusses an approach to securing a hypervisor using cross-layer isolation techniques [37].

Each study adopts unique methodologies and techniques to address the complex problems in their respective fields. In this section, we aim to unpack these methodologies, explore the implementation processes, and identify the challenges encountered along the way. The aim is to develop a comprehensive understanding of the techniques employed by each paper, fostering a greater appreciation of the research and potential for future developments.

In the technique of Low Overhead Security Isolation using Lightweight Kernels and TEEs [34], the implementation methodology begins with the creation of a secure environment. This environment is based on Hafnium and operates on the ARM architecture. Figure 1 depicts the hafnium system architecture.

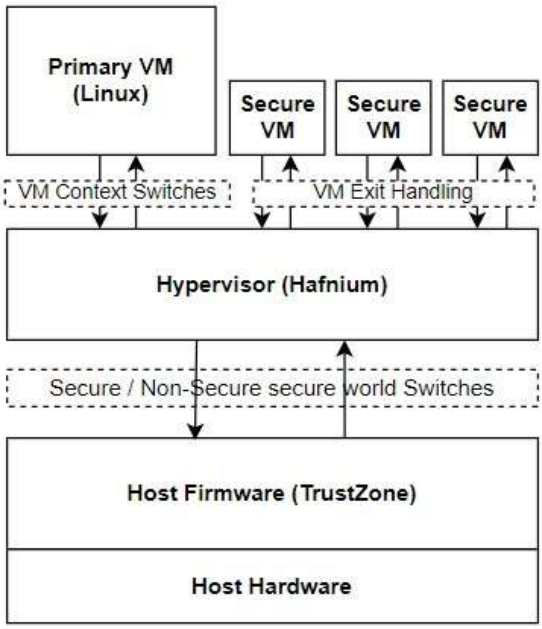

Figure 1: Hafnium VM Configuration

This platform serves as the foundation for deploying the Kitten Lightweight Kernel (LWK) as the primary scheduling VM [38], [39]. The goal is to significantly reduce, if not entirely eliminate, the OS overheads exerted on HPC workloads running in secure VM instances. Furthermore, maintaining a Linux Operating System/Runtime (OS/R) instance is crucial to handle necessary system management operations that modern HPC environments require.

At the heart of this approach is the deployment of Kitten as the primary VM within a Hafnium-based secure virtualization environment. The implementation of Kitten in this way aims to mitigate or significantly reduce the OS overheads that the primary scheduling VM imposes on HPC workloads. This stage of the implementation process encounters a variety of challenges, particularly concerning I/O and device drivers. These challenges stem from Hafnium's design, which anticipates a full-featured OS kernel operating as the primary VM.

Parallelly, to ensure the availability of a Linux OS/R instance, a Linux-based environment is proposed to be hosted as a management VM instance or a "super-secondary" instance. This super-secondary instance exists in a semi-privileged state, positioned between the primary VM and other secondary VMs. It has the ability to directly interact with I/O devices but doesn't have full access to the hypercall API or the capacity to control CPU cores. The super-secondary instance assumes a critical role as it takes over I/O responsibilities from the Kitten primary VM, harnessing the Linux device driver ecosystem.

Finally, the super-secondary incorporates a Linux user space environment, facilitating the easy deployment and accessibility of system management frameworks. The super-secondary VM manages job control responsibilities and has the ability to configure system resources as well as manage the lifecycles of the secondary VMs. This is accomplished through a secure communication channel between the super-secondary and primary VMs, allowing the super-secondary to issue commands to a control task executing in the Kitten VM instance [40]. This innovative approach offers a feasible system architecture that has been demonstrated via a prototype implementation.

For the technique of Machine Learning-Based anomaly detection in Cloud Virtual Machine Resource Usage [35], this work proposes a model. The model leverages an existing machine-learning algorithm to detect deviations in virtual machine resource usage, thereby indicating potential anomalies or threats.

The methodology used in this technique comprises crucial parts - the working model, data collection, and implementation. The model operates under the assumption that anomalies can be detected by tracking deviations from a normative VM resource usage pattern, identified via continuous monitoring of VM metrics like disk read/write throughput, memory, and CPU usage. This data, after feature scaling to ensure metric comparability, forms the basis of time-series data, which is then used to deduce VM usage patterns [35], [37].

For data collection, the authors used the workload trace dataset from the Grid Workload Archive, a compilation of VM resource usage over a month-long period between August and September 2013. This dataset was also used in other research, studying the characteristics of cloud datacenters hosting business-critical workloads. After scaling, the metrics were restructured into hourly and daily time-series for model training and testing [35].

In the implementation stage, the authors utilized two unsupervised machine learning algorithms from the Scikit-Learn library in Python, namely Isolation Forest and One Class Support Vector Machine (One Class SVM) [35]. The former was used to isolate anomalies by constructing an ensemble of isolation trees, and the latter was deployed to establish datapoint boundaries during training and to assess new data points during testing [35]. The training and

testing were done on ten virtual machine datasets, each containing over 5000 data points. These datasets, once aggregated into hourly and daily time series, contained the requisite VM metrics [35].

This detailed methodology forms the bedrock of this research, offering a robust example of applying machine learning for proactive anomaly detection in VM usage patterns.

In the technique of Securing Hypervisor using Cross-layer Isolation (SecFortress) [36], the work is centered on enhancing hypervisor security against threats from untrusted outerOS and VMs. The methodology outlined comprises two main components: the mediator and cross-layer isolation [37]. Figure 2 presents the architecture of SecFortress.

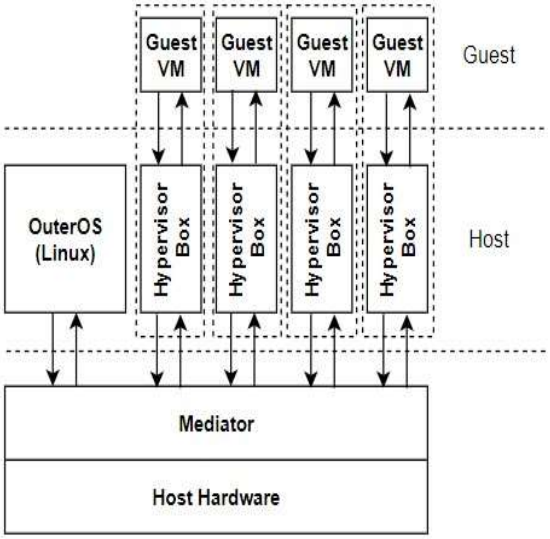

Figure 2: SecFortress Architecture

The mediator serves as the crucial component of SecFortress protection [36]. It ensures tamper-proof security by leveraging nested MMU to create a memory protection domain for monitoring memory mapping updates within the system [36]. By virtualizing the MMU, the mediator configures kernel pagetable-pages as read-only and employs CPU write protection, preventing unauthorized modifications. Furthermore, the mediator provides security services such as memory protection, instruction protection, and enforced control flow [36]. These services enforce security policies and prevent unauthorized access and execution, ensuring the integrity and confidentiality of the system [37].

Cross-layer isolation is established between the outerOS and HypBox, as well as among different HypBox instances [36]. SecFortress creates isolated address spaces for HypBoxes, effectively separating them from the outerOS and preventing unauthorized access [36]. The mediator controls memory mapping updates, ensuring the confidentiality and integrity of each HypBox instance. Additionally, robust isolation mechanisms are enforced among HypBox instances to prevent attacks between VMs [36]. The mediator restricts direct access to sensitive instructions and prevents code injection attacks, ensuring the security and isolation of each HypBox instance.

The methodology of SecFortress, employing the mediator and cross-layer isolation mechanisms, provides a comprehensive approach to enhancing hypervisor security against untrusted components [36]. The implementation of SecFortress on KVM, running on Ubuntu 18.04.5 with Linux 5.2 for Intel VT, and utilizing QEMU 4.0.0 as the virtualization software at the user level, demonstrates the practicality and effectiveness of the proposed approach [36].

5. RESULTS

The experiments from the first study utilized the Pine A64LTS Single Board Computer (SBC) platform for benchmark testing. The evaluation demonstrated that adding a virtualization layer had little impact on the noise profile of the environment. The Stream and Random-Access memory tests showed expected results, with Random-Access performance being most affected by the presence of Hafnium due to low TLB hit rates. The High-Performance Conjugate Gradients (HPCG) mini-app test showed similar performance across the three configurations, with the Hafnium with Kitten scheduler configuration showing slightly better performance [34].

The second study focused on data preprocessing and anomaly detection. Normalizing the data did not alter the distribution of the data points. The use of a One-Class Support Vector Machine (OCSVM) showed an average F1- score of 0.97 for the hourly time series, and 89% for the daily time series, outperforming the Isolation Forest which had average scores of 93% and 80% for hourly and daily time-series respectively [35].

The third study evaluated the performance overhead of SecFortress. Results showed negligible overhead for CPU intensive tasks. Memory-intensive tasks showed a slight performance overhead of 0.3% to 0.7% due to monitoring and checking of memory operations. In terms of I/O

performance, SecFortress had an average overhead of 5% for disk I/O and 5.5% for network I/O, outperforming the nested KVM in these respects. Micro benchmarks demonstrated that the overhead of switching to HypBox was 2626 cycles and to the mediator was 235 cycles [36] [37].

6. DISCUSION

In the technique of Low Overhead Security Isolation using Lightweight Kernels and TEEs, the results showed that Hafnium, when paired with the Kitten scheduler, performed exceptionally well in the HPCG mini-app test. This finding challenges previous literature, suggesting that adding a virtualization layer doesn't necessarily negatively affect performance. However, a slight dip in performance was observed in the Random-Access test. This could point to a need for optimization, especially when dealing with tasks that have low Translation look-aside buffer (TLB) hit rates.

Our study, "Secure and Isolated Computing in Virtualization" expands upon the previous findings by conducting a thorough analysis of secure and isolated computing in virtualization and cloud environments. In contrast to the concentrated investigation of lightweight kernels and TEEs, our systematic literature review covers a wider array of technologies and strategies, providing a comprehensive perspective on the present condition of security in virtualized environments. In addition, our review combines the findings of the machine learning-based anomaly detection study with other emerging trends, resulting in a comprehensive understanding of anomaly detection and its applications in cloud security. This synthesis provides valuable insights into resource usage anomalies in cloud-based VMs.

In addition to the study "SecFortress: Securing Hypervisor using Cross-layer Isolation," our research looks into a range of inventive methods for enhancing hypervisor security, going beyond the scope of cross-layer isolation techniques. By adopting this comprehensive approach, we can identify gaps in existing research and propose potential avenues for future exploration. This, in turn, highlights the originality and impact of our research within the wider scope of cybersecurity and virtual machine performance.

The technique of Machine Learning-Based anomaly detection in Cloud Virtual Machine Resource Usage showed that OCSVM consistently outperforms Isolation Forest in anomaly detection. This was true for both hourly and daily time series. Although data normalization did not affect the distribution, the findings suggest that OCSVM is more robust in handling various data distributions and scales. One limitation of this study, however, is that it only compared two machine learning models. Therefore, the performance of OCSVM could potentially vary when applied to different datasets.

In the evaluation of the SecFortress technique, we found that it offers improved performance compared to nested KVM, particularly in I/O operations. This suggests that SecFortress could be an effective solution for enhancing VM performance in terms of I/O. However, the slight performance overhead observed in memory-intensive tasks is an area that should be addressed in future development.

These studies together provide valuable insights into the performance and security of virtual machines, and the use of machine learning for anomaly detection. Future research could explore the integration of these areas, for example, by developing machine learning models to detect and respond to performance issues in virtual machines.

7. CONTRIBUTIONS AND OPEN RESEARCH ISSUES

This study has made significant contributions to the field of secure and isolated computing in virtualization and cloud environments:

- ✓ **Broadened Scope of Research:** We provide a thorough analysis of security in high-performance computing (HPC) and cloud environments, filling a notable research void that concentrates on these domains. Having a broader perspective is essential for understanding the complex landscape of secure computing.
- ✓ **Synthesis of Emerging Trends:** Through the synthesis of information from multiple studies, we aim to emphasize the emerging patterns, approaches, and methodologies in secure and isolated computing. This will contribute to a more comprehensive comprehension of the present condition of the field.
- ✓ **Identification of Key Challenges:** The study focuses on identifying and analyzing the main obstacles in ensuring secure computing for high-performance computing (HPC) systems and cloud environments. One of the key challenges is maintaining security isolation while minimizing any negative impact on performance.

Despite these contributions, several research issues remain open:

- ✓ **Optimization of Security Isolation:** Further research is required to enhance security isolation in HPC platforms, while minimizing any negative effects on performance.
- ✓ **Machine Learning Applications in Cloud Security:** Further investigation can be conducted to explore a wider array of machine learning models to detect anomalies and enhance security in cloud computing.
- ✓ **Security in Emerging Technologies:** With the advancement of technology, it is imperative for research to adjust and tackle security challenges in emerging fields like quantum computing and edge computing.

These contributions and unresolved research issues create opportunities for future advancements in secure computing, promoting ongoing exploration and progress in this important domain.

8. CONCLUSION

Based on the research presented, it is clear that advancements in high-performance computing (HPC), anomaly detection in cloud computing, and virtual machine performance and security have significant implications for a wide range of applications.

The use of Lightweight Kernels, specifically the Kitten LWK, in conjunction with the Hafnium hypervisor, showcased the potential for securely isolated HPC systems. Initial evaluations indicated minimal performance overhead across various HPC benchmarks. Despite this, the challenge of maintaining security isolation without affecting performance in future HPC platforms was highlighted. This is an area that calls for further research, emphasizing the importance of security isolation and trusted computing in next-generation HPC platforms.

The Machine Learning-Based technique focused on addressing the security challenges in cloud computing, specifically in detecting and mitigating anomalies in user resource usage. The proposed model, which utilizes virtual machine resource metrics, demonstrated a high success rate when tested with the One-Class Support Vector Machine (OCSVM). These results suggest that the model could serve as a viable solution for improving cloud security, possibly by integration into a monitoring-as-a-service module. Additionally, the model's adaptability to changes in VM specifications suggests that it can be continually improved and used, even as technology evolves.

SecFortress introduces a groundbreaking cross-layer isolation approach aimed at bolstering hypervisor runtime security. By implementing features like platform partitioning, restricted memory access, and enhanced integrity and confidentiality of instances, SecFortress proves to be effective in preventing potential security breaches against both the host OS and VMs. Moreover, it accomplishes this with minimal performance overhead, marking it as a viable solution for secure virtualization.

These studies underscore the importance of continual research and development in the realm of secure, efficient computing. The strategies and techniques they present provide promising directions for improving the performance and security of HPC systems, cloud computing, and virtual machines. Future work should further explore and expand on these initial findings, paving the way for even more robust and secure computing environments.

8.1 Strengths and Weaknesses

8.1.1 Strengths:

- ✓ **Comprehensive Scope:** Our study provides a comprehensive examination of security in high-performance computing (HPC) and cloud environments, encompassing emerging patterns and tactics that have not been extensively investigated in prior research.
- ✓ **Innovative Methodologies:** Utilizing machine learning models to identify and resolve performance problems in virtual machines presents an innovative strategy for tackling security concerns in cloud computing.
- ✓ **Practical Insights:** The assessment of Lightweight Kernels, particularly the Kitten LWK in combination with the Hafnium hypervisor, provides practical observations on HPC systems that are securely isolated with minimal impact on performance.

8.1.2 Weaknesses:

- ✓ **Limited Focus on Performance Optimization:** While we acknowledge the difficulty of preserving security isolation without compromising performance, our research would benefit from a more thorough investigation of optimization strategies for upcoming high-performance computing (HPC) platforms.
- ✓ **Narrow Scope in Machine Learning Applications:** a focus on particular

machine learning techniques, while beneficial, may restrict the comprehension of alternative methods in anomaly detection and cloud security.

8.2 FUTURE RESEARCH DIRECTIONS

Building on our findings, future research should:

- ✓ **Explore Optimization Techniques:** Explore sophisticated techniques for enhancing security isolation in high-performance computing (HPC) platforms while maintaining optimal performance.
- ✓ **Broaden Machine Learning Applications:** Expand the investigation of machine learning methodologies to encompass a more diverse array of models and their suitability in various cloud computing scenarios.
- ✓ **Focus on Emerging Technologies:** Analyze the impact of emerging technologies, such as quantum computing and edge computing, boosting the security of virtualized environments.
- ✓ **Evaluate Real-world Implementations:** Perform case studies or real-world evaluations to evaluate the practical efficacy of the proposed strategies in various computing environments.

9. ACKNOWLEDGMENTS

The authors are grateful to the Applied Science Private University, Amman, Jordan, for the full financial support granted to this research.

REFERENCES:

[1] "Arm trustzone technology." https://developer.arm.com/technologies/trustzone. Accessed august 2023.

[2] P. Mell and T. Grance, "SP 800-145: The NIST Definition of Cloud Computing." National Institute of Standards and Technology, September 2011.

[3] C. Meyers, "The biggest cloud breaches of 2019 and how to avoid them for 2020." Lacework Editorial, December 2019. [4] Available at: https://www.lacework.com/top-cloud-breaches-2019/.

[4] M. Gontovnikas, "The 11 biggest data breaches of 2020 ." Auth0 Blog, 2020. Available at: https://auth0.com/blog/the-11-biggest-data-breachesof-2020-so-far/.

[5] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, "kvm: the linux virtual machine monitor," in Proceedings of the Linux symposium, pp. 225–230, Dttawa, Dntorio, Canada, 2007.

[6] A. M. Azab, P. Ning, Z. Wang, X. Jiang, X. Zhang, and N. C. Skalsky, "Hypersentry: enabling stealthy in-context measurement of hypervisor integrity," in Proceedings of the 17th ACM conference on Computer and communications security, pp. 38–49, 2010.

[7] Y. Wu, Y. Liu, R. Liu, H. Chen, B. Zang, and H. Guan, "Comprehensive vm protection against untrusted hypervisor through retrofitted amd memory encryption," in 2018 IEEE International Symposium on High Performance Computer Architecture (HPCA), pp. 441–453, IEEE, 2018.

[8] "Hafnium hypervisor." https://www.trustedfirmware.org/projects/hafnium/. Accessed September 2023.

[9] M. Boubakri, F. Chiatante, and B. Zouari, "Open portable trusted execution environment framework for risc-v," in 2021 IEEE 19th International Conference on Embedded and Ubiquitous Computing (EUC), pp. 1–8, 2021.

[10] F. Bellard, "Quick emulator." https://www.qemu.org, 2001.

[11] N. Dautenhahn, T. Kasampalis, W. Dietz, J. Criswell, and V. Adve, "Nested kernel: An operating system architecture for intra-kernel privilege separation," in Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 191–206, 2015.

[12] T.-S. Chou, "Security threats on cloud computing vulnerabilities," International Journal of Computer Science & Information Technology, vol. 5, no. 3, p. 79, 2013.

[13] A. F. S. Althobaiti et al., "Analyzing security threats to virtual machines monitor in cloud computing environment," Journal of Information Security, vol. 8, no. 01, p. 1, 2017.

[14] K. Hashizume, D. G. Rosado, E. Fernandez-Medina, and E. B. Fernan-´ dez, "An analysis of security issues for cloud computing," Journal of internet services and applications, vol. 4, pp. 1–13, 2013.

[15] M. Prashanthi, "Analysis of security issues in virtualization cloud computing," International Journal of Computer Science and Mobile Computing, vol. 5, no. 8, pp. 274–281, 2016.

[16] Y. Mehmood, M. A. Shibli, U. Habiba, and R. Masood, "Intrusion detection system in cloud computing: challenges and opportunities," in

2013 2nd national conference on information assurance (NCIA), pp. 59– 66, IEEE, 2013.
[17] T. Tiwari, A. Turk, A. Oprea, K. Olcoz, and A. K. Coskun, "Userprofile-based analytics for detecting cloud security breaches," in 2017 IEEE International Conference on Big Data (Big Data), pp. 4529–4535, IEEE, 2017.
[18] C. N. Modi, D. R. Patel, A. Patel, and R. Muttukrishnan, "Bayesian classifier and snort based network intrusion detection system in cloud computing," in 2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12), pp. 1–7, IEEE, 2012.
[19] S. Gupta, P. Kumar, and A. Abraham, "A profile based network intrusion detection and prevention system for securing cloud environment," International Journal of Distributed Sensor Networks, vol. 9, no. 3, p. 364575, 2013.
[20] H. Nguyen, Y. Tan, and X. Gu, "Pal: P ropagation-aware a nomaly l ocalization for cloud hosted distributed applications," in Managing Large-scale Systems via the Analysis of System Logs and the Application of Machine Learning Techniques, pp. 1–8, 2011.
[21] W. Shi, J. Lee, T. Suh, D. H. Woo, and X. Zhang, "Architectural support of multiple hypervisors over single platform for enhancing cloud computing security," in Proceedings of the 9th conference on Computing Frontiers, pp. 75–84, 2012.
[22] F. Zhang, J. Chen, H. Chen, and B. Zang, "Cloudvisor: retrofitting protection of virtual machines in multi-tenant cloud with nested virtualization," in Proceedings of the twenty-third acm symposium on operating systems principles, pp. 203–216, 2011.
[23] D. Kaplan, J. Powell, and T. Woller, "Amd memory encryption," White paper, 2016.
[24] Intel, "Intel trust domain extensions." https://software.intel.com, 2023.
[25] Z. Mi, D. Li, H. Chen, B. Zang, and H. Guan, "(mostly) exitless vm protection from untrusted hypervisor through disaggregated nested virtualization," in Proceedings of the 29th USENIX Conference on Security Symposium, pp. 1695–1712, 2020.
[26] S.-W. Li, J. S. Koh, and J. Nieh, "Protecting cloud virtual machines from hypervisor and host operating system exploits," in Proceedings of the 28th USENIX Security Symposium, 2019.
[27] S.-W. Li, X. Li, R. Gu, J. Nieh, and J. Z. Hui, "Formally verified memory protection for a commodity multiprocessor hypervisor," in 30th USENIX Security Symposium (USENIX Security 21), pp. 3953–3970, 2021.
[28] A. M. Azab, K. Swidowski, R. Bhutkar, J. Ma, W. Shen, R. Wang, and P. Ning, "Skee: A lightweight secure kernel-level execution environment for arm.," in NDSS, vol. 16, pp. 21–24, 2016.
[29] Z. Hua, J. Gu, Y. Xia, H. Chen, B. Zang, and H. Guan, "vtz: Virtualizing arm trustzone.," in USENIX security symposium, pp. 541–556, 2017.
[30] M. Zhu, B. Tu, W. Wei, and D. Meng, "Ha-vmsi: A lightweight virtual machine isolation approach with commodity hardware for arm," ACM SIGPLAN Notices, vol. 52, no. 7, pp. 242–256, 2017.
[31] Y. Xia, Y. Liu, and H. Chen, "Architecture support for guest-transparent vm protection from untrusted hypervisor and physical attacks," in 2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA), pp. 246–257, IEEE, 2013.
[32] Amazon, "AWS Nitro Enclaves." https://docs.aws.amazon.com/enclaves/, 2023.
[33] ARM, "ARM Confidential Compute Architecture." https://www.arm.com, 2023.
[34] J. R. Lange, N. Gordon, and B. Gaines, "Low overhead security isolation using lightweight kernels and tees," in 2021 SC Workshops Supplementary Proceedings (SCWS), pp. 42–49, IEEE, 2021.
[35] P. Ntambu and S. A. Adeshina, "Machine learning-based anomalies detection in cloud virtual machine resource usage," in 2021 1st International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS), pp. 1–6, IEEE, 2021.
[36] Q. Zhou, X. Jia, S. Zhang, N. Jiang, J. Chen, and W. Zhang, "Secfortress: Securing hypervisor using cross-layer isolation," in 2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 212–222, IEEE, 2022.
[37] R. Abulail, S. Murrar, and M. Shkoukani, "An Enhanced Approach for Realizing Robust Security and Isolation in Virtualized Environments," International Journal of Advanced Computer Science and Applications, vol. 14, no. 11, pp. 293-299, 2023. doi: 10.14569/IJACSA.2023.0141129.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[38] J. Ouyang, B. Kocoloski, J. R. Lange, and K. Pedretti, "Achieving performance isolation with lightweight co-kernels," in Proceedings of the 24th International Symposium on High-Performance Parallel and Distributed Computing, pp. 149–160, 2015.

[39] B. Kocoloski and J. Lange, "Better than native: Using virtualization to improve compute node performance," in Proceedings of the 2nd International Workshop on Runtime and Operating Systems for Supercomputers, pp. 1–8, 2012.

[40] R. Pi, "Raspberry pi 3 model b," online].(https://www.raspberrypi.org, 2023).