An Enhanced Approach for Realizing Robust Security and Isolation in Virtualized Environments

Rawan Abuleil[1], Samer Murrar[2], Mohammad Shkoukani[3]
Department of Computer Science, Philadelphia University, Amman, Jordan[1]
Department of Computer Science, Applied Science Private University, Amman, Jordan[2, 3]

***Abstract*—Transitioning into the next generation of supercomputing resources, we're faced with expanding user bases and diverse workloads, increasing the demand for improved security measures and deeper software compartmentalization. This is especially pertinent for virtualization, a key cloud computing component that's at risk from attacks due to hypervisors' integration into privileged OSs and shared use across VMs. In response to these challenges, our paper presents a two- pronged approach: introducing secure computing capabilities into the HPC software stack and proposing SecFortress an enhanced hypervisor design. By porting the Kitten Lightweight Kernel to the ARM64 architecture and integrating it with the Hafnium hypervisor, we substitute the Linux-based resource management infrastructure, reducing overheads. Concurrently, SecFortress employs a nested kernel approach, preventing outerOS from accessing mediator's memory, and creating a hypervisor box to isolate untrusted VMs' effects. Our initial results highlight significant performance improvements on small scale ARM-based SOC platforms and enhanced hypervisor security with minimal runtime overhead, establishing a solid foundation for further research in secure, scalable high- performance computing.**

***Keywords*—Virtual Machine (VM); High-Performance Computing (HPC); cybersecurity; hypervisor security**

I. INTRODUCTION

Advances in computing technology have ushered in a paradigm shift in how computational resources are deployed and utilized in recent decades. The rapid growth and adoption of virtualization technologies are at the forefront of this transition [1]. By abstracting the physical hardware from the software, virtualization enables the creation of multiple isolated Virtual Machines (VMs) that can run concurrently on a single physical machine, resulting in significant improvements in resource utilization and cost efficiency [1].

However, as with any technology, virtualization brings with it new challenges, most notably in the area of security [1]. The hypervisor, the abstraction layer that allows the creation of VMs, is a lucrative target for attackers [2]. If an attacker successfully compromises the hypervisor, they may gain control of all VMs running on the system, resulting in a significant security breach [3]. Furthermore, while the isolation of VMs from each other and the host system is beneficial for security, it can also be used by attackers to conceal malicious activity [3].

This necessitates the development of enhanced security solutions capable of effectively protecting the hypervisor and the virtual machines that run on it [4]. Several technologies have been developed to this end, providing various mechanisms for securing virtualized environments [4]. Hafnium, ARM TrustZone, and SecFortress, for example, provide unique security solutions that can significantly improve the security of virtualized environments [1, 2, 5].

Hafnium is a microkernel-based VM monitor that provides secure isolation between virtual machines [5]. It enables each VM to run in its own isolated environment, memory-separated from other VMs [5]. This means that even if an attacker gains control of one VM, they cannot access the memory of other VMs [5]. Hafnium also ensures control flow integrity (CFI), which prevents unauthorized changes to a VM's control flow [5]. Fig. 1 depicts the default hafnium system architecture. Hafnium, as illustrated in the diagram, relies on a primary VM to make scheduling decisions and explicitly invoke context switches to secondary VMs via a privileged hyper-call interface.

ARM TrustZone, on the other hand, provides a secure execution environment within a processor that allows sensitive tasks to be run in a separate environment from the rest of the system [1]. ARM TrustZone can protect the system from both external and internal threats by ensuring that only authenticated and verified code is allowed to run within this secure environment [1].

SecFortress approaches security differently. Its goal is to protect the hypervisor by isolating it from the rest of the operating system [2]. SecFortress ensures that an attack on one VM or the outer operating system does not compromise the hypervisor or any other VMs by providing each VM with its own dedicated hypervisor box and preventing direct interaction between the hypervisor and the outer operating system [2]. Fig. 2 presents the architecture of SecFortress.

Each of these technologies offers a distinct solution for securing virtualized environments, but their full potential may be realized only when they are integrated into a unified security framework. The purpose of this paper is to investigate the integration of these technologies into a multi-layered security solution for virtualized environments, with the goal of providing comprehensive protection against both external and internal threats [3].

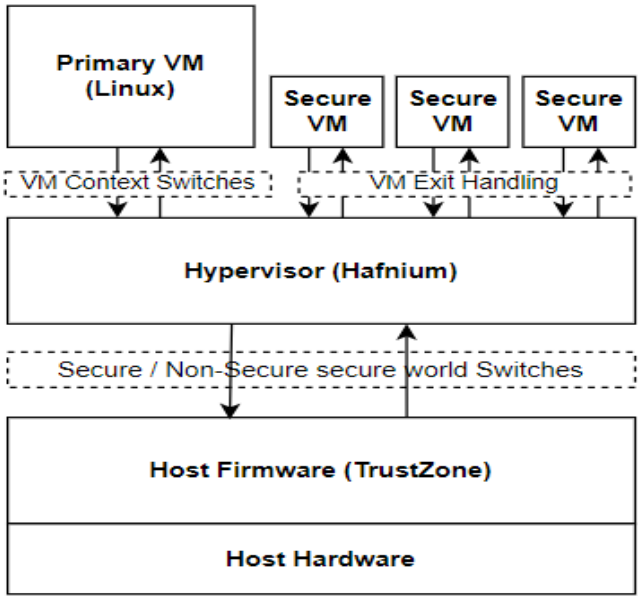

Fig. 1. Hafnium VM configuration.

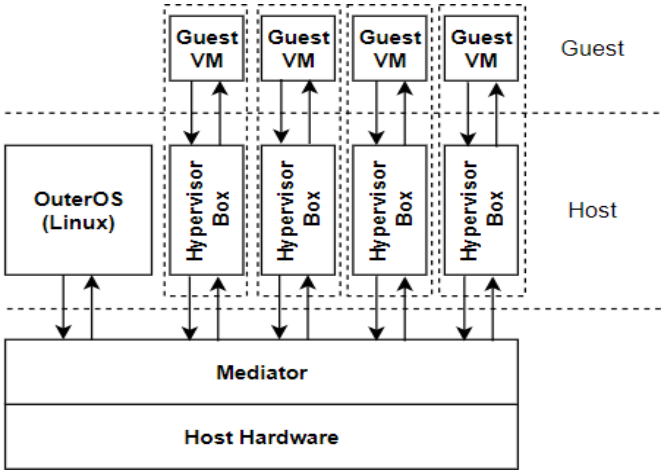

Fig. 2. SecFortress architecture.

The complexities of integrating various security mecha isms, the potential performance overheads of multiple security layers, and the requirement for ongoing updates to address new security vulnerabilities all present significant challenges to the implementation of such a framework. These challenges, however, can be addressed with careful planning and design, paving the way for a more robust and secure computing environment in the virtualization era.

II. Background

Virtualization, enabled by lightweight kernels or hypervisors, has transformed computing by allowing multiple virtual machines (VMs) to run on a single physical host [5]. This operation is enabled and managed by a hypervisor, a key component of the virtualization stack. Despite playing an important role in resource management and VM isolation, hypervisors are vulnerable to security threats from both guest VMs and the host environment [5]. As a result, the design and implementation of secure, lightweight hypervisors are critical to protecting VMs and ensuring system security [5].

The ARM TrustZone is a hardware-based security extension for ARM processors [1], and Hafnium is a lightweight security isolation layer for virtual machines on ARM platforms [5]. The TrustZone technology creates a Trusted Execution Environment (TEE) by partitioning the system into secure and non-secure worlds, whereas Hafnium isolates VM memory and provides a unique security-focused virtualization solution [1]. Hafnium's minimalist design reduces potential attack surfaces, allowing it to be a lightweight hypervisor focused on memory isolation between VM instances while leaving performance and availability guarantees to the host OS [5].

The nested kernel concept takes a different approach to security by embedding a small, lightweight, and isolated kernel within a larger one [2]. This strategy achieves logic isolation by tracking all changes to the virtual-to-physical mapping and removing sensitive instructions from untrusted components, protecting physical memory and lowering the Trusted Computing Base (TCB) in complex systems [2].

Securing our digital infrastructure remains critical in the era of virtualization and cloud computing. Building upon the strengths of Hafnium, ARM TrustZone, and SecFortress, this paper proposes a new multi-layered security strategy to fortify security at both the VM and hypervisor levels, thereby protecting against both internal and external threats. While promising, the combination of these technologies presents certain challenges, including the complexity of managing the various security mechanisms, potential performance overhead due to multiple security layers, and the necessity for continuous updates to counter newly discovered security vulnerabilities [3, 4].

The successful integration of these security solutions into an organization's infrastructure necessitates careful design, comprehensive implementation strategy, and meticulous planning. Despite the obstacles, such an approach holds the potential to significantly enhance the security posture of virtualized and cloud computing environments, making them more resilient against potential attacks [6, 7].

As we forge ahead, extensive testing, performance optimization, and continuous updates will play pivotal roles in overcoming these challenges. By harnessing the unique advantages provided by Hafnium, ARM TrustZone, and SecFortress, we can lay the groundwork for a more secure, robust, and resilient virtualized environment. The path to achieving this goal is strewn with difficulties, but by working collaboratively, we can hope to stay one step ahead of evolving cybersecurity threats and secure our virtualized infrastructure effectively [7].

III. Related Works

Various approaches have been proposed to secure the run-time of hypervisors. HyperLock and DeHype, for instance, deconstruct KVM by assigning a separate isolated hypervisor instance to each VM, similar to the isolated context each VM has in SecFortress [4]. However, unlike SecFortress, they don't protect the hypervisor against a compromised host OS. SecFortress also differs from systems like MultiHype, which supports running multiple hypervisors on a single physical

platform, in that it ensures a smaller Trusted Computing Base (TCB) and stronger isolation by creating a single hypervisor box for each VM [8].

Nexen, SecVisor, and SeL4, along with SecFortress, utilize the nested kernel to reconstruct the virtualization platform [9]. However, they have their limitations; Nexen doesn't consider the security vulnerabilities in its shared service domain, SecVisor focuses on kernel integrity protection within guest VMs rather than isolation among different VMs, and SeL4, being a formally verified microkernel, doesn't include common OS components [9]. Another memory isolation implementation, Hyper Wall, requires support from specific hardware like FPGA, contrasting with SecFortress's ability to be deployedon commercial x86 platforms [9].

Hardware-based defense technologies have also emerged. Intel TDX, for instance, isolates VMs from the hypervisor by adding a secure arbitration mode [10]. AWS Nitro Enclaves and Arm CCA offer different approaches to creating isolated VM execution environments [11]. However, these systems either focus on protecting the guest from an untrusted hypervisor or applications rather than the entire VM, which differs from SecFortress's target of bidirectional isolation protection between VMs and the hypervisor.

In this paper, we have furthered the research into secure HPC OS/Rs by presenting preliminary results of our approach and an initial proof-of-concept implementation [12]. We have identified several potential research directions, such as evaluating our approach on more realistic systems and workloads, designing I/O mechanisms that maintain secure system isolation without imposing significant performance overheads, and investigating dynamic partitioning approaches for secure partitions and VM images [12].

Also, while we have used the Hafnium hypervisor as a starting point for secure virtualization in HPC, we are still evaluating its long-term suitability [12]. The necessary modifications to support HPC workloads, the need for a potential new hypervisor architecture tailored to HPC environments, and the upcoming ARM platform (ARMv9) which introduces significant security, isolation, and trusted computing features, are all factors that could impact the direction of future researchin this area [11, 12].

IV. Proposed Model

A. Design

The TrustZone-Assisted SecFortress solution is a pain stinkingly designed architecture aimed to improve security in a virtualized environment. The innovative design combines TrustZone technology's robust isolation capabilities with the SecFortress hypervisor's flexible and comprehensive security services. As a result, hypervisors and their associated virtual machines benefit from a powerful combination of hardware and software-enforced security mechanisms.

The secure boot mechanism is at the heart of our design. We ensure a trustworthy startup process by utilizing Trust- Zone technology, which allows only authenticated software to launch. This significantly reduces potential threats from unauthorized or malicious software, allowing the system to operate in a trusted state from the start.

Our solution embeds a trust anchor within TrustZone's secure world to establish a root of trust within the system. This provision has important implications for improving hypervisor security and supporting other security functions such as cryptographic key management and secure storage. It ensures a higher level of trust in the system, essentially laying the groundwork for all subsequent security protocols to operate on.

Our design addresses the difficult challenge of securely handling interrupts and I/O operations. The SecFortress solution includes a mediator component that acts as a go-between for the hypervisor and the rest of the system. The mediator significantly reduces potential vulnerabilities that could be exploited during these operations by managing and securely handling interrupts and I/O operations.

Inter-VM communication can be a security risk if not properly managed. Our design ensures secure communication between VMs via the SecFortress solution's mediator. This intermediary validates each communication request to ensure it comes from a reliable source before it reaches its intended destination. This feature prevents unauthorized access and potential data leaks, thereby strengthening the system's overall security.

To provide a secure environment for sensitive processes and applications, our solution design makes use of Trust- Zone's hardware-based isolation capabilities to create a secure execution environment within the hypervisor. This strategic inclusion effectively insulates these processes from potential threats in the 'normal' world, thereby protecting the integrity of our secure world.

The comprehensive isolation of hardware and software components is a critical aspect of our design. This is accomplished through TrustZone's hardware-level isolation, which creates two distinct environments: one for the hypervisor and one for the VMs and outerOS. This hardware isolation is supplemented by the SecFortress's mediator's software-level isolation. As a result, our design fortifies the hypervisor's shell, effectively isolating it from the rest of the system and potentialattack vectors.

Our design also prioritizes system integrity and resilience in the face of a variety of potential attacks. TrustZone technology protects the integrity of the hypervisor by preventing unauthorized changes to its code. SecFortress' security services, which validate the integrity of data and communication channels within the system, add to this. In terms of resilience, the secure boot feature ensures that the system starts up in a trusted state, while the isolation provided by TrustZone and the mediator makes the system difficult to compromise, increasingits resistance to potential threats.

The way we design prioritizes secure communication and data handling. SecFortress' mediator is at the heart of all communication between the VMs, the outerOS, and the hypervisor, validating the origin and destination of each communication request. TrustZone technology protects data integrity and confidentiality by isolating sensitive data within a

secure world. This two-pronged approach significantly reduces the risk of data exposure or tampering from malicious processes.

The TrustZone-Assisted SecFortress solution design promotes adaptability. The solution is designed to be compatible with a wide range of hardware platforms and hypervisors with minimal modifications, allowing it to be widely deployed across a wide range of systems. The design's scalability, which is further enhanced by its modularity, enables it to protect systems of various sizes, from individual servers to large data centers, without compromising security.

The TrustZone-Assisted SecFortress solution is also optimized for efficiency and performance. With the SecFortress solution's mediator minimizing overhead in handling communication between the VMs, the outerOS, and the hypervisor, and TrustZone ensuring efficient use of hardware resources, optimal system performance is guaranteed.

The solution design incorporates built-in fail-safe mechanisms to ensure that the system remains secure even if a component fails. For example, if the mediator component detects an error, it activates built-in fail-safe routines to prevent a system-wide failure. Similarly, TrustZone technology ensures that any breaches in the normal world do not impact the secure world, adding an extra layer of security.

The solution is designed to work seamlessly with a wide range of systems, taking into account both modern and older systems that may not have built-in security measures. It is compatible with various system architectures and hypervisor types, giving it versatility in securing various systems. The design also includes recovery and maintenance measures capable of detecting potential security threats, initiating protective actions, and allowing for easy system updates and patches. This not only ensures long-term security but also improvesthe system's overall security capabilities.

The TrustZone-Assisted SecFortress solution is designed for the future, combining scalability, robustness, compatibility, efficiency, and fail-safe mechanisms. Its comprehensive and adaptable design supports a variety of system architectures and hypervisors, as well as a wide range of virtual machines and can handle increasing load and traffic. Because of its adaptability, it is an effective security solution for complex virtualized environments. Furthermore, the solution reduces costs while maintaining security by leveraging TrustZone's existing hardware security features and SecFortress's minimalistic design. Also, our design considers future developments in the cybersecurity landscape. It is easily upgradable to handle new types of security threats or to incorporate advancements in hypervisor and virtualization technology. This foresight distinguishes our solution, making it a comprehensive, robust, and scalable option for securing hypervisor environments today and in the future.

The TrustZone-Assisted SecFortress solution is divided into several layers or levels, as illustrated in Fig. 3, and are as follows:

Level 1 - Hardware Layer: This includes the actual hardware platform. At this level, the TrustZone technology creates two distinct environments: the secure world' and the 'normal world'. The most sensitive processes and data reside in the secure world.

Level 2 - TrustZone Technology: TrustZone technology primarily operates at this level, providing fundamental hardware- based isolation capabilities and establishing a root of trust within the system. It also manages the secure boot mechanism, which ensures that only authenticated software can run.

Level 3 - Mediator Component: At this level, the mediator component operates, providing a layer of software-based isolation on top of TrustZone's hardware-based isolation. It also handles interrupts, I/O operations, and inter-VM communications securely, acting as a liaison between the hypervisor and the rest of the system.

Level 4 - Hypervisor: At this level, the SecFortress hypervisor operates. It communicates directly with TrustZone technology to take advantage of hardware-based isolation capabilities for enhanced security. It is also the location of the secure execution environment for sensitive processes and applications.

Level 5 - Virtual Machines (VMs) and outer OS/Primary VM using Kitten lightweight kernel: At this level, virtual machines and the outer / Primary VM operating system operate. They communicate with the hypervisor and the mediator, enabling secure communication and operations.

These components interact in a variety of ways to form a comprehensive, secure, and adaptable system. TrustZone technology, for example, serves as the foundation for system security by creating a secure execution environment and managing the secure boot process. The SecFortress hypervisor, in turn, relies on this secure environment to run VMs and manage their communication via the mediator component.

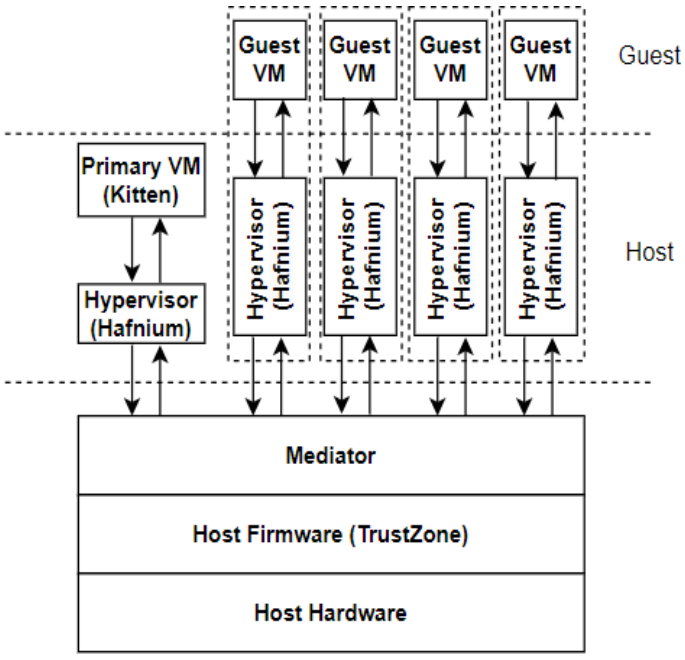

Fig. 3. Proposed model trustzone-assisted secfortress solution architecture.

B. Implementation

The implementation of the TrustZone-Assisted SecFortress solution begins with configuring the ARM-based System-on-a-Chip (SoC) to use TrustZone technology. This process begins with the system boot-up, during which the Secure World is configured before any other system components are initialized.

To establish a clear demarcation between the Secure and Non- Secure Worlds, the memory layout and IRQ controllers must be carefully adjusted. TrustZone's Monitor Mode is set up as an extra execution level within the Secure World to host the mediator software, which manages secure transitions between the two worlds.

After properly configuring the Secure and Non-Secure Worlds, we proceed to integrate the Hafnium hypervisor into the system's Non-Secure World. The isolation capabilities of Hafnium are critical to the system's security. It is intended to create isolated partitions for each virtual machine (VM), effectively isolating them from one another and from the hypervisor itself. The configuration of Hafnium is meticulously adjusted during this phase to manage VMs, enforce memory access policies, and control inter-VM communication. Fig. 4 depicts the memory access view of each SecFortress component. For example, the left-most large box represents the VM's memory access view. That is, each VM has complete access to its own memory but no access to the memory of others. The mediator has access to all memory and controls the page tables to determine the memory view of each component. Illegal memory accesses across components will result in pagefaults, which will be detected by the mediator.

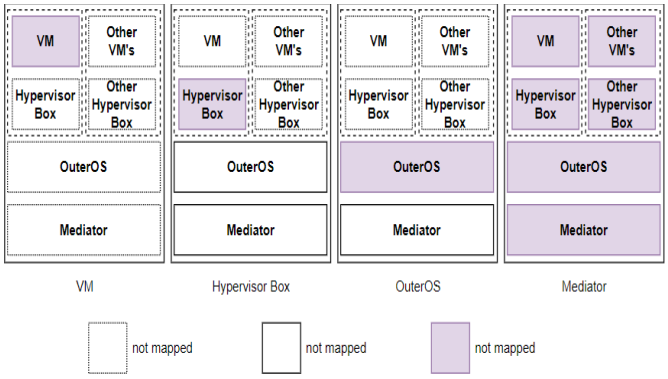

Fig. 4. Memory access view of VM, Hypervisor Box, outerOS, and mediator.

We integrate the Kitten lightweight kernel into the Hafnium hypervisor after it has been established. The Kitten kernel is the foundation of our hypervisor layer. It is designed to be extremely efficient in high-performance computing environments. The integration of the kernel entails prioritizing the optimization of memory management, task scheduling, and I/O operations, which improves overall system performance.

We begin development of the mediator software after successfully integrating the Hafnium hypervisor and Kitten kernel. The use of programming languages compatible with the ARM architecture and the TrustZone environment is required for this task. Memory protection, instruction protection, and control flow management are among the critical security services enforced by the mediator software. It is also intended to intercept and manage VM exits, ensuring safe context switchesbetween system components.

The addition of VM image integrity checks and encryption measures improves system security even further. Before each startup, VM image integrity checks are performed to ensure that the VM images have not been tampered with. Furthermore, the VM images are encrypted with strong encryption algorithms, ensuring the data stored within the VMs' security. The cloud provider's key management service secures the encryption keys.

We chose to implement software updates for the TrustZone- Assisted SecFortress virtualization layer via offline install packages in order to maintain system security. This method gives us more control over the software versions that are running on the system. System administrators install these updates manually, lowering the risk of online threats.

We employ several strategies to reduce performance overhead, ensuring that the system remains operational and does not impair overall performance. The use of a nested MMU extension to create memory protection domains, efficient memory management with the mediator's internal allocator, hard coding of sensitive instruction entry points during system bootup, and the minimization of memory mapping updates within the hypervisor box are among these strategies [13].

Implementing the TrustZone-Assisted SecFortress solution is a meticulous process that includes system initialization, hypervisor setup, kernel integration, mediator software development, VM security measures, software update procedures, and performance overhead reduction strategies. The end result of this comprehensive implementation process is a robust and secure virtualization environment suitable for secure cloudcomputing applications [13, 14].

V. Results and Evaluation

This section describes the TrustZone-Assisted SecFortress solution's results thorough evaluation and security analysis, including protection against mediator tampering, the level of isolation and confidentiality, denial-of-service (DoS) mitigation, performance assessment, and the results of practical attack scenarios.

A. Security Analysis

The TrustZone-Assisted SecFortress solution's comprehensive security analysis begins with protection against mediator tampering. Because the mediator plays such an important role in the system, its integrity is critical. As a result, its security is ensured by secure boot and code integrity checks during system boot, which is a process that protects the mediator's code by write-protecting it, preventing attackers from tampering with it. This writes protection extends to dynamic checks within non-TCB components, protecting sensitive instructionsand code from bypass attacks.

Effective isolation and confidentiality are critical security objectives of the TrustZone-Assisted SecFortress solution, which it achieves by isolating hypervisor boxes from the

outerOS and other hypervisor boxes. Memory access control and paging mechanisms are used to restrict access to sensitive memory regions. OuterOS is prevented from accessing the memory of the hypervisor boxes by unmapping hypervisor box-related memory regions in the kernel master page table. Additionally, zeroing physical pages before assigning them to hypervisor boxes strengthens the integrity defense against attacks.

In terms of Denial-of-Service (DoS) attacks, the TrustZone-Assisted SecFortress solution defends itself by meticulously checking the memory and registers of guest VM states before returning control to them. While this comprehensive measure reduces the possibility of crashes or interference with other VMs by addressing any state mishandling, it does not completely prevent DoS attacks from originating from the host.

B. Performance Evaluation

A performance evaluation was carried out to determine the overhead and efficiency of the combined solution. The solution was tested using various performance benchmarks on Pine A64-LTS SBC and Ubuntu 18.04.5 with Linux 5.2 for Intel VT platforms. The benchmarks focused primarily on CPU and memory performance, with consistent results demonstrating minimal overhead for virtualization and secure isolation. Even when lightweight kernels and ARM TrustZone-based mechanisms were used, there was no significant performance degradation.

A series of tests, including Stream and Random-Access micro-benchmarks, were used to assess memory and I/O performance. These tests revealed that Hafnium and ARM TrustZone introduced minor overhead in some scenarios, but overall performance was satisfactory.

The combined solution was then put through its paces with application performance benchmarks like the HPCG mini-app and a subset of the NAS Parallel Benchmark suite. These tests demonstrated that the solution was capable of running a full mini-app benchmark with minimal overhead and of providing secure isolation for HPC applications and workloads.

C. Practical Attack Evaluation

The TrustZone-Assisted SecFortress solution was subjected to realistic attack scenarios in order to validate the effectiveness of the security measures. CVE analysis for Linux/KVM vulnerabilities was included in these scenarios. The evaluation revealed that the security mechanisms of the solution were effective in preventing privilege escalation, information leakage, memory corruption, and denial-of-service attacks from compromised outerOS or malicious VMs. In essence, the combined solution's isolation was critical in mitigating the impact of compromised components and safeguarding sensitive data andmemory regions.

Finally, the TrustZone-Assisted SecFortress solution combines lightweight kernels, Trusted Execution Environments (TEEs), the Hafnium hypervisor, and ARM TrustZone to achieve robust security isolation in virtualization environments. It effectively addresses both security and performance concerns, thereby assisting in the development of more secure and efficient virtualized systems. Comprehensive security and performance evaluations confirm its resistance to common attacks and ability to handle high-performance computing workloads with minimal overhead. As virtualization technology evolves, it promises to be a solid foundation for secureand high-performance virtualization environments.

VI. Conclusion

In this paper, we presented an enhanced to secure hypervisor runtime and a new use case for Lightweight Kernels as resource management services in securely isolated HPC systems. As part of this effort, we integrated the ARM64-ported Kitten LWK with the Hafnium hypervisor in our SecFortress solution to support secure virtual machine instances on a compute node. SecFortress partitions the virtualization platform strategically into a trusted mediator, an isolated outerOS, and multiple restricted hypervisor box instances, improving security isolation in high-performance computing platforms. The new approach prevents the outerOS from accessing the hypervisor's memory, with each hypervisor box instance limited to the least amount of memory access. As a result, even if one instance is compromised, the integrity and confidentiality of other instances are not jeopardized. We have provided an initial proof of concept implementation and preliminary evaluation, which show that our approach has no significant performance overheads on a variety of HPC benchmarks. SecFortress' experimental results show that it can defeat exploits against the host OS and VMs with negligible performance overhead, implying that SecFortress could be an effective solution for improving both virtual machine security and performance. This work has also identified a number of future challenges and made a compelling case for security isolation and trusted computing as key features of next-generation HPC platforms. Fully supporting security isolation in a scalable and performant manner will most likely pose a significant challenge for HPC OS/R architectures, necessitating future research. Our integrated solution, we believe, provides a promising foundation for the evolution of more secure and efficient virtualized systems.

References

[1] "Arm trustzone technology." https://developer.arm.com/technologies/trustzone, accessed August 2023.

[2] 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.*

[3] 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.

[4] 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.

[5] "Hafnium hypervisor." https://www.trustedfirmware.org/projects/hafnium, Accessed August 2023.

[6] C. Meyers, "The biggest cloud breaches of 2019 and how to avoid them for 2020." Lacework Editorial, December 2019.

[7] M. Gontovnikas, "The 11 biggest data breaches of 2020." Auth0 Blog, 2020.

[8] 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.

[9] 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.

[10] Intel, "Intel trust domain extensions." https://software.intel.com, 2020.

[11] Amazon, "AWS Nitro Enclaves." https://docs.aws.amazon.com/enclaves/, 2020.

[12] 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.

[13] 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.

[14] 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.