Kata Container: Influence of Security onto Network Performance

George Jin, Florian Wiedner*
*Chair of Network Architectures and Services
School of Computation, Information and Technology, Technical University of Munich, Germany
Email: george.jin@tum.de, wiedner@net.in.tum.de

***Abstract*—The introduction of technologies like 5G has increased the demand for high-performance, scalable networks. Network Function Virtualization (NFV) has emerged as a solution by virtualizing network functions on standard servers to meet these standards. However, it relies on robust security measures which simultaneously provide high network performance. This paper discusses whether Kata Containers integrated with Trusted Execution Environments (TEEs) is a viable solution for securing NFV environments. We dive into a comprehensive background of these technologies, analyze their security benefits and drawbacks, and their impact on network performance in NFV. We found that although Kata Containers and TEEs provide enhanced security through virtualization and hardware-based protection, they introduce performance overheads, especially regarding network latency and scalability. We conclude that further research is necessary to analyze the concrete security and performance implications of Kata Containers integrated with TEEs in NFV environments. This includes research in performance optimization techniques and strategies for determining the right balance between security and performance in NFV environments.**

***Index Terms*—Kata Containers, TEE, NFV**

1. Introduction

In recent years, the emergence of technologies like 5G and its use cases like IoT and edge computing have become increasingly relevant. With the progress they bring, they have introduced new requirements and challenges, particularly the need for highly scalable networks capable of delivering high network throughput and low latency while simultaneously dealing with a large number of devices in real-time [1], [2]. Consequently, NFV has become a key technology to meet these standards by virtualizing network functions on standard servers instead of relying on specialized hardware [3].

NFV, in turn, requires robust virtualization and containerization solutions to provide both high performance and high security [3]. Kata Containers has emerged as a promising approach to meet these requirements by combining the lightweight performance of traditional containers with the enhanced security and isolation characteristics of virtual machines (VMs) [4]. This makes them a potential solution for securing NFV environments. To further enhance the isolation of these containers, TEEs can be integrated to offer an additional layer of hardware-based security [5]. Together, they can provide an effective method for enhancing the security of virtualized network functions (VNFs), though their full impact requires further analysis.

However, the increased security and isolation offered by Kata Containers and TEEs come at the cost of performance overhead [6]. In environments where high network performance is critical, balancing the trade-off between enhanced security and optimal performance remains a major challenge. Therefore, understanding these challenges is crucial for evaluating the viability of Kata Containers and TEEs in NFV environments.

This paper provides an overview of the current research on Kata Containers and TEEs in NFV environments, investigating the security benefits and network performance drawbacks. The goal is to determine whether Kata Containers, with or without TEEs, present a viable solution for securing NFV environments while maintaining the necessary network performance requirements. To achieve this, Section 2 will provide a comprehensive background on Kata Containers, TEE, and NFV. In Section 3, we will explore the security benefits and drawbacks of Kata Containers and TEEs. This will be followed by Section 4, which discusses their network performance implications. And finally, we will compare the security implications with the performance drawbacks in Section 5.

2. Background

We will explore three key technologies important to our discussion: Kata Containers, TEEs, and NFV. Kata Containers aims to improve security by utilizing lightweight VMs. TEEs use a separated area on the CPU to provide hardware-level protection. NFV is a concept that moves network functions from specialized hardware to standard servers. A solid understanding of these technologies is essential for our evaluation.

2.1. Kata Containers

Traditional container runtimes like RunC achieve performance close to native levels [7]. However, by only relying on Linux namespaces and control groups (cgroups) while sharing the same host kernel, they offer little to no isolation, making them vulnerable to attacks. On the other hand, VMs offer strong isolation with their isolated kernel but introduce a significant performance overhead [8]. To bridge this gap, Kata Containers, an open-source container runtime introduced in 2017, aims to combine the high performance of lightweight containers with the security and isolation of VMs. This approach is the result of merging

 doi: 10.2313/NET-2025-05-1_05

Intel Clear Containers and Hyper.sh runV, technologies that run each container in its own optimized VM [4].

The main idea is to run each container within its own lightweight VM, highly optimized to minimize performance overhead and resource consumption using technologies like Guest Kernel Minimal and Guest Image [4] while providing kernel-level isolation. Additionally, it uses a specialized QEMU version named qemu-lite as the default hypervisor, which improves boot time and reduces memory footprint with features like Machine Accelerators, Kernel same-page merging, Hot Plug Devices, and Fast Template [4]. The architecture of Kata Containers consists of three main components: the Kata-runtime, Kata-agent, and Kata-shim. The Kata-runtime on the host creates the VM for running the container. The Kata-agent process, running in the guest kernel in the VM, sets up the environment and runs the container and processes within the container while receiving instructions from the host via gRPC [9]. The Kata-shim is a process on the host that is responsible for all container I/O streams [4]. An additional advantage of Kata Containers is their OCI-compliance, making them seamlessly integrate with containerization and container orchestration platforms like Docker, Kubernetes, and OpenStack by simply replacing runC with Kata-runtime, ensuring easy deployment for organizations and removing the need for major modifications of existing workflows [4].

2.2. Trusted Execution Environment

A TEE is a secure area within the processor that allows the safe execution of code and storage of data. It is segregated from the rest of the CPU, protecting its data from unauthorized access or tampering by code outside of that environment [10]. TEEs establish a chain of trust during the boot process, which guarantees the authenticity of the running software, the integrity of the runtime states, and the confidentiality of the code and data. TEEs also support remote attestation, allowing third parties to verify the integrity and trustworthiness of the TEE [11]. Prominent examples are Intel SGX and ARM TrustZone, each with its own approach to security. Intel SGX isolates data and code in areas called enclaves, while ARM TrustZone separates the whole processor into a secure and normal world [12]. In general, the core features of a TEE include strong hardware-based isolation, efficient scheduling and secure communication between secure and rich environment, and secure updates [5].

2.3. Network Function Virtualization

In traditional networks, network functions like firewalls, load balancers, and routers rely on specialized hardware. In times of fast-paced digital innovations and declining lifecycles of hardware, the frequent replacement and scaling of these systems pose a serious challenge for network service providers, particularly in fields like 5G [3]. NFV aims to solve this problem by virtualizing these network functions and running them on commercial off-the-shelf servers instead of specialized equipment, resulting in several advantages [3]. Firstly, it increases scalability and flexibility by enabling providers to scale the number of VNFs up or down depending on demand. Secondly, it offers better operating performance by dynamically allocating resources based on a given network load. Thirdly, it leads to shorter development cycles by replacing the necessity of installing new physical devices with the deployment of software updates. These benefits significantly reduce both capital expenditures and operational expenditures, making NFV a highly flexible and cost-effective solution for modern networks [3].

The NFV architecture consists of three components. A physical server provides computing and storage resources, a hypervisor that manages the virtual environment, and a virtualized environment for executing VNFs [3]. Even though NFV brings several advantages, the virtualization of network functions is expected to increase the potential for security attacks [3], requiring more sophisticated security measures. Firstly, it requires a protected hypervisor to prevent unauthorized access or data leakage [3]. Secondly, data communication and VM migration need a secure environment [13]. Thirdly, NFVs use application programming interfaces (APIs), which pose another security threat [14]. At the same time, the network performance must be comparable to traditional networks despite the additional virtualization layer. Therefore, a well-balanced trade-off between security and performance is necessary.

3. Kata Containers and Security

One of the main vulnerabilities of traditional containers is their dependence on the shared host kernel. If one container gets compromised, the attacker could potentially get access to the host and other containers. While the Linux kernel uses cgroups to isolate and limit the usage of physical resources for each container, it is shown that out-of-band workloads can break the cgroups' confinement [15], potentially making the whole system vulnerable to resource exhaustion attacks, such as Denial-of-Service. Next to the weak resource confinement, the shared kernel also poses the risk of privacy leakage through pseudo-filesystems, enabling attackers to gather sensitive information about the environment for further exploitation [16], [17]. In addition to the kernel layer, the container layer faces vulnerabilities like improper handling of symbolic links or insecure API handling, making container escapes possible [18]. In fact, 56.82% of vulnerability exploits could launch successfully from within a container [19], indicating the need for more advanced security solutions.

Kata Containers makes container escapes more difficult with its additional layer of security offered by lightweight VMs and kernel-level isolation, but research shows that escapes are still possible [18]. In particular, according to research, Kata Containers have three key vulnerabilities. First, Kata Containers did not properly enforce device cgroups, allowing attackers to access the /dev files on the VM inside the container, leading to CVE-2020-2023 [20]. The attacker could then overwrite the kata-agent, leading to container escapes and further compromises. The Kata Containers reuse the corrupted kata-agent, leading to CVE-2020-2025 [21]. Lastly, kata-runtime does not validate mount points in shared folders. That means it resolves any symbolic link and conducts the mount operation, leading to CVE-2020-2026 [22], which would allow the attacker to mount the root file

system to any part of the host system, thereby breaking the virtualized container despite the hardware virtualization in use [18]. To mitigate these vulnerabilities, Kata Containers can be integrated with TEEs. The isolation of the lightweight VM in Kata Containers, in combination with hardware-based protection by TEEs, can significantly increase the overall security of the system, offering protection not only from container escapes but also from malicious code from the host system itself. Additionally, remote attestation ensures that only verified code runs within the TEE, further reducing the attack surface [11]. However, the use of TEEs comes with new challenges as well. Even though TEEs provide secure hardware-based security, they are not immune to side-channel attacks, which make use of indirect system information like memory access, CPU load, or power consumption to infer information about other aspects of the system for further exploitation [11]. Nevertheless, while there is limited research in this area, combining Kata Containers and TEEs could be particularly useful for securing VNFs, making them less vulnerable compared to traditional network functions.

4. Network Performance Impact of Kata Containers

The shift from traditional network functions to an NFV environment introduces performance challenges, especially when implemented with Kata Containers. The critical performance metrics in NFV are network latency and network throughput. Latency refers to the time needed for data to travel from one point to another, while throughput refers to the amount of data able to be transmitted within a specified time window. Research has shown that the additional virtualization layer of Kata Containers notably reduces the network performance in certain cases compared to traditional containers like runC, introducing potential bottlenecks [6], [7], [9]. In particular, Kata Containers only shows slightly lower throughput compared to runC, less than 1% [6]. In more specific cases regarding TCP and HTTP throughput, it showed 1-18% lower throughput depending on the scenario [6], [7]. Similarly, Kata Containers only scored 20% of the throughput of runC in simple GET operations from in-memory data [8]. Similarly, latency is also affected, showing a performance loss of up to 35% compared to RunC [6], [9]. Despite these shortcomings, Kata Containers performs better than gVisor in both network throughput and network latency [6], [7]. These findings are further illustrated in Figure 1, adapted from [6], which compares the network performance of different container runtimes with an emphasis on latency. In particular, TCP_RR and UDP_RR are request-response tests, while TCP_CRR additionally includes the connection time and teardown time. runC and bare metal show the best performance, while gVisor shows the highest overhead, with Kata Containers performing in between.

Furthermore, in [7], the authors compared the scalability benchmarks of runC and gVisor across different container counts. The results of these scalability benchmarks are summarized in Figure 2, adapted from [7], which shows that while runC scales efficiently, gVisor suffers from significant network performance overhead with an increasing number of containers [7]. However, we observe that containerd/runsc does not show the same scaling issues as crio/runsc, which could be the result of specific optimizations in the containerd runtime. Nevertheless, gVisor relies on an additional isolation layer with its own user-space kernel. It can be expected that Kata Containers show similar behavior due to its additional VM layer. This means that as the number of containers grows, the network performance of Kata Containers could drop significantly. This scaling limitation could pose a serious challenge in NFV environments, where a large number of VNFs run simultaneously while requiring high network throughput and latency.

A way for addressing the performance drop of Kata Containers could be technologies like Single Root I/O Virtualization (SR-IOV), which allows direct access to network hardware, bypassing the virtualization layer [23]. Even though SR-IOV gives containers direct access to hardware, it does not directly violate the principles of NFV, as it allows the hardware to be shared between multiple VNFs [23]. Research shows that it can drastically reduce the performance overhead of the virtualization layer, almost achieving native network performance [23]. Consequently, SR-IOV could be a key solution to solve the network performance overhead and the scaling limitations of Kata Containers. Next to SR-IOV, there are software-based performance optimization techniques. One of them is the use of the kernel driver vhost-net, which offloads network packet handling from the VM to the host. This reduces context switches between guest and host, leading to higher throughput and lower latency while also allowing a higher number of VMs to run on the same host. This means more VNFs can run in parallel without performance degradation, improving scalability [24]. Another technique is a set of optimized libraries and drivers called Data Plane Development Kit (DPDK), which enables the network packets to bypass the host kernel entirely. By allowing distributed processing of network packets across multiple CPU cores, DPDK further increases network performance [25]. In terms of performance optimizations for TEEs, current improvements in ARM TrustZone include

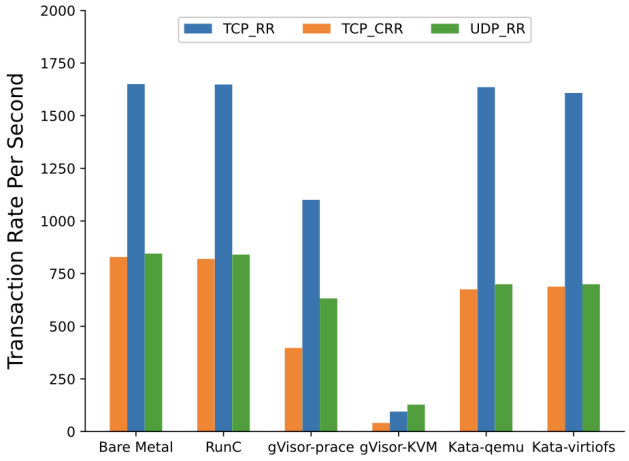
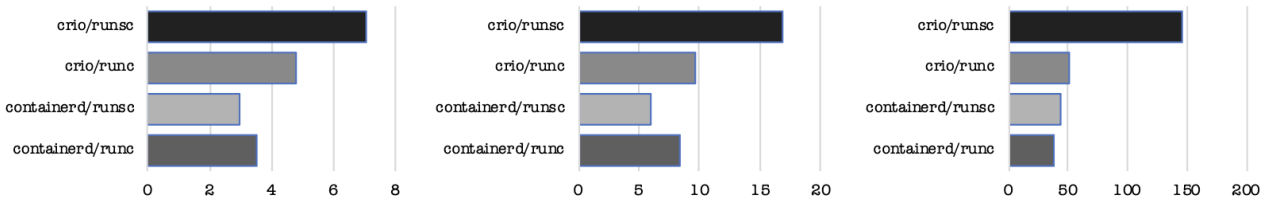


Figure 1: TCP_RR, TCP_CRR, UDP_RR transaction rates per second for different container runtimes. Adapted from [6].



(a) 5 Containers. (b) 10 Containers. (c) 50 Containers.

Figure 2: Benchmark comparing the network performance (in seconds) of RunC-based runtimes (crio/runc, containerd/runc) and gVisor-based runtimes (crio/runsc, containerd/runsc) across 5, 10, and 50 containers, based on 10 runs. Adapted from [7].

the simultaneous access to I/O devices for both secure and rich environment. This reduces context switching and increases the performance and scalability of I/O-heavy tasks like network packet processing in NFV [26]. However, Kata Containers could still be limited in their use in large-scale NFV environments because of their scalability limitations, and even optimization techniques like SR-IOV and vhost-net, which mitigate this limitation, could be insufficient to fully address this issue.

Additionally, when integrating Kata Containers with TEEs for additional security, there are also trade-offs in terms of network performance. One of the key issues is the frequent context switches between the TEE and the rich environment, especially in network-intensive tasks where large amounts of data are transported. These can lead to significant performance overheads and thus challenge the combined use of Kata Containers and TEEs in NFV environments. A solution is to minimize the performance overhead by limiting the TEE usage to critical parts of the network, thereby reducing the performance overhead while maintaining security [27]. Because of the additional overhead introduced by Kata Containers and TEEs, their usage in performance-critical applications in 5G networks could be limited.

5. Kata Containers and NFV

NFV environments allow the virtualization of network functions, leading to significant improvements in scalability, flexibility, and cost reductions in comparison to traditional networks. However, this shift comes at the cost of higher vulnerability to attacks. To mitigate these security risks, Kata Containers could offer a promising solution. In these environments, potentially multiple VNFs run on the same hardware, consequently requiring a strong level of isolation and security. While traditional containers like runC show high network performance, they are flawed regarding security and isolation, making them vulnerable to critical attacks like container escapes which could compromise the whole system. By using Kata Containers as an alternative, we could significantly reduce the risk of container escapes. This is the result of encapsulating each container in its own lightweight VM, offering kernel-level isolation. Running these Kata Containers in TEEs provides the benefit of an additional layer of hardware-based protection, even making attacks from a compromised host difficult. Nevertheless, TEEs are vulnerable to side-channel attacks, making use of secondary information for further compromise.

When it comes to network performance, VNFs require high throughput and low latency. However, securing NFV environments using Kata Containers comes at the cost of performance loss caused by the additional virtualization layer, making it a potential bottleneck for high-traffic scenarios. More specifically, while Kata Containers show comparable throughput to runC in the majority of cases, the network latency seems highly affected by the additional virtualization layer. Moreover, NFV requires easy scaling, but we have shown that Kata Containers are potentially limited in that regard, causing increasing performance overhead with a growing number of containers. This makes adopting Kata Containers difficult when handling high numbers of VNFs. Proposed solutions would be SR-IOV, vhost-net, or DPDK, which would enable Kata Containers to maintain high performance while retaining the security gains. However, with the use of SR-IOV or vhost-net, direct access to hardware could undermine the benefits of the additional virtualization layer and should only be considered when the need for performance outweighs the risk of reduced security and isolation. Additionally, Kata Containers could require more hardware resources than traditional containers due to the virtualization layer, increasing operational costs in terms of computing power and memory. Similarly, TEEs like Intel SGX and ARM TrustZone require specialized processors, potentially driving up capital costs. The decision whether to use Kata Containers and TEEs to secure NFV depends heavily on the specific use case. In areas where security is paramount, it could be a strong solution. On the contrary, in situations where performance is paramount, Kata Containers could pose a limitation with its performance overhead caused by the hardware virtualization and possible integration of TEEs, which further reduces the performance through the need for frequent context switches.

6. Conclusion and Future Work

This paper aims to provide a broad overview of the current research on Kata Containers, TEEs, and their security and performance implications on NFV environments. While Kata Containers make use of lightweight virtual machines to improve their isolation compared to traditional containers, they are still susceptible to vulnerabilities. These can be mitigated with the integration

of TEEs, which add an additional layer of hardware-based security, thereby creating a robust combination that can significantly reduce attacks in NFV environments. However, this enhanced security comes at the cost of overheads in network performance, particularly in network latency and scalability. Since VNFs rely on low latency, high throughput, and high scalability, these additional security layers could become a bottleneck in performance-critical scenarios. Therefore, further research is necessary to examine the exact security benefits and performance drawbacks in real-world deployments of Kata Containers integrated with TEEs in NFV environments. This includes further research in optimization techniques like SR-IOV for increased performance while maintaining security, as well as potential improvements, like the reduction of the frequency or cost of encryption cycles in Intel SGX [12]. Based on these results, future work should aim to determine the right balance of security and performance for secure NFV.

References

[1] N. Hassan, K.-L. A. Yau, and C. Wu, "Edge computing in 5g: A review," *IEEE Access*, vol. 7, pp. 127 276–127 289, 2019.

[2] I. Alawe, Y. Hadjadj-Aoul, A. Ksentini, P. Bertin, and D. Darche, "On the scalability of 5g core network: The amf case," in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2018, pp. 1–6.

[3] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "Nfv: State of the art, challenges and implementation in next generation mobile networks (vepc)," *IEEE network*, vol. 28, no. 6, pp. 18–26, 2014.

[4] A. Randazzo and I. Tinnirello, "Kata containers: An emerging architecture for enabling mec services in fast and secure way," in *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*. IEEE, 2019, pp. 209–214.

[5] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: What it is, and what it is not," in *2015 IEEE Trustcom/BigDataSE/Ispa*, vol. 1. IEEE, 2015, pp. 57–64.

[6] X. Wang, J. Du, and H. Liu, "Performance and isolation analysis of runc, gvisor and kata containers runtimes," *Cluster Computing*, vol. 25, no. 2, pp. 1497–1513, 2022.

[7] L. Espe, A. Jindal, V. Podolskiy, and M. Gerndt, "Performance evaluation of container runtimes," in *Proceedings of the 10th International Conference on Cloud Computing and Services Science (CLOSER)*. IEEE, 2020, pp. 273–281.

[8] W. Viktorsson, C. Klein, and J. Tordsson, "Security-performance trade-offs of kubernetes container runtimes," in *2020 28th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. IEEE, 2020, pp. 1–4.

[9] H. Z. Cochak, G. P. Koslovski, M. A. Pillon, and C. C. Miers, "runc and kata runtime using docker: A network perspective comparison," in *2021 IEEE Latin-American Conference on Communications (LATINCOM)*. IEEE, 2021, pp. 1–6.

[10] Microsoft, "Trusted execution environment in azure," 2024, accessed: 2024-09-16. [Online]. Available: https://learn.microsoft.com/en-us/azure/confidential-computing/trusted-execution-environment

[11] S. Brenner, "Enhancing cloud security with trusted execution," Doctoral Dissertation, Technische Universität Carolo-Wilhelmina zu Braunschweig, Braunschweig, Germany, December 2020, disputation on 14.12.2020.

[12] N. Buchner, H. Kinkelin, and F. Rezabek, "Survey on trusted execution environments," 2022, accessed: 2024-10-14.

[13] V. Ashktorab, S. R. Taghizadeh *et al.*, "Security threats and countermeasures in cloud computing," *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, vol. 1, no. 2, pp. 234–245, 2012.

[14] C. S. Alliance, "The notorious nine cloud computing top threats in 2013," 2013, accessed: 2024-09-16. [Online]. Available: https://cloudsecurityalliance.org/artifacts/the-notorious-nine-cloud-computing-top-threats-in-2013

[15] X. Gao, Z. Gu, Z. Li, H. Jamjoom, and C. Wang, "Houdini's escape: Breaking the resource rein of linux control groups," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1073–1086.

[16] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *2010 proceedings IEEE infocom*. IEEE, 2010, pp. 1–9.

[17] H. Li, Y. Yang, Y. Dou, J.-M. J. Park, and K. Ren, "Pedss: Privacy enhanced and database-driven dynamic spectrum sharing," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1477–1485.

[18] Y. Yang, W. Shen, B. Ruan, W. Liu, and K. Ren, "Security challenges in the container cloud," in *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. IEEE, 2021, pp. 137–145.

[19] X. Lin, L. Lei, Y. Wang, J. Jing, K. Sun, and Q. Zhou, "A measurement study on linux container security: Attacks and countermeasures," in *Proceedings of the 34th annual computer security applications conference*, 2018, pp. 418–429.

[20] "CVE-2020-2023: Kata Containers device cgroup enforcement issue," https://nvd.nist.gov/vuln/detail/CVE-2020-2023, accessed: 2024-10-14.

[21] "CVE-2020-2025: Kata Containers reuse of corrupted kata-agent," https://nvd.nist.gov/vuln/detail/CVE-2020-2025, accessed: 2024-10-14.

[22] "CVE-2020-2026: Kata-runtime mount point validation failure," https://nvd.nist.gov/vuln/detail/CVE-2020-2026, accessed: 2024-10-14.

[23] Y. Dong, X. Yang, J. Li, G. Liao, K. Tian, and H. Guan, "High performance network virtualization with sr-iov," *Journal of Parallel and Distributed Computing*, vol. 72, no. 11, pp. 1471–1480, 2012.

[24] Red Hat, "Deep dive into virtio networking and vhost-net," https://www.redhat.com/en/blog/deep-dive-virtio-networking-and-vhost-net, 2019, accessed: 2024-10-14.

[25] Trenton Systems, "What is dpdk?" https://www.trentonsystems.com/en-us/resource-hub/blog/what-is-dpdk, 2023, accessed: 2024-10-14.

[26] N. Zhang, B. D. R. Hafshejani, D. Forte, and M. Tehranipoor, "Self-secured devices: Trustzone-based management of shared resources," p. 101934, 2021, accessed: 2024-10-14.

[27] F. Schwarz, "Trustedgateway: Tee-assisted routing and firewall enforcement using arm trustzone," in *Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses*. New York, NY, USA: Association for Computing Machinery, 2022, pp. 56–71. [Online]. Available: https://doi.org/10.1145/3545948.3545961