Poster: RICe: Runtime Integrity for Containers leveraging eBPF and IMA Appraisal

Suh Ho Lee
National Security Research Institute
Daejeon, South Korea
Email: suhho1993@gmail.com

***Abstract*—Runtime Integrity for Containers Leveraging eBPF and IMA Appraisal (RICe) is a runtime integrity enforcement system for containerized applications that combines eBPF with the Linux *Integrity Measurement Architecture* (IMA). Prior work has primarily focused on namespacing the IMA measurement list for remote attestation, whereas IMA signature appraisal remains underexplored in containerized environments. RICe repurposes the IMA signature appraisal as a container-specific, real-time integrity enforcement mechanism. In doing so, it proactively preserves container immutability at runtime and helps bridge the gap between supply-chain trust (e.g., image signatures) and runtime integrity. Furthermore, RICe serves as a practical and immediate alternative to IMA namespacing by providing container-specific integrity enforcement on unmodified commodity kernels.**

1. Introduction

The shift toward containerized microservices has made supply-chain integrity a central security concern. Although deployment-time verification [1] ensures that only signed images are admitted to a cluster, it leaves a significant trust gap during the container lifecycle. Once a container is running, it remains vulnerable to post-deployment tampering through container escapes (e.g., CVE-2019-5736 in RunC [2]), remote code execution (RCE) vulnerabilities (e.g., Log4Shell [3], React2Shell [4]), and malware targeting containerized workloads (e.g., Kinsing [5]). Such attacks can modify or inject binaries directly into the container file system, thereby bypassing static image scanning. Without continuous runtime integrity enforcement, a previously verified container can execute malicious code, undermining the supply-chain trust model.

Industrial tools such as *Tetragon* [6] leverage eBPF and LSM hooks to provide runtime security monitoring and enforcement. However, these systems primarily focus on monitoring and detection, generating alerts when suspicious activity is observed. Even when they provide real-time enforcement, they typically rely on behavioral heuristics or path-based access control, neither of which guarantees the integrity of the executed binaries themselves.

Several prior efforts have attempted to apply IMA to containers due to its robust integrity guarantees. Luo et al. [7] proposed Container-IMA to provide independent integrity measurement of containers for remote attestation. Blanchard et al. [8] developed a namespaced IMA using BPF-LSM to intercept file events and create a container-specific measurement list for remote attestation as well. IMA Namespacing [9] proposes isolating IMA by namespace to support containers, but doing so requires substantial modifications to the Linux kernel.

We introduce ***Runtime Integrity for Containers leveraging eBPF and IMA Appraisal* (RICe)**, a system that repurposes the Linux kernel's IMA signature appraisal mechanism as a container-aware runtime enforcement engine. RICe uses eBPF to observe the internal state of the IMA appraisal process, while a BPF-LSM enforces the execution decision at the `execve()` boundary. Unlike reactive monitoring systems, RICe proactively preserves container immutability by blocking the execution of unauthorized binaries in real time. Moreover, it does not require extensive kernel modifications like IMA namespacing. As a result, RICe provides a practical mechanism for extending supply-chain trust into the runtime phase.

RICe introduces several key technical contributions:

1) **Appraisal-Driven Enforcement:** RICe shifts from passive measurement to active enforcement by leveraging IMA appraisal results through eBPF.
2) **An Alternative to IMA Namespacing:** RICe provides a practical and immediately deployable alternative to IMA namespacing, operating on unmodified commodity kernels.
3) **Mitigating Risks in Supply Chain Security:** By verifying cryptographic signatures at the point of execution, RICe reduces the window of vulnerability inherent in deployment-time verification.

2. System Design and Implementation

RICe is implemented as an eBPF-based system deployed on the container host. To enable integrity enforcement, container image vendors must cryptographically sign executable binaries within container images prior to deployment. The host kernel must support IMA signature appraisal and be provisioned with the corresponding public keys for verification.

The RICe architecture consists of two components: (1) an *observation module* that monitors IMA appraisal outcomes during binary execution by container processes, and (2) an *enforcement module* that dynamically permits or denies execution based on the observed integrity state.

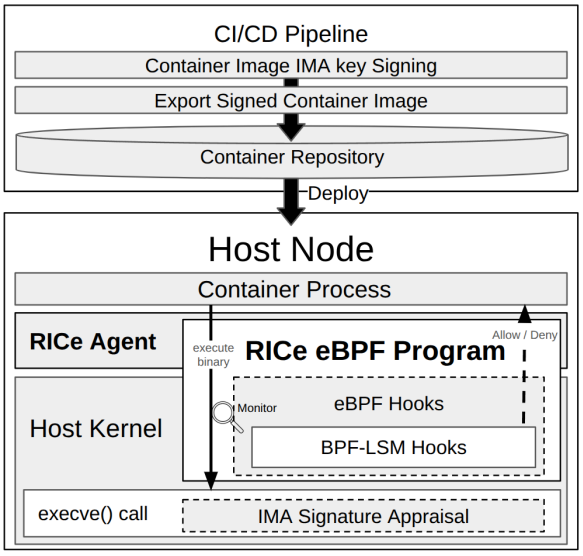

Figure 1. RICe Architecture

RICe requires the host node to be configured with the Linux Integrity Measurement Architecture (IMA). First, IMA key provisioning must be performed: a Certificate Authority (CA) key is registered in the kernel's trusted keyring. This CA signs subordinate IMA signing keys (e.g., container vendor keys), which can be dynamically loaded as needed. Second, a custom IMA policy is installed with the rule `appraise func=BPRM_CHECK mask=MAY_EXEC appraise_type=imasig`, instructing the kernel to perform cryptographic signature appraisal on all executable files. Finally, the system must be booted with the kernel parameter `ima_appraise=log`. This configuration places IMA in a permissive mode, where appraisal results are computed and logged but not enforced by the kernel. This design allows RICe to selectively enforce integrity decisions for container workloads without disrupting host processes.

To obtain the kernel's ground-truth integrity decisions without modifying kernel code, RICe attaches BPF `fexit` probes to internal IMA functions. Specifically, probes on `ima_appraise_measurement` and `ima_inode_get` intercept the `ima_iint_cache` structure immediately after appraisal. The resulting integrity status (e.g., `INTEGRITY_PASS` or `INTEGRITY_FAIL`) is extracted and stored in an eBPF hash map. The inode number is used as the key for this map, binding the integrity state to the file object. This design ensures content-based integrity enforcement, in contrast to prior approaches that rely on path- or behavior-based policies.

Enforcement is executed synchronously during the `execve()` system call by attaching a BPF-LSM hook to `bprm_creds_from_file`. This hook serves as a mandatory access-control checkpoint before binary execution. When triggered, the hook determines whether the process belongs to a container by inspecting its namespace ID. For container processes, the hook queries the eBPF hash map using the binary's inode number. If the integrity status indicates a missing or invalid signature, the hook returns `-EACCES`, causing the kernel to deny execution immediately.

3. Conclusion and Future Work

RICe demonstrates that proactive runtime integrity enforcement is achievable using existing kernel primitives. By combining BPF-LSM with IMA appraisal, RICe provides a practical and deployable mechanism for enforcing container immutability and narrowing the gap between supply-chain trust and runtime integrity.

Despite these results, RICe remains an early-stage system, and several challenges must be addressed to enable broader deployment. One architectural limitation is its incompatibility with UEFI Secure Boot. Secure Boot restricts the use of the `ima_appraise=log` kernel parameter, a configuration on which RICe currently relies to selectively enforce integrity for container workloads.

Future work will explore alternative designs that eliminate this dependency, such as fine-grained IMA policies that enable selective enforcement. In addition, we plan to conduct a comprehensive performance evaluation of RICe to quantify overhead and execution latency under realistic container workloads.

References

[1] Sigstore, "cosign: Container Signing, Verification and Storage in an OCI registry," https://github.com/sigstore/cosign, 2026, accessed: Mar. 20, 2026.

[2] National Vulnerability Database, "CVE-2019-5736 Detail," https://nvd.nist.gov/vuln/detail/CVE-2019-5736, 2019, accessed: Mar. 19, 2026.

[3] ——, "CVE-2021-44228 Detail," https://nvd.nist.gov/vuln/detail/cve-2021-44228, 2021, accessed: Mar. 19, 2026.

[4] ——, "CVE-2025-55182 Detail," https://nvd.nist.gov/vuln/detail/CVE-2025-55182, 2025, accessed: Mar. 19, 2026.

[5] G. Singer, "Threat Alert: Kinsing Malware Attacks Targeting Container Environments," https://www.aquasec.com/blog/threat-alert-kinsing-malware-container-vulnerability/, Apr. 2020, accessed: Mar. 19, 2026.

[6] Cilium Project, "Tetragon: eBPF-based Security Observability and Runtime Enforcement," https://github.com/cilium/tetragon, 2022, accessed: Mar. 19, 2026.

[7] W. Luo, Q. Shen, Y. Xia, and Z. Wu, "Container-IMA: A privacy-preserving integrity measurement architecture for containers," in *22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019)*. Chaoyang District, Beijing: USENIX Association, Sep. 2019, pp. 487–500. [Online]. Available: https://www.usenix.org/conference/raid2019/presentation/luo

[8] A. Blanchard, "container-ima: Container IMA using eBPF," https://github.com/avery-blanchard/container-ima, 2023, accessed: Mar. 19, 2026.

[9] S. Berger, "ima: Namespace IMA with audit support in IMA-ns," https://lwn.net/Articles/900311/, 2022, accessed: Mar. 19, 2026.

RICe: Runtime Integrity for Containers leveraging eBPF and IMA Appraisal

Suhho Lee
Email: suhho1993@gmail.com

47th IEEE Symposium on Security and Privacy

Abstract

Runtime Integrity for Containers Leveraging eBPF and IMA Appraisal (RICe) is a runtime integrity enforcement system for containerized applications that combines eBPF with the Linux Integrity Measurement Architecture (IMA). Prior work has primarily focused on namespacing the IMA measurement list for remote attestation, whereas IMA signature appraisal remains underexplored in containerized environments. RICe repurposes the IMA signature appraisal as a container-specific, real-time integrity enforcement mechanism. In doing so, it proactively preserves container immutability at runtime and helps bridge the gap between supply-chain trust (e.g., image signatures) and runtime integrity. Furthermore, RICe serves as a practical and immediate alternative to IMA namespacing by providing container-specific integrity enforcement on unmodified commodity kernels.

Main Contributions

➢ Appraisal-Driven Enforcement

RICe shifts from passive measurement to active enforcement by leveraging IMA appraisal results through eBPF.

➢ An Alternative to IMA Namespacing

RICe provides a practical and immediately deployable alternative to IMA namespacing, operating on unmodified commodity kernels.

➢ Mitigating Risks in Supply Chain Security

By verifying cryptographic signatures at the point of execution, RICe reduces the window of vulnerability inherent in deployment-time verification.

Motivation

➢ The Supply Chain Trust Gap

- Container image verification is typically limited to deployment time.
- Existing security tools focus on monitoring, not real-time enforcement.

➢ Container IMA support

- IMA support for containers is focused on measurement isolation and remote attestation, not appraisal.
- IMA signature appraisal is underexplored.

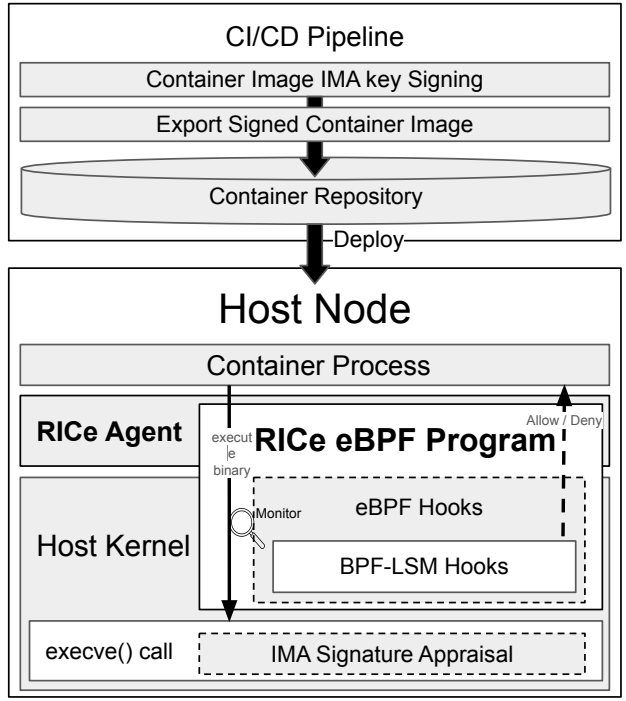

Figure 1. RICe Architecture

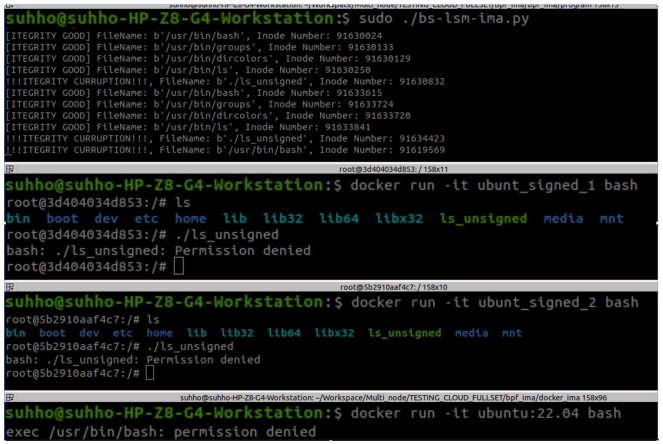

Figure 2. RICe Integrity Enforcement Demonstration

Design & Implementation

➢ Signed Container Image:

- The CI/CD pipeline must cryptographically sign the executable binaries within container images prior to deployment.

➢ IMA Enabled Host Node:

- Provision IMA keys to the kernel.
- Apply a custom IMA policy to support IMA signature appraisal on all binary executions:
  func=BPRM_CHECK mask=MAY_EXEC appraise_type=imasig
- The boot parameter *ima_appraise=log* must be set.
  - *This parameter conflicts with UEFI Secure Boot.*
- Setting up IMA is a major limitation for real-world adoption of RICe and should be addressed in future work.

➢ Monitor IMA appraisal:

- RICe attaches eBPF hooks to monitor IMA appraisal routines (e.g., *ima_appraise_measurement* and *ima_inode_get*) during the execve() system call.
- It caches the IMA appraisal status in an eBPF hash map, using the inode number as the index.

➢ Enforce Integrity:

- Leverage *bprm_creds_from_file* BPF-LSM hook to enforce integrity within the execve() system call boundary.
- If a file fails integrity verification, execution is denied immediately.
- Integrity enforcement is applied exclusively to container processes by checking the namespace ID.

➢ Agent Deployment:

- The RICe agent can be installed on the container host as a container.
- This design allows RICe to be deployed to a Kubernetes environment.

Conclusion & Future Work

RICe combines BPF-LSM and IMA to provide a portable solution for container immutability, mitigating the supply chain trust gap.

Future Work:

➢ Investigate compatibility solutions for UEFI Secure Boot.
- Solutions such as fine-grained IMA policy design can be considered.

➢ Simplify IMA setup for container host nodes.

➢ Evaluate RICe's overhead in real-world container environments.