Secure Kernel Extensibility With eBPF

Soo Yee Lim, Xuechun Cao, Thomas Pasquier

University of British Columbia, Canada

PROBLEM

The safety of eBPF program is not always guaranteed at runtime.

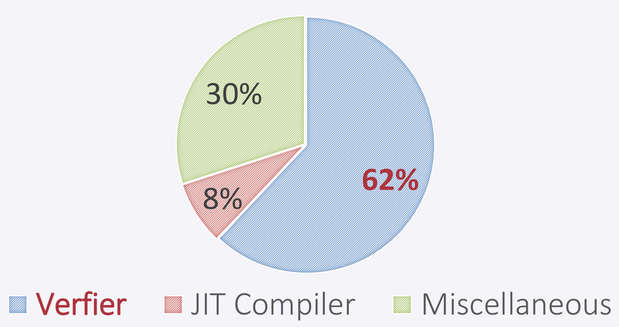
Implementation Bugs

CVE-2021-3490: A bounds tracking bug in the eBPF verifier did not properly update bounds. *Out-of-bounds memory access can lead to privilege escalation.*

Specification Bugs

CVE-2017-17856: The verifier did not enforce strict alignment checks for stack pointers. *Unaligned stack accesses can lead to corruption of spilled registers*

Vulnerabilities in eBPF (2010-2022)



Restricted Use Case For Containers Due To Security Concerns

On most Linux distributions, unprivileged users (e.g., containers) cannot use eBPF.

eBPF

A Linux framework that allows *any* users to extend the kernel without any kernel modifications.

GOAL

To preserve the *confidentiality* and *integrity* of kernel memory at runtime by dynamically sandboxing eBPF programs.

SOLUTION

We leverage state-of-the-art kernel isolation techniques to sandbox eBPF:

- Software fault isolation (completed)
- Hardware-assisted isolation with ARM Pointer Authentication and ARM Memory Tagging Extension (in progress)
- Hardware-assisted software fault isolation with Intel Control-flow Enforcement Technology (future work)

PROLIFERATION OF EBPF

Google	Security auditing Performance monitoring
Meta	Load balancing

USE CASES

Kernel Auditing for Containers

Lim SY, Stelea B, Han X, Pasquier T. Secure Namespaced Kernel Audit for Containers. In Proceedings of the ACM Symposium on Cloud Computing 2021 Nov 1 (pp. 518-532).

Customized Prefetching for Containers

Motivation:

The default prefetching policy in Linux works well for simple sequential memory accesses. However, it does not perform well with complex spatial stride patterns, and there is no universal solution for optimizing all applications' memory access latency.

Solution:

We extend the eBPF framework to allow containers to customize prefetching policies for their own needs.

Contact: sooyee@cs.ubc.ca