Contribution ID: **96** Type: **not specified**

eBPF in kernel lockdown mode

Wednesday, August 26, 2020 9:45 AM (45 minutes)

Linux has a new 'lockdown' security mode where changes to the running kernel requires verification with a cryptographic signature and restrictions to accesses to kernel memory that may leak to userspace.

Lockdown's 'integrity' mode requires just the signature, while in 'confidentiality' mode in addition to requiring a signature the system can't leak confidential information to userspace.

Work needs to be done to add cryptographic signatures for eBPF bytecode. The signature be then passed to the kernel via sys_bpf() reusing the kernel module signing infrastructure.

The main eBPF loader, libbpf, may perform relocations on the received bytecode for things like CO-RE (Compile Once, Run Everywhere), thus tampering with the signature made with the original bytecode.

It is thus needed to move such modifications to the signed bytecode from libbpf to the kernel, so that it may be done after the signature is verified.

This presentation is intended to provide a problem statement, some ideas being discussed, provide a reading list, and to foster awareness about this security feature so that BPF can be used in environments where 'lockdown' mode is required.

I agree to abide by the anti-harassment policy

I agree

Primary author: MELO, Arnaldo (Red Hat)

Presenter: MELO, Arnaldo (Red Hat)

Session Classification: Networking and BPF Summit

Track Classification: Networking & BPF Summit