Contribution ID: **389** Type: **not specified**

BpfJailer: eBPF based Mandatory Access Control

At Meta, due to the proliferation of AI workloads, increased security was needed around key services. In particular, two use cases were jailing untrusted code, and preventing insider and attacker tampering with user data.

AI training and execution of prompts involves executing untrusted code. Meta's network is flat, leading to untrusted workloads operating in the same space as sensitive workloads. Meta built a jailed stack based on microVMs, with a key component being bpf based jailing of the microVM. BpfJailer blocks access to the network, filesystem, IPC, etc. using bpf LSM, in a flexible way tailored to the needs of this workloads.

Another workload involves securing user data in Meta Private Processing. User data is processed in a CVM. BpfJailer is used in this case to prevent tampering with the binaries that can access the CVM, enforcing signed binaries as well as command line argument validation. BpfJailer prevents even root users from tampering with the processes interacting with the CVM, blocking debuggers, /proc access, etc.

This talk will focus on the challenges of supporting these two workloads with bpf LSM, as well as the unsolved problems in this space. Specific discussion will involve:
- Jailing techniques used with bpf LSM in BpfJailer
- Protecting bpf LSM programs and agents from tampering
- Implementing binary and integrity checks and protections
- Managing and orchestrating a bpf LSM agent at scale
- Integrating bpf based enforcement into containerized workloads

Primary author: WISEHART, Liam (Meta)

Presenter: WISEHART, Liam (Meta)

Session Classification: eBPF Track

Track Classification: eBPF Track