
Contribution ID: **184** Type: **not specified**

# Kernel-Resident Regex and Jails: DFA-powered eBPF filtering and certificate-safe agent isolation at fleet scale

Abstract: At Meta's scale, high-signal telemetry competes with overwhelming noise. We present a pragmatic approach that pushes policy into the kernel to eliminate noise at the source and enforce controls before user space is involved. First, we show how we compile regex patterns into deterministic finite automata (DFAs) and execute them in eBPF at Linux Security Module (LSM) and fentry attach points, matching file paths and other identifiers in-kernel to short-circuit events that previously traversed user-space filters. The result is "no-emit"security: irrelevant events are never emitted, cutting CPU, I/O, and downstream processing costs and enabling faster responses.

Second, we connect this to BpfJailer, an eBPF-based enforcement framework used to dynamically sandbox processes (including internal AI agents). Beyond certificate lockdown, we show how the same kernel-first approach extends to executables and privilege boundaries, e.g. blocking execution of untrusted binaries, constraining unauthorized privilege escalation, and guarding sensitive kernel surfaces. For agentic workflows, we also explore DNS-aware enforcement: using in-kernel DNS inspection and network hooks to constrain name resolution to policy-bound allowlists, detect exfil-friendly domains, and coordinate with connection-level blocks to prevent prompt-injected egress. Together, these controls establish agent-specific, low-privilege identity and bounded egress across developer workstations, ephemeral on-demand compute (e.g., containerized batch jobs), and continuous integration (CI) environments.

We'll share the engineering journey: a regex => AST => DFA pipeline for kernel-friendly execution; layered filtering and dynamic configuration; MetArmor's orchestration (BpfHandler, map updaters, event buffers) for fleet rollouts; and response actions (file/process quarantine, targeted network blocks, isolation).

We conclude with operational wins and open questions around verifier limits, DFA size/memory trade-offs, path normalization across filesystems (e.g., btrfs device IDs, mounts, symlinks), uniform kernel/userspace filtering semantics, false positives management, and standardizing "kernel-first"controls for agentic workflows.

**Primary author:** NGAI, Justin (Meta)

**Presenter:** NGAI, Justin (Meta)

**Session Classification:** eBPF Track

**Track Classification:** eBPF Track