BPF LSM etc.

KP Singh, LSF/MM/BPF 2024

Updates

- We have a new maintainer and they are a real person
  - Welcome Matt Bobrowski from Google
- Lots of security use-cases (detection, policy enforcement) with projects like
  - Tetragon
  - Kubarmor
  - Systemd file system restriction
- Prefer use tracing hooks and modify return programs:
  - Overhead
  - Backward compatibility
- BPF token is great for security
  - Principle of least privilege
- Coming soon: Non-zero offset pointers to TRUSTED_ARGS

Topics

- Signed or trusted BPF
- LSM Static calls, what’s going on? Will it get merged before 3025?
- What about kfuncs?

Have we got a handle on signed BPF?

It’s better to call this trusted BPF…

What is trusted BPF?

Trusted BPF loaders

“I trusted this loader to not load malicious BPF programs”

Trusted BPF programs

I trust this BPF program is not malicious and, potentially built in a secure environment

Trusted loaders load trusted programs

How is trust represented?

- Use a private key to sign the loader program
- The kernel verifies the signature of the program and allows BPF operations
- The **Cilium** case:
  - The signature represents the trust that Cilium will not generate malicious programs
- The **bpftrace** case:
  - Add support for signed scripts
  - Trusted bpftrace will only load scripts signed with a private key
  - No -e and command line scripts

Tying it together..

- Use fs-verity
- Created a signed digest of the loader
- Store this signed digest in an extended attribute
- Use `bpf_file_get_xattr` to retrieve the hash in `bprm_committed_creds`
- Use `bpf_verify_pkcs7_signature` BPF LSM hooks to only allow operations by trusted binaries (e.g. `prog_load, map_create, token_create`)
- Pass on the policy at fork (with the `task_alloc`) LSM hook

Setup

On the a trusted host (or install phase of the machine / package)

```
fsverity sign --key signing_key.pem ${loader} "${loader}.sig"

${LOAD_SIGNATURE_IN_XATTR} "${loader}" "${loader}.sig"
```

On the machine

```
fsverity enable "${loader}"
```

Tokens + Trusted load

- Only allow the BPF token to be created for trusted loaders
- The loader creates the token and drops privileges
- Good for long-running loaders (e.g Cilium) as it reduces the attack surface at run-time
- Go further and ensure that BPF operations happen only via the token from trusted loaders.

Static calls in LSM: Why?

Performance

- The branch predictor does not like indirect calls (address of a branch known at runtime, loaded from memory)
- Need expensive mitigations for spectre_v2 (Branch Target Injection) i.e. Retpolines
- Empty BPF LSM hooks everywhere

Correctness

- BPF LSM hooks have side effects
- Need some logic to not invoke the hook when there is no BPF program attached

Status

- “But honestly, this series needs to be turned to 11” - Linus (context)
-

kfunc Discussion

- An effective LSM needs kfuncs to be powerful
- Right now getting kfuncs is hard, Matt tried this and ran into issues with alignment
- We need buy-in / collaboration from other subsystems to allow helpers