Check out our website

bpfman

A Cloud-Native eBPF Program Manager

bpfman

CLOUD NATIVE
COMPUTING FOUNDATION

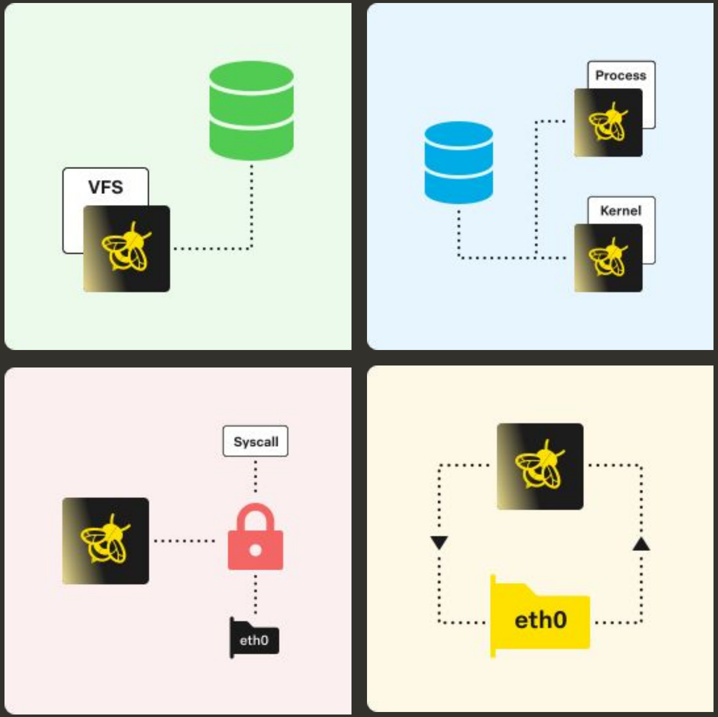
Dave Tucker and Daniel Mellado

What is eBPF and how does it work?

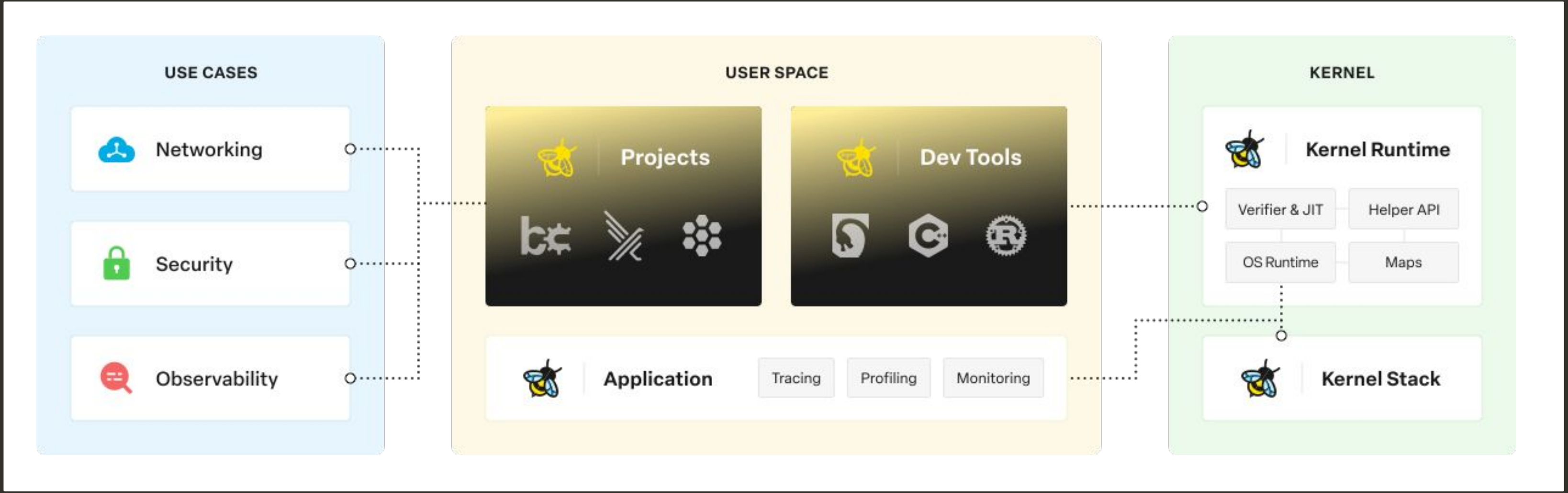
What is eBPF?

eBPF is a technology that allows you to dynamically program the kernel for efficient networking, observability, tracing, and security.

Head over to ebpf.io to learn more



How does it work?



How does it work?

Kernel Space

- Attaches to an eBPF Hook
- Performs use-case specific functions
- May write data into **eBPF Maps**

User Space

- Deploys the Kernel Space program
- May read data from **eBPF Maps**

eBPF Maps

- Data storage that spans userspace and kernel space
- Different types of maps with different properties
- Storage space is limited

Why do we need an eBPF Manager?

Rising Demand

Many projects are choosing to use eBPF:

- Cilium and Calico CNIs
- Pixe: Open source observability
- KubeArmor: Container-aware Runtime Security Enforcement System
- Blixt: Gateway Api L4 conformance implementation
- NetObserv: Open Source Operator for network observability

To name just a few...

But with the rising demand for eBPF, there are still a few issues that are preventing wider adoption.

Security

- All programs that load eBPF probes have effective **root access** to the entire system

- The linux capabilities system isn’t fine-grained enough to sufficiently constrain access to eBPF features

- There is currently no signing for eBPF programs

Co-operation

- Some eBPF hooks in the kernel are exclusive - for example some networking program types

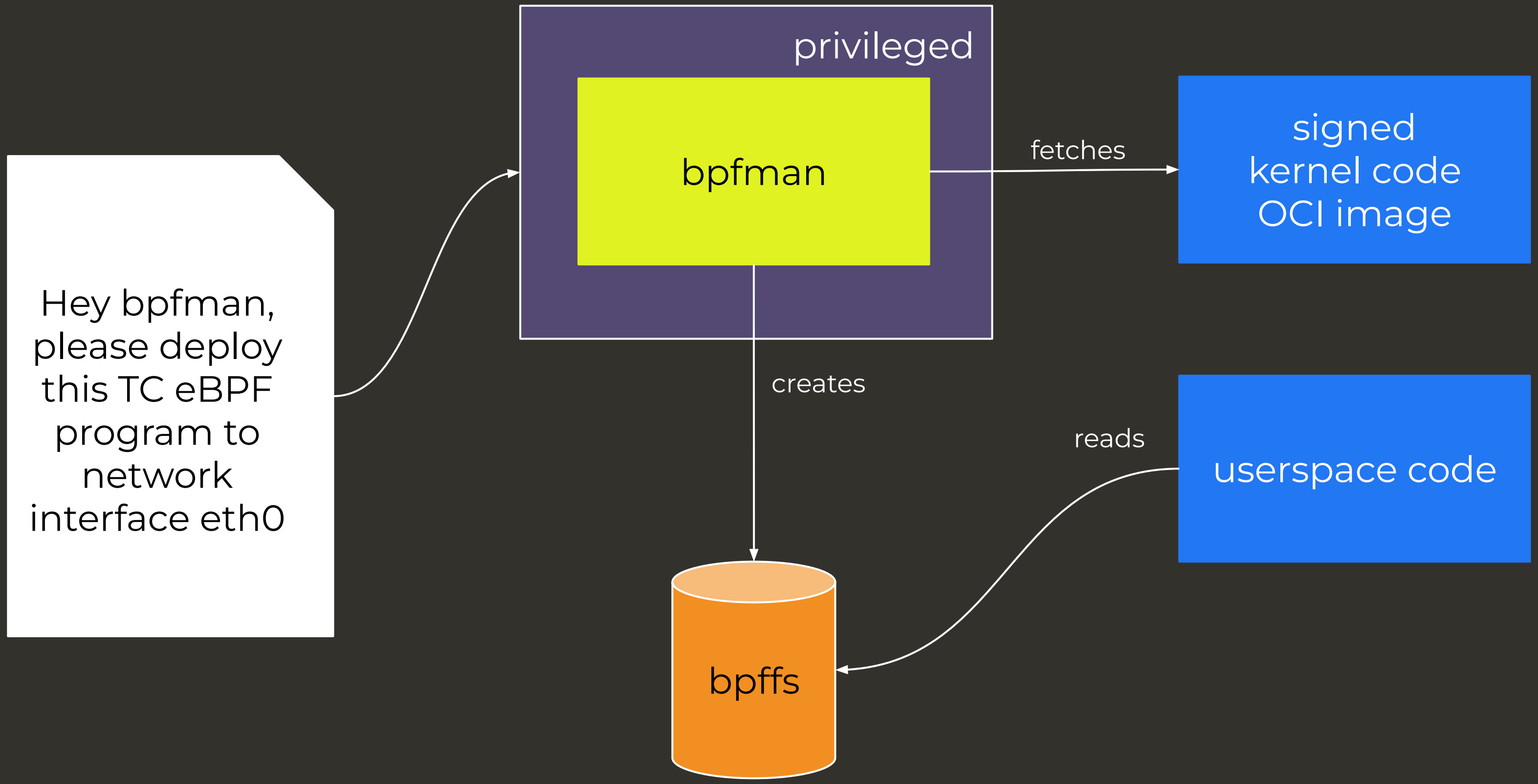
- Even if fixed in the kernel, some entity needs to prioritize programs that are sharing the same hook for the correct effect i.e to run your firewall before your load-balancer.

Enter, bpfman

What is bpfman?

Open source project started in the Red Hat Emerging Tech Networking Group

How does it work?



Hey bpfman, please deploy this TC eBPF program to network interface eth0

Cloud Native Integrations

- Integrates with Kubernetes
  - Provides Custom Resource Definitions (CRDs) to deploy your eBPF bytecode
  - eBPF Filesystems can be provided to applications that need them via our CSI plugin
  - RBAC can be used to restrict which users can use which eBPF features
  - All packaged in an Operator which can be installed from Operator Hub
- Integrates with OCI Registries
  - Both bytecode and userspace components can be stored in OCI registries, greatly simplifying the packaging process
- Integrates with Sigstore
  - eBPF bytecode images can be signed and bpfman can verify the signatures
- Integrates with OpenTelemetry
  - Exposes metrics from the Kernel eBPF subsystem to help troubleshoot eBPF-related issues
  - Exports kernel audit messages as logs in OTEL format

fedora eBPF

- A new sig group was created in late 2023 to gather interest around eBPF in Fedora. Fedora eBPF Special Interest Group

- Identified bpfman as a useful tool to user as a bpf manager and decided to push for it to be included in Fedora, aiming for Fedora 40.
    - There’s currently a Self Contained Change proposed.

Packaging bpfman

- Bpfman main component is written in rust

- Currently in review for addition at https://bugzilla.redhat.com/show_bug.cgi?id=2257948.
  - Built mainly using rust2rpm
  - Dependencies…

Bpfman rust dependencies and dependency tree

- Missing in Fedora
  - oci-distribution
  - sigstore
  - sled
  - systemd-journal-logger
  - Tonic
- Too new in Fedora
  - netlink-packet-route
  - rtnetlink
- Too old in Fedora
  - comfy-table
  - prost / prost-types

Current status

- Added several new rust packages
  - rust-cache
  - rust-poly135
  - ...
- Thanks to:
  - #rust-sig-group and Fabio Valentini
  - Mikel Olasagasti
  - Fedora Steering Committee
  - ebpf-sig-group

DEMO TIME!

What’s next for bpfman?

- We’re in the CNCF Sandbox Queue
- We’re part of the Fedora eBPF Special Interest Group and expect to ship in Fedora 40
- Continuing to work with the Kubernetes community to broaden adoption and establish best practices for eBPF deployment

Release 0.4.0 will be the first release as “bpfman”, due Q1 2024

- No more daemon
- Experimental OTEL metrics exporter and log exporter

Later in 2024 we expect to work on:

- Integration with BPF Tokens to secure applications that don’t load eBPF via bpfman
- Deeper Sigstore integration

Thank you!

Contact us at #bpfman in k8s slack
Check out our website