The Inspektor Gadget Project

An eBPF systems inspection tool and framework

CNCF Sandbox Project

Goals

eBPF

You have a high-level understanding of eBPF and its superpowers

Inspektor Gadget

You learn about Inspektor Gadget and how it “supercharges” eBPF

Fun

You are engaged during this talk and enjoy learning about these items

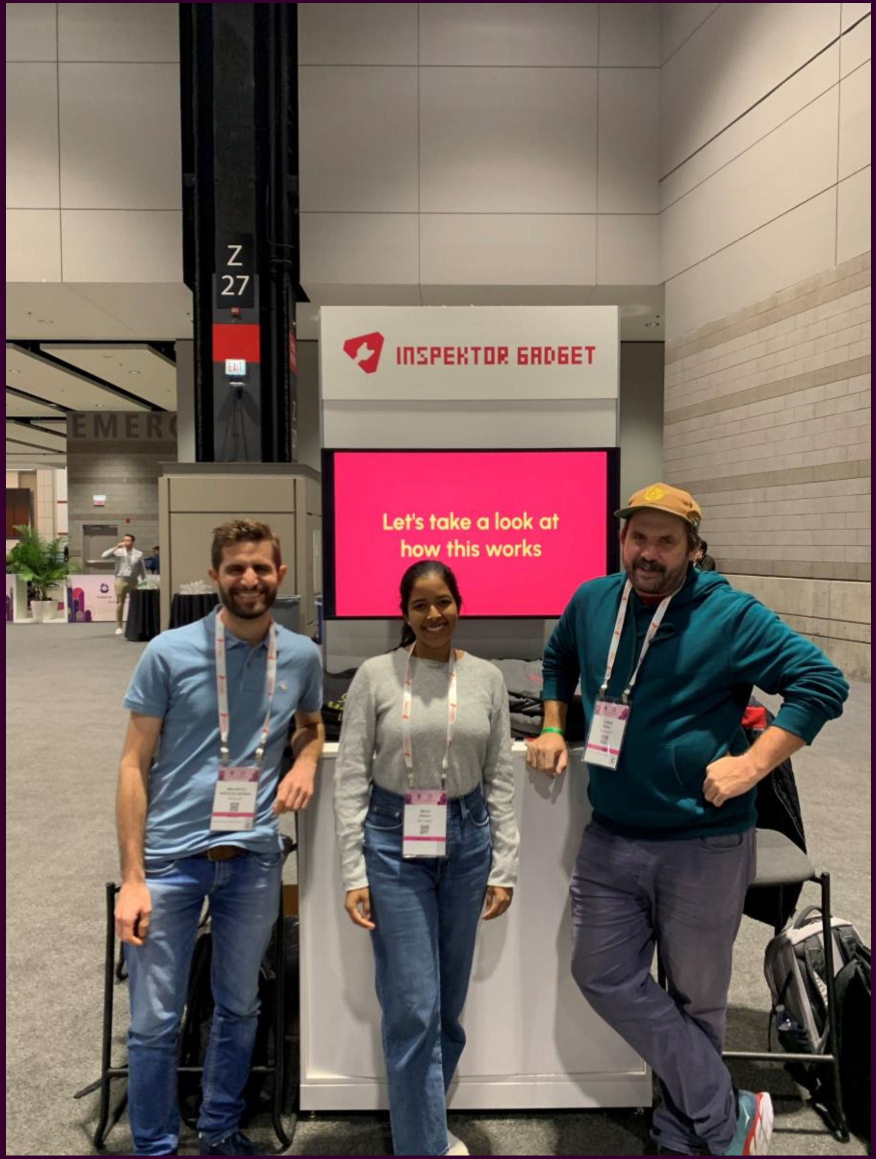
Agenda

Hello!

Maya Singh

Product Manager

@ Microsoft



Mauricio Vasquez Bernal Chris Kuehl

Disclaimer:

Under Active development!

+ Some features are behind an experimental flag or in progress
+ Others still in development branches
+ Will stabilize over the next few releases/months
+ Eager to have your feedback on functionality and UX

What do you think of when you hear "eBPF?"

Menti.com | 2209 6641

What is eBPF?

What is eBPF?

eBPF is in-kernel bytecode runtime used for tracing, security, networking etc…

eBPF Capabilities

+ Brings flexibility to the kernel
+ Low strain from a performance perspective
+ Won’t crash your kernel

Examples of eBPF use cases

Tracing

eBPF can be used to measure CPU usage, memory allocation, and similar metrics which can be used for performance troubleshooting

Security

eBPF can be used to enforce access control policies, you can whitelist/blacklist specific system calls, network connection etc…

Networking

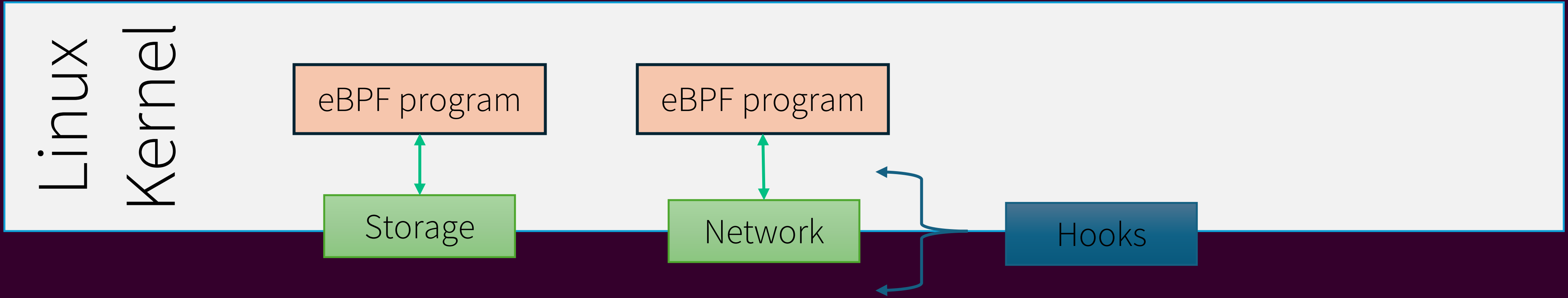
eBPF allows for packet filtering and modification within the Linux kernel (Firewall rules)

eBPF traffic monitoring | Android Open Source Project

Keynote: eBPF - Everything You Need to Know in 5 Minutes - Thomas Graf, CTO, Isovalent (youtube.com)

eBPF Hooks

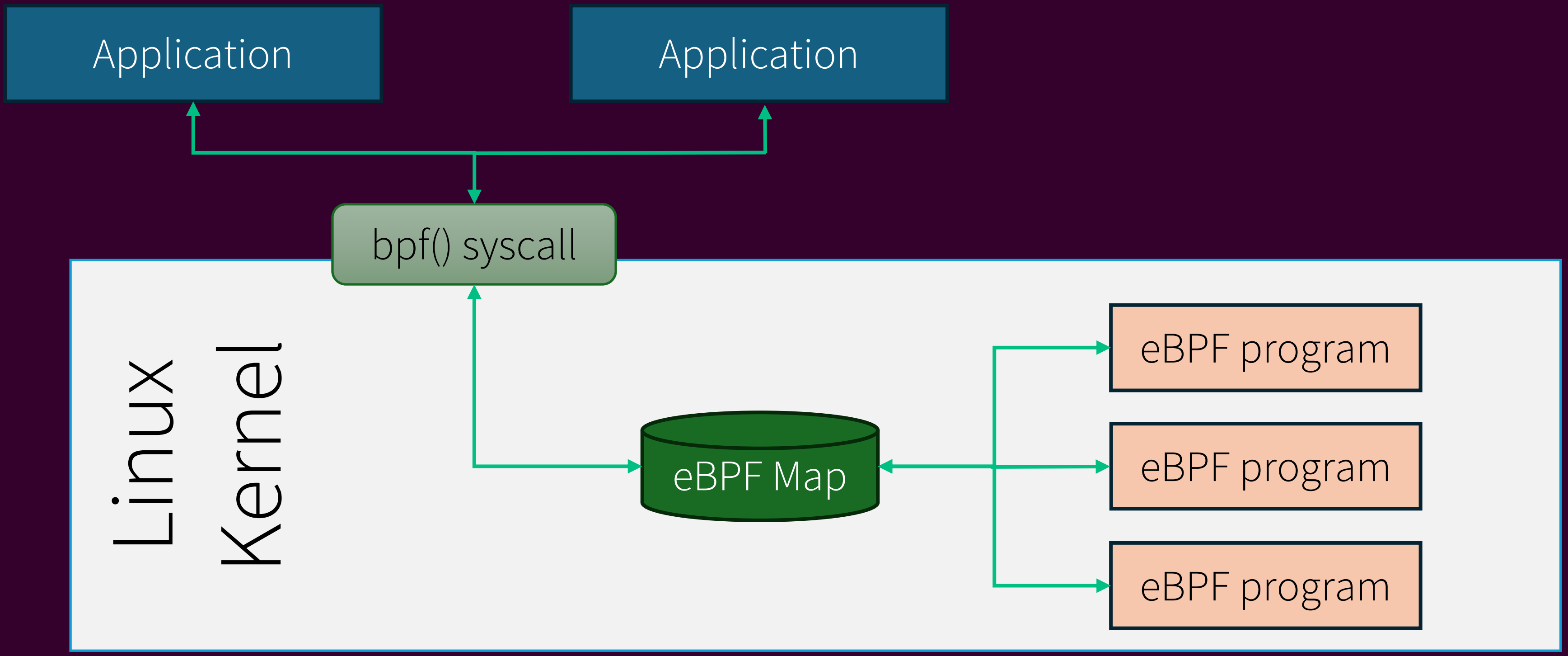
+ eBPF is event driven, when “hooks” are passed, eBPF programs are executed



Source: Mauricio Vasquez Bernal & Chris Kuehl

eBPF Maps

- Key/Value structures to share information between eBPF programs and user space applications



Source: Mauricio Vasquez Bernal & Chris Kuehl

What is eBPF?

eBPF is in-kernel bytecode runtime used for tracing, security, networking etc…

eBPF Capabilities

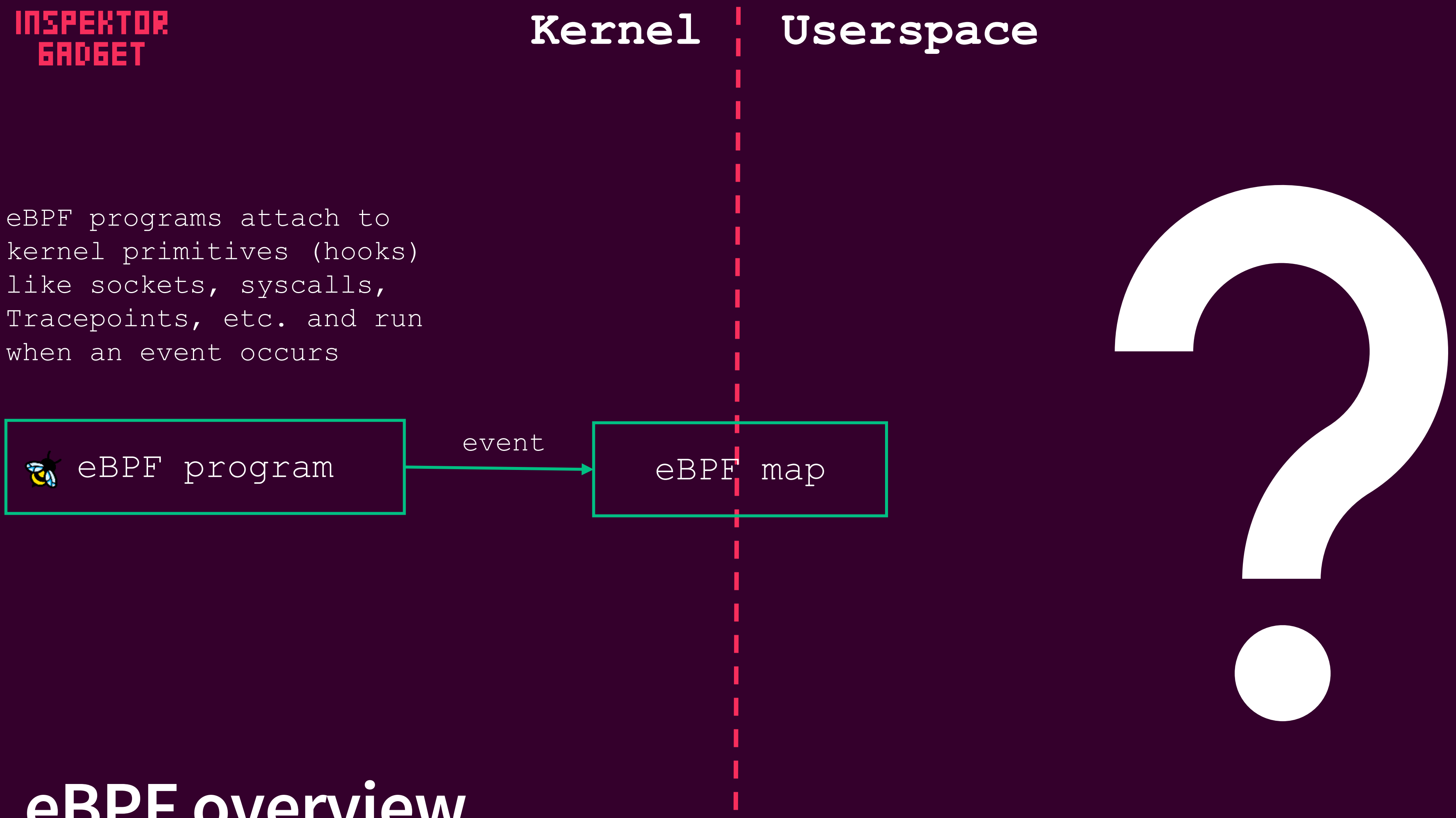
+ Brings flexibility to the kernel
+ Low strain from a performance perspective
+ Safe way to access the kernel

eBPF Challenges

+ Steep learning curve
+ Requires deep level of low-level systems troubleshooting
+ Limited higher level context

eBPF overview

eBPF programs attach to kernel primitives (hooks) like sockets, syscalls, Tracepoints, etc. and run when an event occurs



Source: Mauricio Vasquez Bernal & Chris Kuehl

What is Inspektor Gadget?

What is Inspektor Gadget?

Tool

A set of tools (gadgets) that empower users to inspect Linux and Kubernetes systems using eBPF programs in an accessible way

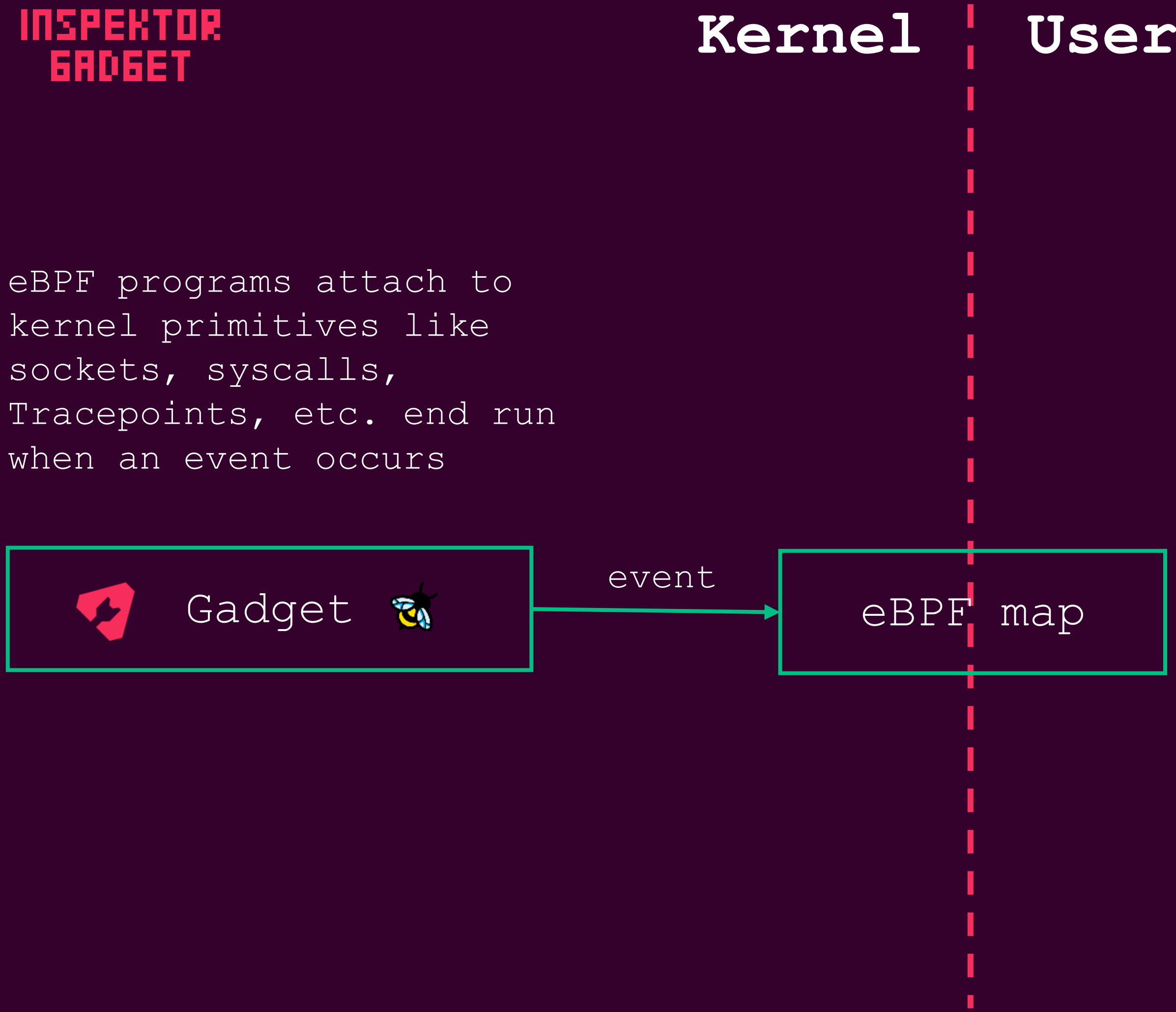
Framework

A method through which eBPF developers can easily build, package, deploy, and run "gadgets"

Community

Bridging the gap between highly technical eBPF concepts and the everyday developer who wants visibility into Linux and Kubernetes systems

eBPF with Inspektor Gadget


Kernel
Userspace
eBPF programs attach to kernel primitives like sockets, syscalls, Tracepoints, etc. end run when an event occurs
Gadget
event
eBPF map

- Enrichment
- Filtering
- Userspace processing
- Data export
- Sharing & distribution
- Many modes of use

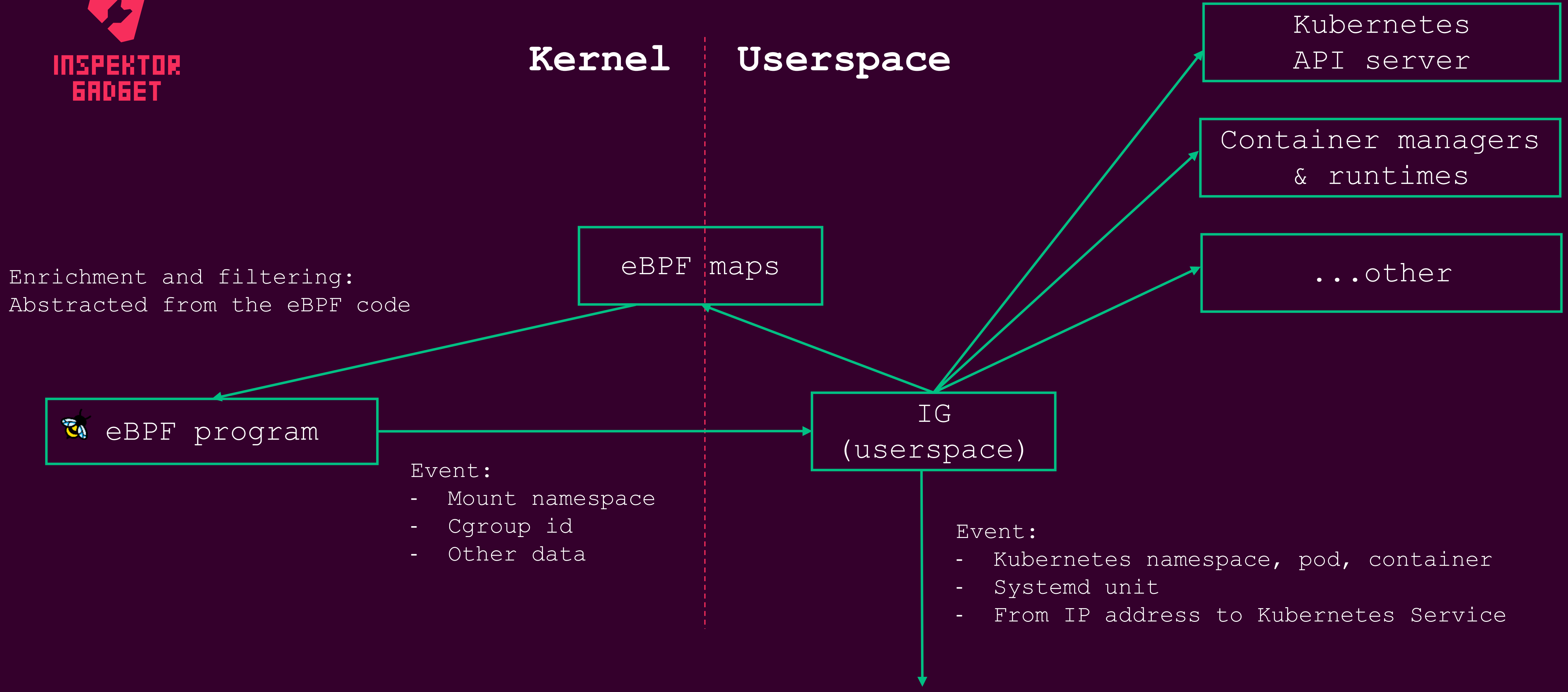
Source: Mauricio Vasquez Bernal & Chris Kuehl

Event enrichment and filtering

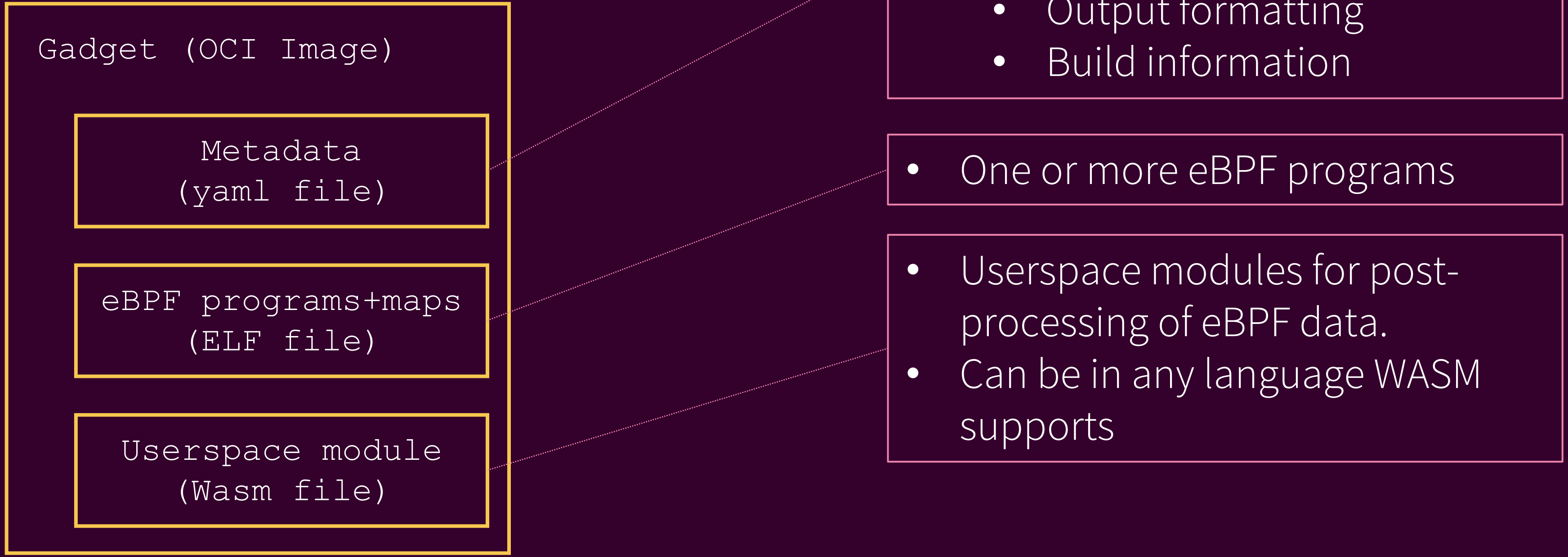
- Problem: events from eBPF give low-level data:
  - Kernel namespaces
  - cgroups
- Solution: event enrichment adds high-level data:
  - Kubernetes pods, containers
  - Domain names or Kubernetes services from IP
  - Container information
- Event filtering: showing a subset of events
  - From selected containers, Kubernetes pod, namespace, labels
  - Filtered in eBPF for performance, but abstracted for gadget authors

Source: Mauricio Vasquez Bernal & Chris Kuehl

Event enrichment and filtering


Kernel
Userspace
Kubernetes API server
Container managers & runtimes
...other
eBPF maps
Enrichment and filtering: Abstracted from the eBPF code
eBPF program
IG (userspace)
Event:
- Mount namespace
- Cgroup id
- Other data
Event:
- Kubernetes namespace, pod, container
- Systemd unit
- From IP address to Kubernetes Service

Source: Mauricio Vasquez Bernal & Chris Kuehl



- Information about
  - The gadget
  - Capabilities
  - Output formatting
  - Build information

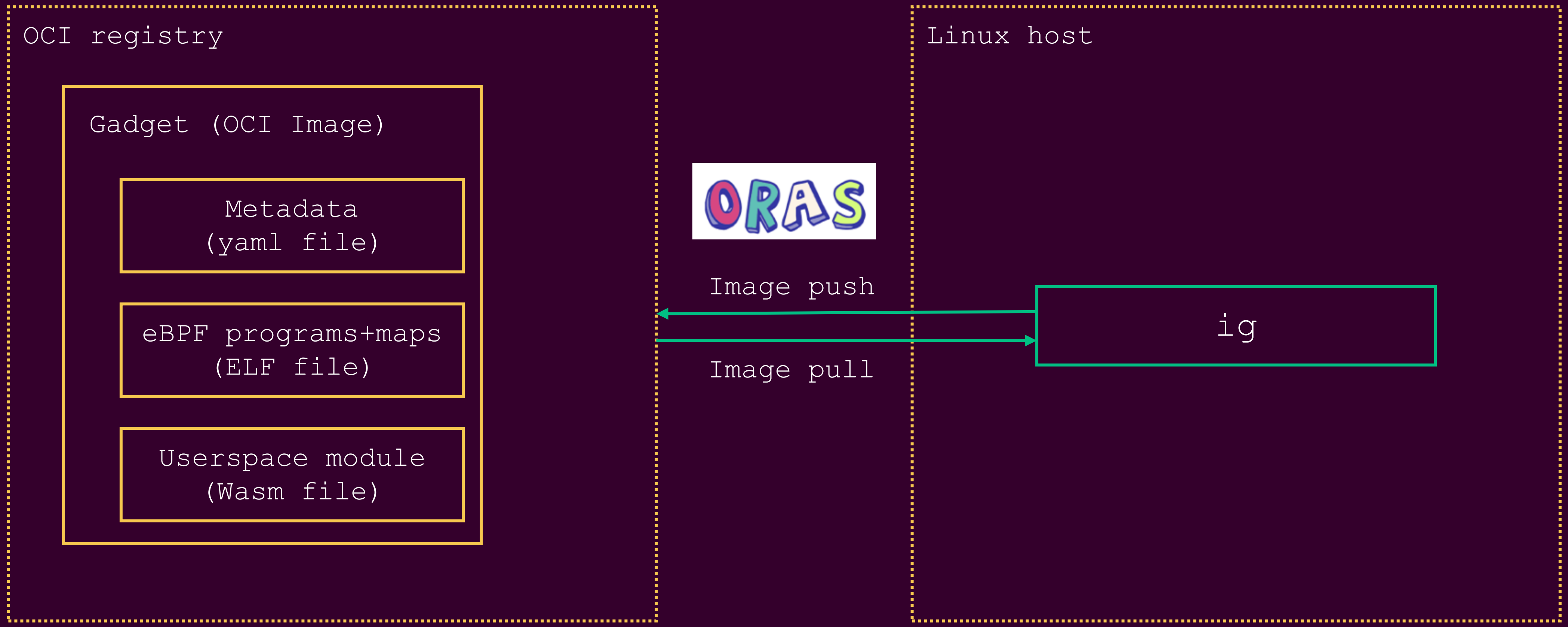
- One or more eBPF programs

- Userspace modules for post-processing of eBPF data.
- Can be in any language WASM supports

* Also looking to include
- Documentation
- Source code
- Logo
- etc.

Anatomy of a gadget

Source: Mauricio Vasquez Bernal & Chris Kuehl



Anatomy of a gadget

Source: Mauricio Vasquez Bernal & Chris Kuehl

“Official” Gadgets

- **Advise**: Recommend system configurations based on collected information
  - seccomp-profile, network-policies
- **Audit**: Audit a subsystem
  - seccomp
- **Profile**: Profile different subsystems
  - block-io, cpu
- **Snapshot**: Take a snapshot of a subsystem and print it
  - process, socket
- **Top**: Gather, sort and periodically report events according to a given criteria
  - file, tcp
- **Trace**: Trace and print system events
  - bind, dns, exec, mount, oomkill, tcp{drop, retrans}, open, few more…

Sample Gadgets

`trace DNS`

Prints information about DNS queries and responses sent and received by different pods

`top block io`

Used to visualize containers generating the most block device input/output

`snapshot process`

Gets a list of running processes on the host

Which type of gadget seems the most useful to you?

Menti.com | 2209 6641

Why Inspektor Gadget?

Background

+ Started in 2019
+ Wanted to bring eBPF and BCC tools to Kubernetes
  + …meaning it was Kubernetes only
+ Now supports…
  + Linux hosts with the *ig* cli tool
  + Kubernetes with the *kubectl-gadget* kubectl plugin
+ Discovered we had essentially built a framework
+ Has been transitioning from a collection of “built-in” *gadgets* to tool for building, packaging and running *gadgets* packaged as OCI images

Why Inspektor Gadget?

+ eBPF is an extremely powerful tool for gathering system information
+ But eBPF is hard – technically and intuitively
+ Once you have data, it’s still not immediately useful
  + How does this kernel data relate to my system as I understand it?
  + Where do I send the data?
+ Lots of additional tooling needed for…
  + Managing eBPF programs
  + Mapping kernel data to higher-level resources (K8s, container runtimes, etc.)
  + Doing userspace processing
  + Exporting data / providing data via API

Some examples of IG use cases ☺

ARMO

ARMO and the Opensource project Kubescape use IG to enhance detecting vulnerabilities in containers

MS Defender

Inspektor Gadget is used in MS Defender for Containers to collect security events, generate insights and real-time threat detection alerts.

Amazon EKS

Amazon EKS Users leverage Inspektor Gadget to inspect their Kubernetes environment with eBPF tools

Source: Empowering Kubernetes Observability with eBPF on Amazon EKS | Containers
CNCF On demand webinar: eBPF-based Kubernetes Security | CNCF

How you can leverage IG

Inspektor Gadget Modes of Operation

- Linux host
  - Ig binary
  - Ig inside a container
- Client-server setup
  - Ig runs as a service inside the host
  - We use a Client called gadgetctl to control the service (via API call)
- Kubernetes
  - Ig is deployed via daemon set
  - Kubectl-gadget plugin used to control the daemon set
- Go library API
  - Work in progress

Inspektor Gadget Data Export Options

Raw Data

We provide a json file of data that you can then do with what you please

Prometheus

We are working towards a more efficient exporting of metrics to Prometheus

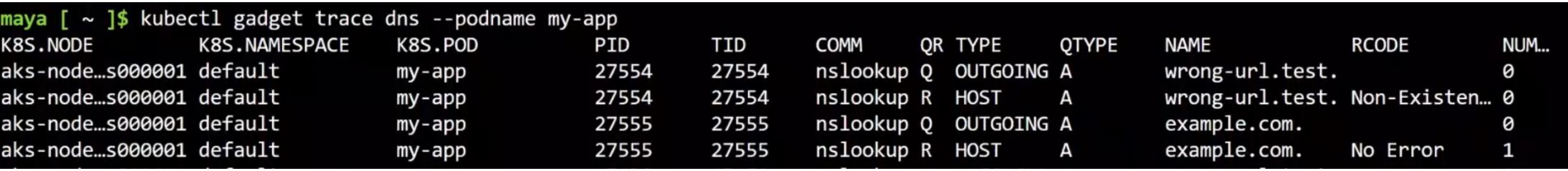
OpenTelemetry

Ultimately, we want to support logs, metrics, and traces through OTel

Demo – trace DNS and top TCP

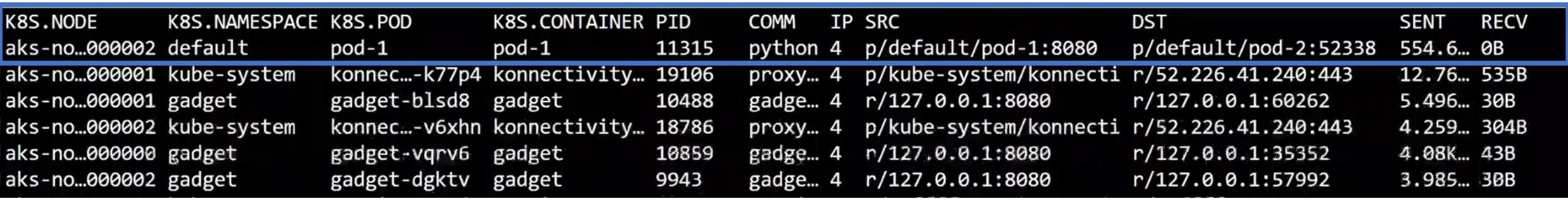
Demo – trace DNS and top TCP

trace DNS output



```
maya [ ~ ]$ kubectl gadget trace dns --podname my-app
K8S.NODE         K8S.NAMESPACE    K8S.POD          PID       TID       COMM     QR TYPE     QTYPE    NAME            RCODE        NUM…
aks-node…s000001 default          my-app           27554     27554     nslookup Q  OUTGOING A        wrong-url.test.              0
aks-node…s000001 default          my-app           27554     27554     nslookup R  HOST     A        wrong-url.test. Non-Existen… 0
aks-node…s000001 default          my-app           27555     27555     nslookup Q  OUTGOING A        example.com.                 0
aks-node…s000001 default          my-app           27555     27555     nslookup R  HOST     A        example.com.    No Error     1
```

top TCP output

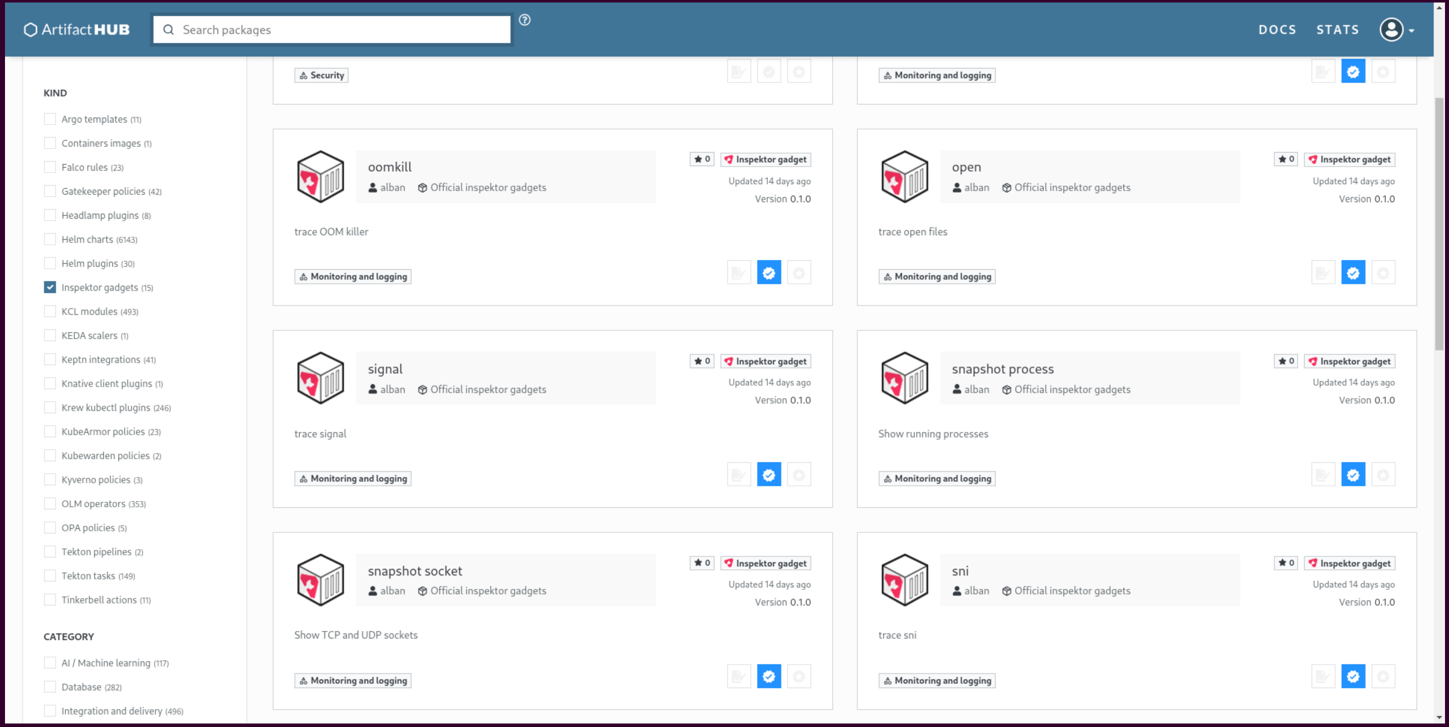


```
K8S.NODE       K8S.NAMESPACE K8S.POD        K8S.CONTAINER PID     COMM     IP SRC                      DST                      SENT     RECV
aks-no…000002  default       pod-1          pod-1         11315   python   4  p/default/pod-1:8080     p/default/pod-2:52338    554.6…   0B
aks-no…000001  kube-system   konnec…-k77p4  konnectivity… 19106   proxy…   4  p/kube-system/konnecti   r/52.226.41.240:443      12.76…   535B
aks-no…000001  gadget        gadget-blsd8   gadget        10488   gadge…   4  r/127.0.0.1:8080         r/127.0.0.1:60262        5.496…   30B
aks-no…000002  kube-system   konnec…-v6xhn  konnectivity… 18786   proxy…   4  p/kube-system/konnecti   r/52.226.41.240:443      4.259…   304B
aks-no…000000  gadget        gadget-vqrv6   gadget        10859   gadge…   4  r/127.0.0.1:8080         r/127.0.0.1:35352        4.08K…   43B
aks-no…000002  gadget        gadget-dgktv   gadget        9943    gadge…   4  r/127.0.0.1:8080         r/127.0.0.1:57992        3.985…   30B
```

What’s Next?

Looking ahead

- Support declarative way to run gadgets
  - Configuration file
- Support of various export options
- Golang API full support of image based gadgets
- Gadgets in Artifacthub.io
- Understand community priorities
- Proper documentation for all this ;-)



Artifact HUB · Search packages · DOCS · STATS

KIND
- Argo templates (11)
- Containers images (1)
- Falco rules (23)
- Gatekeeper policies (42)
- Headlamp plugins (8)
- Helm charts (6143)
- Helm plugins (30)
- [✓] Inspektor gadgets (15)
- KCL modules (493)
- KEDA scalers (1)
- Keptn integrations (41)
- Knative client plugins (1)
- Krew kubectl plugins (246)
- KubeArmor policies (23)
- Kubewarden policies (2)
- Kyverno policies (3)
- OLM operators (353)
- OPA policies (5)
- Tekton pipelines (2)
- Tekton tasks (149)
- Tinkerbell actions (11)

CATEGORY
- AI / Machine learning (117)
- Database (282)
- Integration and delivery (496)

Security

Monitoring and logging

oomkill — alban — Official inspektor gadgets — ★ 0 — Inspektor gadget — Updated 14 days ago — Version 0.1.0
trace OOM killer
Monitoring and logging

open — alban — Official inspektor gadgets — ★ 0 — Inspektor gadget — Updated 14 days ago — Version 0.1.0
trace open files
Monitoring and logging

signal — alban — Official inspektor gadgets — ★ 0 — Inspektor gadget — Updated 14 days ago — Version 0.1.0
trace signal
Monitoring and logging

snapshot process — alban — Official inspektor gadgets — ★ 0 — Inspektor gadget — Updated 14 days ago — Version 0.1.0
Show running processes
Monitoring and logging

snapshot socket — alban — Official inspektor gadgets — ★ 0 — Inspektor gadget — Updated 14 days ago — Version 0.1.0
Show TCP and UDP sockets
Monitoring and logging

sni — alban — Official inspektor gadgets — ★ 0 — Inspektor gadget — Updated 14 days ago — Version 0.1.0
trace sni
Monitoring and logging

INSPEKTOR GADGET

Think about how eBPF could be used to enhance the projects you’re working on and see if we have a gadget that could help you!

Call to Action

INSPEKTOR GADGET

Web: inspektor-gadget.io

Slack: #inspektor-gadget on the Kubernetes Slack

Github: **github.com/inspektor-gadget/inspektor-gadget**

Hope to see you around SCaLE!

Thank you