Use Falco and eBPF to protect your applications

Thomas Labarussias

June 08, 2023

Who am I?

Thomas Labarussias
OSS/Ecosystem Advocate at Sysdig
SRE for over 8 years

Contributor to Falco
Creator of Falcosidekick/UI

github.com/Issif
@TLabarussias
untappd.com/user/Issif

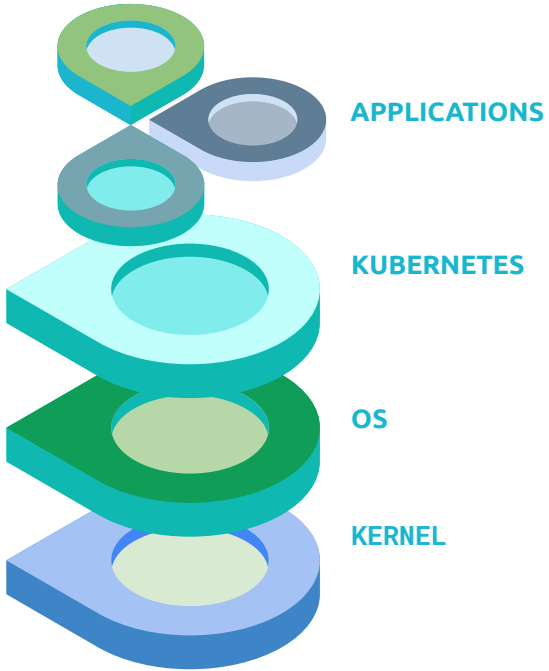
Runtime Security?

All the **tools** and **procedures** put in place to **secure** an **application**, **containerised** or not, during its lifetime in **production**

Syscalls

The **System calls** are the way for the programs to ask to the Kernel accesses to the resources

- process
- network
- IO files
- And more...



Falco what's that?

Falco

CLOUD NATIVE COMPUTING FOUNDATION

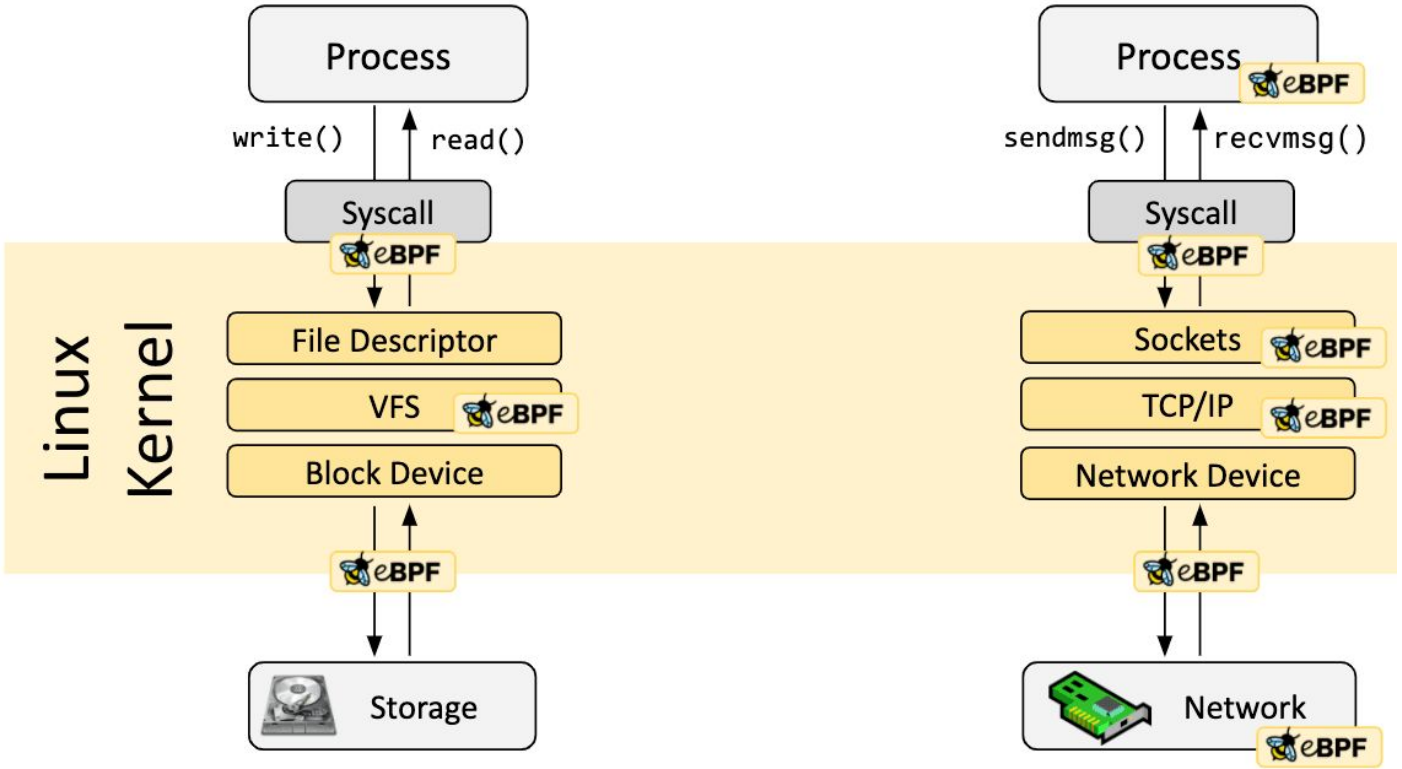
CNCF incubation-level project

Falco is a **cloud native project** securing running applications, it's the most advanced **threat detection engine** in **Kubernetes**

5.8k on Github
60M+ pulls

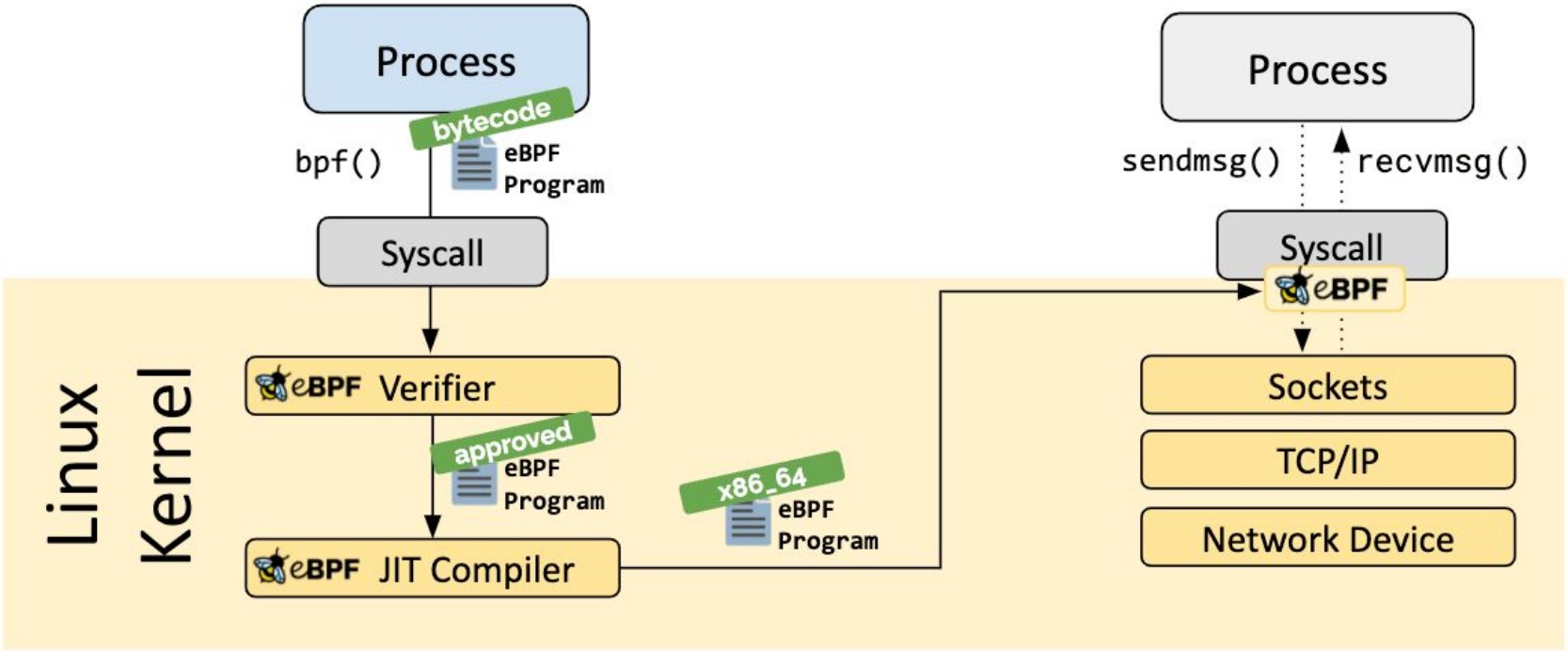
- **Linux Kernel feature** which allows to run programs **in the Linux Kernel without changing** its code or loading a module
- **Access Kernel activity** without risking **system stability** or **security**
- Useful for **security**, **monitoring** and **troubleshooting**
- A new probe for Falco is in development (**CoRe**)

eBPF The hooks



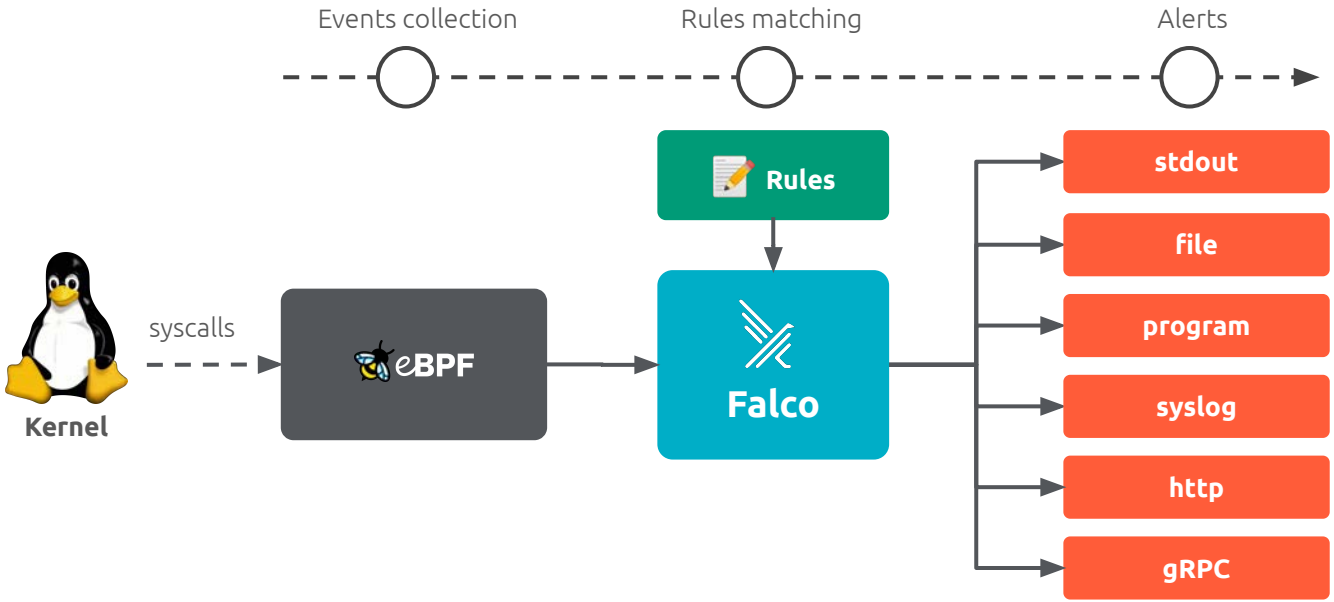
source: ebpf.io

eBPF The verification

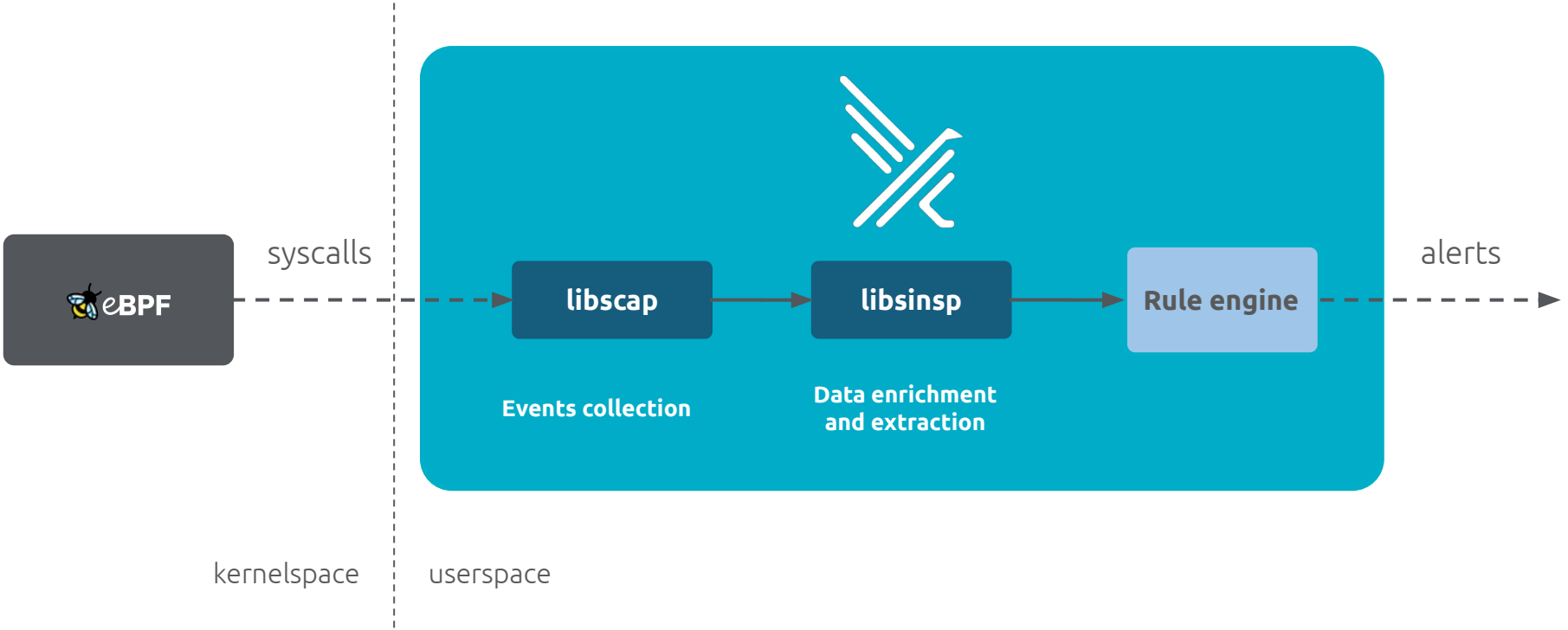


source: ebpf.io

Falco's architecture



Falco's architecture



libscap aka **lib**rary for **S**ystem **CAP**ture

- **userspace** library
- communicates with the drivers
- reads syscall events from the ring buffer (where drivers place them)
- forwards them up to **libsinsp**

https://github.com/falcosecurity/libs/

libsinsp aka **lib**rary for **S**ystem **INSP**ection

- **userspace** library
- receives events from **libscap**
- **enriches** events with machine states
- performs events filtering

https://github.com/falcosecurity/libs/

Falco: The rule engine

```
- rule: Terminal shell in container
  desc: A shell has been spawned in a container.
  condition: >
    spawned_process and container
    and shell_procs
  output: >
    A shell was spawned in a container (user=%user.name
    user_loginuid=%user.loginuid %container.info shell=%proc.name
    parent=%proc.pname cmdline=%proc.cmdline container_id=%container.id)
  priority: WARNING
  tags: [container, shell, mitre_execution]
```

Falco: The rule engine

```
- rule: Terminal shell in container
  desc: A shell has been spawned in a cont
  condition: >
    spawned_process and container
    and shell_procs
  output: >
    A shell was spawned in a container (us
user_loginuid=%user.loginuid %container.in
parent=%proc.pname cmdline=%proc.cmdline c
  priority: WARNING
  tags: [container, shell, mitre_execution
```

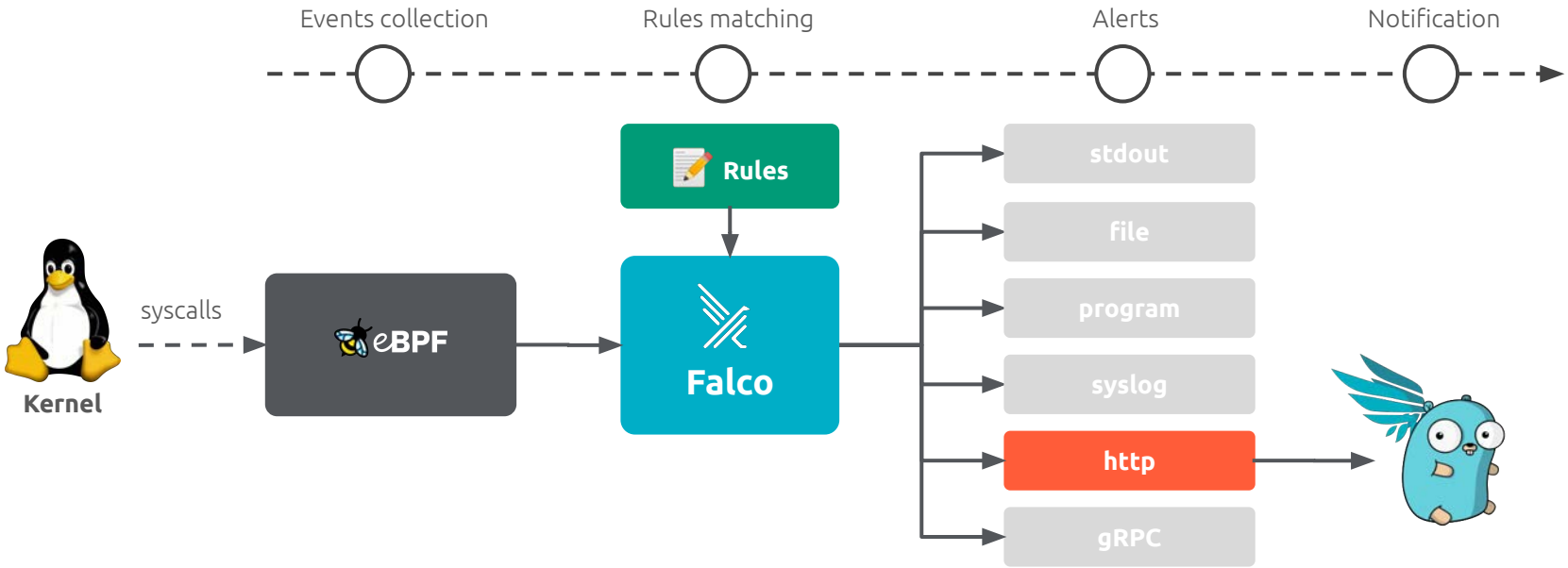
```
- list: shell_binaries
  items: [ash, bash, csh, ksh, sh,
tcsh, zsh, dash]

- macro: shell_procs
  condition: proc.name in
(shell_binaries)

- macro: container
  condition: (container.id != host)

- macro: spwaned_process
  condition: >
        evt.type in (execve, execveat)
        and evt.dir=<
```

Falco: The default ruleset

- Privilege escalation
- R/W to sensitive directories
- Executing shell
- Execute SSH binaries
- Mutating binaries
- Creating symlinks
- ...

  ~70 system rules

Connect Falco: Falcosidekick



Falcosidekick

Connect Falco to your ecosystem

chat

logs

queue/streaming

faas

metrics

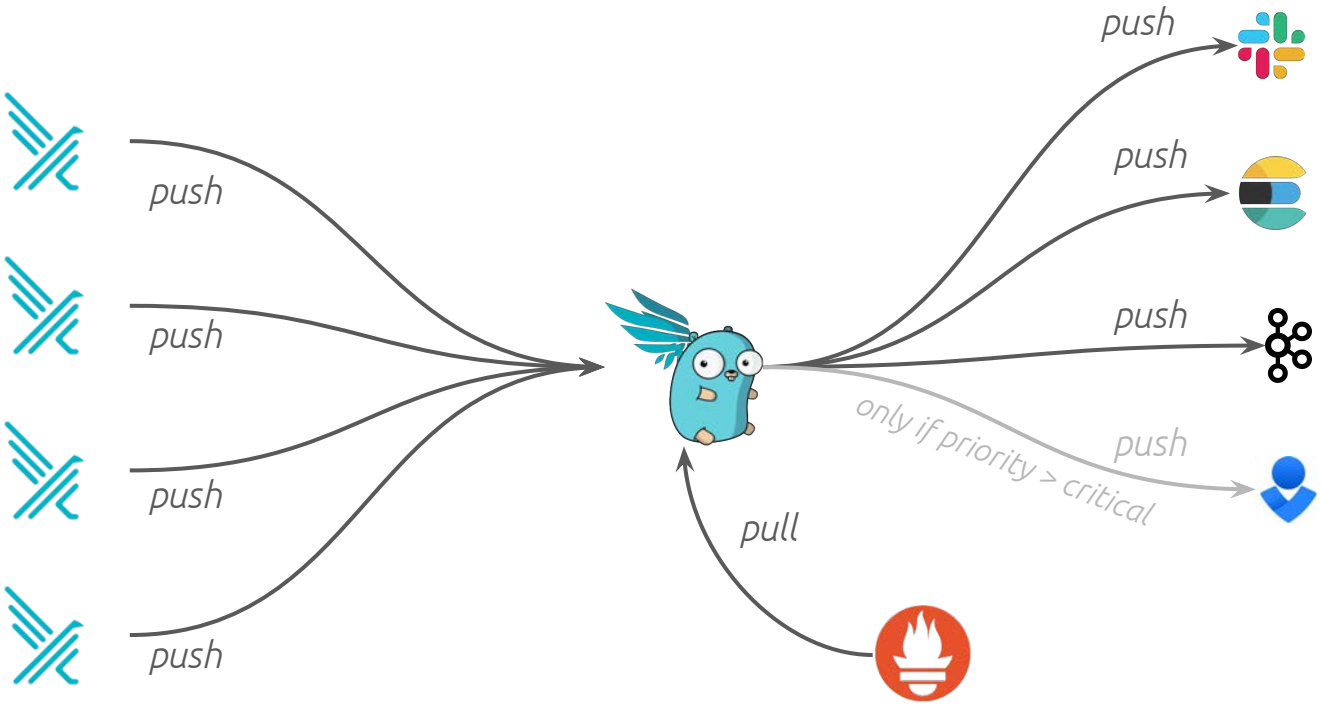
alerts

storage

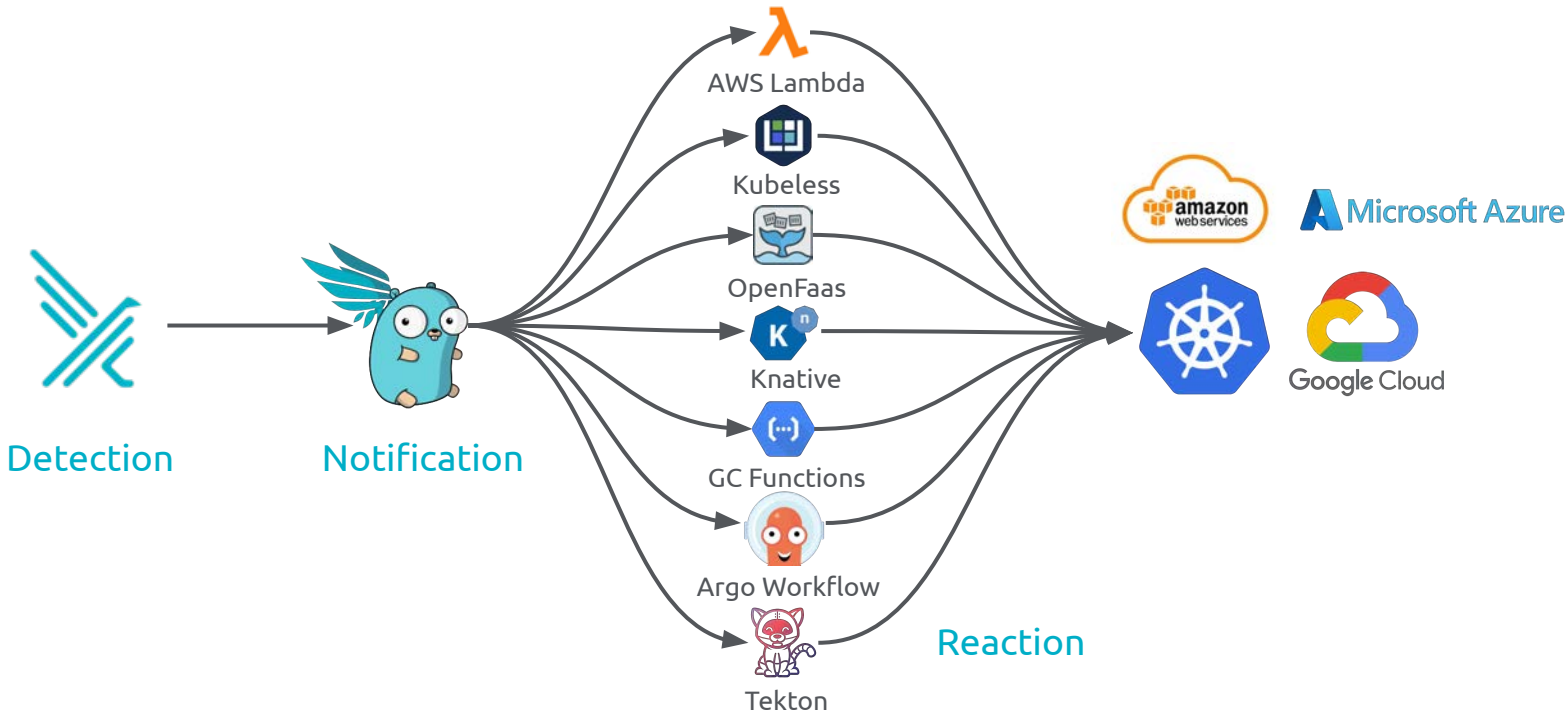
and more…

github.com/falcosecurity/falcosidekick

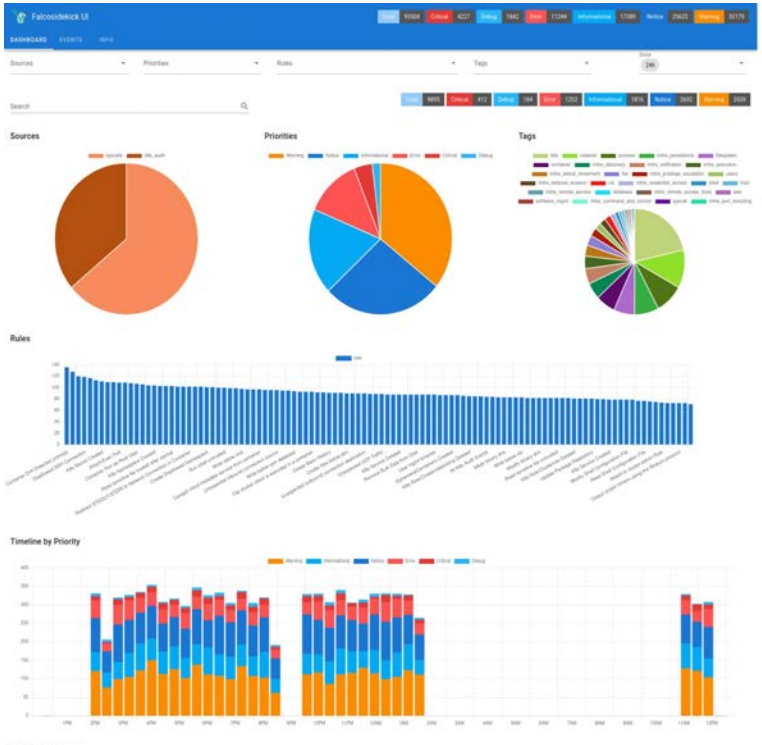
Falcosidekick

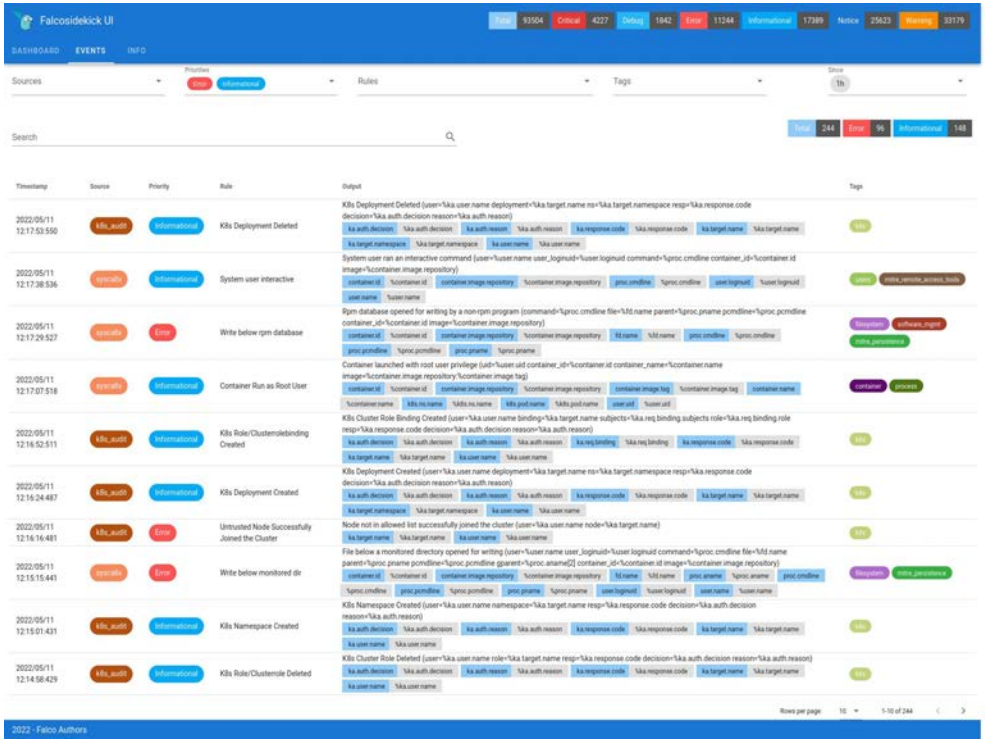


React to events



Falcosidekick UI





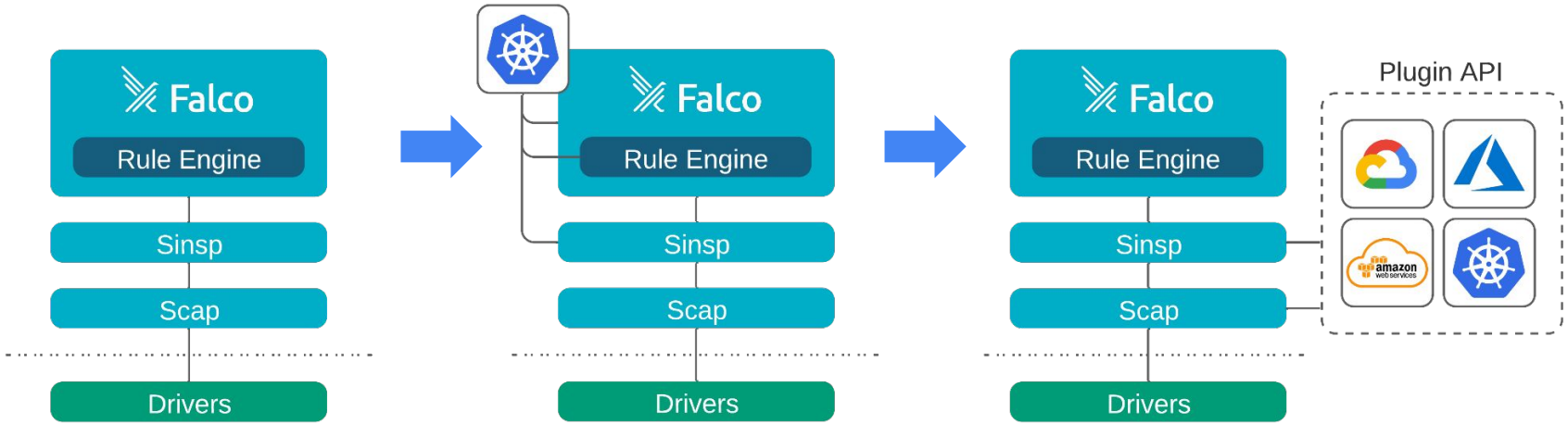
github.com/falcosecurity/falcosidekick-ui

The Evolution: the Plugins

Monitoring **only system events** coming from the Kernel

Monitoring **K8S Audit Logs**
New event source & data enrichment

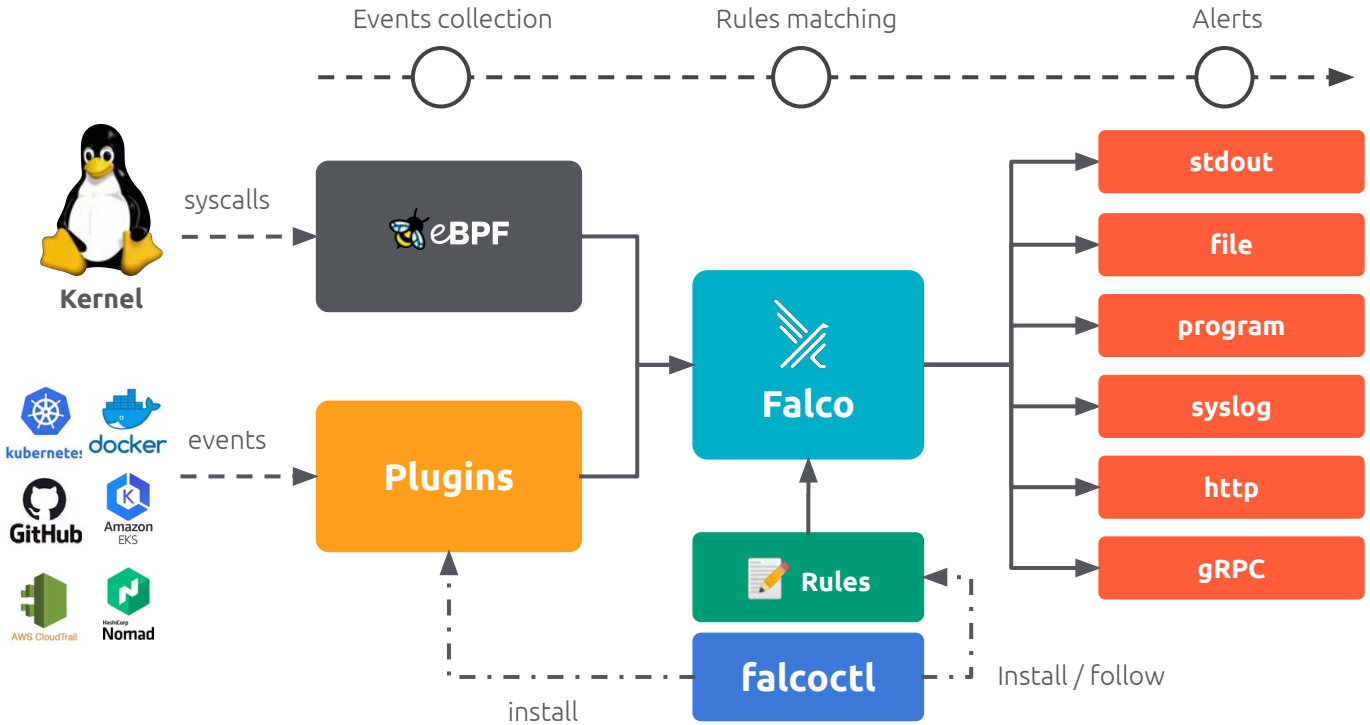
Monitoring **Cloud events**
(or any sort of events, technically)
Standard API definitions for adding new sources & event data enrichment



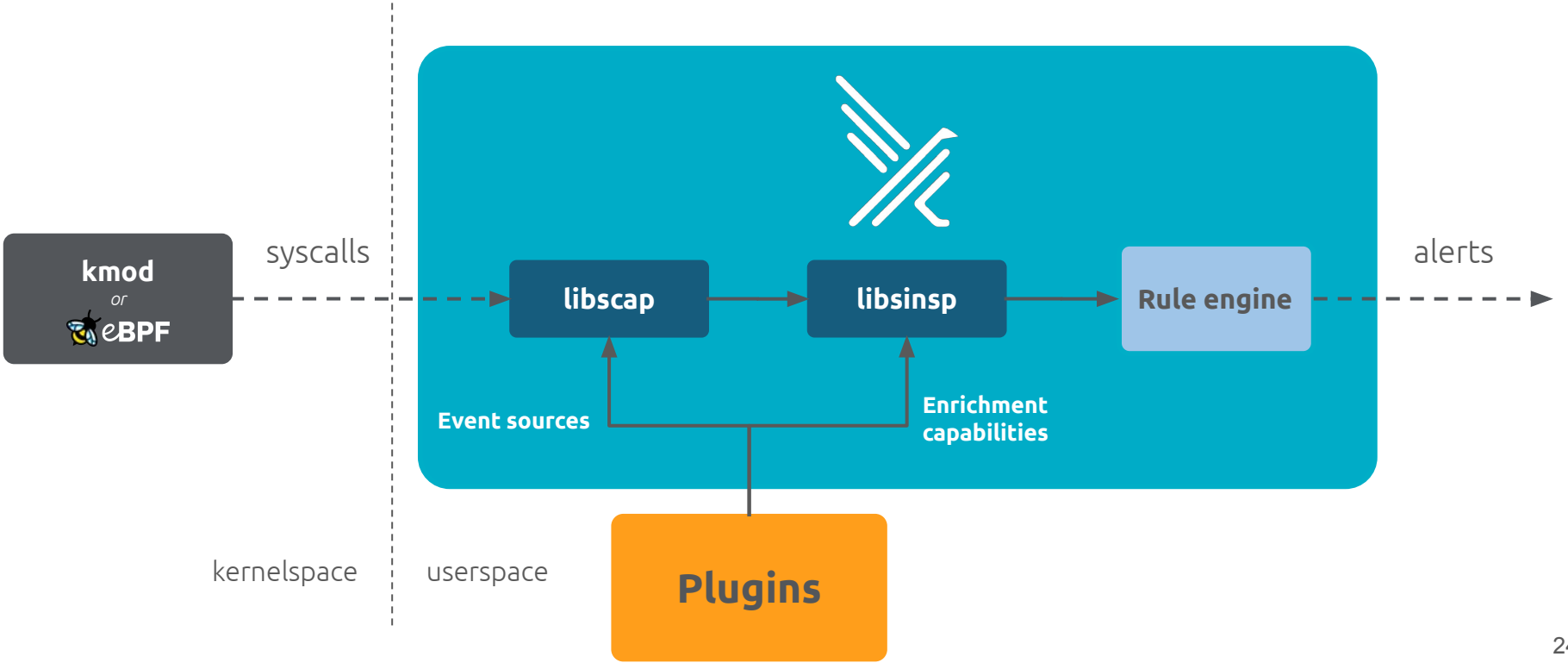
The Plugins

Plugins are **dynamic shared libraries**
which allow **Falco** to **collect** and **extract fields**
from **streams of events**

The Evolution: the Plugins



Falco's current architecture



DEMO

Getting started

```
helm repo add falcosecurity https://falcosecurity.github.io/charts

helm install falco falcosecurity/falco \
    --set falcosidekick.enabled=true \
    --set falcosidekick.webui.enabled=true \
    --set driver.kind=ebpf \
    -n falco --create-namespace
```

How to contribute

Get Started in Falco.org

Check out the Falco project in Github

Get involved in the Falco community

Develop a Plugin

Meet the maintainers on the Falco Slack

Follow @falco_org on Twitter

Thank you

Merci