Introducción práctica a Runtime Security con Falco

Miguel Hernández Boza
Vicente J. Jiménez Miras

Logistics

- Wi-fi: **FZC_AULAS** Password: **invitadas**
- Slides: bit.ly/41XCyGU
- Lab: bit.ly/3Lx73x9
- Slack
- Length: 90m
- Three series of lecture and lab iteration

Agenda

- Runtime security
- Falco engine
- Falco ecosystem
- Closing remarks

sysdig

Runtime security

sysdig

Once, there was a perimeter

You had a perimeter **guarded by a firewall**

Detecting intrusions was your breach indicator

sysdig

Now, there is no perimeter in the cloud

Cloud providers own external connections

Cloud is exposed to the outside world

You need to control access to services your team uses

You need to detect unusual activity

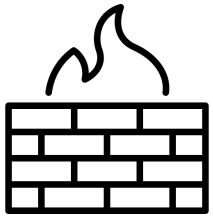
sysdig

Without a perimeter, a security camera is more important than a good lock

- Watch for changes that create security gaps
- Identify intruders and suspicious insider behavior
- Send an alert and take immediate action

sysdig

Security

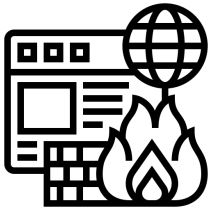


Firewalls

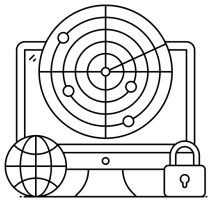
Authentication

EDR

Antivirus



Web Application Firewall



Threat Detection and Response

Vulnerability Scanning

sysdig

Containers

- Containers changed the game.
- Increased connectivity and surface attack.
- Less visibility.
- Containers complicated the security space.
- Old tools don't work anymore.

sysdig

Runtime security

- Runtime security protects workloads after containers have been deployed.
- Some common threats are:
  - outbound connections
  - spawned shells
  - cryptomining
- Kubernetes emerges as the main container orchestrator.
- Runtime security becomes a priority in cloud-native environments.

sysdig

Falco

Falco, the cloud-native **runtime security** project, is the de facto **Kubernetes threat detection engine**.

CNCF Incubation-Level Project
(applied to graduation Nov 2022)

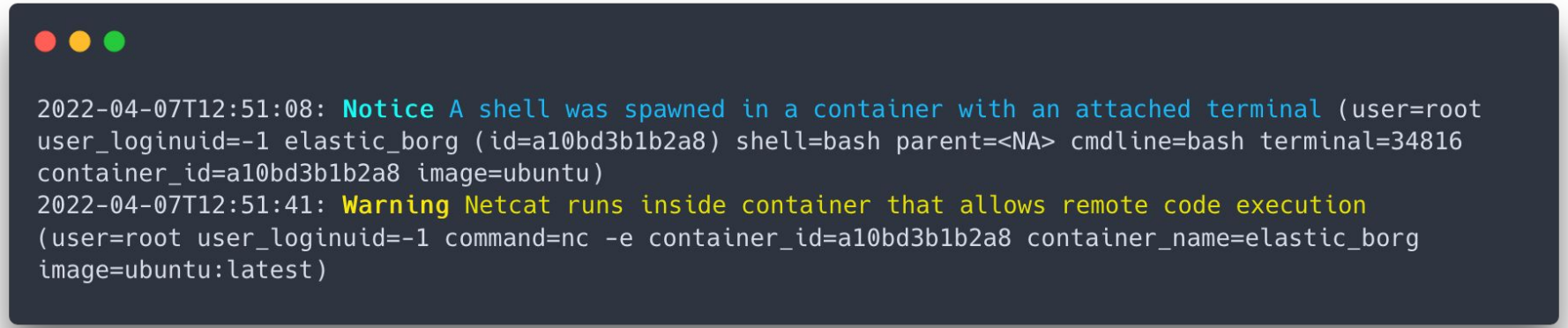
☆ **5.2k**

50M+ pulls

CLOUD NATIVE COMPUTING FOUNDATION

sysdig

Falco



sysdig

Set up Falco

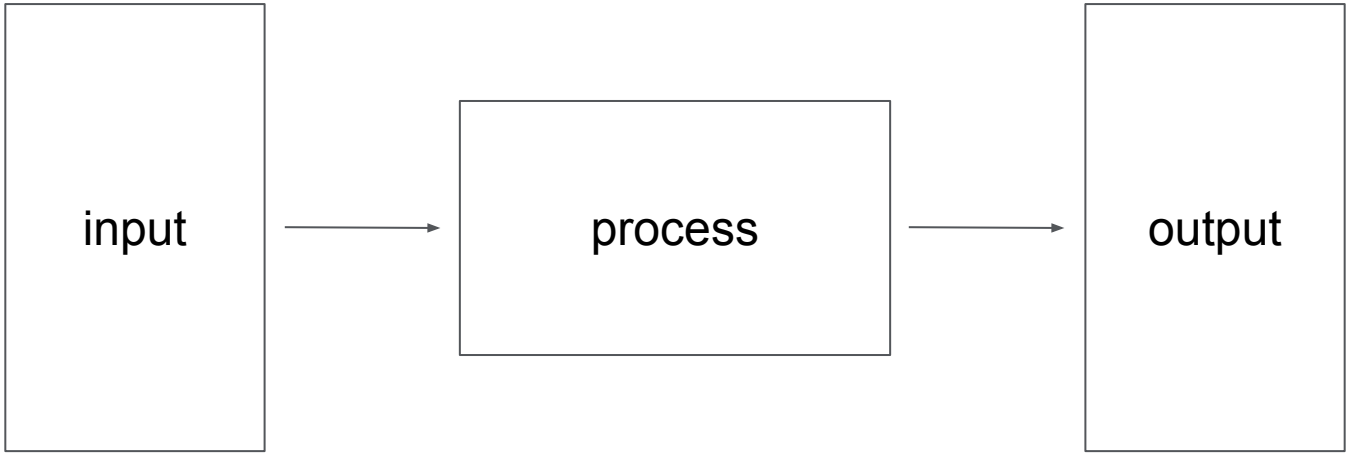
Lab time!

sysdig

Falco engine

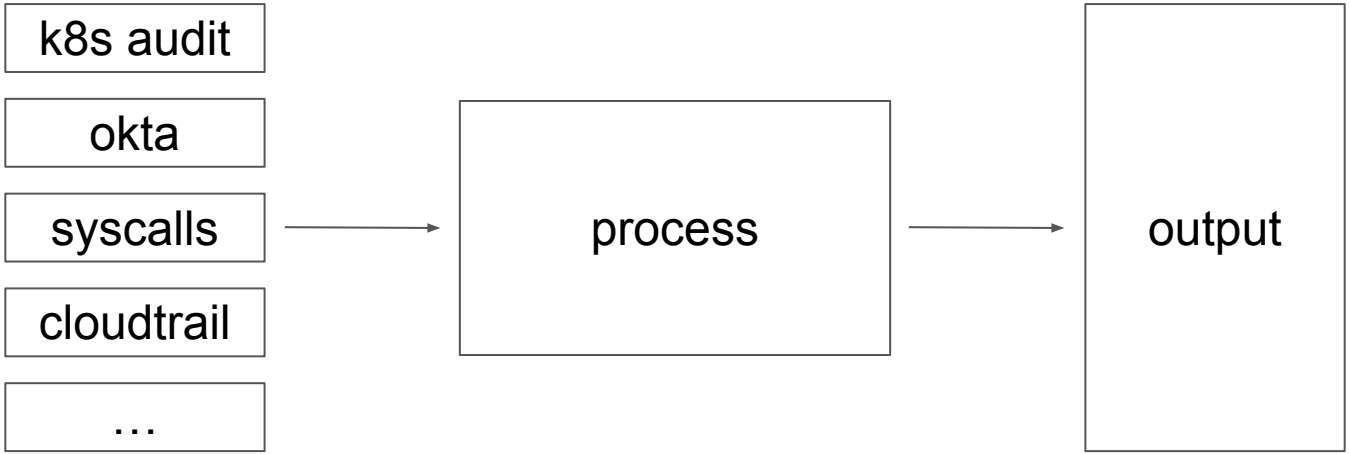
sysdig

Falco architecture



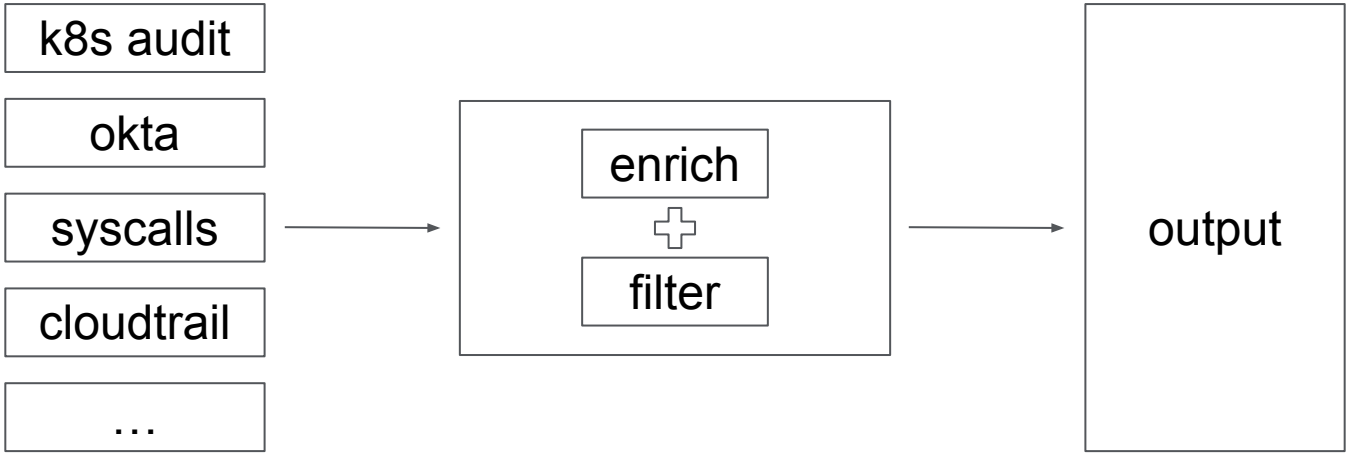
sysdig

Falco architecture



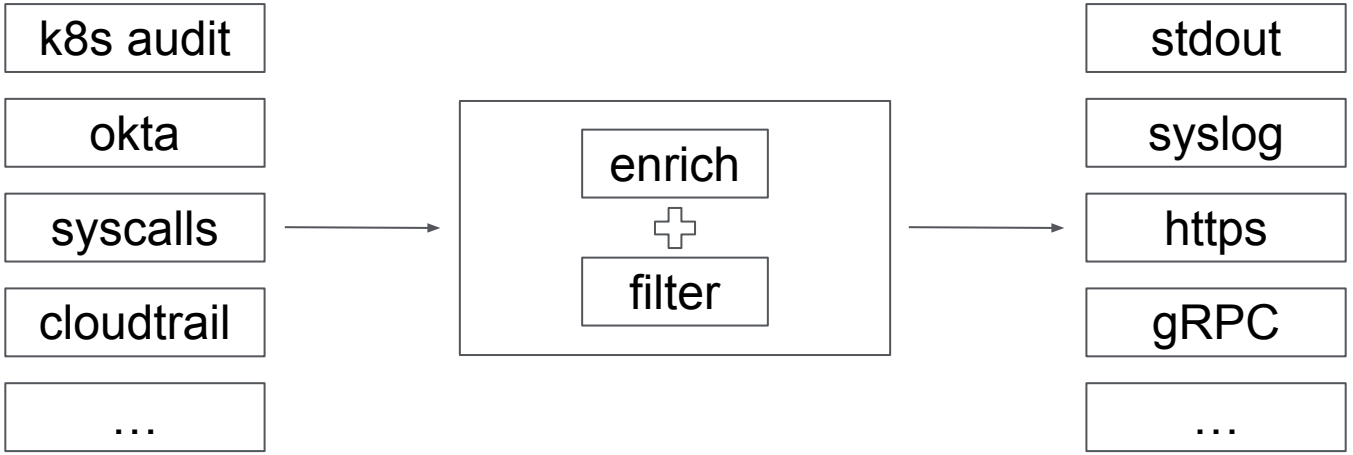
sysdig

Falco architecture



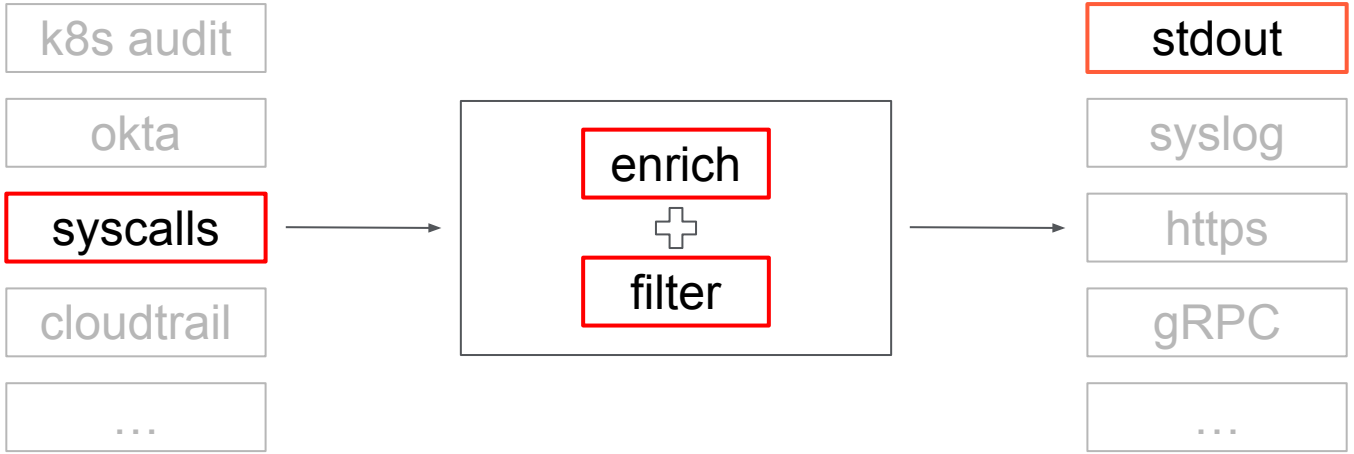
sysdig

Falco architecture



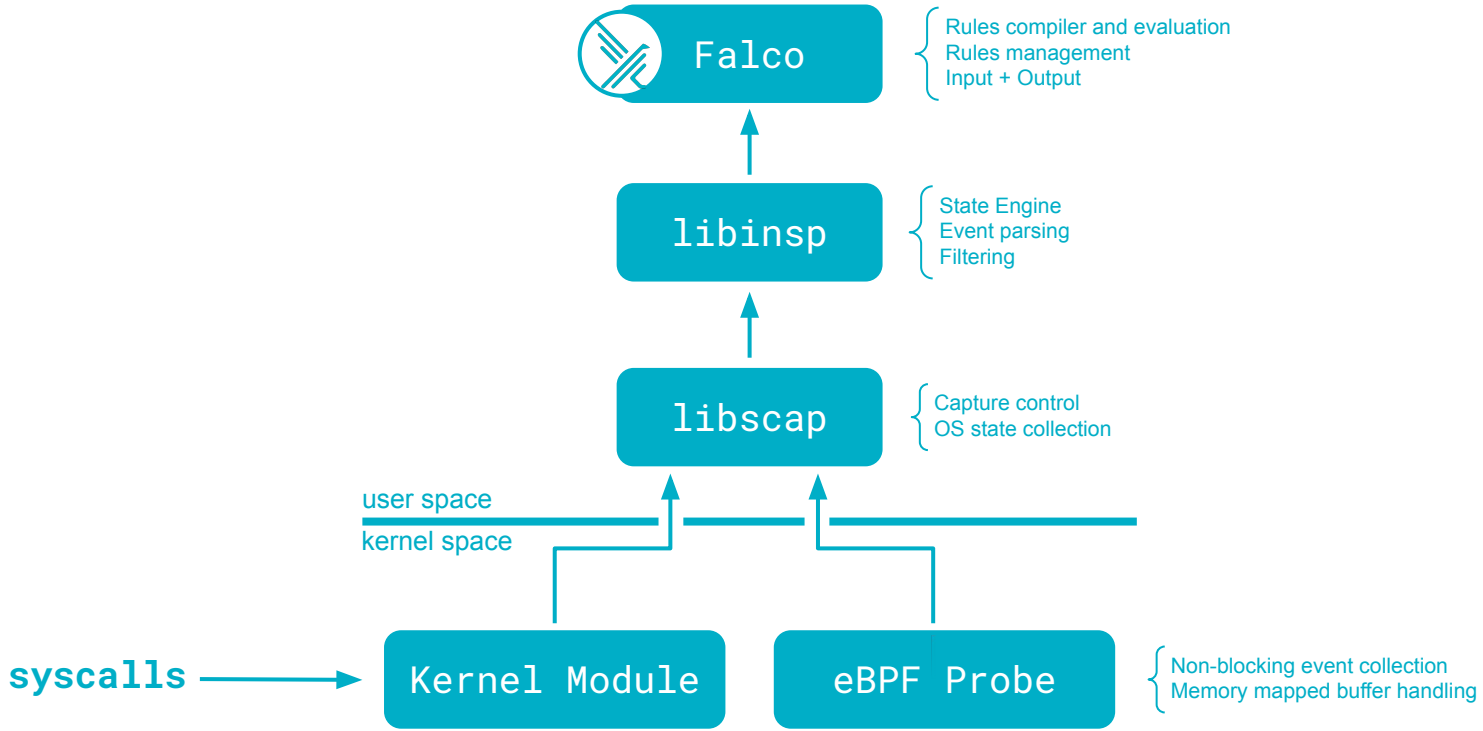
sysdig

Falco architecture



sysdig

Falco architecture



sysdig

eBPF

Kernel instrumentation made simple!

- Extended Berkeley Packet Filter.
- Extend the kernel capabilities safely and efficiently.
  - without changing kernel source code.
  - without loading kernel modules.
- Fast and safe in-kernel, register based, bytecode VM.

Modern, low-overhead, production-ready observability for monitoring and security in containers and Linux systems.

sysdig

Falco default rules

- Falco ships with more than 70 default rules:
  - Privilege escalation
  - R/W to sensitive directories
  - Executing shell
  - Execute SSH binaries
  - Mutating binaries
  - Creating symlinks
  - …

sysdig

Falco syntax

```
- 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]
```

https://falco.org/docs/reference/rules/rule-fields/

sysdig

Falco syntax

```
- rule: Terminal shell in container
  desc: A shell has been spawned in
  condition: >
    spawned_process and container ar
  output: >
    A shell was spawned in a contain
user_loginuid=%user.loginuid %contai
parent=%proc.pname cmdline=%proc.cmd
  priority: WARNING
  tags: [container, shell, mitre_exe
```

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

https://falco.org/docs/reference/rules/supported-fields/

Macros and Lists

- **Macros** allow you to define conditions and reuse them wherever you want.
- **Lists** help you organize your rules files with naming and segmentation.
- They have four main benefits:
  - code reuse
  - avoid long strings of conditions
  - rules are easier to understand
  - easier to extend

Falco default rules

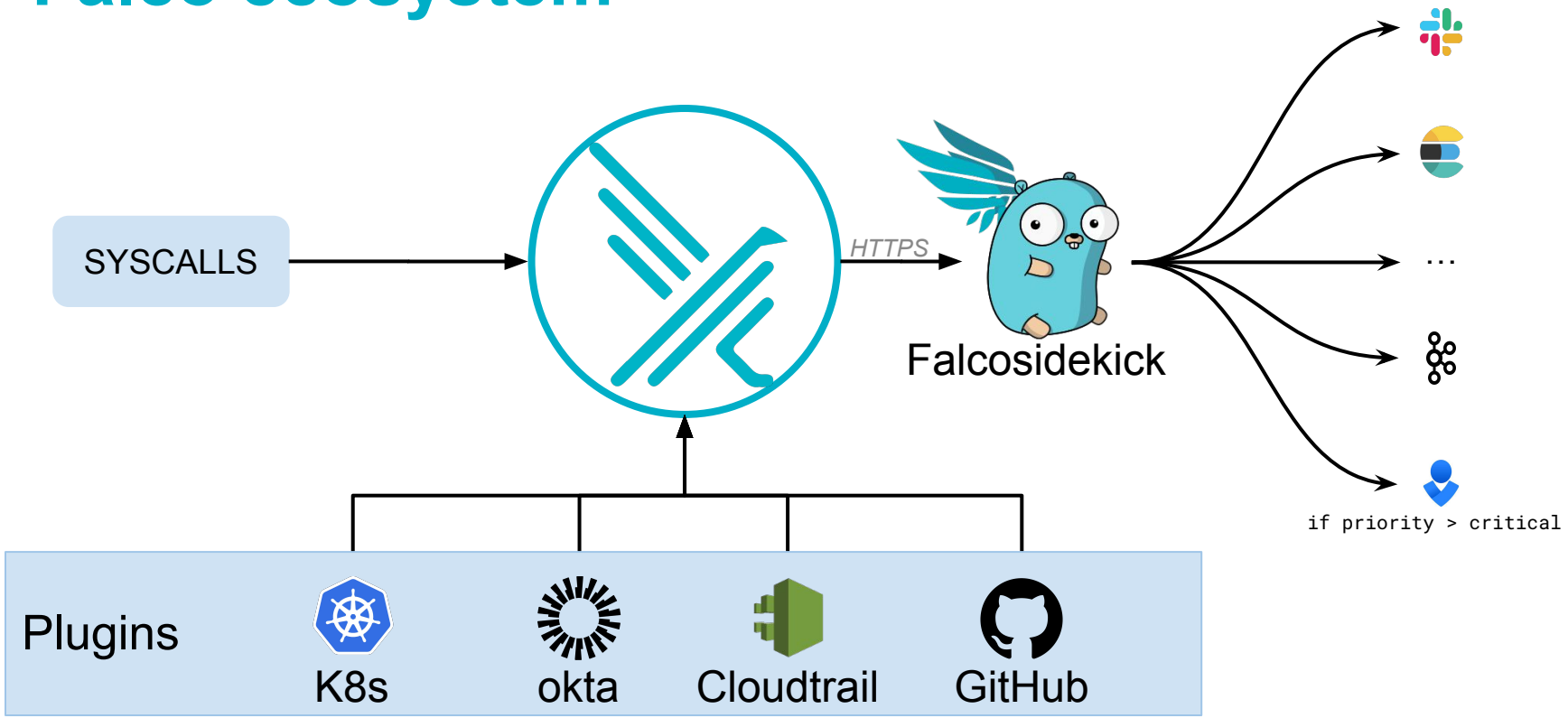
Lab time!

sysdig

Falco ecosystem

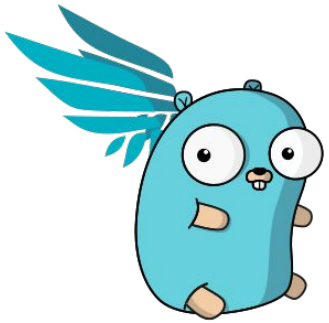
sysdig

Falco ecosystem



sysdig

Falcosidekick



chat

logs

queue/streaming

faas

metrics

alerting

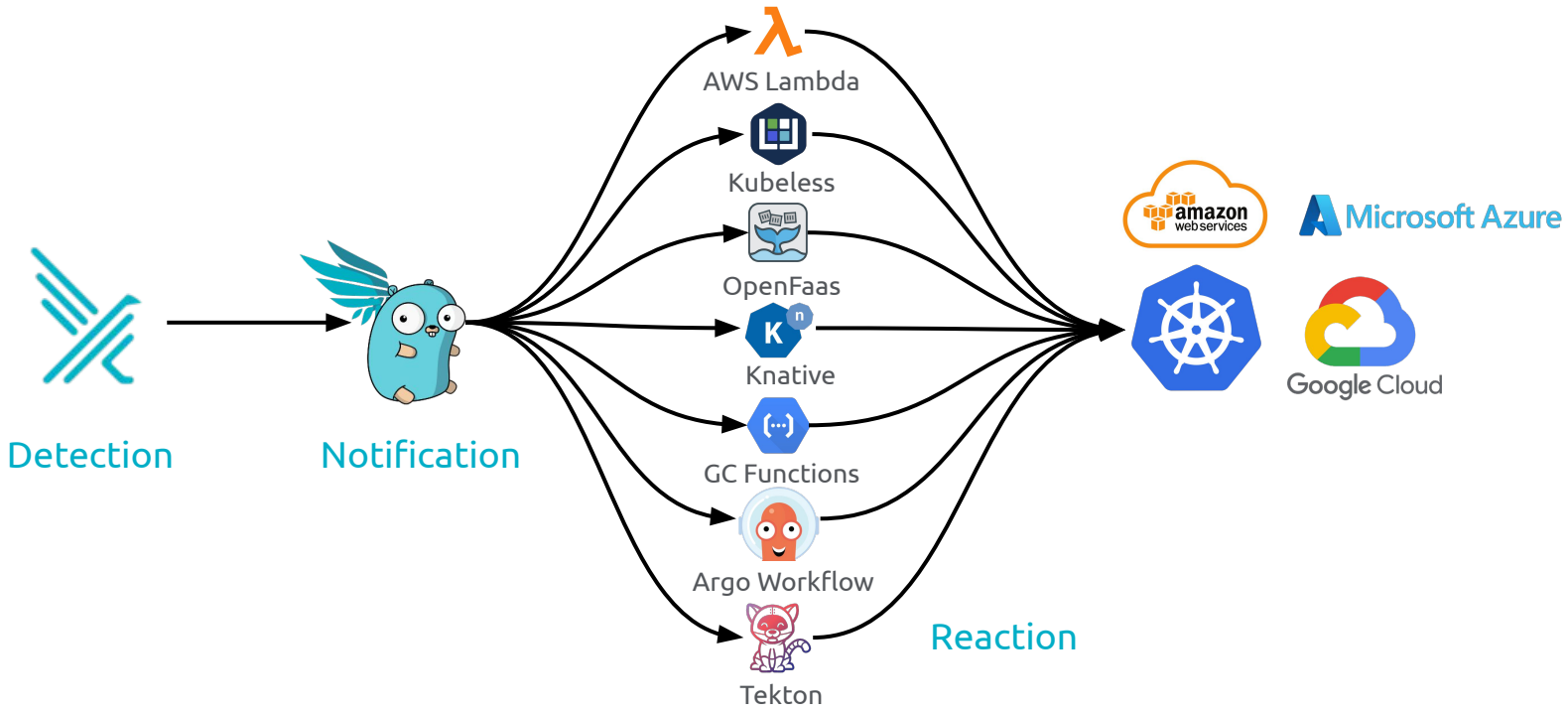
storage

and more ...

https://github.com/falcosecurity/falcosidekick

sysdig

React to events



sysdig

Falcosidekick install

Standalone:

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

helm install falcosidekick -n falco --set config.debug=true falcosecurity/falcosidekick
```

With Falco:

```
helm upgrade falco -n falco falcosecurity/falco \
  --set falcosidekick.enabled=true \
  --set falcosidekick.webui.enabled=true
```

Event generator

- Generates a variety of suspect actions that are detected by Falco rulesets.
- Good to test Falco rulesets:
  - syscalls
  - kubernetes audit
- Run it within Docker or Kubernetes,
  - as some commands might alter your system.

```
docker run -it --rm falcosecurity/event-generator run
```

```
helm install event-generator falcosecurity/event-generator \
  --namespace event-generator \
  --create-namespace \
  --set config.actions=""
```

https://github.com/falcosecurity/event-generator

sysdig

Falco ecosystem

Lab time!

sysdig

Closing remarks

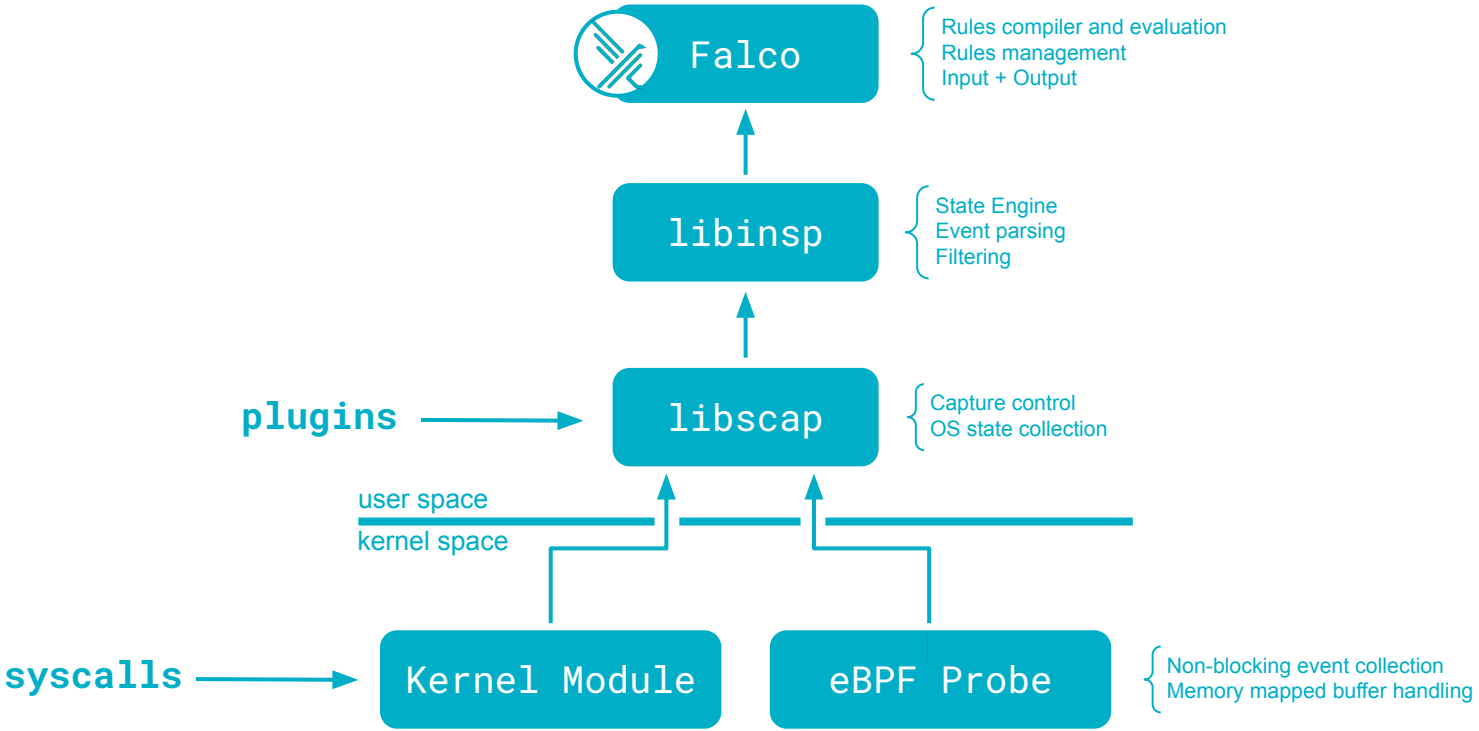
Runtime security

Guard perimeter

Detect unusual activities

sysdig

Falco architecture

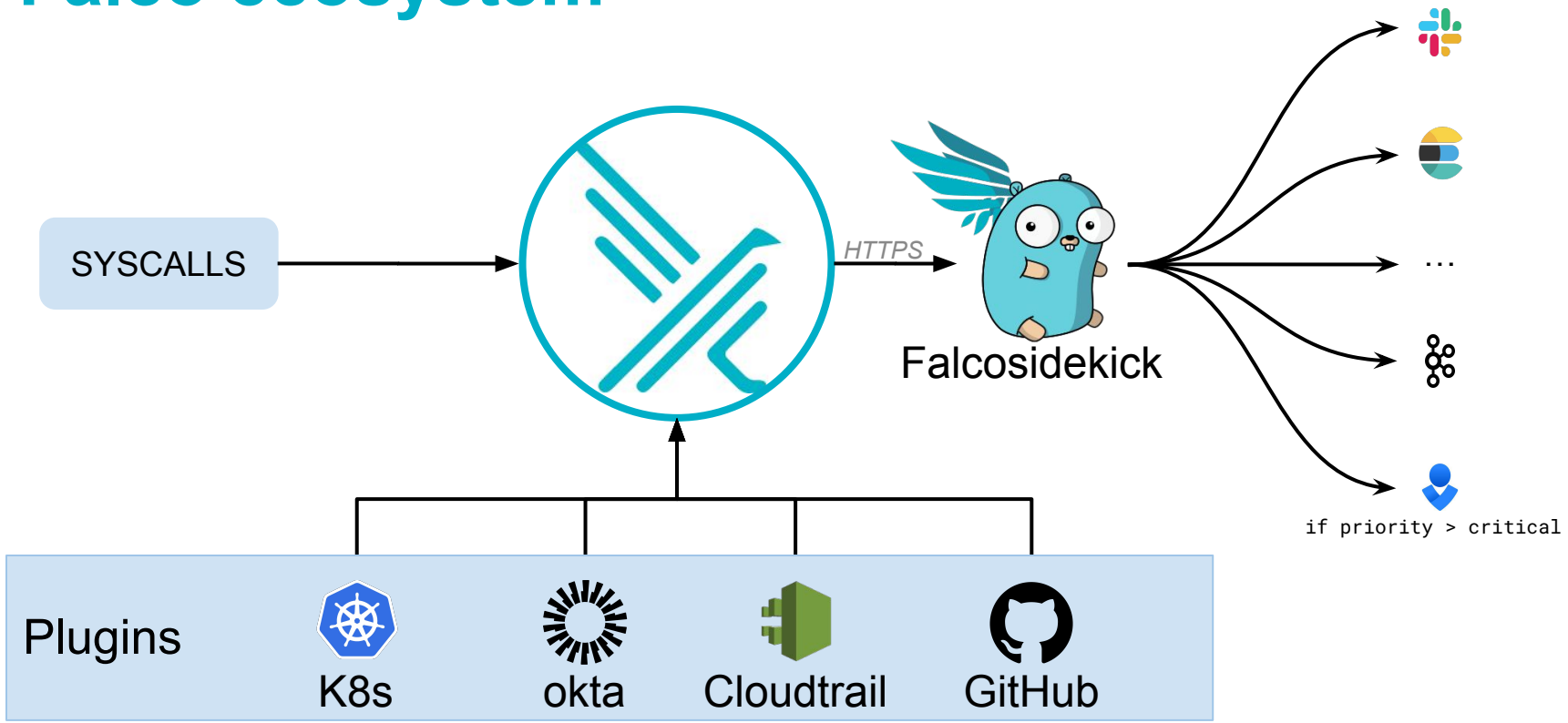


sysdig

Falco rules

```
- 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 ecosystem



sysdig

Reference

"Practical Cloud Native Security with Falco", O'Reilly book

https://falco.org/docs

https://falco.org/training

https://falco.org/blog/extend-falco-outputs-with-falcosidekick/

sysdig

Survey

https://t.ly/-pMv

sysdig


sysdi
Seeing is Secu