Getting Started with Runtime Security using Falco

Loris Degioanni, CTO and founder, Sysdig

Falco

Loris Degioanni
CTO, Founder
Sysdig
@lorisdegio

sysdig

Open by design

- Founded by Wireshark co-creator
- Contributed Falco to CNCF
- Supported open-source sysdig (10M+ downloads)

Ecosystem integration

- Cloud-native security and monitoring
- Provides visibility and control for secure operations

Strong momentum

- Customer expansion mirrors cloud-native adoption
- Trusted by the largest enterprises

Falco

Agenda

- Runtime security overview
- Comparing runtime security technologies
- Falco overview and rules
- Demo
- History and roadmap

Falco

Runtime Security Overview

Falco

Why runtime security?

Detect malicious behavior

- ❑ Drift from image scanned
- ❑ Only present in runtime
- ❑ Unknown/0-day threats

Incident response

Alert on detections right when they happen

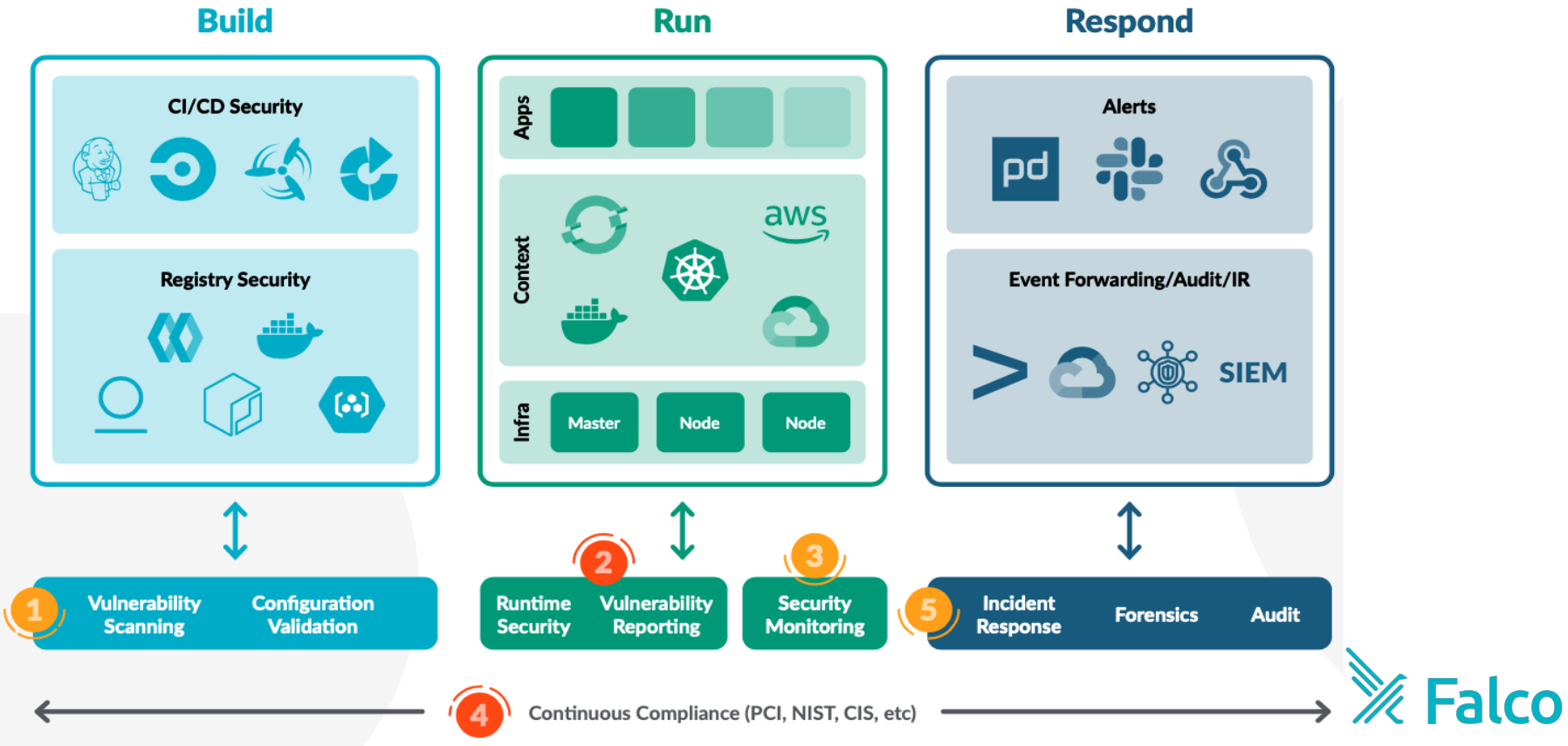
Forensics

Audit activity and gain knowledge of extent

Compliance with security frameworks from PCI, NIST, SOC

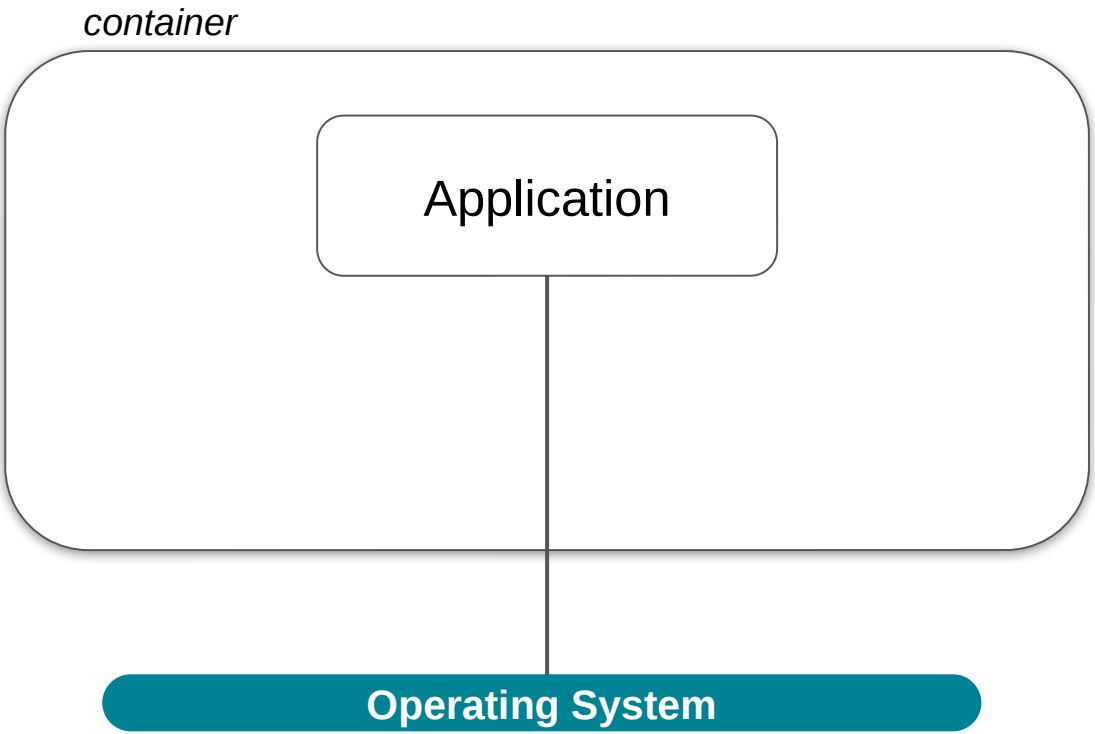
Falco

How runtime security fits into the workflow



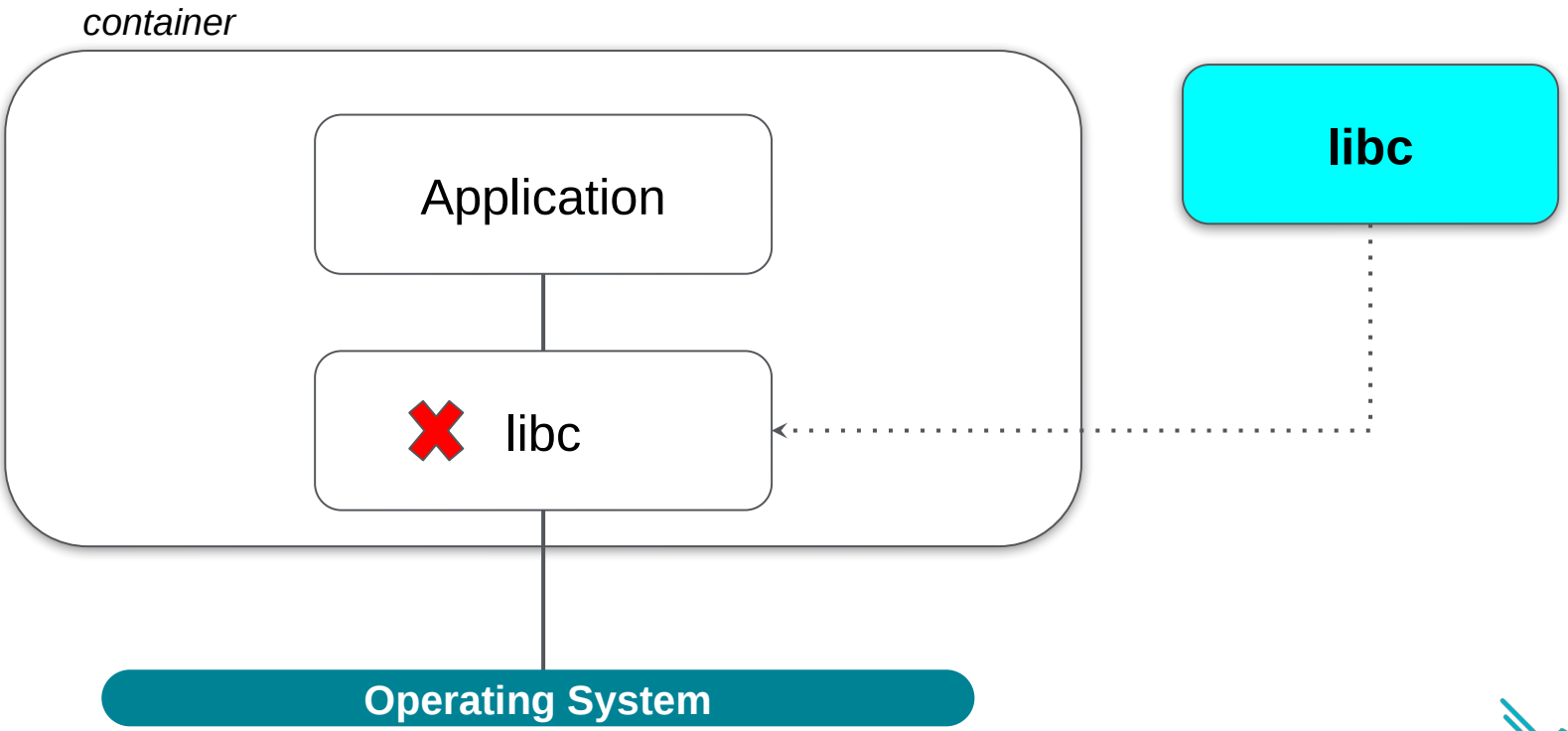
Comparing Runtime Security Collection Technologies

Falco



Falco

LD_PRELOAD



Falco

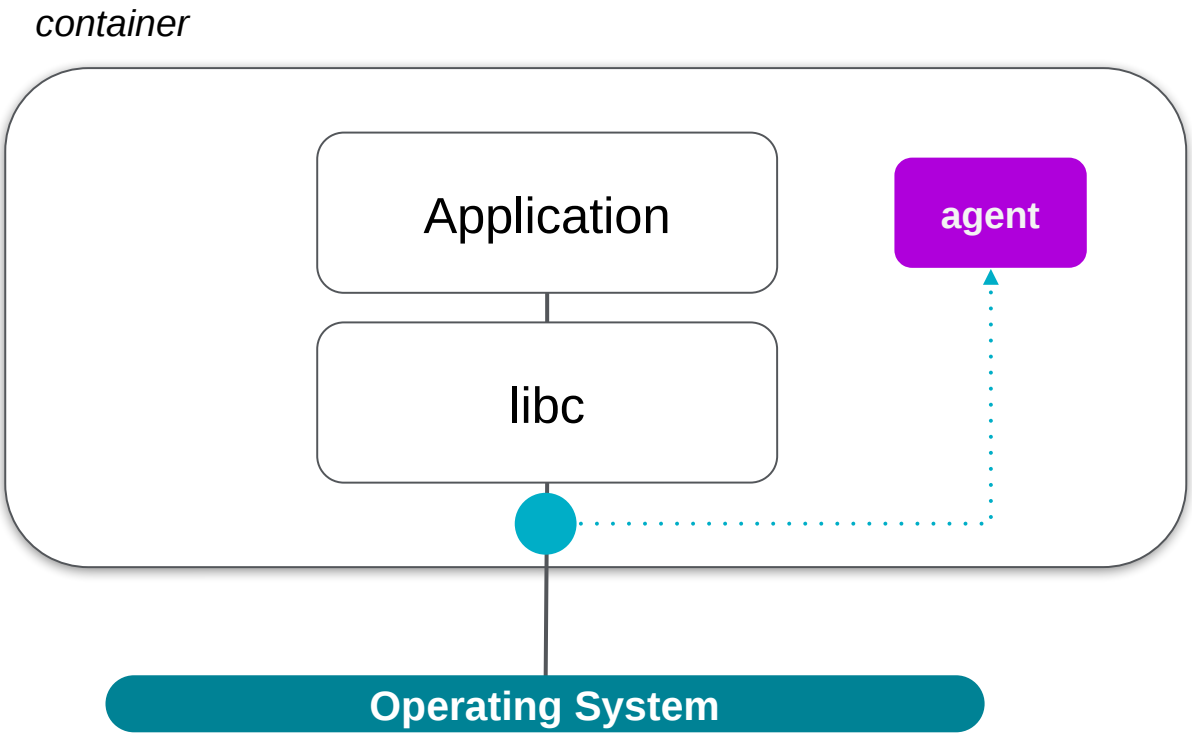
Pros vs Cons: LD_PRELOAD

- Can be applied to serverless and non-privileged environments

- Not accurate, as it is an out of kernel based instrumentation
- Can crash the target process
- Limited support (e.g. doesn’t work with Go)
- Requires instrumenting every container

Falco

ptrace



Falco

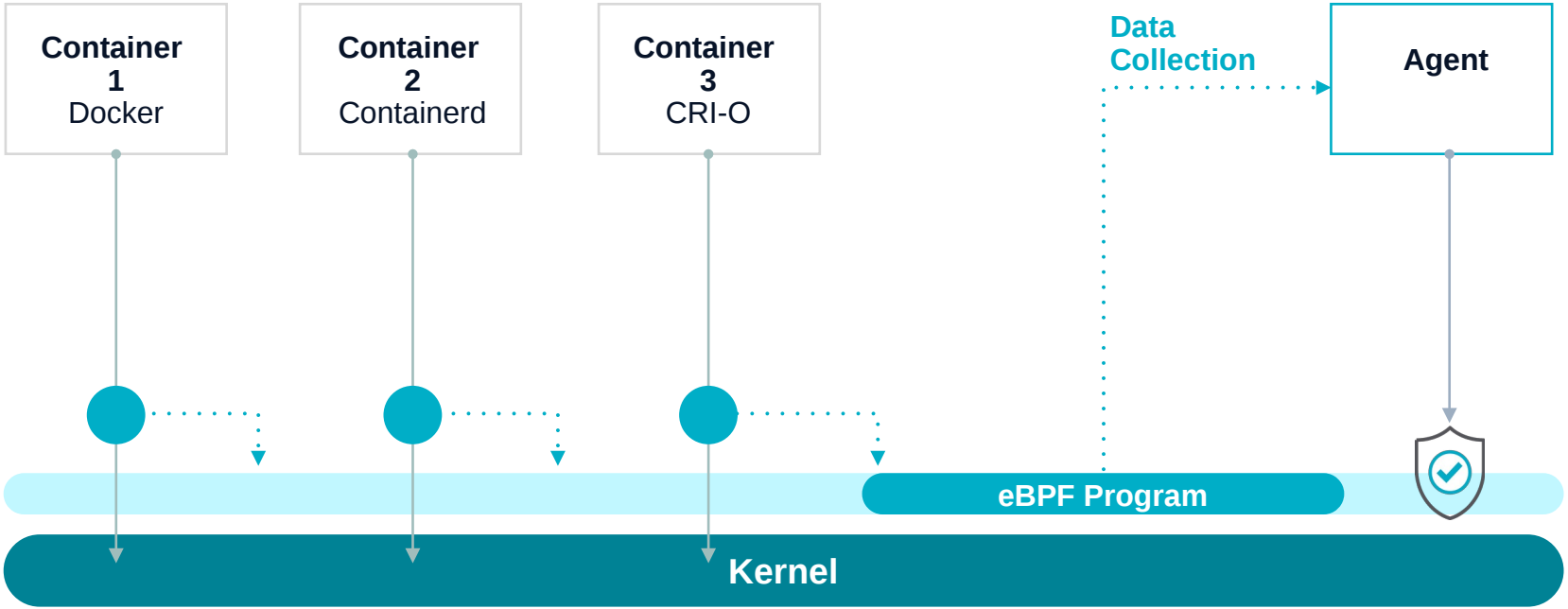
Pros vs Cons: ptrace

- Accurate
- Language and stack independent
- Safer than LD_PRELOAD b/c it leverages the Operating System

- Inefficient
- Requires instrumenting every container

Falco

Kernel based instrumentation



Falco

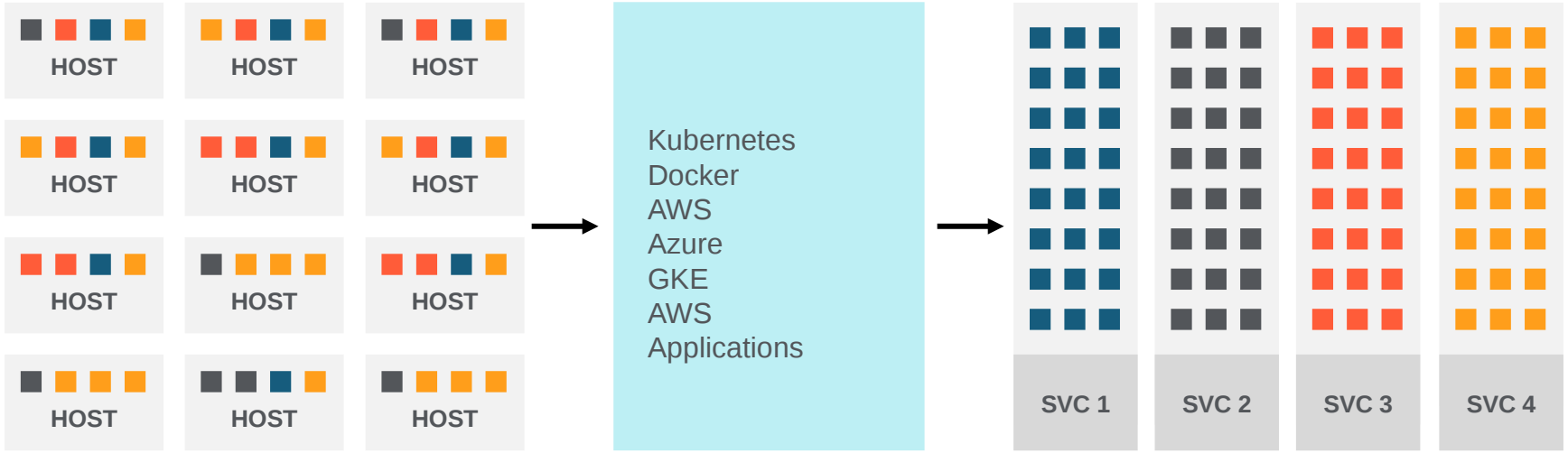
Pros vs Cons: Kernel instrumentation

- Greater accuracy
- Performance without compromise
- Highly scalable (doesn’t need to run in every container)

- Limited ability to collect data in serverless environments (Fargate, Lambda)

Falco

Enriched Context via Cloud/K8s Metadata instrumentation



Falco

Falco: Open-source runtime security engine

Falco

What is Falco?

- Runtime security engine
- Observability from the kernel
- Built on kmod/eBPF
- Consumable / Modular

github.com/falcosecurity

Cloud Native Runtime Security

Container-centric Runtime Security

- **Kernel Events** as source of truth
- **Enriched** with metadata
- **Assert against rules** at runtime
- **Alert/Alarm** during violation events

CNCF Incubating Project

- **Jan 2016** First Commit
- **Oct 2018** Donated CNCF
- **Dec 2019** Promoted to Incubation

Core Principles

Community Driven

- Deeply integrated with Kubernetes and CNCF communities
- Decision making in the open
- Integrations built and supported by the community

End-Users

- Consumers of Falco and the Falco ecosystem
- Contributors to the Falco ecosystem
- PCI Compliance, SOC2, HIPAA
- Observability, CVEs, Exploits, 0Day events

Vendors

- Sysdig Secure (scale)
- SumoLogic
- SkyScanner (scale)
- PCI Compliance, HIPAA
- Kubernetes Audit
- Application Integration

Falco

Falco in Production at shopify

Production Environment

- Running Falco in PCI-compliant environment on AWS EC2 / EKS
- Every day, Falco protects $100-150M in Shopify transactions
- Falco enabled PCI-compliant lift and shift to AWS from data center

Involvement with Open Source

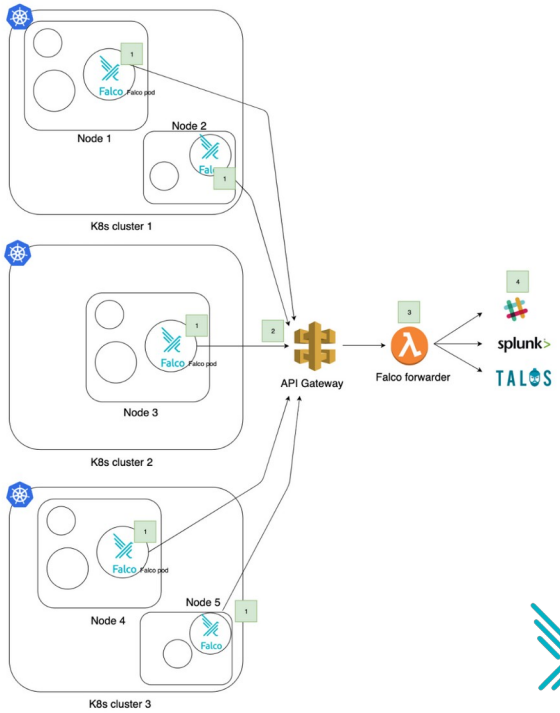
- Integrated with Falco community and maintainers
- Deploys Falco artifacts regularly as released
- Presented production use case at KubeCon Europe Virtual keynote (see the replay on CNCF's YouTube channel)

Falco in Production at Skyscanner

Production Environment

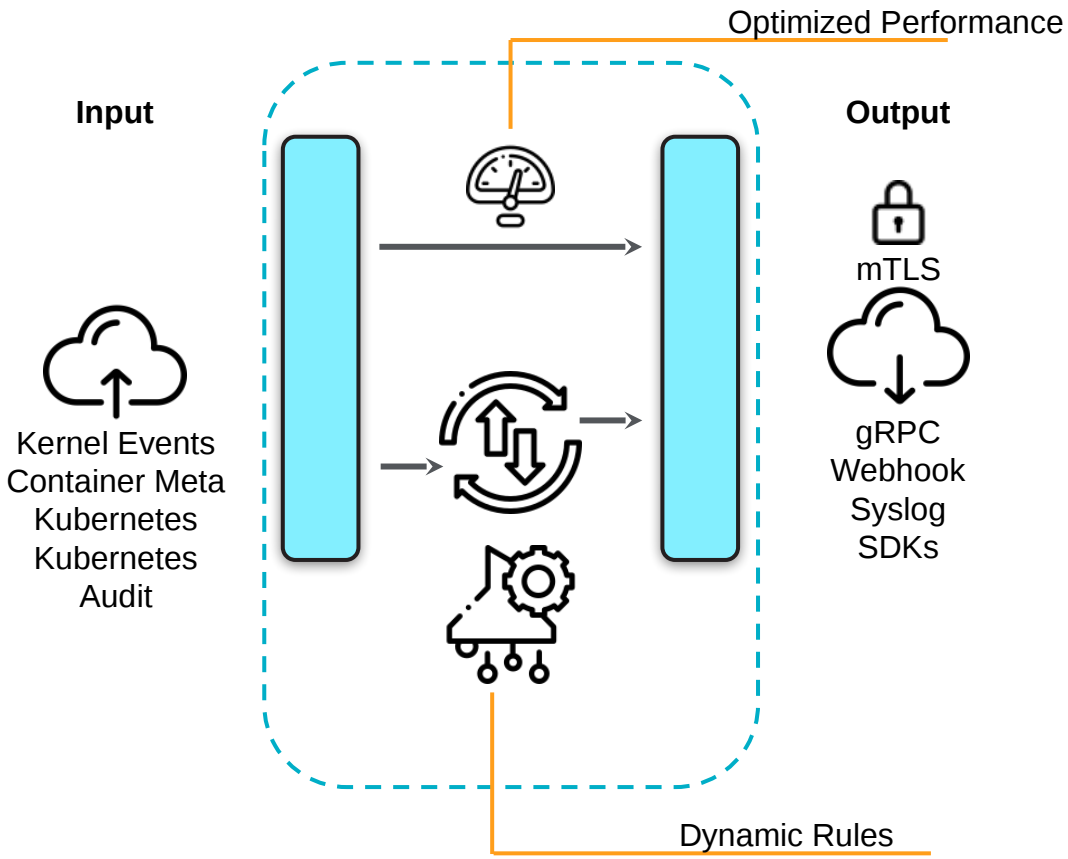
- Running Falco with 2,000+ nodes across 30 clusters powering 160+ services on AWS Lambda
- Core requirements: detect malicious activity at scale without hindering performance, integrate with service mapping tools
- Read their use case - Medium, 1/29/20
  https://medium.com/@SkyscannerEng/kubernetes-security-monitoring-at-scale-with-sysdig-falco-a60cfdb0f67a

High Level Architecture



Falco

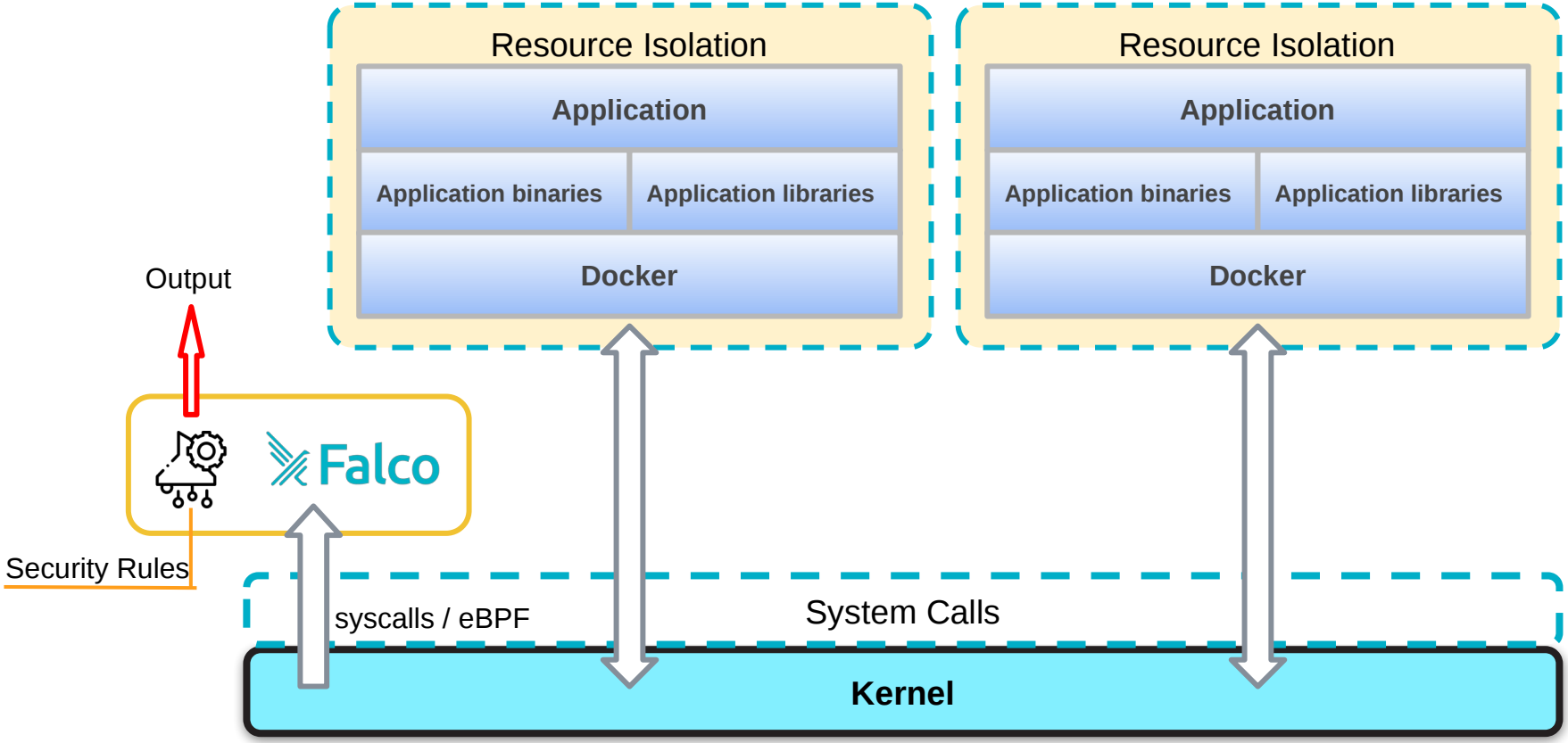
Falco Architecture

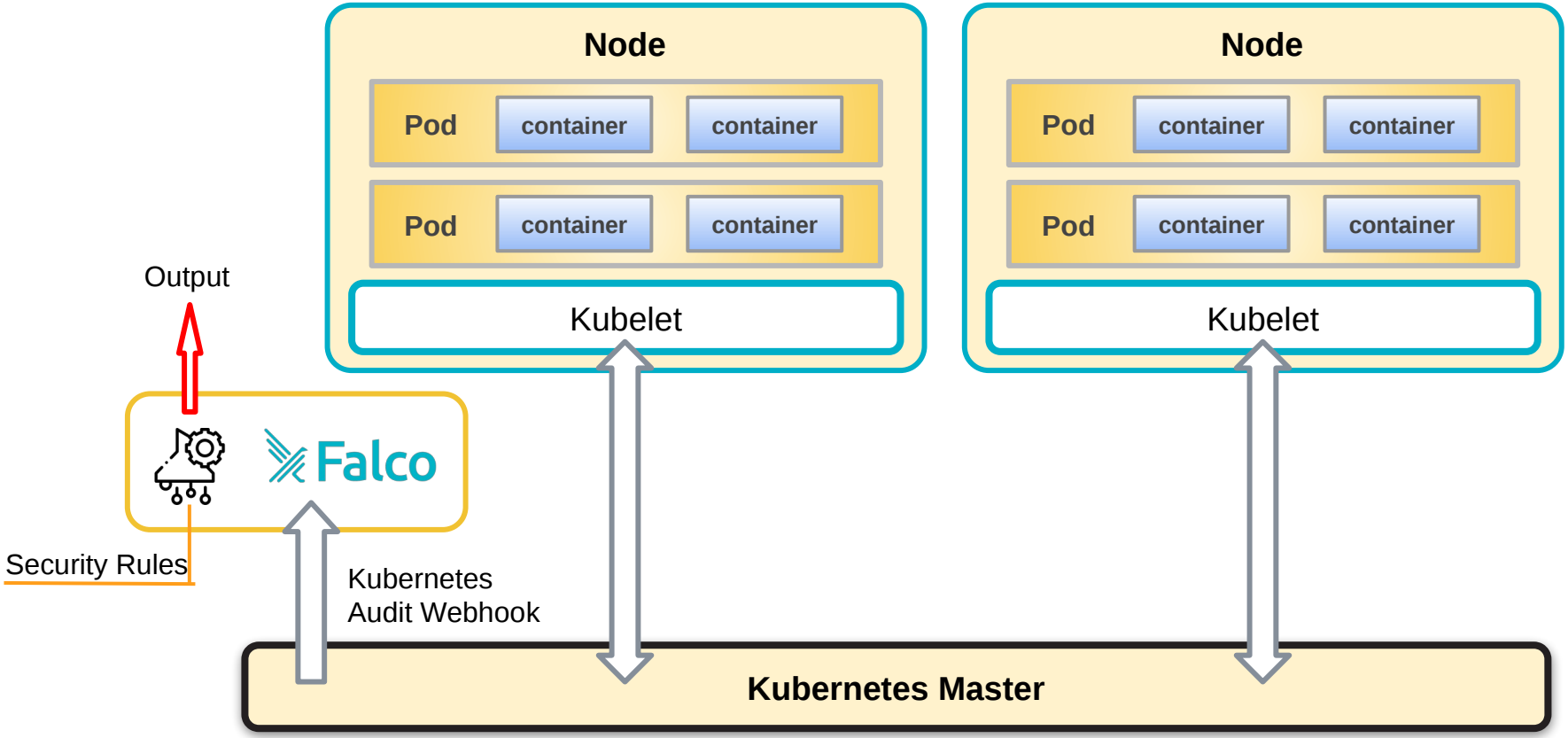


API Layers

Integration
Hooks

Falco


Resource Isolation
Application
Application binaries
Application libraries
Docker
Resource Isolation
Application
Application binaries
Application libraries
Docker
Output
Falco
Security Rules
syscalls / eBPF
System Calls
Kernel


Node
Pod
container
container
Pod
container
container
Kubelet
Node
Pod
container
container
Pod
container
container
Kubelet
Output
Falco
Security Rules
Kubernetes
Audit Webhook
Kubernetes Master

Falco Rules

Rule Examples

A shell is run in a container	container.id != host and proc.name = bash
Overwrite system binaries	fd.directory in (/bin, /sbin, /usr/bin, /usr/sbin) and write
Container namespace change	evt.type = setns and not proc.name in (docker, sysdig)
Non-device files written in /dev	(evt.type = create or evt.arg.flags contains O_CREAT) and proc.name != blkid and fd.directory = /dev and fd.name != /dev/null
Process tries to access camera	evt.type = open and fd.name = /dev/video0 and not proc.name in (skype, webex)

Falco

Sample Rule: DB spawns a shell

```
- rule: Database spawns a shell
  condition: >
    proc.pname in (db_server_binaries) and spawned_process
    and not proc.name in (db_server_binaries)
    and not postgres_running_wal_e
  output: >
    Database-related program spawned process other than itself
    (user=%user.name program=%proc.cmdline parent=%proc.pname)
  source: syscall
  desc: >
    Database-server program spawned a new process other than itself.
    This shouldn't occur and is a follow on from some SQL injection attacks.
  priority: WARNING
  tags: [process, database]
```

Falco

Lists and Macros

```
- list: sensitive_file_names
  items: [/etc/shadow, /etc/sudoers,
/etc/pam.conf, /etc/security/pwquality.conf]

- macro: sensitive_files
  condition: >
    fd.name startswith /etc and
    (fd.name in (sensitive_file_names)
     or fd.directory in (/etc/sudoers.d,
/etc/pam.d))
```

Falco

Falco MITRE Rule Matrix

Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Exfiltration
DB program spawned process	Modify Shell Configuration File	Launch Privileged Container	Clear Log Activities	Read sensitive file trusted after startup	Read Shell Configuration File	Launch Privileged Container	System procs network activity
Run shell untrusted	Schedule Cron Jobs	Non sudo setuid	Delete Bash History	Read sensitive file untrusted	Read ssh information	Launch Sensitive Mount Container	Interpreted procs inbound network
Terminal shell in container	Update Package Repository			Search Private Keys or Passwords	Read sensitive file untrusted	Launch Disallowed Container	Interpreted procs outbound network
Netcat Remote Code Execution in Container	Write below binary dir Write below monitored dir				Contact K8S API Server From Container		Unexpected UDP Traffic
	Write below etc Write below root Write below rpm database				Launch Suspicious Network Tool in Container		Launch Suspicious Network Tool in Container
	Modify binary dirs Mkdir binary dirs				Launch Suspicious Network Tool on Host		Launch Suspicious Network Tool on Host
	User mgmt binaries						
	Create files below dev						
	Launch Package Management Process in Container						
	Remove Bulk Data from Disk Set						
	Create Hidden Files or Directories						
	Setuid or Setgid bit						

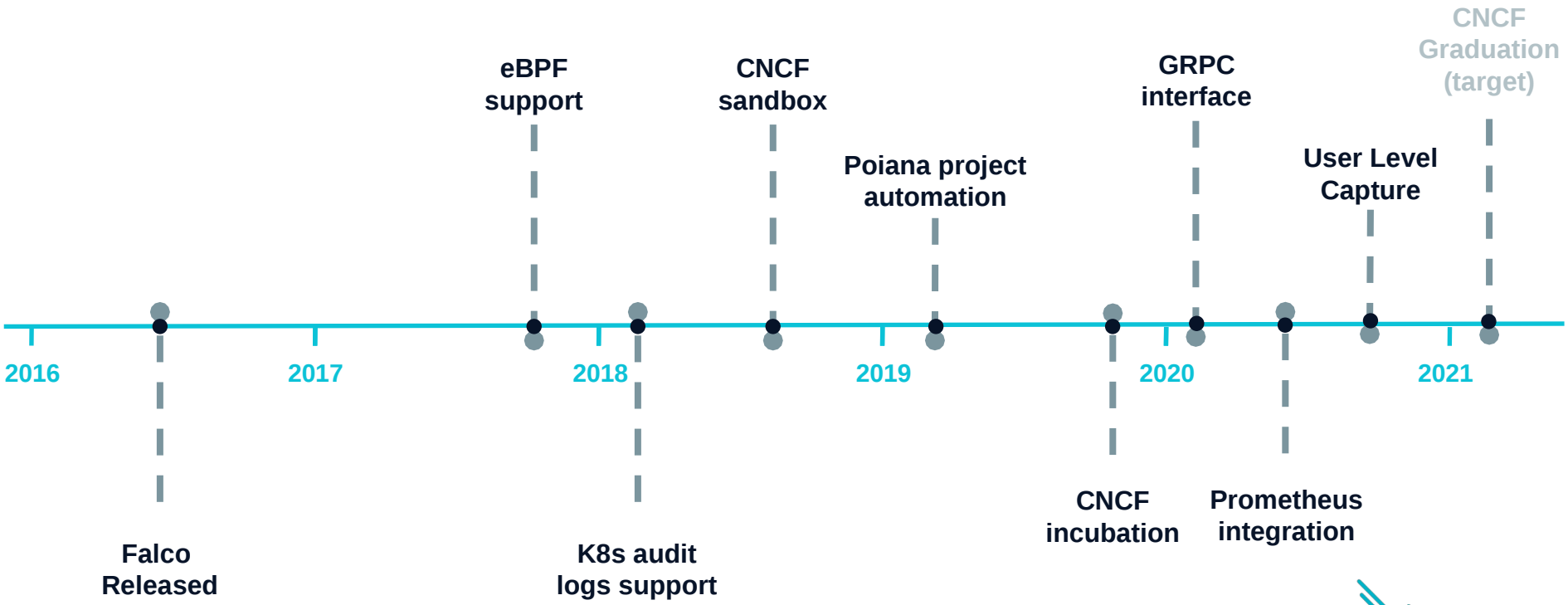
More info at:
https://sysdig.com/blog/mitre-attck-framework-for-container-runtime-security-with-sysdig-falco

Demo

Falco Releases and Roadmap

Falco

Falco's History



Falco

Recently Added Features

- Rules Improvements
  - PSPs, MITRE framework, cryptoming
- gRPC input/output interface
- Integrations
  - Prometheus, Slack, ElasticSearch, AWS Lambda
- Helm Chart
- ptrace instrumentation

https://www.cncf.io/blog/2020/08/17/falco-update-whats-new-in-falco-0-25/

Falco

Roadmap

- Expand our community by delighting our users
- Lowering the barrier
  - Stability
  - Ease of deployment
  - Performance
- Integrations
- Platform coverage
  - AWS Fargate

Falco

Resources

The Falco project

falco.org

github.com/falcosecurity/falco

Try it yourself

https://falco.org/docs/installation/

Join the community

https://github.com/falcosecurity/community

Questions?

Falco