

# Falco

Cloud native security & observability monitoring

Dive in! →



# Petr Michalec

Speaker

Pracuje jako SRE v F5 Czech Republic s.r.o.

- 147+ services, ~30 regional datacenters
- (Volterra.io, Mirantis, IBM, ...)

n(vi)m lover • maker • golfer • quad fpv pilot

- [twitter.com/epcim](https://twitter.com/epcim)
- [github.com/epcim](https://github.com/epcim), [Gists](#)



# Security monitoring

SecOps coverage

- detect intrusion
- prevent intrusion
- configuration enforcement
- audit all critical events
- compliance checks & reporting
- ...



# Security monitoring

Purpose & requirements

- ...
- Compliance with security standards:
  - PCI DSS, CIS, GDPR
  - HIPPA, NIST, FIPS, FedRAMP
  - evidence collection
  - data availability and traceability
  - measurement and support resources
  - tools, policies, processes, reporting

# Traditional approach ~2015

## Topics and tools

- network IDS, WAF
  - filesystem integrity
  - system/service/user audit/logs
  - data access & encryption
  - security threats mitigation
    - CVEs, vulnerabilities
  - malicious activity detection
  - auditd, aide
  - PAM, SELinux, AppArmour
  - OSSEC, OpenSCAP, Inspec, ...
  - Enterprise SIEM tools
- SaaS approaching (Snyk, Whitesource, Graylog, Thread stack, ...)

# New challenges

Microservices

Containers

- namespace isolation
  - container images
  - 3rd party libraries

Distributed applications

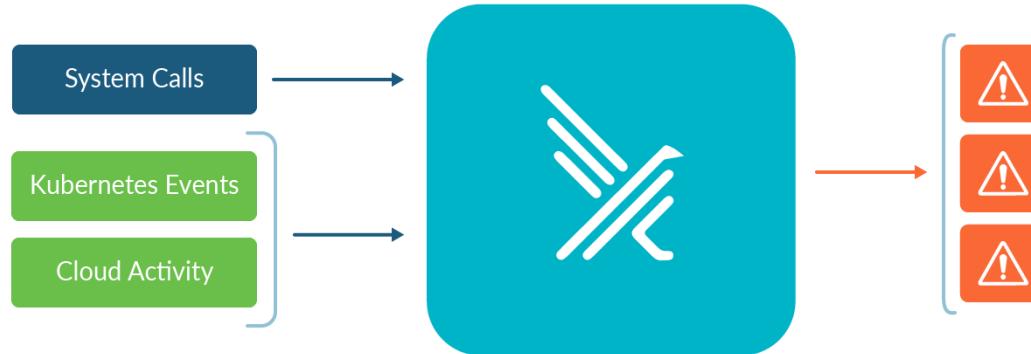
- cloud environments ^n
- 3rd party base images for OS
- 3rd party accessing servers

Distributed data



# Falco

May 16, 2016 - Sysdig introducing open source, behavioral security



[Falco.org](https://Falco.org) runtime security project detecting unexpected behavior, intrusions, and data theft in real time!

# Overview

Falco

- Kernel integration
- Highly granular rules to check for activities involving
  - file and network activity
  - process execution
  - IPC, ...
- Real-time metrics & notification when these rules are violated
- Less complex & faster

*There are a million ways a burglar can break into your home, but once they do they're going to steal your jewelry.*

...

*You only need to detect the things that an attacker does once they have access to a system, rather than all the ways an attacker can gain access to a system.*

# Comparison to existing approaches

Falco

File integrity monitoring: (checksums)

Watch for any OS activity that is writing to a file of interest, and be alerted in real-time.

Network monitoring (signatures)

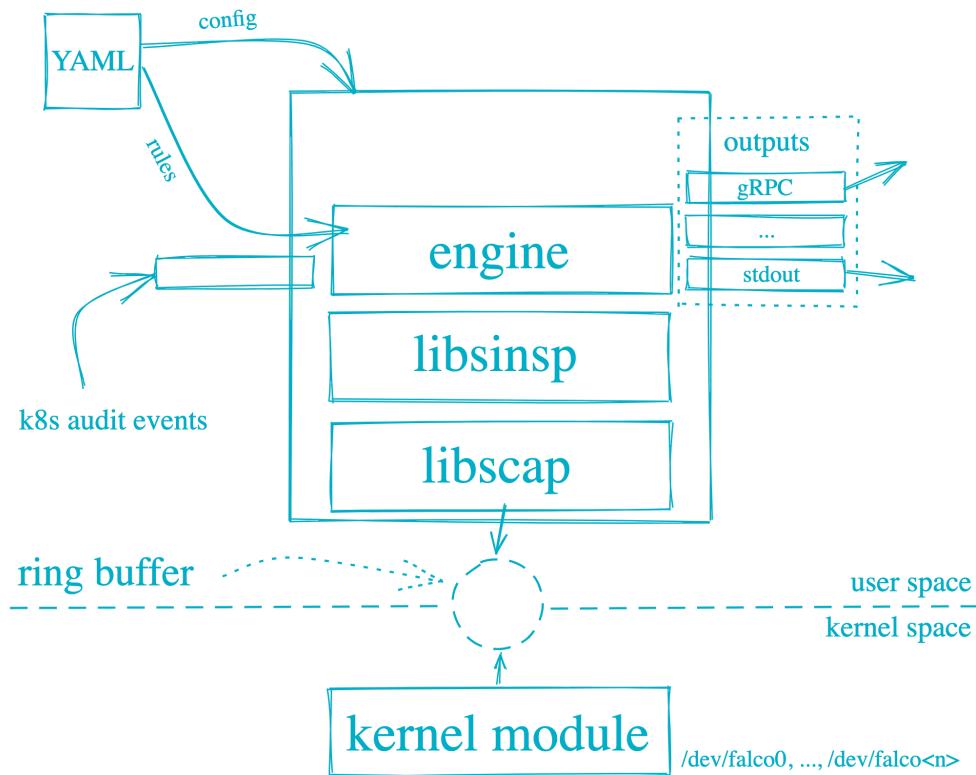
Falco see I/O "from the inside" with an immediate correlation between applications and traffic.

Linux has multiple security modules ~ advanced access control systems with sophisticated policies and concepts. As a result, understanding and configuring them is a rather complex undertaking.

Falco is far simpler to understand and configure, "detection-only".

# How it works

Falco architecture



# Kernel integration

Falco

- eBPF
- Built-in
- Module `falco.ko` (w/ DKMS)
- Userspace instrumentation (based on PTRACE2)

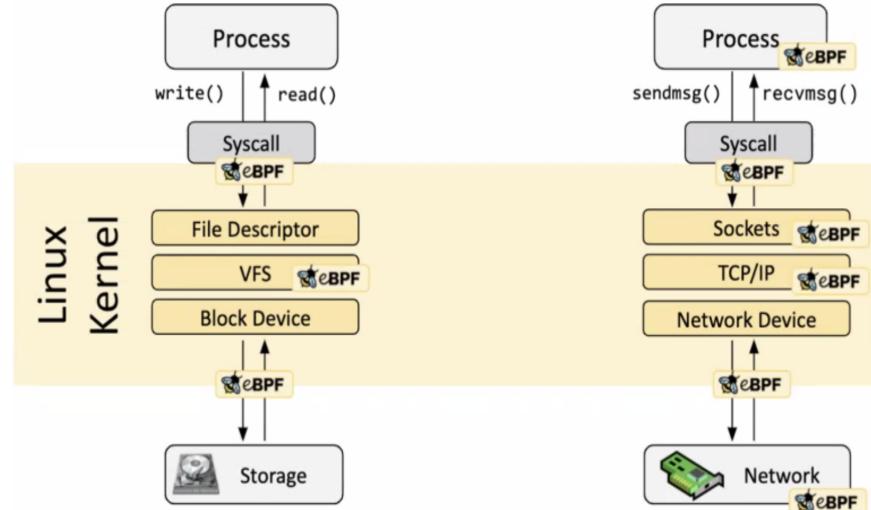
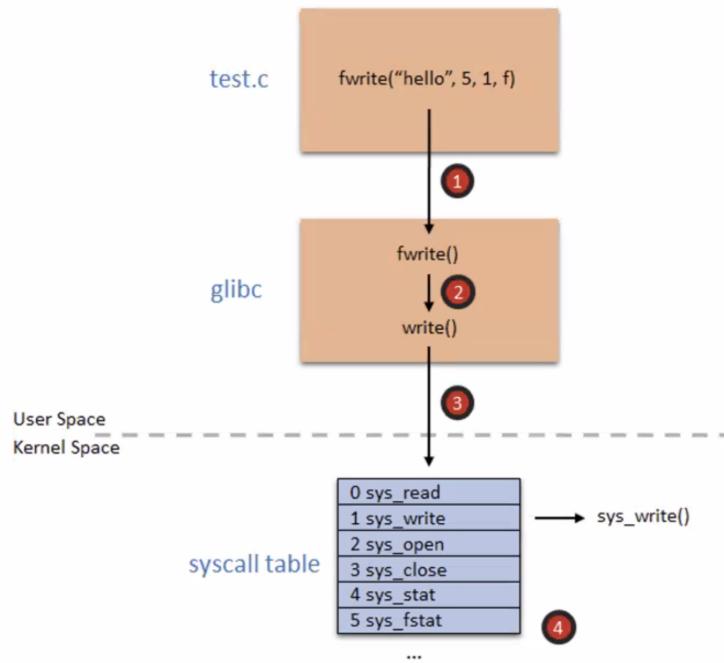
# eBPF

- Legacy "Berkeley Packet Filter" (BPF) - technology that among other things allows programs to analyze network traffic (and eBPF is an extended BPF JIT virtual machine in the Linux kernel).
  - raw interface to data link layers
  - permitting raw link-layer packets to be sent and received
  - can run sandboxed programs in a privileged context

*BPF is a highly flexible and efficient virtual machine-like construct in the Linux kernel allowing to execute bytecode at various hook points in a safe manner. It is used in number of Linux kernel subsystems (networking, tracing, security (sandboxing))"*

# SysCalls

eBPF



# Why?

eBPF

Enhanced Telemetry Collection -> annotation

- kernel and syscall attributes
- socket info

Performance

- avoid transfer of all audit data to userspace
- lower resource impact (net, file, proc)
- real time processing

eBPF Verifier verifies the safety of eBPF programs

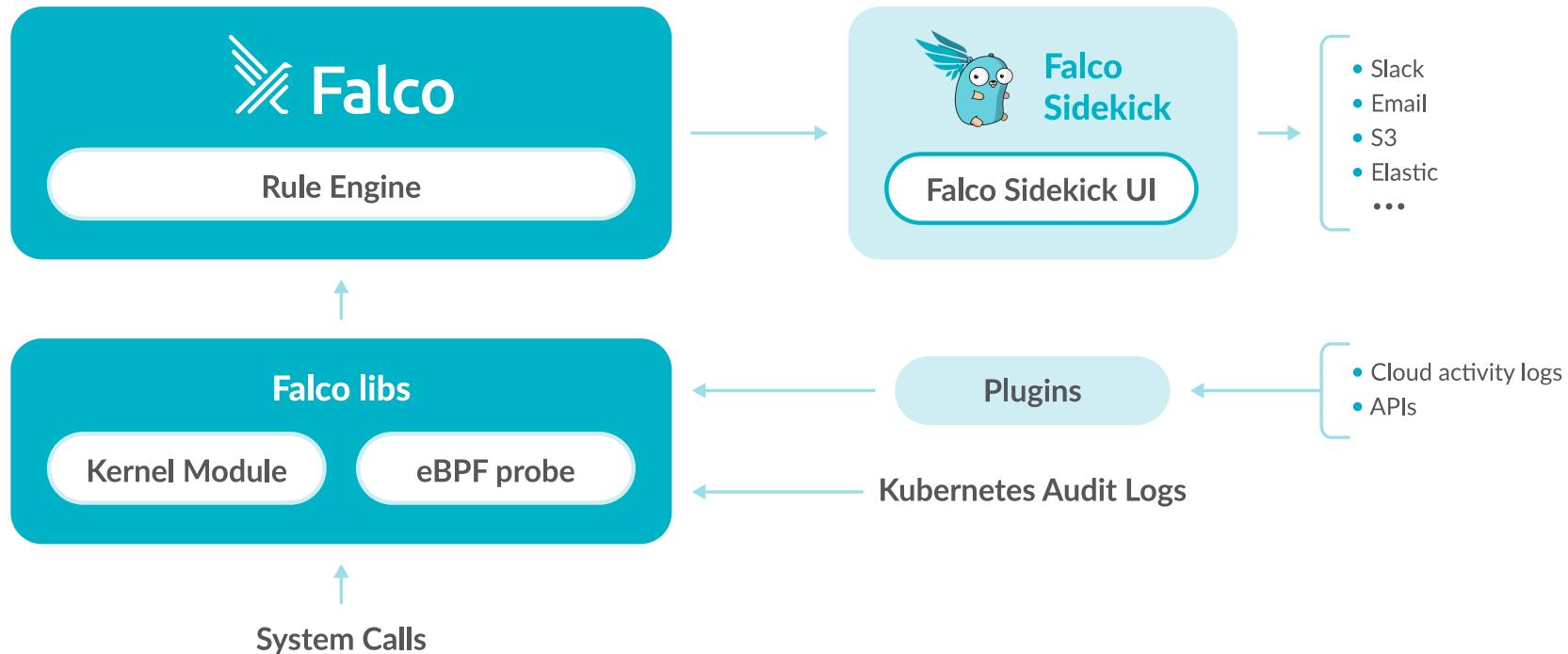
## Field Class: process

Additional information about the process and thread executing the syscall event.

| Name         | Type    | Description   |
|--------------|---------|---|
| proc.pid     | INT64   | the id of the process generating the event.   |
| proc.exe     | CHARBUF | the first command line argument (usually the executable name or a custom one).  |
| proc.name    | CHARBUF | the name (excluding the path) of the executable generating the event.   |
| proc.args    | CHARBUF | the arguments passed on the command line when starting the process generating the event.                                |
| proc.env     | CHARBUF | the environment variables of the process generating the event.  |
| proc.cmdline | CHARBUF | full process command line, i.e. proc.name + proc.args.  |
| proc.exeline | CHARBUF | full process command line, with exe as first argument, i.e. proc.exe + proc.args.                                       |
| proc.cwd     | CHARBUF | the current working directory of the event.   |
| proc.threads | UINT32  | the number of threads that the process generating the event currently has, including the main process thread.           |
| proc.childs  | UINT32  | the number of child threads that the process generating the event currently has. This excludes the main process thread. |
| proc.ppid    | INT64   | the pid of the parent of the process generating the event.  |
| proc.pname   | CHARBUF | the name (excluding the path) of the parent of the process generating the event.  |

# Deployment

Falco components



# Deployment

## K8s & Configuration

### Deployment

- Falco-sidekick, prom. exporter
- Falco-sidekick UI
- Grafana dashboards
- Alertmanager, Loki, ES, Kibana
- ..., Plugins

### Daemonset

- Falco

(only ``falco-driver-loader`` needs to be run with  
``securityContext: privileged``)

### What to enable?

- driver-loader (DKMS, private builds)
- docker, containerd, cri-o
- w/k8s metadata
- custom rules
- readiness, maxBurst, eventDrops
- priority/severity level
- plugins
- k8s audit, ...

# Language

Syntax, [github.com/falcosecurity/charts/falco/rules](https://github.com/falcosecurity/charts/falco/rules)

## Macros

- name (identifier)
- condition (filter)

## Lists

- name (identifier)
- items:

## Rules

- `name` (identifier)
- desc
- `condition` (filter expression, macro)
- `output` (formated message with **core details**)
- priority (severity of rule)
- tag
- append
- exceptions (new, not used in upstream)

# Primitives

## Rules

Shell executed in container

```
container.id != host and proc.name = bash
```

Overwirite system bins

```
fd.directory in (/bin, /bin/sbin, /usr/bin, /usr/sbin)  
and write
```

Container namespace change

```
evt.type = setns and not proc.name in (docker)
```

Process access camera

```
ect.type = open and fd.name = /dev/video0 and not proc.name in (skype, zoom, webex)
```

# Macros & Lists

## Rules

```
- list: _container_engine_binaries
  items: [dockerd, containerd, containerd-shim, "runc:[0:PARENT]", "runc:[1:CHILD]", "runc:[2:INIT]"]

- macro: docker_authorized_binaries
  condition: >
    proc.name in (_container_engine_binaries)
    or proc.pname in (_container_engine_binaries)

" [CVE-2019-11246 on Mitre](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-11246)
- macro: safe_kubectl_version
  condition: (
    jevt.value[/useragent] startswith "kubectl/v1.20" or
    jevt.value[/useragent] startswith "kubectl/v1.19"
    ...
  )
```

# Filesystem integrity

## Rules

- rule: Detect Write Below /etc/hosts  
desc: an attempt to write to /etc/hosts file (CVE-2020-8557)  
condition: open\_write and container and fd.name=/etc/hosts  
output: "File /etc/hosts opened for writing (user=%user.name command=%proc.cmdline parent=%proc.pname \  
pc cmdline=%proc.pc cmdline file=%fd.name program=%proc.name qparent=%proc.aname[2] \  
ggparent=%proc.aname[3] gggparent=%proc.aname[4] container\_id=%container.id image=%container.image.repository)  
priority: ERROR  
tags: [filesystem, mitre\_persistence]
  
- macro: open\_write  
condition: evt.type in (open,openat,openat2) and evt.is\_open\_write=true and fd.typechar='f' and fd.num>=0

# Detect su, sudo

## Rules

```
- rule: Detect su or sudo
  desc: detect sudo activities
  condition: >
    spawned_process
    and activity_under_escalated_privilege
    and not in (sre_known_infraops_actions, sre_known_managed_cloud_actions)
  output: >
    Privilege escalation activity (user=%user.name auser=%user.loginname command=%proc.cmdline ppid=%proc.ppid apid=%proc.apid priority: WARNING
  tags: [process, sudo, su]

- macro: activity_under_escalated_privilege
  condition: >
    proc.name in (sudo, su)
    or proc.pname in (sudo, su)
    or proc.aname[1] in (sudo, su)
    ...
  ...

- rule: Privilege escalation
  condition: sf.pproc.uid != 0 and sf.proc.uid = 0 and not entrypoint
```

# Compromised server process

## Rules

HitchSQL injection attack?

```
condition: spawn_process and proc.name = mysql and not proc_is_new
```

```
- macro: spawn_process
  condition: syscall.type = execve

- macro: proc_is_new
  condition: proc.duration <= 5000000000
```

# Howto rules

What we tweaked?

```
# override upstream defined macro
- macro: user_known_read_sensitive_files_activities
  condition: >
    (sre_authorized_activities)
```

```
rules/
├── falco_rules.preload.yaml
├── falco_rules-10-exceptions.yaml
├── falco_rules-20-security.yaml
├── falco_rules-30-apps.yaml
├── falco_rules-40-fim.yaml
└── falco_rules-50-cve.yaml
```

- macro: `failed\_k8s\_annotation`
- macro: `sre|host|infra\_authz\_activities`
- macro: `sre|aws|gcp\_known\_vendoractions`
- macro: `sre\_known\_ports`
- list: `sre|aws|gcp\_known\_commands`
- list: ...

# Falco sidekick

## Integrations

<https://github.com/falcosecurity/falcosidekick>

```
alertmanager:  
  hostport: http://{{domain or ip}}:{{port}}  
  minimumpriority: "error"          # emergency|alert|critical|error|warning|notice|informational|debug  
  endpoint: "/api/v2/alerts"  
  expiresafter: "900"
```

Slack • Rocketchat • Mattermost • Teams • Datadog • Discord • AlertManager • Elasticsearch • Loki • NATS • STAN (NATS Streaming) • Influxdb • AWS Lambda • AWS SQS • AWS SNS • AWS CloudWatch • AWS S3 • SMTP (email) • Opsgenie • StatsD • DogStatsD • Webhook • Azure Event Hubs • Prometheus • GCP PubSub • GCP Storage • Google Chat • Apache Kafka • PagerDuty • Kubeless • OpenFaaS

# Dashboards and alerting

The screenshot displays the Falcosidekick UI dashboard, which includes a header with navigation tabs (Dashboard, Events, Info), a search bar, and a summary of total events across various categories. Below this is a table listing specific alerts with columns for Timestamp, Source, Priority, Rule, Output, and Tags.

| Timestamp               | Source   | Priority      | Rule  | Output  | Tags                                      |
|-------------------------|----------|---------------|---|---|---|
| 2022/05/11 12:18:18:565 | syscalls | Critical      | Polkit Local Privilege Escalation Vulnerability (CVE-2021-4034) | Detect Polkit pkexec Local Privilege Escalation Exploit (CVE-2021-4034) (user=%user.loginname uid=%user.loginuid command=%proc.cmdline args=%proc.args)<br>proc.args proc.cmdline %proc.cmdline user.loginname %user.loginname user.loginuid %user.loginuid   | process mitre_privilege_escalation        |
| 2022/05/11 12:18:12:561 | syscalls | Warning       | Mount Launched in Privileged Container                          | Mount was executed inside a privileged container (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline %container.info image=%container.image.repository %container.image.tag)<br>container.image.repository %container.image.repository container.image.tag %container.image.tag container.info %container.info proc.cmdline %proc.cmdline user.loginuid %user.loginuid user.name %user.name   | container cis mitre_lateral_movement      |
| 2022/05/11 12:17:58:555 | syscalls | Warning       | Create Symlink Over Sensitive Files                             | Symlinks created over sensitive files (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline target=%evt.arg.target linkpath=%evt.arg.linkpath parent_process=%proc.pname)<br>evt.arg.linkpath %evt.arg.linkpath evt.arg.target %evt.arg.target proc.cmdline %proc.cmdline proc.pname %proc.pname user.loginuid %user.loginuid user.name %user.name  | file mitre_exfiltration                   |
| 2022/05/11 12:17:40:538 | syscalls | Notice        | Launch Ingress Remote File Copy Tools in Container              | Ingress remote file copy tool launched in container (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline parent_process=%proc.pname container_id=%container.id container_name=%container.name image=%container.image.repository %container.image.tag)<br>container.id %container.id container.image.repository %container.image.repository container.image.tag %container.image.tag container.name %container.name k8s.ns.name %k8s.ns.name k8s.pod.name %k8s.pod.name proc.cmdline %proc.cmdline proc.pname %proc.pname user.loginuid %user.loginuid user.name %user.name | network process mitre_command_and_control |
| 2022/05/11 12:17:38:536 | syscalls | Informational | System user interactive   | System user ran an interactive command (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline container_id=%container.id image=%container.image.repository)<br>container.id %container.id container.image.repository %container.image.repository proc.cmdline %proc.cmdline user.loginuid %user.loginuid user.name %user.name  | users mitre_remote_access_tools           |
| 2022/05/11 12:17:29:528 | syscalls | Notice        | DB program spawned process                                      | Database-related program spawned process other than itself (user=%user.name user_loginuid=%user.loginuid program=%proc.cmdline parent=%proc.pname container_id=%container.id image=%container.image.repository)<br>container.id %container.id container.image.repository %container.image.repository proc.cmdline %proc.cmdline proc.pname %proc.pname user.loginuid %user.loginuid user.name %user.name  | process database mitre_execution          |

# Elasticsearch

## Record detail

|                 |  |
|-----------------|--|
| ⌚ output_fields | <pre>{<br/>    "user.uid": 2201,<br/>    "proc.cmdline": "uname -o",<br/>    "proc.ppid": 6271,<br/>    "proc.env": "XDG_SESSION_ID=5471 HOSTNAME=master-0 SELINUX_ROLE_REQUESTED= TERM=xterm-256color SHELL=/bin/bash HISTSIZE=1000 S<br/>SH_CLIENT=10.54.80.50 51390 22 SELINUX_USE_CURRENT_RANGE= SSH_TTY=/dev/pts/0 USER=vesop MAIL=/var/spool/mail/vesop PATH=/usr/loc<br/>al/bin:/usr/bin:/usr/local/sbin:/usr/sbin PWD=/home/vesop SELINUX_LEVEL_REQUESTED= HISTCONTROL=ignoredups SHLVL=1 HOME=/home/ves<br/>op DNSServiceIP=10.3.0.10 LOGNAME=vesop SSH_CONNECTION=10.54.80.50 51390 10.63.51.10 22 KUBERNETES_VERSION=v1.15.0 XDG_RUNTIME_D<br/>IR=/run/user/2201 _=/usr/bin/uname",<br/>    "proc.aname[2]": "bash",<br/>    "container.id": "host",<br/>    "proc.aname[3]": "sshd",<br/>    "proc.pid": 6273,<br/>    "proc.aname[4]": "sshd",<br/>    "user.loginuid": 2201,<br/>    "group.gid": 2201,<br/>    "proc.pcmdline": "bash",<br/>    "evt.time": 1654426400432897300<br/>}</pre> |
| ⌚ severity      | warning  |
| ⌚ site          | ty8-tky  |
| ⚠ source        | syscall  |
| ⌚ source_type   | kafka  |
| ⌚ stream        | stdout   |
| ⌚ tag           | kube.app.falco.falco   |
| ⌚ tags          | group, process, user   |

# Falco Audit in Grafana

Volterra Security Services / Falco Audit

Last 30 minutes

Audit Events

| time                | severity | message                                    | Cluster             | hostname                                     | namespace | pod | procCmdline   | fdName                       | id           |
|---------------------|----------|--|---------------------|--|-----------|-----|---|------------------------------|--------------|
| 2021-12-13 20:22:41 | warning  | Detect File Permission or Ownership Change | gc01-cle-int-ves-io | ip-172-16-130-194.us-east-2.compute.internal |           | -   | chmod 700 /tmp/awsagent.8k0fdvcy  |                              | KaQ_tX0Bsryk |
| 2021-12-13 20:22:40 | warning  | Clear Log Activities                       | gc01-cle-int-ves-io | ip-172-16-130-194.us-east-2.compute.internal |           | -   | bash -c /bin/sleep \${((RANDOM % 3000) + 1)}; rm -f /var/log/awsagent-update.log; umask 037 && /opt/aws/awsagent/bin/update > /var/log/awsagent-update.log 2>&1 | /var/log/awsagent-update.log | KKQ_tX0Bsryk |
| 2021-12-13 20:21:50 | warning  | Clear Log Activities                       | gc01-cle-int-ves-io | ip-172-16-132-140.us-east-2.compute.internal |           | -   | bash -c /bin/sleep \${((RANDOM % 3000) + 1)}; rm -f /var/log/awsagent-update.log; umask 037 && /opt/aws/awsagent/bin/update > /var/log/awsagent-update.log 2>&1 | /var/log/awsagent-update.log | 5Bw-tX0BRw1  |
| 2021-12-13 20:18:29 | error    | Write below rpm database                   | gc01-cle-int-ves-io | ip-172-16-147-217.us-east-2.compute.internal |           | -   | python  | /var/lib/rpm/.dbenv.lock     | oRw7tX0BRw   |
| 2021-12-13 20:18:28 | error    | Write below rpm database                   | gc01-cle-int-ves-io | ip-172-16-147-217.us-east-2.compute.internal |           | -   | python  | /var/lib/rpm/__db.003        | HqQ7tX0Bsryl |
| 2021-12-13 20:18:28 | error    | Write below rpm database                   | gc01-cle-int-ves-io | ip-172-16-147-217.us-east-2.compute.internal |           | -   | python  | /var/lib/rpm/__db.002        | HaQ7tX0Bsryl |

# Plugins

Added recently (>= v0.31)

## External sources

- API boundaries, hardly extensible
- Falco must expose a web server
- TLS to manage
- Doesn't work with managed K8s

## Features

- dynamic shared libraries -> any language
- allows falco to collect and extract fields from streams of events
- source / extractor plugins

## Available plugins:

- K8s audit
- AWS CloudTrail
- JSON
- comming (okta, github, docker, seccompagent)

# K8s audit rules

<https://github.com/falcosecurity/plugins/tree/master/plugins/k8saudit>

An attempt to start a pod using the host pid NS.

```
condition: kevt and pod and kcreate  
and ka.req.pod.host_pid intersects (true)
```

Updated role binding

```
condition: kevt  
and clusterrolebinding  
and kcreate and ka.req.binding.role=cluster-admin
```

Detect pod starting a privileged container

```
condition: kevt  
and pod  
and kcreate  
and ka.req.pod.containers.privileged intersects (true)  
and not ka.req.pod.containers.image.repository  
in (falco_privileged_images)
```

Credentials in configmap

```
- macro: contains_private_credentials  
condition: >  
(ka.req.configmap.obj contains "access_key" or  
ka.req.configmap.obj contains "access-key" or  
ka.req.configmap.obj contains "token" or  
ka.req.configmap.obj contains "secret" or  
ka.req.configmap.obj contains "pass")
```

# CloudTrail

## Plugin

```
- rule: Console Login Without MFA
  desc: Detect a console login without MFA.
  condition:
    ct.name=="ConsoleLogin" and not ct.error exists
    and ct.user.identitytype!="AssumedRole"
    and json.value[/responseElements/ConsoleLogin]=="Success"
    and json.value[/additionalEventData/MFAUsed]=="No"
  output:
    Detected a console login without MFA
    (requesting user=%ct.user,
     requesting IP=%ct.srcip,
     AWS region=%ct.region)
  priority: CRITICAL
```

# What is the next step?

```
- rule: Pet detection, custom plugin
  condition: video.entities[animal] > 0
```

## Sysflow.io

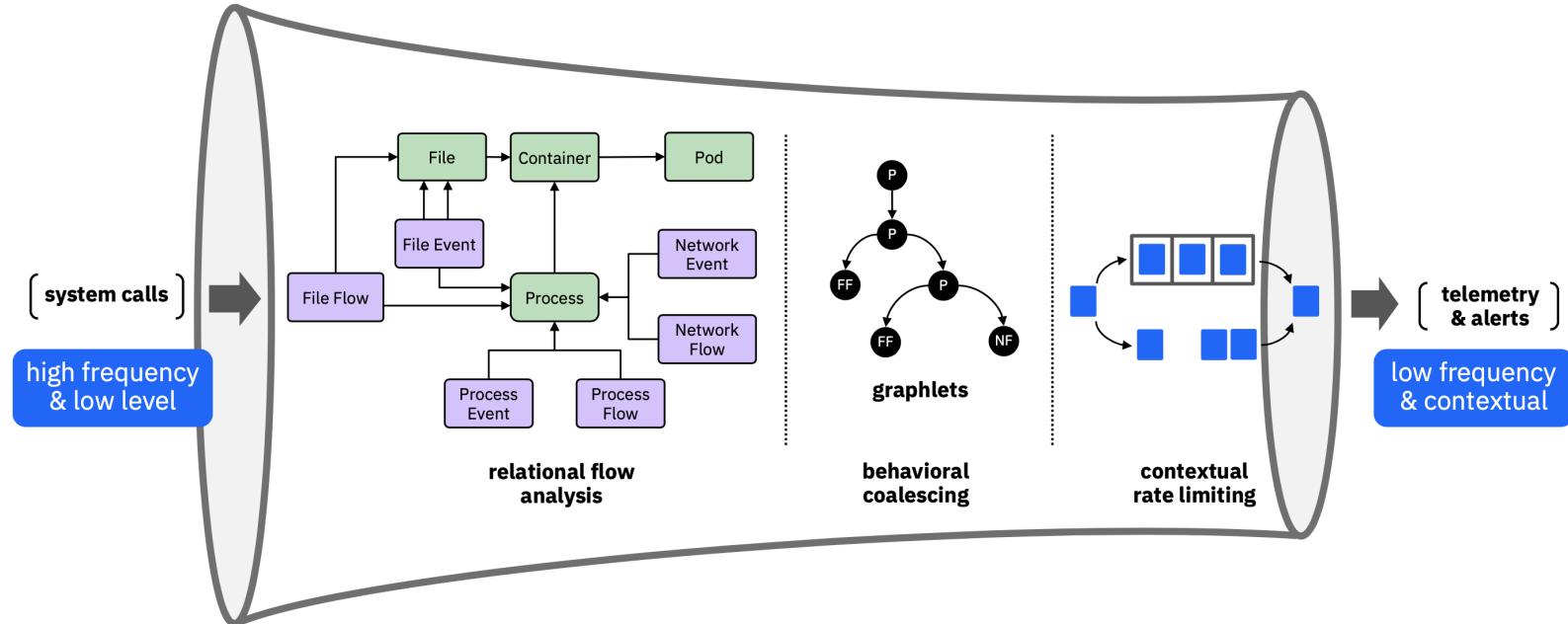
```
- rule: Impair Defenses: Disable or Modify System Firewall
  desc: Detects disabling security tools
  condition: sf.opflags = EXEC and
    ((sf.proc.name in (service_cmds) and
      sf.proc.args pmatch (security_services) and sf.proc.args pmatch (stop_cmds)) or
    (sf.proc.name = setenforce and sf.proc.args = '0'))
  prefILTER: [PE]

- rule: Large network data transfer with database endpoint
  condition: ( sf.opflags contains RECV and sf.net.dport = 3306 and sf.flow.rbytes > 1024 ) or
    ( sf.opflags contains SEND and sf.net.sport = 3306 and sf.flow.wbytes > 1024 )
  prefILTER: [NF]

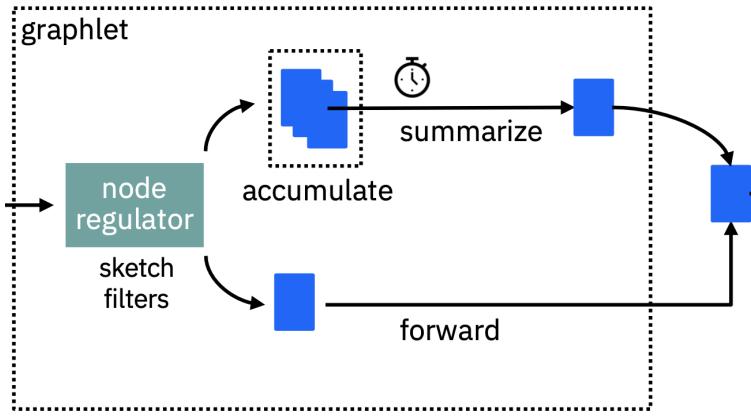
- rule: Privilege escalation
  condition: sf.pproc.uid != 0 and sf.proc.uid = 0 and not entrypoint
```

# SysFlow.io

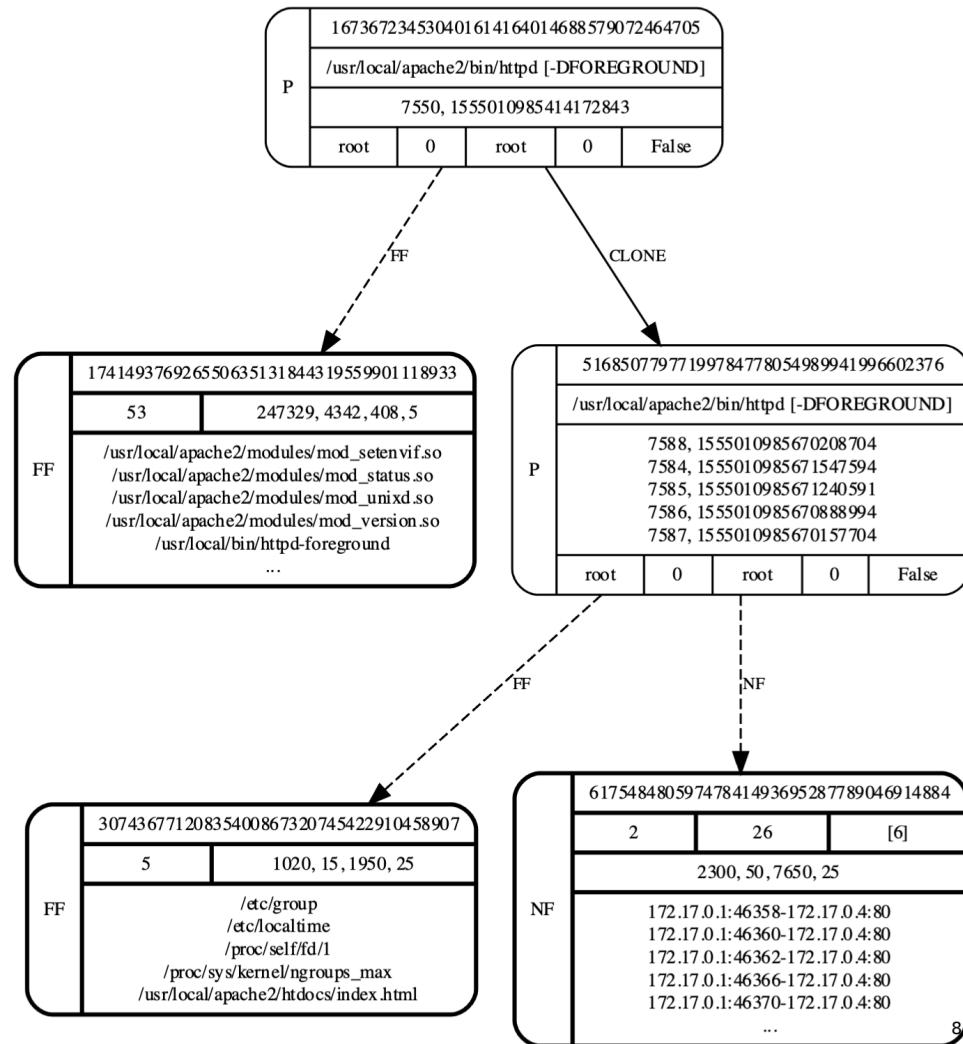
cloud-native system telemetry framework



# SysFlow.io



- Rate modulation
- Node-level regulators
  - HyperLogLog sketch
  - Count-min sketch
  - Tries



# Learn More

[Documentation](#) · [GitHub](#) · [Blog](#)

- [Falco & Plugins CloudNativeCon 2022](#)
- [SysFlow is a cloud-native system telemetry framework that enables the creation of security analytics on a scalable, pluggable open-source platform](#)
  - [SysFlow telemetry](#)
  - [SysFlow & Sidekick analytics PoC](#)
  - [SysFlow policies & examples](#)
- [Plugin Pet surveillance with falco PoC](#)
- [Employ AI/ML for anomaly detection](#)



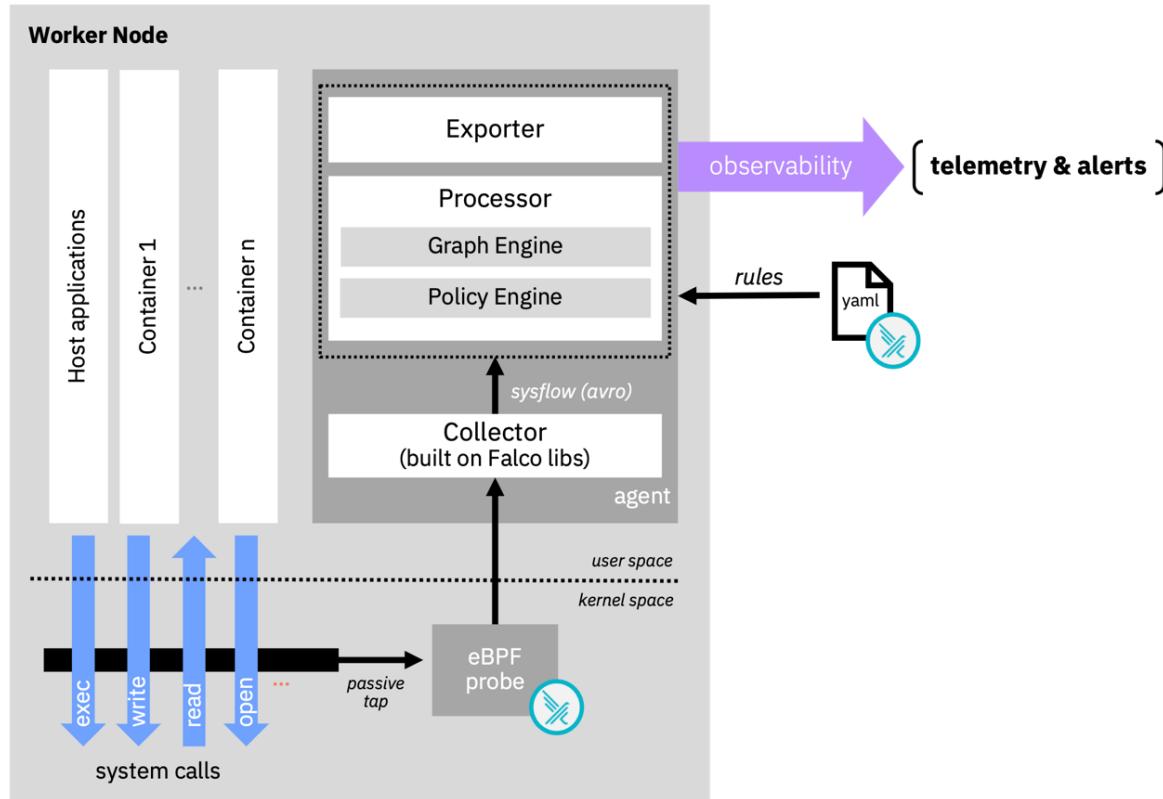
END

# Backup slides

Falco architectural overview

# Backup slides

## SysFlow architectural overview



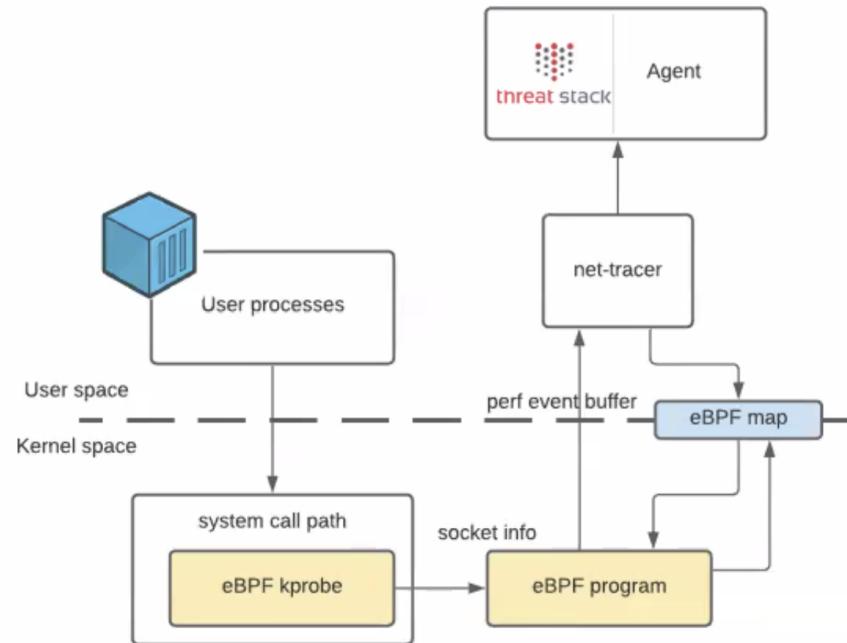
# Backup slides

## eBPF network observability

The agent attaches eBPF programs to kprobes to trace TCP (connect/accept) and UDP (connect) activity.

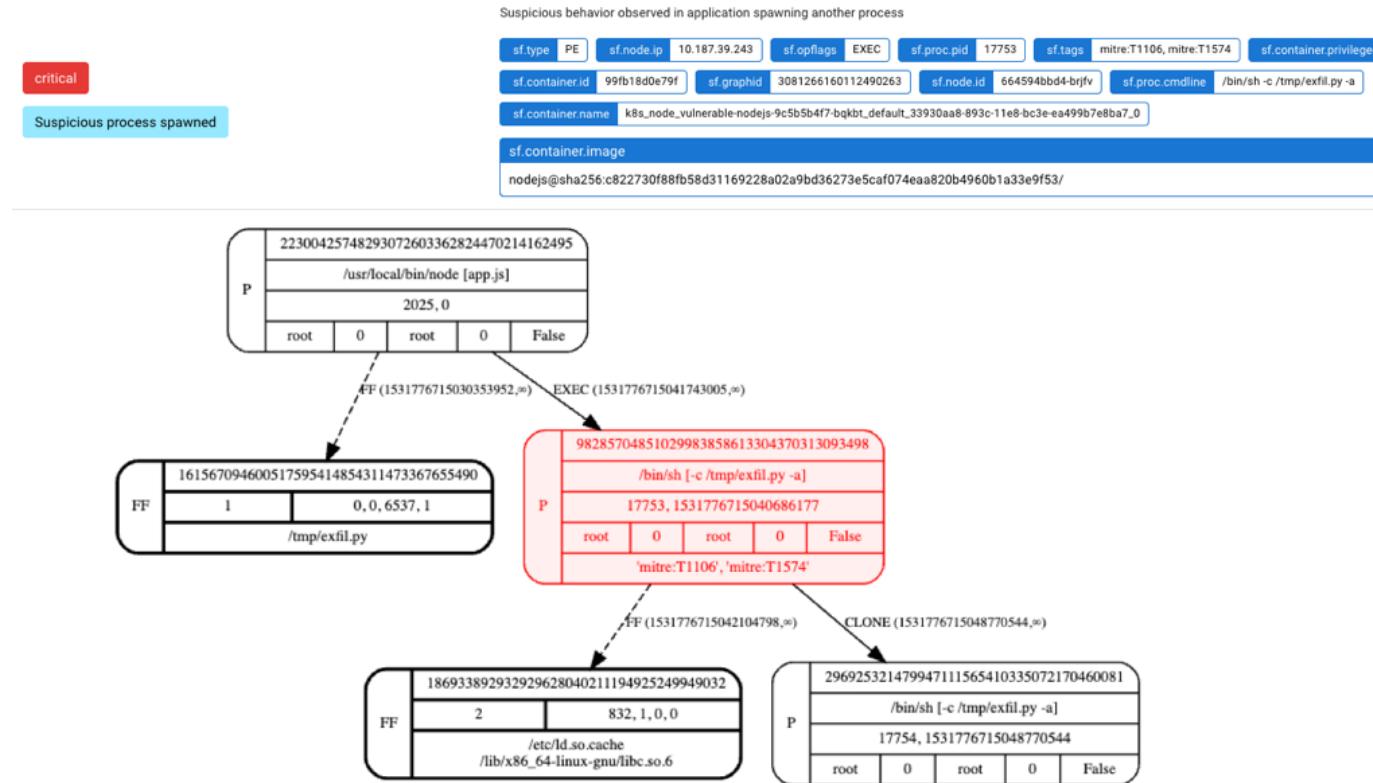
Uses telemetry from eBPF sensor to augment audit events.

Additional DNS sensor provides reverse DNS information



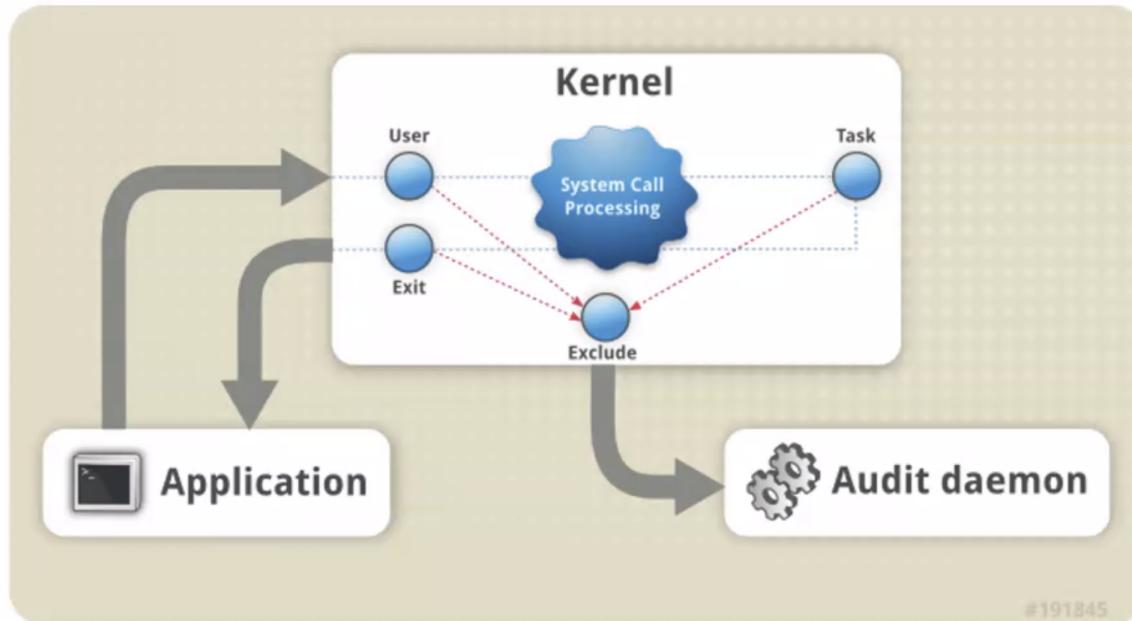
# Backup slides

SysFlow.io integration to sidekick



# Backup slides

AuditD comparison



# Backup slides

## Inbound ssh rule

```
- rule: Inbound SSH Connection
  desc: Detect Inbound SSH Connection
  condition: >
    ((evt.type in (accept,listen) and evt.dir=<) or
     (evt.type in (recvfrom,recvmsg))) and ssh_port
     and not is_kubernetes
  output: >
    Inbound SSH connection (user=%user.name client_ip=%fd.cip client_port=%fd.cport server_ip=%fd.sip)
  priority: WARNING
  tags: [ssh, network]
```

# Backup slides

K8s audit -> plugin

- Removed K8S audit logs from Falco [#1952] (<https://github.com/falcosecurity/falco/pull/1952>)
- Now under plugins: <https://github.com/falcosecurity/plugins>

```
- rule: Attach/Exec Pod
desc: Detect any attempt to attach/exec to a pod
condition: |
  kevt_started and pod_subresource and kcreate and ka.target.subresource in (exec,attach)
  and not user_known_exec_pod_activities

- list: falco_hostpid_images
items: []

- rule: Create HostPid Pod
desc: Detect an attempt to start a pod using the host pid namespace.
condition: |
  kevt and pod and kcreate and ka.req.pod.host_pid intersects (true)
  and not ka.req.pod.containers.image.repository in (falco_hostpid_images)
```

# Backup slides

Rules from helm chart

<https://github.com/falcosecurity/falco/tree/master/rules>

```
rules-traefik.yaml: |-  
  - macro: traefik_consider_syscalls  
    condition: (evt.num < 0)  
  
  - macro: app_traefik  
    condition: container and container.image startswith "traefik"  
  
  # Restricting listening ports to selected set  
  
  - list: traefik_allowed_inbound_ports_tcp  
    items: [443, 80, 8080]
```

# Dashboards and alerting

