sysdig

Kubernetes Runtime Security with Falco and Sysdig

Jorge Salamero
@bencerillo

About me

Jorge Salamero

- Technical and Product Marketing @ Sysdig
- Used to be a speaker, DevOps and Debian Developer
- Behind many of the Falco integrations and Sysdig content and launches
- GitHub: bencer
- Twitter: @bencerillo

About Sysdig

- The OSS project:
  - 2013 Linux kernel tracing tool
  - Evolution of tcpdump and Wireshark into the system
  - Easy to use (no code required), asynchronous, production performance
  - Container and Kubernetes support

- The company:
  - 2014 Sysdig Monitor
  - 2017 Sysdig Secure
  - Committed to OSS: sysdig, Sysdig Inspect, Falco, eBPF and Prometheus contributor

Falco, a CNCF project

- Created originally by Sysdig the company
- Currently run independently by the Falco community
- Under the umbrella of the CNCF since 2018
- Sandbox level project, Incubation proposal WIP

  https://github.com/cncf/toc/pull/307

sysdig

Why we need runtime security?

Runtime security

CI/CD vulnerability scanning + ID management, is that enough?

1. Prevention / enforcement
2. Detection / audit
3. Blocking
4. Incident response and forensics

Runtime security: prevention

- Who can do what within Kubernetes?
- Kubernetes native controls:
  - Admission controller / OPA
  - RBAC
  - Network policies
  - Pod Security Policies
    - seccomp
    - SELinux / AppArmor

Runtime security: detection

- What happens when a control fails? Last line of defense!
- What do we count on when a catastrophe strikes?
- How do we tell a story when the unexpected happens?
- Applicable very few times, but critical when we need it.

But also:

- How can I validate enforcement works?
- Does it break my applications?

Runtime security: use cases for detection

- Unpatched vulnerabilities, not public vulnerabilities, 0-day exploits
- Insecure configurations
- Leaked or insecure credentials
- Internal threats
- Compliance

Existing approaches

- LD_PRELOAD
  - Dependency on glibc
  - Changes your app, possible in unknown ways
- ptrace
  - Single PID, captures every system call
  - Changes your app, possible in unknown ways
- sidecars
  - Shared namespaces (process, network, storage, etc)
  - Instrumentation overhead, complexity, limited scope to the pod
- Kernel based

Kernel module vs eBPF

Kernel module

- Close to total system visibility
- Doesn’t change containers or processes
- Asynchronous tracing, lowest impact
- Requires kernel-headers
- Can potentially crash the kernel

eBPF probe

- Close to total system visibility
- Doesn’t change containers or processes
- Asynchronous tracing, low impact
- Requires kernel-headers
- Safe to run in eBPF VM

https://sysdig.com/blog/sysdig-and-falco-now-powered-by-ebpf/

sysdig

Falco:
Kubernetes runtime security

sysdig

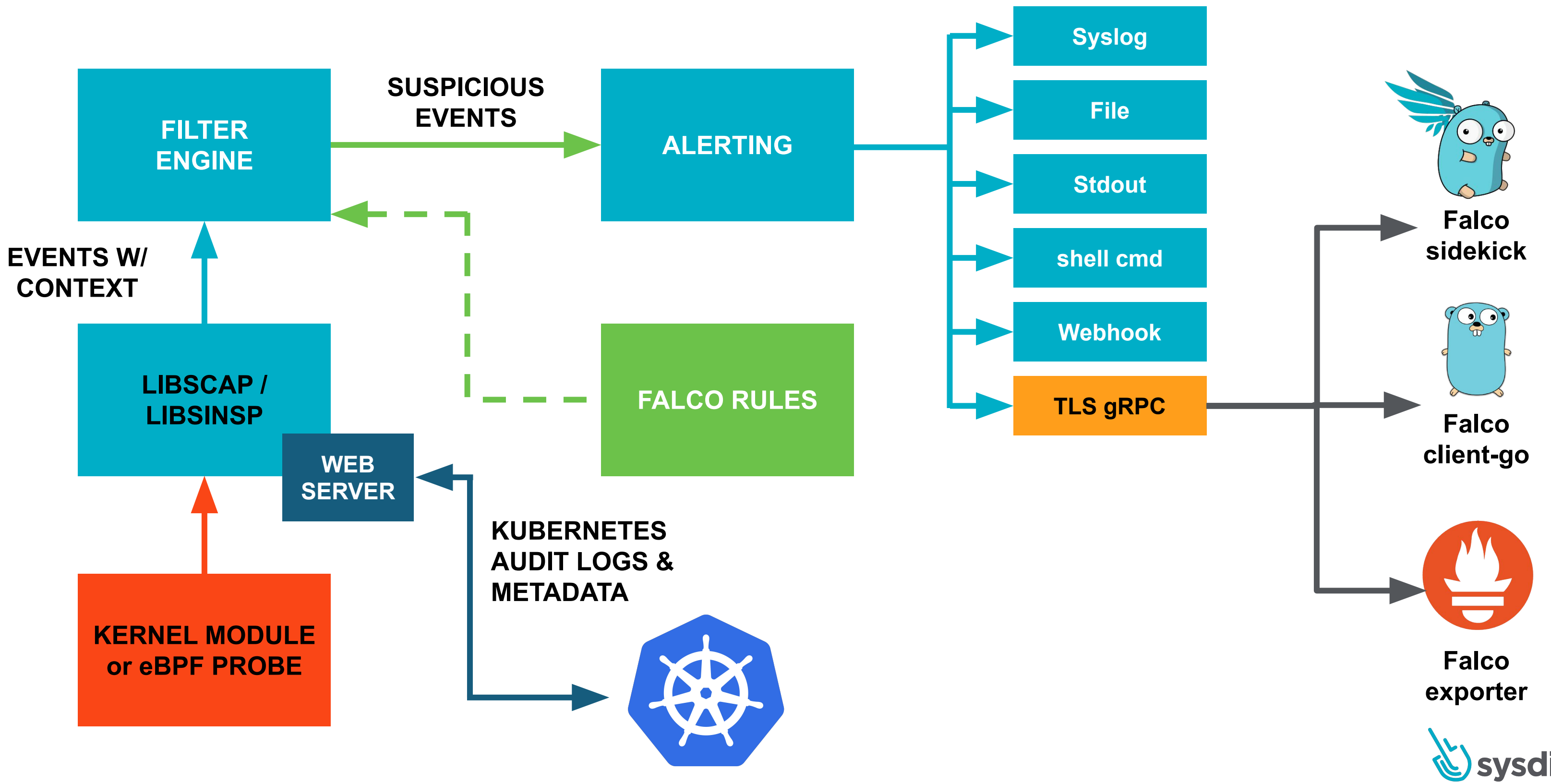
What is Falco?

- Kubernetes runtime security tool
- Detection engine for anomalous activity in hosts and containers
- Rules built using tcpdump like syntax
- Leverages `libscap` and `libsinsp`
- Kubernetes native support (context, kube-apiserver audit)

What kind of problems does it solve?

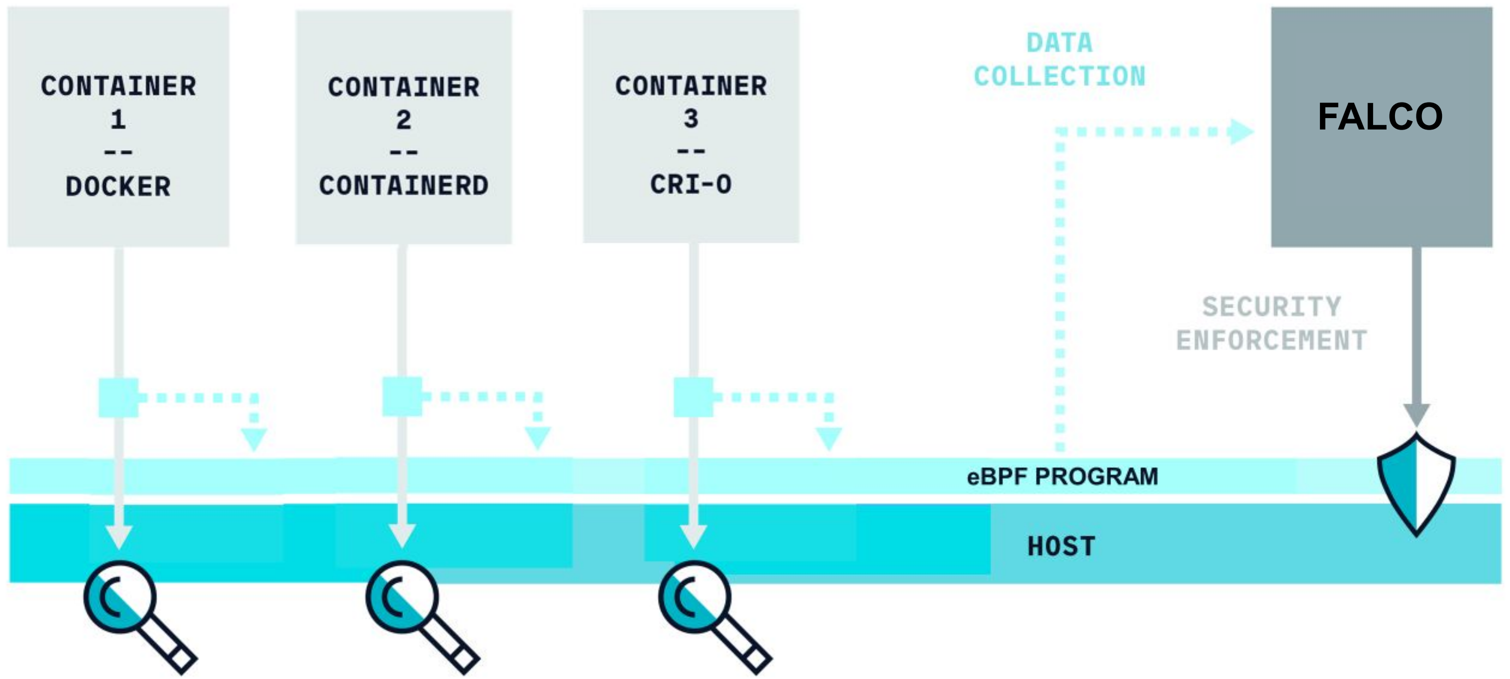
- Are my hosts and containers doing something they shouldn’t?
- Spawned processes:
  - Did my PostgreSQL container spawn an unexpected process?
- File system reads, writes:
  - Did someone install a new package or change configuration in a running container?
- Network activity:
  - Did my Nginx container open a new listening port or unexpected outgoing connection?
- User or orchestration activity:
  - Did any K8s user spawn a shell into a privileged container?

Falco architecture

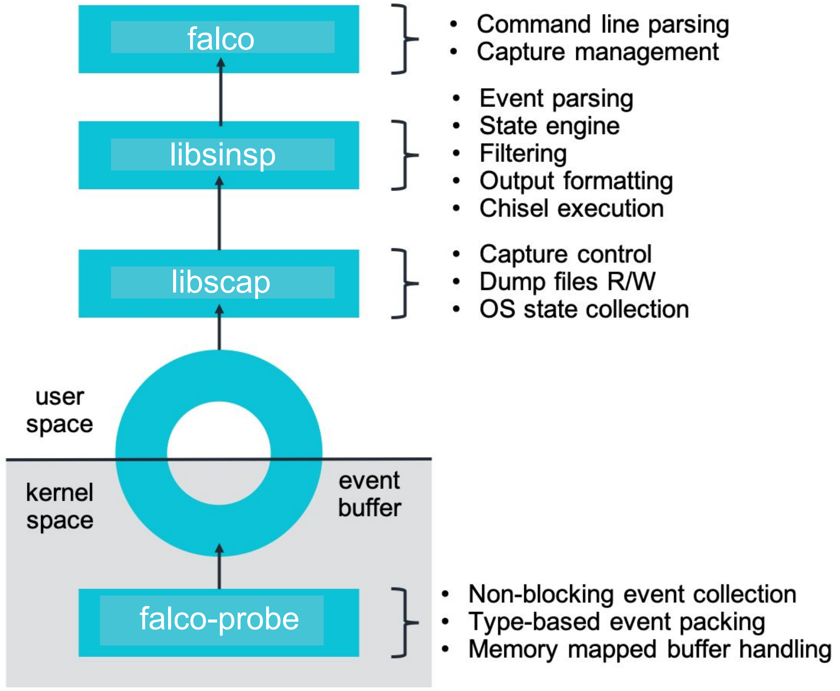


sysdig

Falco low level architecture



Falco lower level architecture



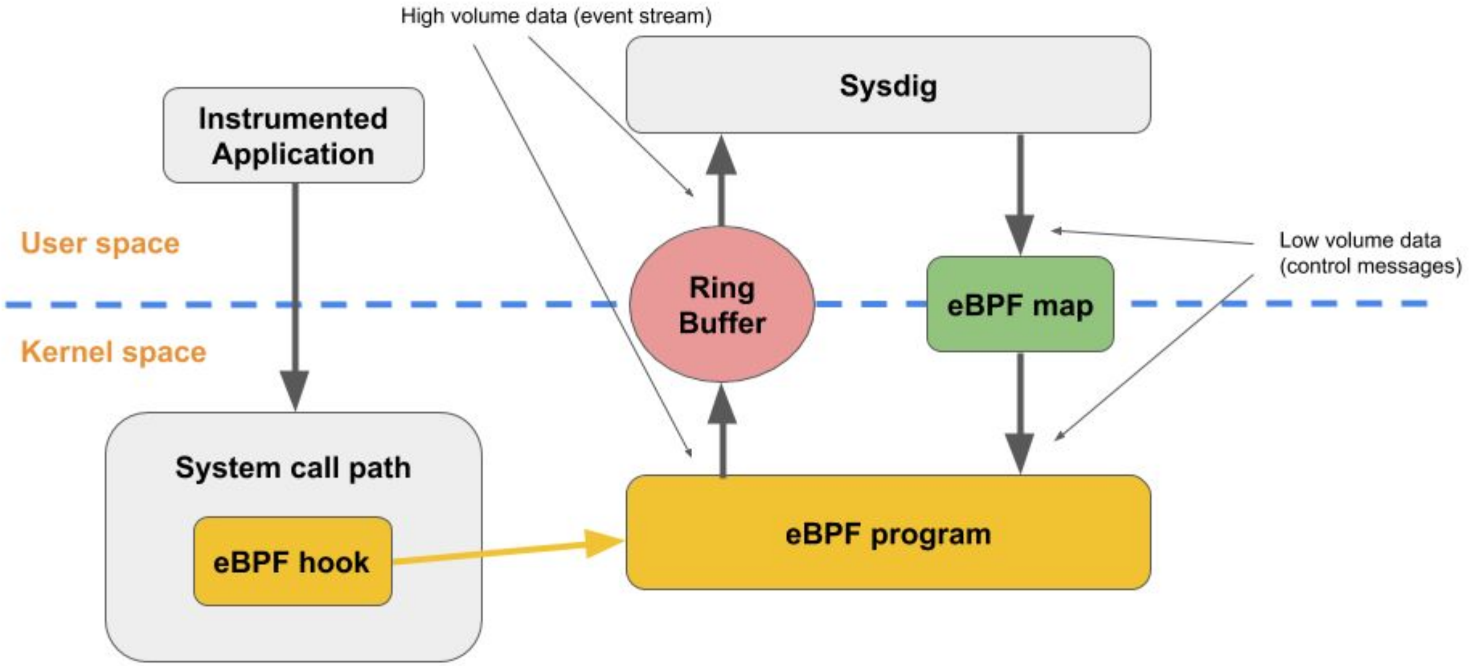
- Command line parsing
- Capture management

- Event parsing
- State engine
- Filtering
- Output formatting
- Chisel execution

- Capture control
- Dump files R/W
- OS state collection

- Non-blocking event collection
- Type-based event packing
- Memory mapped buffer handling

Falco lowest level architecture



Falco rule: container activity

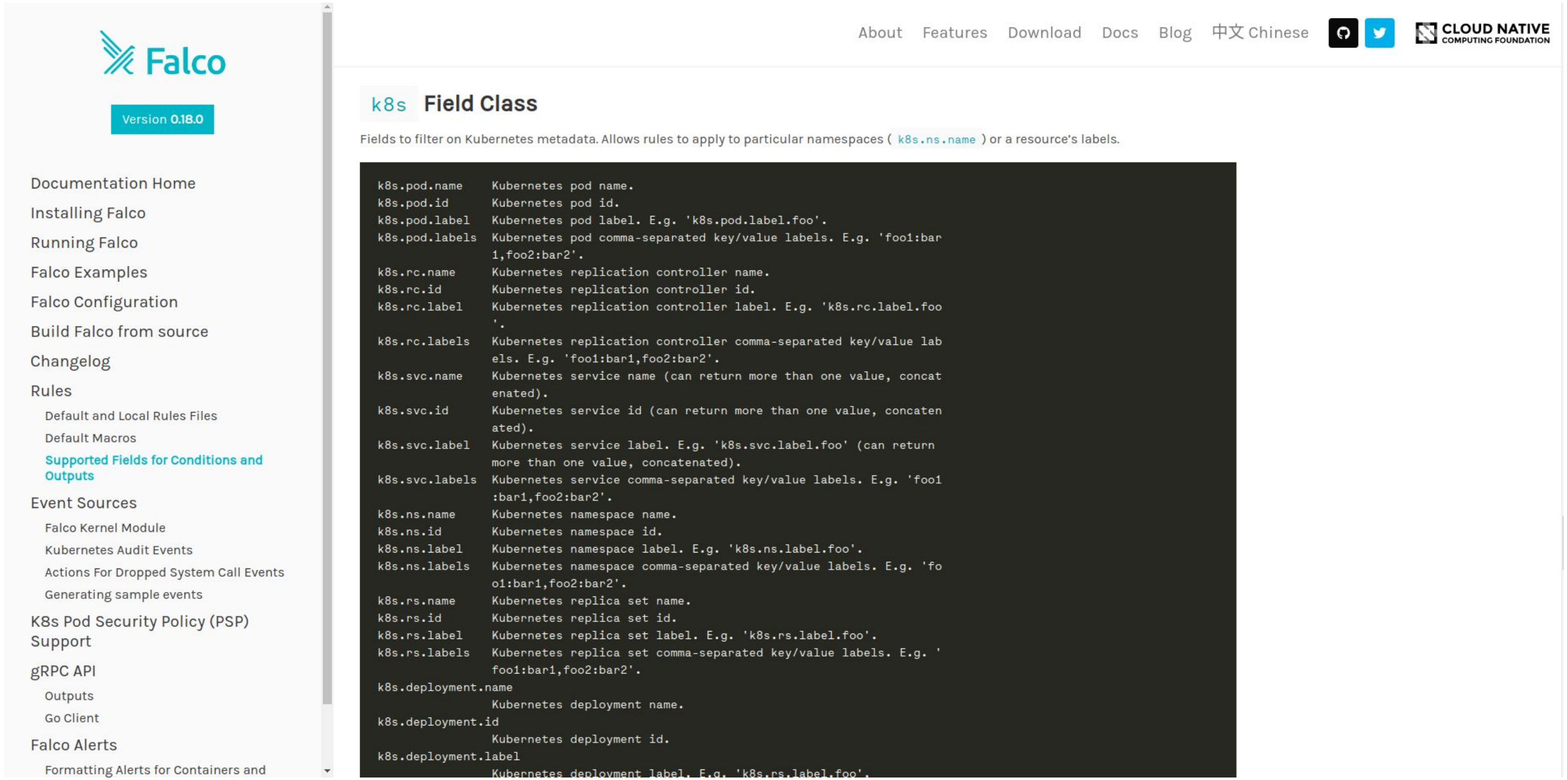
```
- rule: Node container runs Node binary
  desc: Detect a process that’s not node started in a Node container.
  condition: evt.type=execve and k8s.deployment.name=my-node-app and proc.name!=node
  output: Node container started unexpected process
                (user=%user.name command=%proc.cmdline %container.info)
  priority: INFO
  tags: [container, apps]
```

Something is executing a program

In a container in my Kubernetes deployment for my-node-app

And the process name isn’t node

Kubernetes filters



`k8s` Field Class

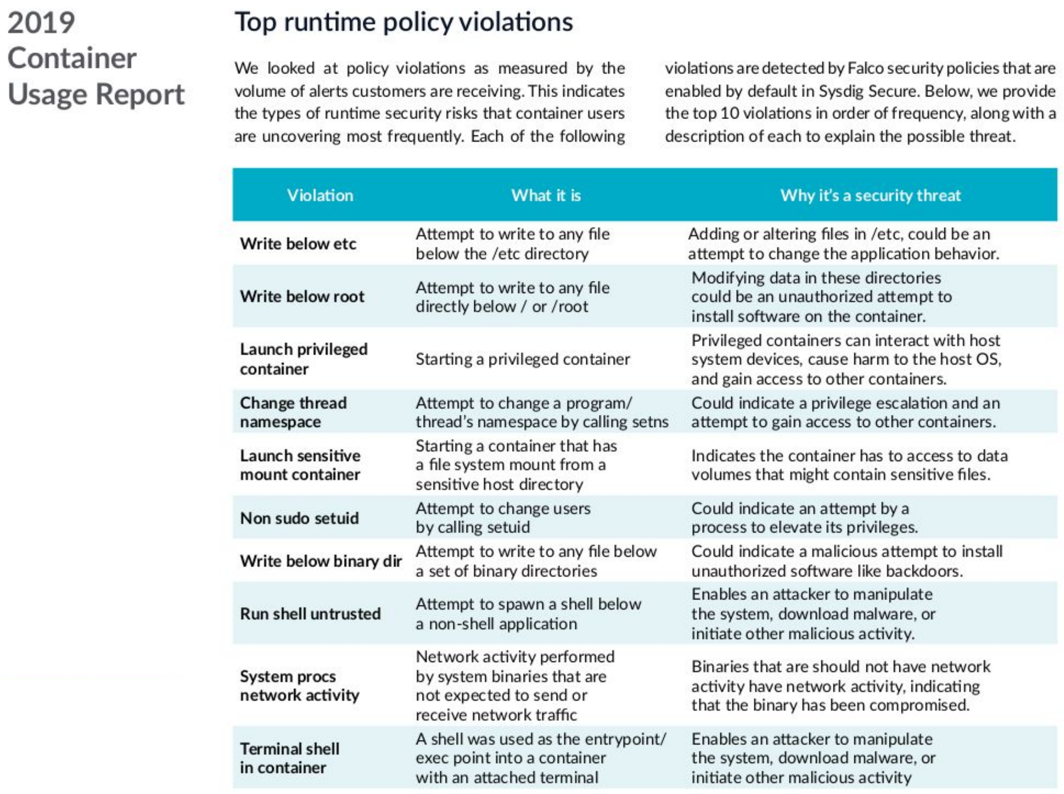
Fields to filter on Kubernetes metadata. Allows rules to apply to particular namespaces (`k8s.ns.name`) or a resource's labels.

```
k8s.pod.name    Kubernetes pod name.
k8s.pod.id      Kubernetes pod id.
k8s.pod.label   Kubernetes pod label. E.g. 'k8s.pod.label.foo'.
k8s.pod.labels  Kubernetes pod comma-separated key/value labels. E.g. 'foo1:bar
                1,foo2:bar2'.
k8s.rc.name     Kubernetes replication controller name.
k8s.rc.id       Kubernetes replication controller id.
k8s.rc.label    Kubernetes replication controller label. E.g. 'k8s.rc.label.foo
                '.
k8s.rc.labels   Kubernetes replication controller comma-separated key/value lab
                els. E.g. 'foo1:bar1,foo2:bar2'.
k8s.svc.name    Kubernetes service name (can return more than one value, concat
                enated).
k8s.svc.id      Kubernetes service id (can return more than one value, concaten
                ated).
k8s.svc.label   Kubernetes service label. E.g. 'k8s.svc.label.foo' (can return
                more than one value, concatenated).
k8s.svc.labels  Kubernetes service comma-separated key/value labels. E.g. 'foo1
                :bar1,foo2:bar2'.
k8s.ns.name     Kubernetes namespace name.
k8s.ns.id       Kubernetes namespace id.
k8s.ns.label    Kubernetes namespace label. E.g. 'k8s.ns.label.foo'.
k8s.ns.labels   Kubernetes namespace comma-separated key/value labels. E.g. 'fo
                o1:bar1,foo2:bar2'.
k8s.rs.name     Kubernetes replica set name.
k8s.rs.id       Kubernetes replica set id.
k8s.rs.label    Kubernetes replica set label. E.g. 'k8s.rs.label.foo'.
k8s.rs.labels   Kubernetes replica set comma-separated key/value labels. E.g. '
                foo1:bar1,foo2:bar2'.
k8s.deployment.name
                Kubernetes deployment name.
k8s.deployment.id
                Kubernetes deployment id.
k8s.deployment.label
                Kubernetes deployment label. E.g. 'k8s.rs.label.foo'.
```

Falco rule: Kubernetes activity

```
- macro: contains_private_credentials
  condition: >
    (ka.req.configmap.obj contains "aws_access_key_id" or
     ka.req.configmap.obj contains "aws_s3_access_key_id" or
     ka.req.configmap.obj contains “password")

- macro: configmap
  condition: ka.target.resource=configmaps

- macro: modify
  condition: (ka.verb in (create,update,patch))

- rule: Create/modify Configmap with private credentials
  desc: Detect creating/modifying a configmap containing a private credential
    (aws key, password, etc.)
  condition: configmap and modify and contains_private_credentials
  output: K8s configmap with private credential (user=%ka.user.name
          verb=%ka.verb name=%ka.req.configmap.name
          configmap=%ka.req.configmap.name config=%ka.req.configmap.obj)
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```

Top runtime security violations



2019 Container Usage Report

Top runtime policy violations

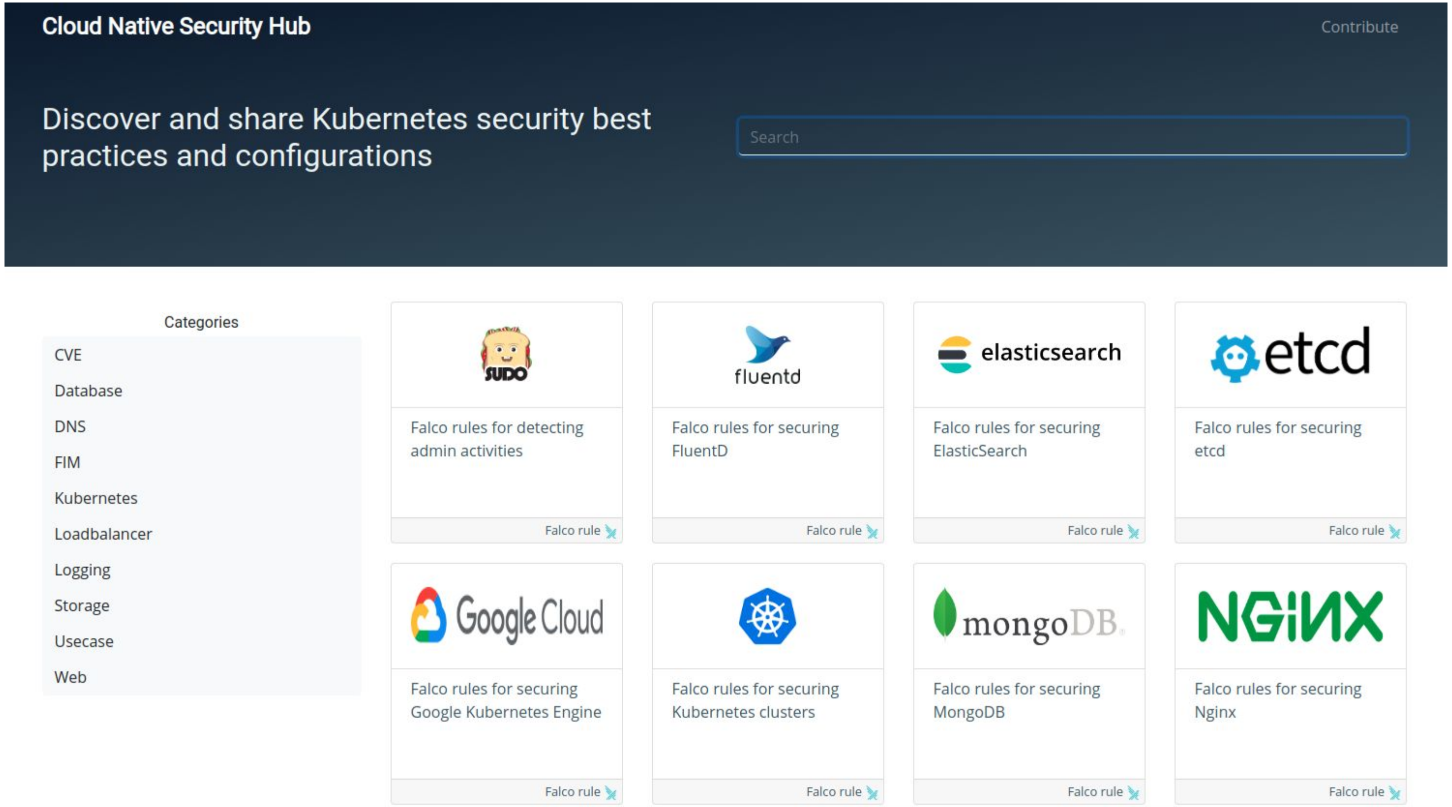
We looked at policy violations as measured by the volume of alerts customers are receiving. This indicates the types of runtime security risks that container users are uncovering most frequently. Each of the following violations are detected by Falco security policies that are enabled by default in Sysdig Secure. Below, we provide the top 10 violations in order of frequency, along with a description of each to explain the possible threat.

Violation	What it is	Why it's a security threat
Write below etc	Attempt to write to any file below the /etc directory	Adding or altering files in /etc, could be an attempt to change the application behavior.
Write below root	Attempt to write to any file directly below / or /root	Modifying data in these directories could be an unauthorized attempt to install software on the container.
Launch privileged container	Starting a privileged container	Privileged containers can interact with host system devices, cause harm to the host OS, and gain access to other containers.
Change thread namespace	Attempt to change a program/ thread's namespace by calling setns	Could indicate a privilege escalation and an attempt to gain access to other containers.
Launch sensitive mount container	Starting a container that has a file system mount from a sensitive host directory	Indicates the container has to access to data volumes that might contain sensitive files.
Non sudo setuid	Attempt to change users by calling setuid	Could indicate an attempt by a process to elevate its privileges.
Write below binary dir	Attempt to write to any file below a set of binary directories	Could indicate a malicious attempt to install unauthorized software like backdoors.
Run shell untrusted	Attempt to spawn a shell below a non-shell application	Enables an attacker to manipulate the system, download malware, or initiate other malicious activity.
System procs network activity	Network activity performed by system binaries that are not expected to send or receive network traffic	Binaries that are should not have network activity have network activity, indicating that the binary has been compromised.
Terminal shell in container	A shell was used as the entrypoint/ exec point into a container with an attached terminal	Enables an attacker to manipulate the system, download malware, or initiate other malicious activity

Popular Falco detection rules

Best practices	Compliance	Vulnerabilities	Cloud Native Stack
Update packages	FIM	CVE-2019-11246	K8s control plane
Modify /bin /usr	Privileged pod	kubectl cp	Nginx
Write below /etc	ConfigMap creds		Elasticsearch
Read sensitive file	kubectl exec/attach	CVE-2019-5736	Redis
DB spawned proc	Role changes audit	runc breakout	HAproxy
Change namespace	PCI		Rook
Privileged container	NIST	CVE-2019-14287	MongoDB
Sensitive mount		sudo bypass	PostgreSQL
Terminal shell			

SecurityHub.dev



Cloud Native Security Hub

Contribute

Discover and share Kubernetes security best practices and configurations

Search

Categories

- CVE
- Database
- DNS
- FIM
- Kubernetes
- Loadbalancer
- Logging
- Storage
- Usecase
- Web

Falco rules for detecting admin activities

Falco rule

Falco rules for securing FluentD

Falco rule

Falco rules for securing ElasticSearch

Falco rule

Falco rules for securing etcd

Falco rule

Falco rules for securing Google Kubernetes Engine

Falco rule

Falco rules for securing Kubernetes clusters

Falco rule

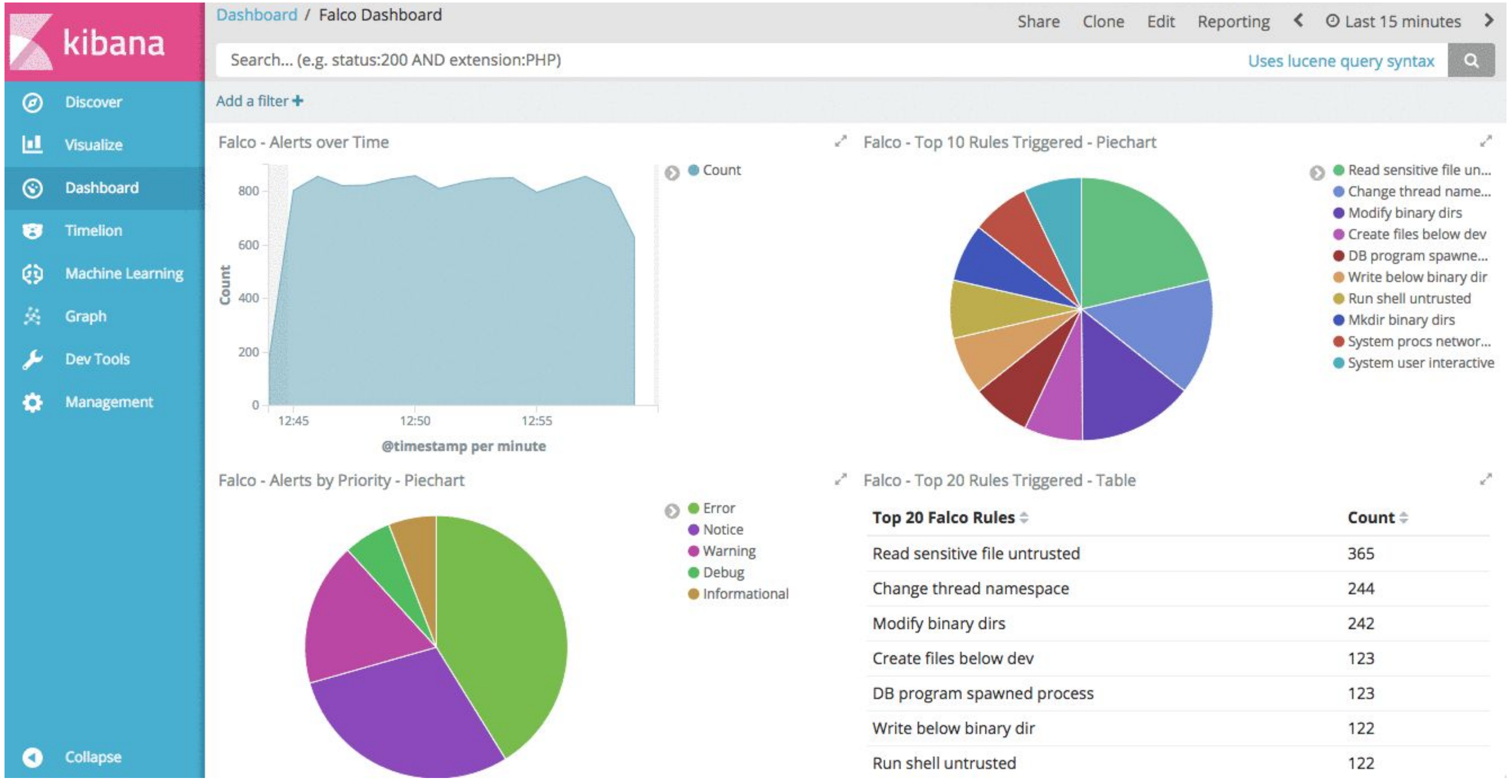
Falco rules for securing MongoDB

Falco rule

Falco rules for securing Nginx

Falco rule

SIEM

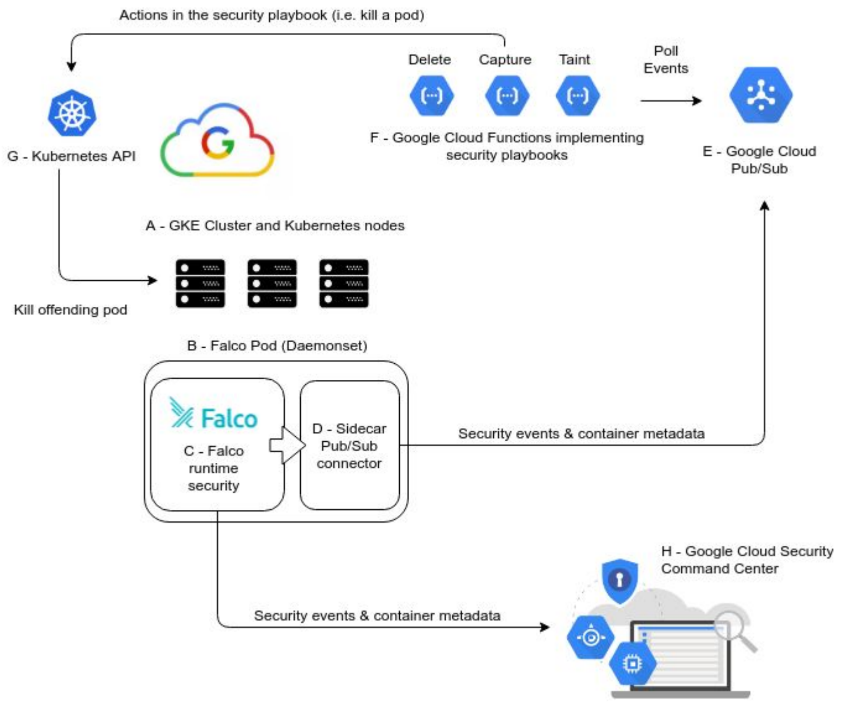


Top 20 Falco Rules	Count
Read sensitive file untrusted	365
Change thread namespace	244
Modify binary dirs	242
Create files below dev	123
DB program spawned process	123
Write below binary dir	122
Run shell untrusted	122

Response Engine

Trigger automated reactions to events
Blocking component of runtime security
Security playbooks executed as FaaS

- Taint a node NoSchedule
- Isolate pod via Network Policy
- Delete offending pod
- Scale down deployment to 0 pods
- Trigger a Sysdig capture
- Send notifications


Actions in the security playbook (i.e. kill a pod)
Delete
Capture
Taint
Poll Events
G - Kubernetes API
F - Google Cloud Functions implementing security playbooks
E - Google Cloud Pub/Sub
A - GKE Cluster and Kubernetes nodes
Kill offending pod
B - Falco Pod (Daemonset)
Falco
C - Falco runtime security
D - Sidecar Pub/Sub connector
Security events & container metadata
H - Google Cloud Security Command Center
Security events & container metadata

ADOPTERS.md

Booz Allen Hamilton
Frame.io
League
Preferral
Shopify
Sight Machine
Sumo Logic
Sysdig

And what about you? Using Falco in production? Talk about it!
https://github.com/falcosecurity/falco/blob/dev/ADOPTERS.md

Getting involved

Website
https://falco.org/

Blogs
https://falco.org/blog/
https://sysdig.com/blog/

Github
https://github.com/falcosecurity/

Slack
http://slack.sysdig.com/

Docs
https://falco.org/docs/

SecurityHub
https://securityhub.dev/

Sysdig Secure extends Falco functionality

Extending Falco Across the Lifecycle

Falco

- Runtime detection

Build | **Run** | **Respond**

Build

- Image Scanning
- Configuration Validation

Run

- Runtime prevention
  - Automated policy creation
  - Policy editor and rules library
  - Threat blocking

Respond

- Incident Response
- Forensics
- Audit

← Continuous Compliance (PCI, NIST, CIS, etc.) →

sysdig

Platform Built on an Open Foundation

Build | Run | Respond

Sysdig Secure DevOps Platform

Adds scale, workflow, K8s, and cloud context

Prometheus

Falco

sysdig

Image scanning
Vulnerability analysis

Monitoring
Infrastructure and application metrics

Runtime security
Detection rules and alerts

Forensics/troubleshooting
Deep visibility into container activity

Secure DevOps Across Cloud-Native Lifecycle

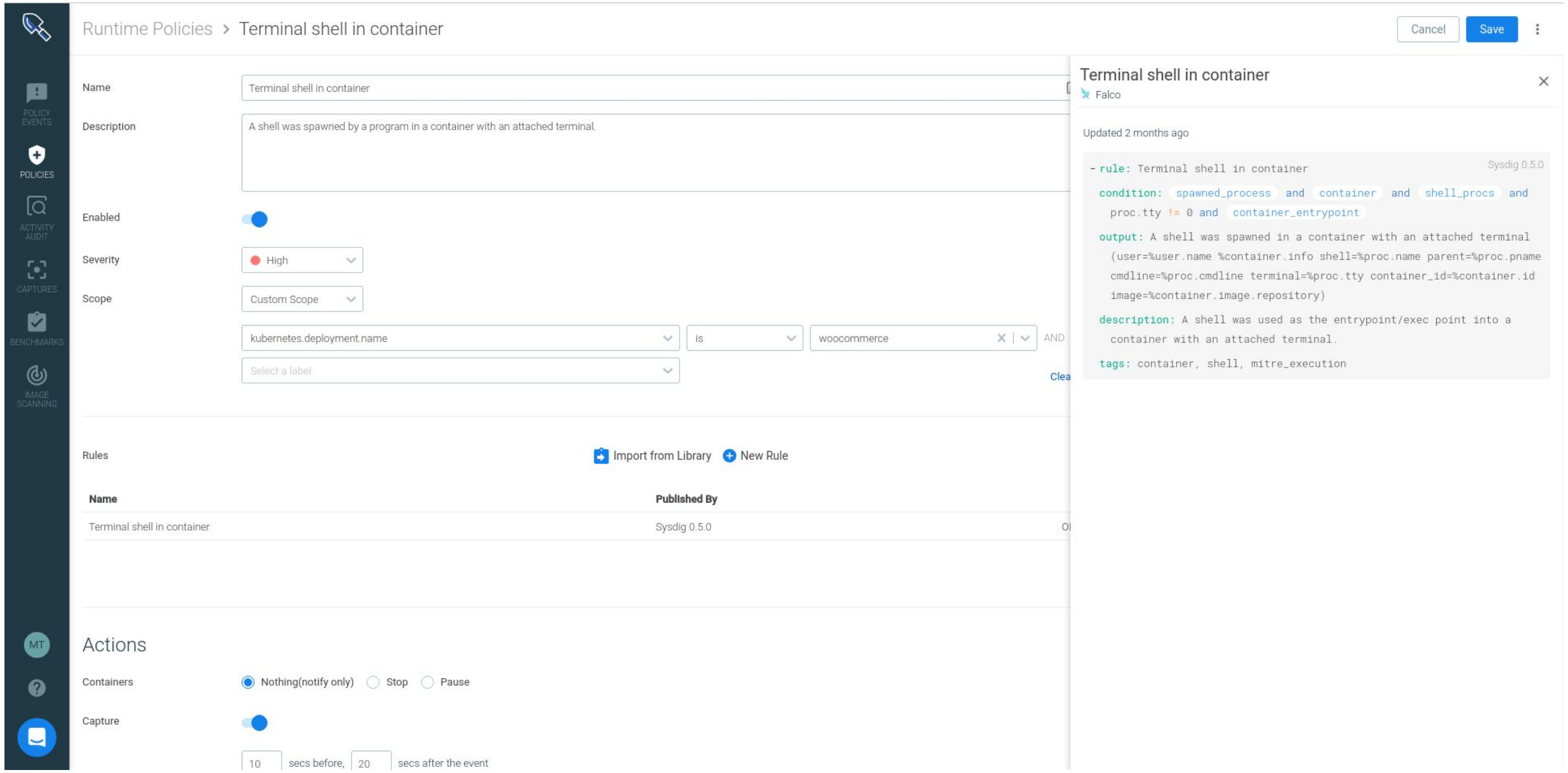
	Build	Run	Respond
	• Image Scanning	• Runtime Security • Vulnerability Reporting	• Incident Response • Forensics • Audit
Secure DevOps	• Configuration Validation	• Infrastructure Monitoring • Application Monitoring	• Troubleshooting

← Continuous Compliance (PCI, NIST, CIS, etc.) →

Unified platform for security and DevOps use cases

sysdig

Inside Sysdig Secure: Falco editor



Runtime Policies > Terminal shell in container

Name	Terminal shell in container
Description	A shell was spawned by a program in a container with an attached terminal.
Enabled	●
Severity	High
Scope	Custom Scope
	kubernetes.deployment.name · is · woocommerce · AND
	Select a label

Rules

Name	Published By
Terminal shell in container	Sysdig 0.5.0

Actions

Containers: [✓ Nothing(notify only)] Stop Pause

Capture: ●

10 secs before, 20 secs after the event

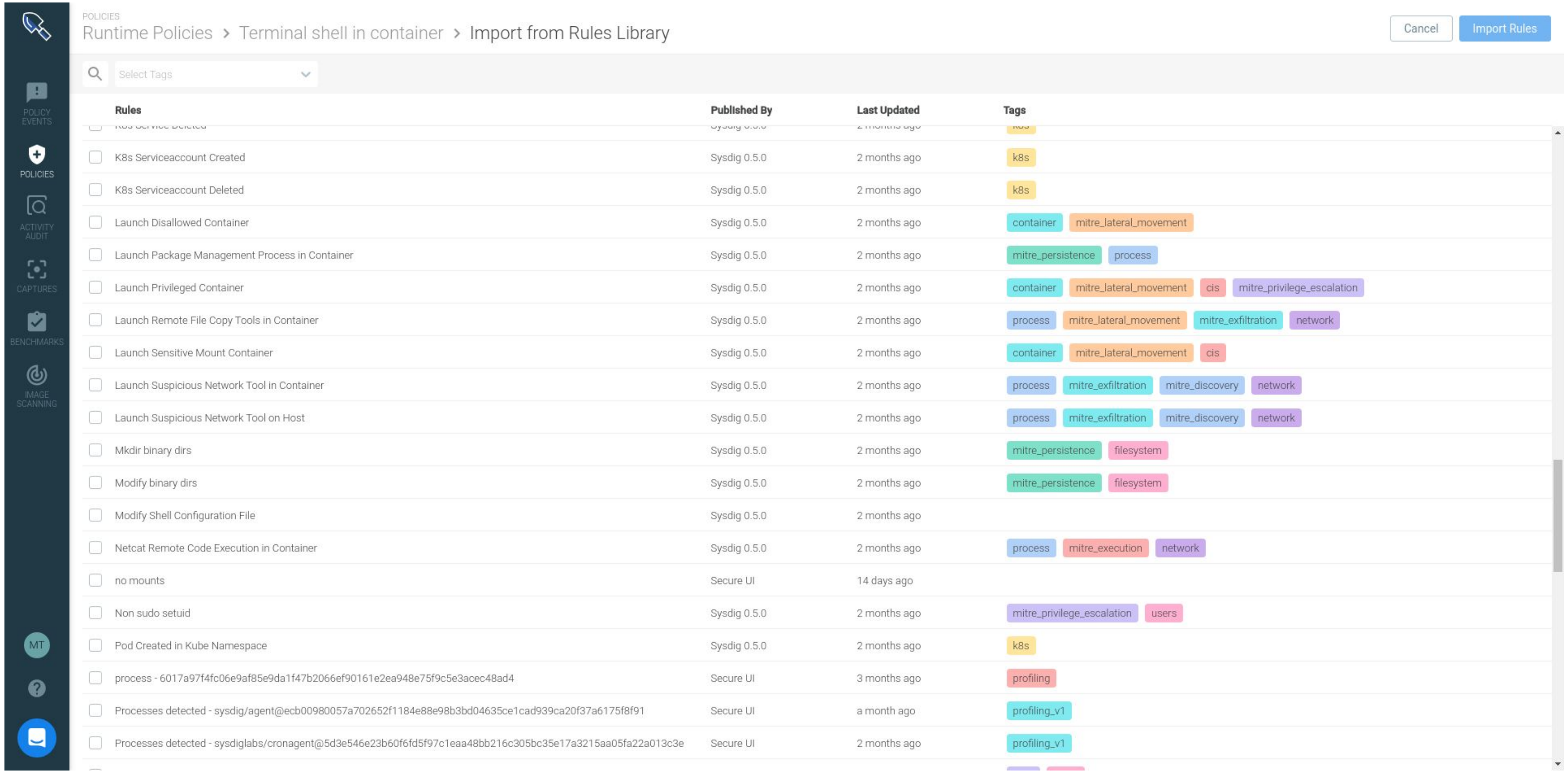
Terminal shell in container
Falco

Updated 2 months ago

Sysdig 0.5.0

```
- rule: Terminal shell in container
  condition: spawned_process and container and shell_procs and
    proc.tty != 0 and container_entrypoint
  output: A shell was spawned in a container with an attached terminal
    (user=%user.name %container.info shell=%proc.name parent=%proc.pname
    cmdline=%proc.cmdline terminal=%proc.tty container_id=%container.id
    image=%container.image.repository)
  description: A shell was used as the entrypoint/exec point into a
    container with an attached terminal.
  tags: container, shell, mitre_execution
```

Inside Sysdig Secure: Falco library



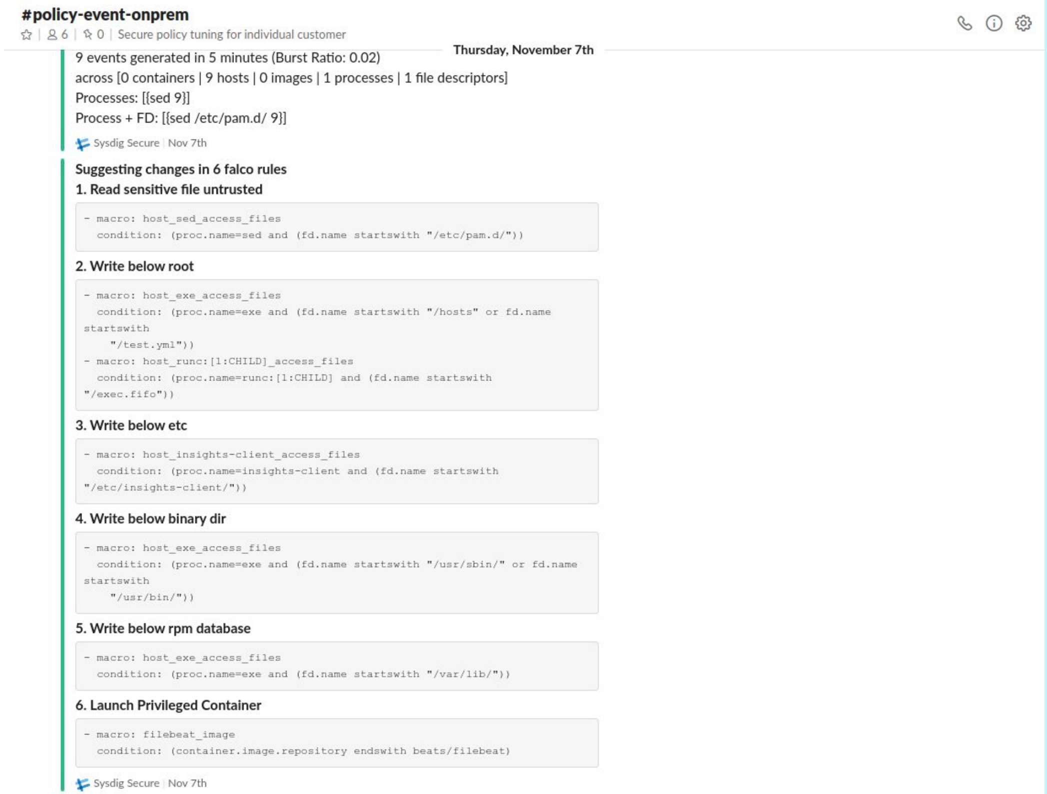
POLICIES
Runtime Policies > Terminal shell in container > Import from Rules Library

Cancel | Import Rules

Select Tags

Rules	Published By	Last Updated	Tags
K8s Serviceaccount Created	Sysdig 0.5.0	2 months ago	k8s
K8s Serviceaccount Deleted	Sysdig 0.5.0	2 months ago	k8s
Launch Disallowed Container	Sysdig 0.5.0	2 months ago	container, mitre_lateral_movement
Launch Package Management Process in Container	Sysdig 0.5.0	2 months ago	mitre_persistence, process
Launch Privileged Container	Sysdig 0.5.0	2 months ago	container, mitre_lateral_movement, cis, mitre_privilege_escalation
Launch Remote File Copy Tools in Container	Sysdig 0.5.0	2 months ago	process, mitre_lateral_movement, mitre_exfiltration, network
Launch Sensitive Mount Container	Sysdig 0.5.0	2 months ago	container, mitre_lateral_movement, cis
Launch Suspicious Network Tool in Container	Sysdig 0.5.0	2 months ago	process, mitre_exfiltration, mitre_discovery, network
Launch Suspicious Network Tool on Host	Sysdig 0.5.0	2 months ago	process, mitre_exfiltration, mitre_discovery, network
Mkdir binary dirs	Sysdig 0.5.0	2 months ago	mitre_persistence, filesystem
Modify binary dirs	Sysdig 0.5.0	2 months ago	mitre_persistence, filesystem
Modify Shell Configuration File	Sysdig 0.5.0	2 months ago	
Netcat Remote Code Execution in Container	Sysdig 0.5.0	2 months ago	process, mitre_execution, network
no mounts	Secure UI	14 days ago	
Non sudo setuid	Sysdig 0.5.0	2 months ago	mitre_privilege_escalation, users
Pod Created in Kube Namespace	Sysdig 0.5.0	2 months ago	k8s
process - 6017a97f4fc06e9af85e9da1f47b2066ef90161e2ea948e75f9c5e3acec48ad4	Secure UI	3 months ago	profiling
Processes detected - sysdig/agent@ecb00980057a702652f1184e88e98b3bd04635ce1cad939ca20f37a6175f8f91	Secure UI	a month ago	profiling_v1
Processes detected - sysdiglabs/cronagent@5d3e546e23b60f6fd5f97c1eaa48bb216c305bc35e17a3215aa05fa22a013c3e	Secure UI	2 months ago	profiling_v1

Inside Sysdig Secure: Falco tuning



#policy-event-onprem

☆ | 6 | 0 | Secure policy tuning for individual customer

Thursday, November 7th

9 events generated in 5 minutes (Burst Ratio: 0.02)
across [0 containers | 9 hosts | 0 images | 1 processes | 1 file descriptors]
Processes: [{sed 9}]
Process + FD: [{sed /etc/pam.d/ 9}]

Sysdig Secure | Nov 7th

Suggesting changes in 6 falco rules

1. Read sensitive file untrusted

```
- macro: host_sed_access_files
  condition: (proc.name=sed and (fd.name startswith "/etc/pam.d/"))
```

2. Write below root

```
- macro: host_exe_access_files
  condition: (proc.name=exe and (fd.name startswith "/hosts" or fd.name
startswith
    "/test.yml"))
- macro: host_runc:[1:CHILD]_access_files
  condition: (proc.name=runc:[1:CHILD] and (fd.name startswith
"/exec.fifo"))
```

3. Write below etc

```
- macro: host_insights-client_access_files
  condition: (proc.name=insights-client and (fd.name startswith
"/etc/insights-client/"))
```

4. Write below binary dir

```
- macro: host_exe_access_files
  condition: (proc.name=exe and (fd.name startswith "/usr/sbin/" or fd.name
startswith
    "/usr/bin/"))
```

5. Write below rpm database

```
- macro: host_exe_access_files
  condition: (proc.name=exe and (fd.name startswith "/var/lib/"))
```

6. Launch Privileged Container

```
- macro: filebeat_image
  condition: (container.image.repository endswith beats/filebeat)
```

Sysdig Secure | Nov 7th

- Makes suggestions on what to change on the Falco rules (default rules or your own rules).
- You decide what you merge or not (easy with Policy Editor!).
- This Falco rule triggers all the time, these are false positives, it’s very noisy creating alert fatigue, how we can fix that at scale for all policies?

Inside Sysdig Secure: Profiling

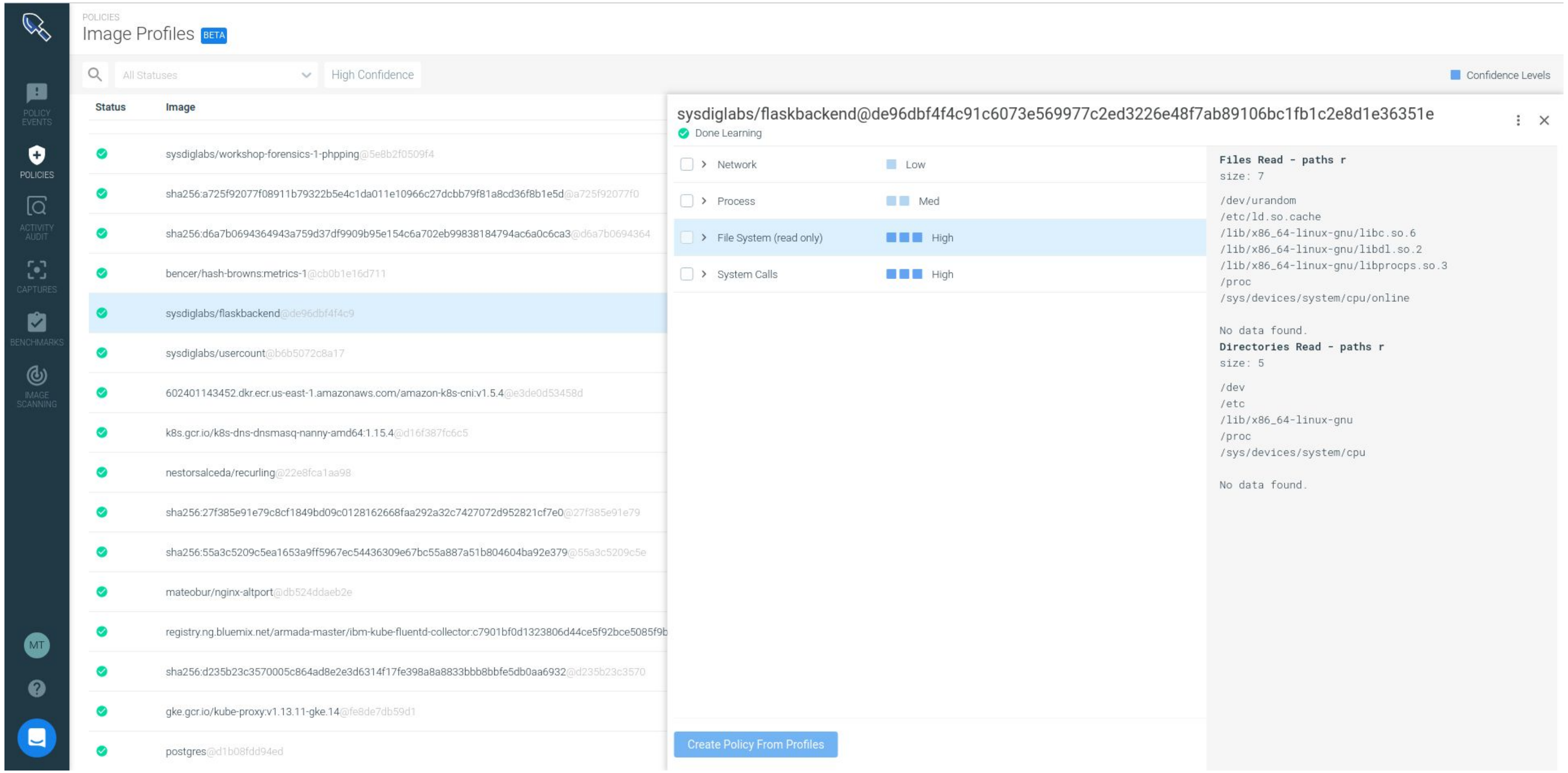


Image Profiles BETA

Status	Image
✓	sysdiglabs/workshop-forensics-1-phpping@5e8b2f0509f4
✓	sha256:a725f92077f08911b79322b5e4c1da011e10966c27dcbb79f81a8cd36f8b1e5d@a725f92077f0
✓	sha256:d6a7b0694364943a759d37df9909b95e154c6a702eb99838184794ac6a0c6ca3@d6a7b0694364
✓	bencer/hash-browns:metrics-1@cb0b1e16d711
✓	sysdiglabs/flaskbackend@de96dbf4f4c9
✓	sysdiglabs/usercount@b6b5072c8a17
✓	602401143452.dkr.ecr.us-east-1.amazonaws.com/amazon-k8s-cni:v1.5.4@e3de0d53458d
✓	k8s.gcr.io/k8s-dns-dnsmasq-nanny-amd64:1.15.4@d16f387fc6c5
✓	nestorsalceda/recurling@22e8fca1aa98
✓	sha256:27f385e91e79c8cf1849bd09c0128162668faa292a32c7427072d952821cf7e0@27f385e91e79
✓	sha256:55a3c5209c5ea1653a9ff5967ec54436309e67bc55a887a51b804604ba92e379@55a3c5209c5e
✓	mateobur/nginx-altport@db524ddaeb2e
✓	registry.ng.bluemix.net/armada-master/ibm-kube-fluentd-collector:c7901bf0d1323806d44ce5f92bce5085f9b
✓	sha256:d235b23c3570005c864ad8e2e3d6314f17fe398a8a8833bbb8bbfe5db0aa6932@d235b23c3570
✓	gke.gcr.io/kube-proxy:v1.13.11-gke.14@fe8de7db59d1
✓	postgres@d1b08fdd94ed

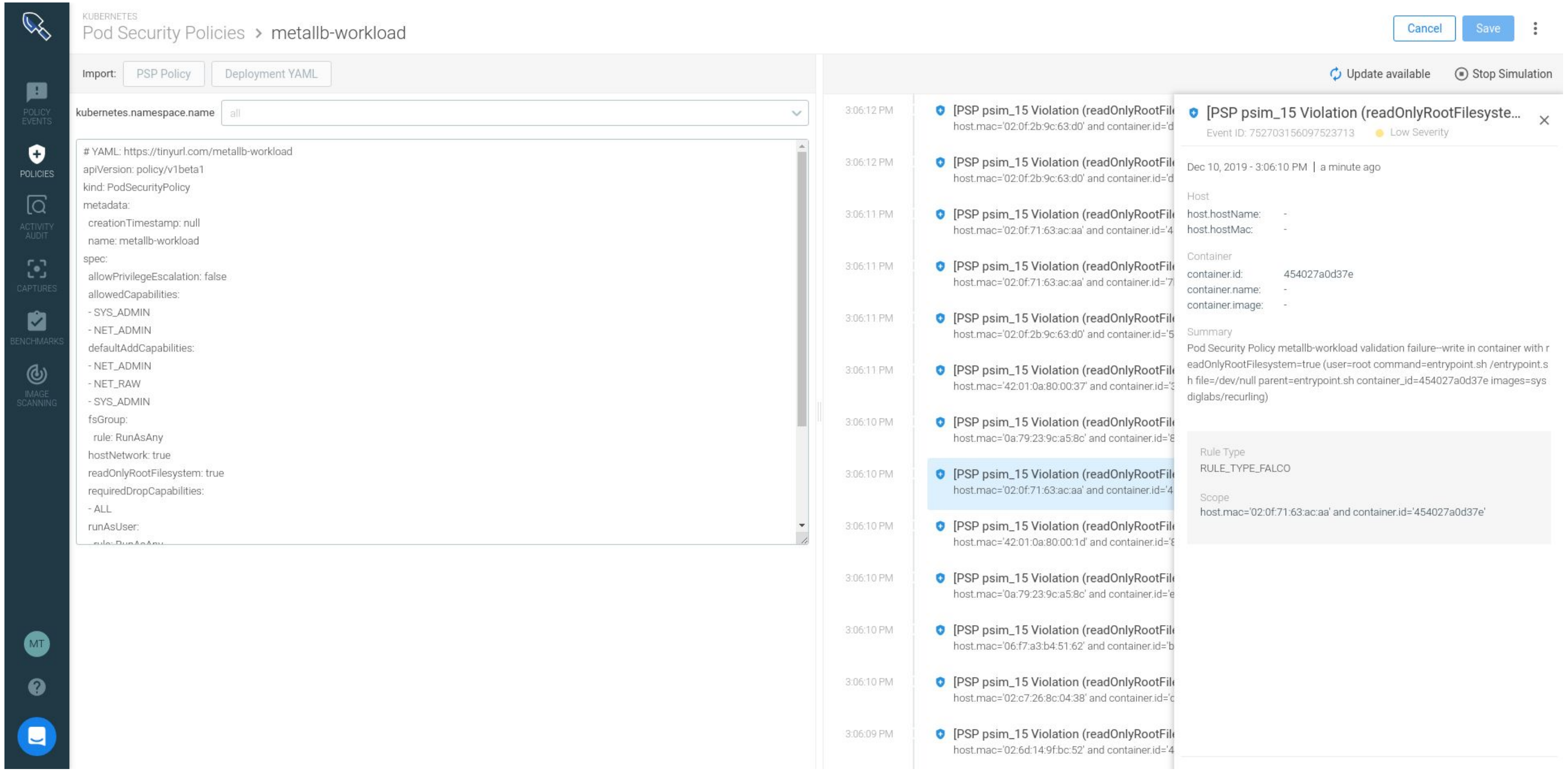
sysdiglabs/flaskbackend@de96dbf4f4c91c6073e569977c2ed3226e48f7ab89106bc1fb1c2e8d1e36351e

✓ Done Learning

Network	Low
Process	Med
File System (read only)	High
System Calls	High

```
Files Read - paths r
size: 7

/dev/urandom
/etc/ld.so.cache
/lib/x86_64-linux-gnu/libc.so.6
/lib/x86_64-linux-gnu/libdl.so.2
/lib/x86_64-linux-gnu/libprocps.so.3
/proc
/sys/devices/system/cpu/online

No data found.
Directories Read - paths r
size: 5

/dev
/etc
/lib/x86_64-linux-gnu
/proc
/sys/devices/system/cpu

No data found.
```

Inside Sysdig Secure: Policy Advisor



kubernetes.namespace.name: all

```
# YAML: https://tinyurl.com/metallb-workload
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  creationTimestamp: null
  name: metallb-workload
spec:
  allowPrivilegeEscalation: false
  allowedCapabilities:
  - SYS_ADMIN
  - NET_ADMIN
  defaultAddCapabilities:
  - NET_ADMIN
  - NET_RAW
  - SYS_ADMIN
  fsGroup:
    rule: RunAsAny
  hostNetwork: true
  readOnlyRootFilesystem: true
  requiredDropCapabilities:
  - ALL
  runAsUser:
    rule: RunAsAny
```

```
3:06:12 PM  [PSP psim_15 Violation (readOnlyRootFil
            host.mac='02:0f:2b:9c:63:d0' and container.id='d
3:06:12 PM  [PSP psim_15 Violation (readOnlyRootFil
            host.mac='02:0f:2b:9c:63:d0' and container.id='d
3:06:11 PM  [PSP psim_15 Violation (readOnlyRootFil
            host.mac='02:0f:71:63:ac:aa' and container.id='4
3:06:11 PM  [PSP psim_15 Violation (readOnlyRootFil
            host.mac='02:0f:71:63:ac:aa' and container.id='7
3:06:11 PM  [PSP psim_15 Violation (readOnlyRootFil
            host.mac='02:0f:2b:9c:63:d0' and container.id='5
3:06:11 PM  [PSP psim_15 Violation (readOnlyRootFil
            host.mac='42:01:0a:80:00:37' and container.id='3
3:06:10 PM  [PSP psim_15 Violation (readOnlyRootFil
            host.mac='0a:79:23:9c:a5:8c' and container.id='8
3:06:10 PM  [PSP psim_15 Violation (readOnlyRootFil
            host.mac='02:0f:71:63:ac:aa' and container.id='4
3:06:10 PM  [PSP psim_15 Violation (readOnlyRootFil
            host.mac='42:01:0a:80:00:1d' and container.id='8
3:06:10 PM  [PSP psim_15 Violation (readOnlyRootFil
            host.mac='0a:79:23:9c:a5:8c' and container.id='e
3:06:10 PM  [PSP psim_15 Violation (readOnlyRootFil
            host.mac='06:f7:a3:b4:51:62' and container.id='b
3:06:10 PM  [PSP psim_15 Violation (readOnlyRootFil
            host.mac='02:c7:26:8c:04:38' and container.id='c
3:06:09 PM  [PSP psim_15 Violation (readOnlyRootFil
            host.mac='02:6d:14:9f:bc:52' and container.id='4
```

[PSP psim_15 Violation (readOnlyRootFilesyste...
Event ID: 752703156097523713 ● Low Severity

Dec 10, 2019 - 3:06:10 PM | a minute ago

Host

host.hostName:	-
host.hostMac:	-

Container

container.id:	454027a0d37e
container.name:	-
container.image:	-

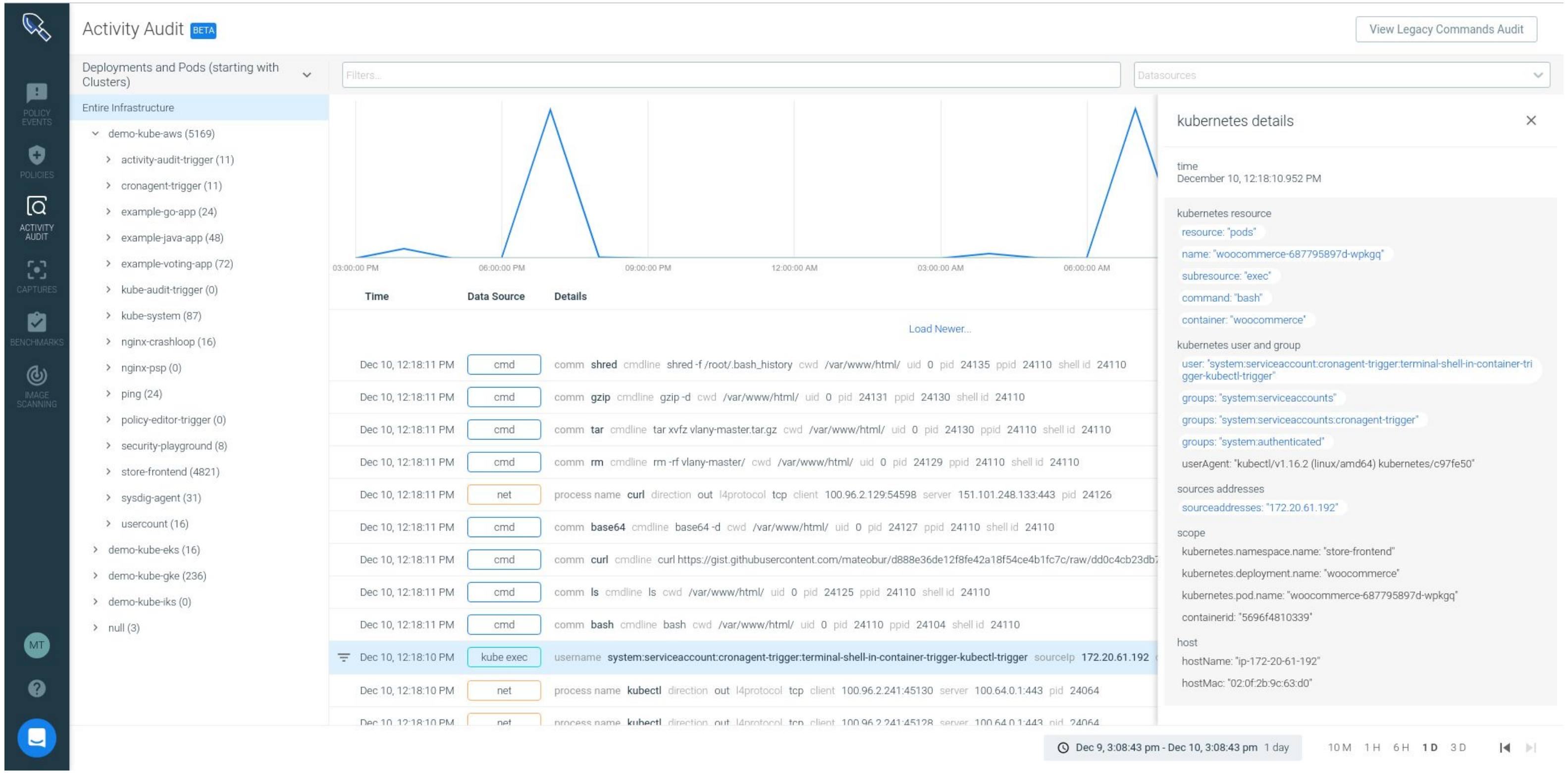
Summary

Pod Security Policy metallb-workload validation failure--write in container with readOnlyRootFilesystem=true (user=root command=entrypoint.sh /entrypoint.sh file=/dev/null parent=entrypoint.sh container_id=454027a0d37e images=sysdiglabs/recurling)

Rule Type
RULE_TYPE_FALCO

Scope
host.mac='02:0f:71:63:ac:aa' and container.id='454027a0d37e'

Cloud-native Incident Response / Forensics



Entire Infrastructure
- demo-kube-aws (5169)
  - activity-audit-trigger (11)
  - cronagent-trigger (11)
  - example-go-app (24)
  - example-java-app (48)
  - example-voting-app (72)
  - kube-audit-trigger (0)
  - kube-system (87)
  - nginx-crashloop (16)
  - nginx-psp (0)
  - ping (24)
  - policy-editor-trigger (0)
  - security-playground (8)
  - store-frontend (4821)
  - sysdig-agent (31)
  - usercount (16)
- demo-kube-eks (16)
- demo-kube-gke (236)
- demo-kube-iks (0)
- null (3)

Time	Data Source	Details
Dec 10, 12:18:11 PM	cmd	comm shred cmdline shred -f /root/.bash_history cwd /var/www/html/ uid 0 pid 24135 ppid 24110 shell id 24110
Dec 10, 12:18:11 PM	cmd	comm gzip cmdline gzip -d cwd /var/www/html/ uid 0 pid 24131 ppid 24130 shell id 24110
Dec 10, 12:18:11 PM	cmd	comm tar cmdline tar xvfz vlany-master.tar.gz cwd /var/www/html/ uid 0 pid 24130 ppid 24110 shell id 24110
Dec 10, 12:18:11 PM	cmd	comm rm cmdline rm -rf vlany-master/ cwd /var/www/html/ uid 0 pid 24129 ppid 24110 shell id 24110
Dec 10, 12:18:11 PM	net	process name curl direction out l4protocol tcp client 100.96.2.129:54598 server 151.101.248.133:443 pid 24126
Dec 10, 12:18:11 PM	cmd	comm base64 cmdline base64 -d cwd /var/www/html/ uid 0 pid 24127 ppid 24110 shell id 24110
Dec 10, 12:18:11 PM	cmd	comm curl cmdline curl https://gist.githubusercontent.com/mateobur/d888e36de12f8fe42a18f54ce4b1fc7c/raw/dd0c4cb23db7
Dec 10, 12:18:11 PM	cmd	comm ls cmdline ls cwd /var/www/html/ uid 0 pid 24125 ppid 24110 shell id 24110
Dec 10, 12:18:11 PM	cmd	comm bash cmdline bash cwd /var/www/html/ uid 0 pid 24110 ppid 24104 shell id 24110
Dec 10, 12:18:10 PM	kube exec	username system:serviceaccount:cronagent-trigger:terminal-shell-in-container-trigger-kubectl-trigger sourceIp 172.20.61.192
Dec 10, 12:18:10 PM	net	process name kubectl direction out l4protocol tcp client 100.96.2.241:45130 server 100.64.0.1:443 pid 24064
Dec 10, 12:18:10 PM	net	process name kubectl direction out l4protocol tcp client 100.96.2.241:45128 server 100.64.0.1:443 pid 24064

kubernetes details

time
December 10, 12:18:10.952 PM

kubernetes resource
- resource: "pods"
- name: "woocommerce-687795897d-wpkgq"
- subresource: "exec"
- command: "bash"
- container: "woocommerce"

kubernetes user and group
- user: "system:serviceaccount:cronagent-trigger:terminal-shell-in-container-trigger-kubectl-trigger"
- groups: "system:serviceaccounts"
- groups: "system:serviceaccounts:cronagent-trigger"
- groups: "system:authenticated"
- userAgent: "kubectl/v1.16.2 (linux/amd64) kubernetes/c97fe50"

sources addresses
- sourceaddresses: "172.20.61.192"

scope
- kubernetes.namespace.name: "store-frontend"
- kubernetes.deployment.name: "woocommerce"
- kubernetes.pod.name: "woocommerce-687795897d-wpkgq"
- containerid: "5696f4810339"

host
- hostName: "ip-172-20-61-192"
- hostMac: "02:0f:2b:9c:63:d0"

sysdig

Upcoming events

- Join us Dec. 19 for **Star Wars** premieres:

  https://go.sysdig.com/starwars2019

- In 2020

  RSA ~ February 24-28 in San Francisco

  KubeCon ~ March 31-April 2 in Amsterdam

  Red Hat Summit ~ April 27-29 in San Francisco

Learn more

- Falco

  https://falco.org/

- Container Usage Report and webinar

  https://sysdig.com/resources/papers/2019-container-usage-report/

- Sysdig blog

  https://sysdig.com/blog/

- What’s new in Kubernetes 1.17

  https://sysdig.com/blog/whats-new-kubernetes-1-17/

sysdig

Dig deeper