
KubeCon
CloudNativeCon
Europe 2023
TiKV

Code of Conduct

KubeCon CloudNativeCon Europe 2023

Remember the **Golden Rule**: Treat others as you would want to be treated - with kindness and respect

Scan the QR code to access and review the **CNCF Code of Conduct**:

Virtual Audience Closed Captioning

KubeCon CloudNativeCon Europe 2023

Closed captioning for the virtual audience is available during each session through Wordly. The Wordly functionality can be found under the “Translations” tab on the session page.

Wordly will default to English. If another language is needed, simply click the dropdown at the bottom of the “Translations” tab and choose from one of 26+ languages available so you don’t miss a beat from our presenters.

*Note: Closed captioning is ONLY available during the scheduled live sessions and will not be available for the recordings on-demand within the virtual conference platform.

Session Q+A

- Virtual attendees may submit questions to speakers through the CNCF Slack channel: **#2-Kubecon-sessions**
- Please create a thread and tag the speaker(s) with questions about their talk.
- Questions will be answered by the speaker and/or other community members after the session concludes.

Sponsor Shout-Out!

KubeCon CloudNativeCon Europe 2023

Thank you to our Session Recording Sponsor:


KubeCon
CloudNativeCon
Europe 2023
Enjoy the Session!

Cluster Grey Zone

Risks in managed cluster middleware

Shay Berkovich

Threat Research @ Wiz

Barak Sharoni

@whoami

- Wiz Threat Research
  Previously in BlueCoat, Symantec and BlackBerry
- Previous research in
  - Published papers and journal articles in Runtime Verification (??what??)
  - UBCIS – benchmark for container image scanning
  - Falco bypasses
  - Everything Kubernetes security
- Usenix and EuroSys AEC member


KubeCon
CloudNativeCon
Europe 2023
Intro
Definition

The Premise

- The core K8s cluster components are a necessity
- The user workloads are a necessity
- What about everything else?

Components - Examples

YES	NO
container-watcher on GKE	kube-proxy
osm-controller on AKS	coredns
aws-node on EKS	OMI agent

Components and Where to Find Them

- A parent list: https://github.com/kubernetes/kubernetes/tree/master/cluster/addons/
- Azure own repo: https://github.com/Azure/aks-engine/tree/master/examples/addons
- Other non-centralized sources

Components - SBOM

- Components in numbers:

GKE v1.25	AKS v1.25	EKS v1.25
5 deployments + 4 daemonsets	6 deployments + 7 daemonsets	1 deployment + 2 daemonsets*

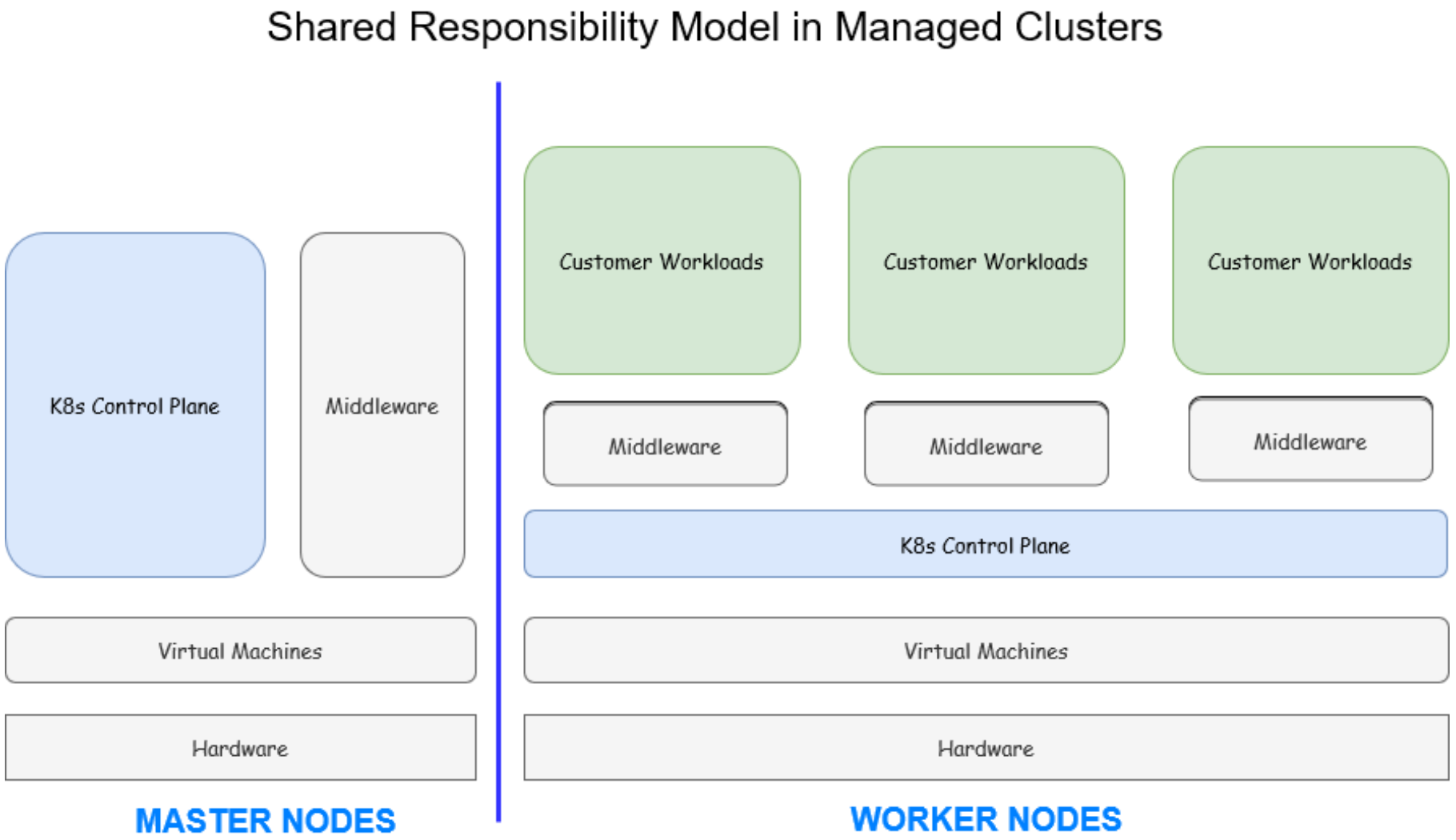
(total 25 daemonsets, replicasets and statefulsets)

- Only native K8s components are considered in this table

- More host-level components

* EBS CSI driver and EBS CSI controller are considered even though they are not ON by default *

The Premise


Shared Responsibility Model in Managed Clusters
K8s Control Plane
Middleware
Virtual Machines
Hardware
MASTER NODES
Customer Workloads
Customer Workloads
Customer Workloads
Middleware
Middleware
Middleware
K8s Control Plane
Virtual Machines
Hardware
WORKER NODES

The Problems - Upgrade

- Cluster users’ focus on workload security, not middleware security
- Cluster middleware is a part of master and worker nodes as well
- The vulnerability patching process is unclear

CSP wants to upgrade but requires user action

user wants to upgrade but the component is controlled by CSP

WIZ


KubeCon
CloudNativeCon
Europe 2023
Intro
Definition
Expectations

This talk is NOT a …

A scoped vulnerability research ✖

A security audit of Kubernetes components ✖

A threat model of Kubernetes components ✖

This talk is about…

A scoped vulnerability research ✖

A security audit of Kubernetes components ✖

A threat model of Kubernetes components ✖

A bit of everything above ✓

A risk assessment of a previously unnoticed surface ✓

An attempt to draw conclusions and an initial call-to-arms ✓


KubeCon
CloudNativeCon
Europe 2023
Intro
Definition
Expectations
Hypothesis

The Hypothesis

Middleware increases attack surface

The Hypothesis

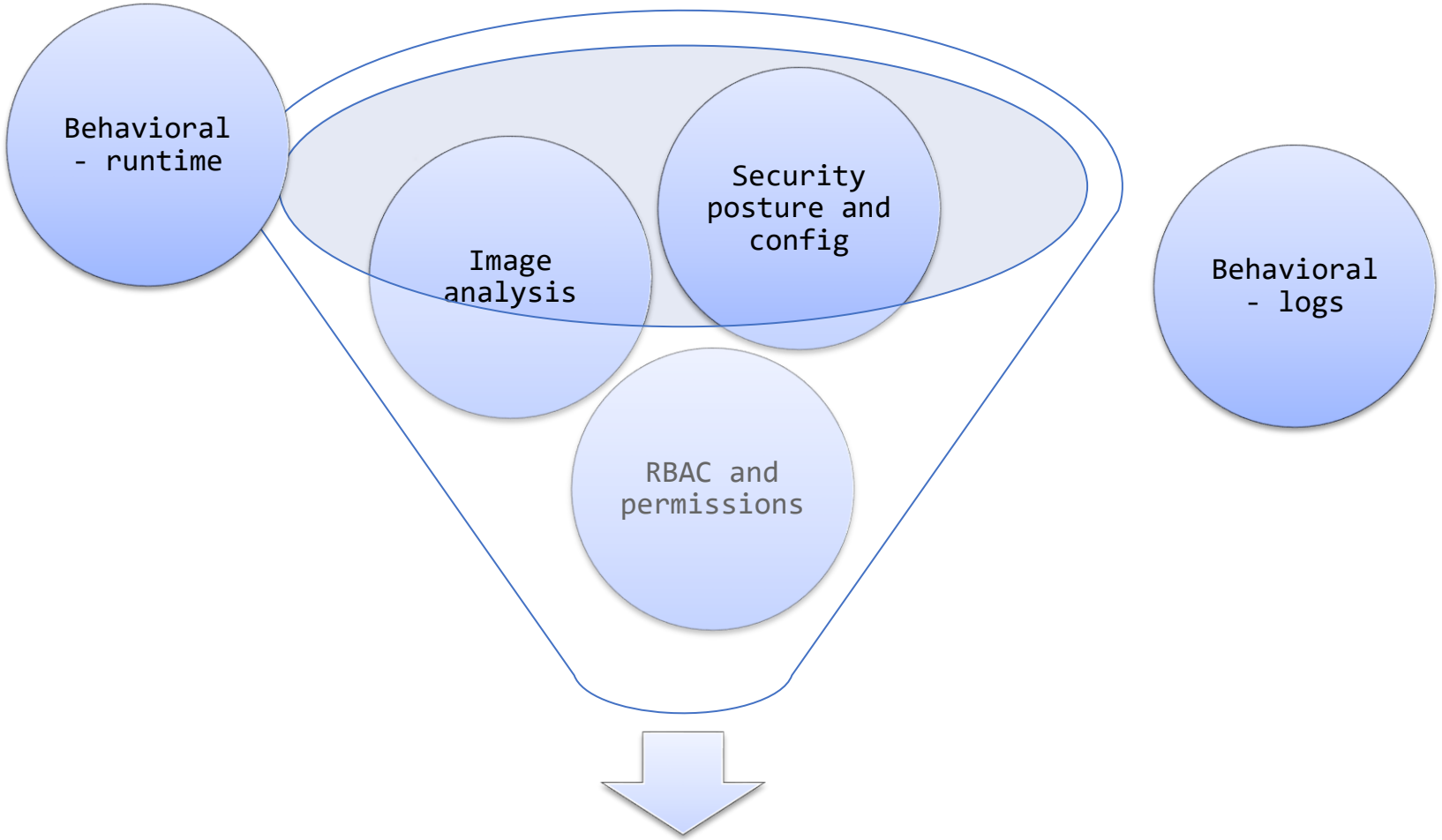
Middleware increases attack surface significantly

The Hypothesis

Middleware increases risk in a non-trivial way


KubeCon
CloudNativeCon
Europe 2023
Intro
Definition
Expectations
Hypothesis
Method

The Approach



SBOM and Basic Security Posture

KubeCon CloudNativeCon Europe 2023

- Components in numbers:

GKE v1.25	AKS v1.25	EKS v1.25
5 deployments + 4 daemonsets	6 deployments + 7 daemonsets	1 deployment + 2 daemonsets*

(total 25 daemonsets, replicasets and statefulsets)

- Shared namespaces: 32% (8 out of 25)
- Privileged / added capabilities: 32% (8 out of 25)
- Mounted sensitive host volumes: 32% (8 out of 25)

* EBS CSI driver and EBS CSI controller are considered even though they are not ON by default *
** Only native K8s components are considered **

WIZ

Image Analytics

- Middleware images have lots of vulnerabilities (wizcli and grype)

```
shay@Ubuntu-2204:~$ grype registry.k8s.io/node-problem-detector/node-problem-detector:v0.8.12
✓ Vulnerability DB [no update availab]
New version of grype is available: 0.5
✓ Parsed image
✓ Cataloged packages [173 packages]
✓ Scanned image [227 vulnerabilities]
```

```
shay@Ubuntu-2204:~$ grype registry.k8s.io/networking/ip-masq-agent-amd64:v2.6.0 | grep -i critical | wc -l
✓ Vulnerability DB [no update available]
New version of grype is available: 0.60.0 (currently running: 0.57.1)
✓ Parsed image
✓ Cataloged packages [83 packages]
✓ Scanned image [182 vulnerabilities]
```

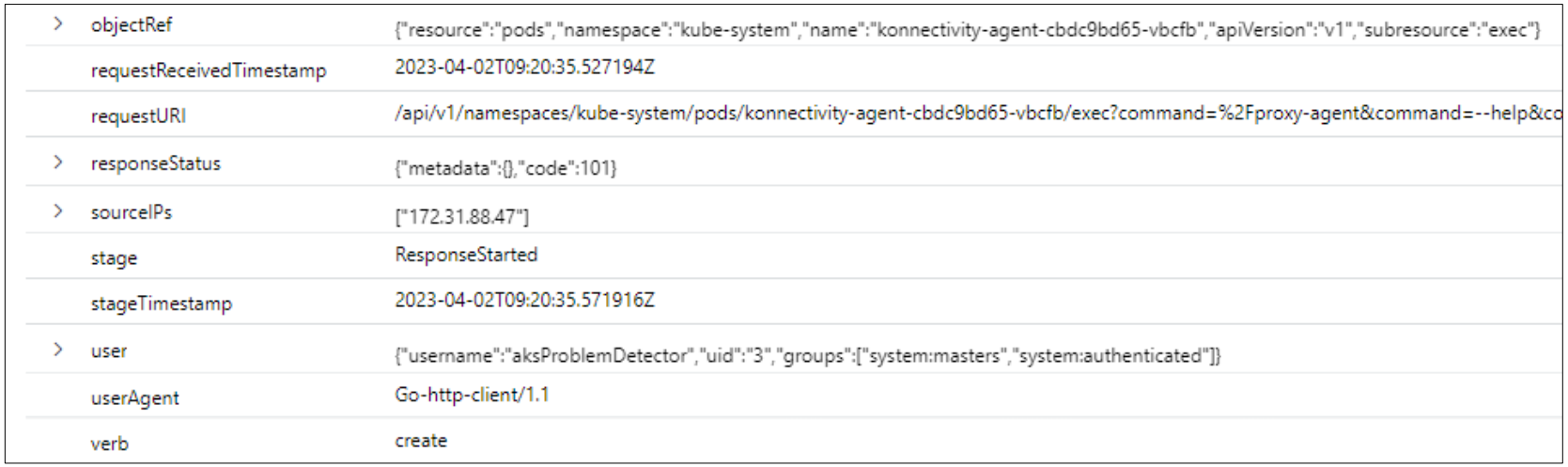
- But so do the core images in control plane
- # of vulnerabilities is proportional to # of packages
- Cannot state middleware images are worse of than core components

Behavioural – logs and runtime

KubeCon CloudNativeCon Europe 2023

Logs are always an interesting source:

- Unexpected principals acting
- Unexpected permissions
- Discrepancies between the CSPs



objectRef	{"resource":"pods","namespace":"kube-system","name":"konnectivity-agent-cbdc9bd65-vbcfb","apiVersion":"v1","subresource":"exec"}
requestReceivedTimestamp	2023-04-02T09:20:35.527194Z
requestURI	/api/v1/namespaces/kube-system/pods/konnectivity-agent-cbdc9bd65-vbcfb/exec?command=%2Fproxy-agent&command=--help&co
responseStatus	{"metadata":{},"code":101}
sourceIPs	["172.31.88.47"]
stage	ResponseStarted
stageTimestamp	2023-04-02T09:20:35.571916Z
user	{"username":"aksProblemDetector","uid":"3","groups":["system:masters","system:authenticated"]}
userAgent	Go-http-client/1.1
verb	create

WIZ

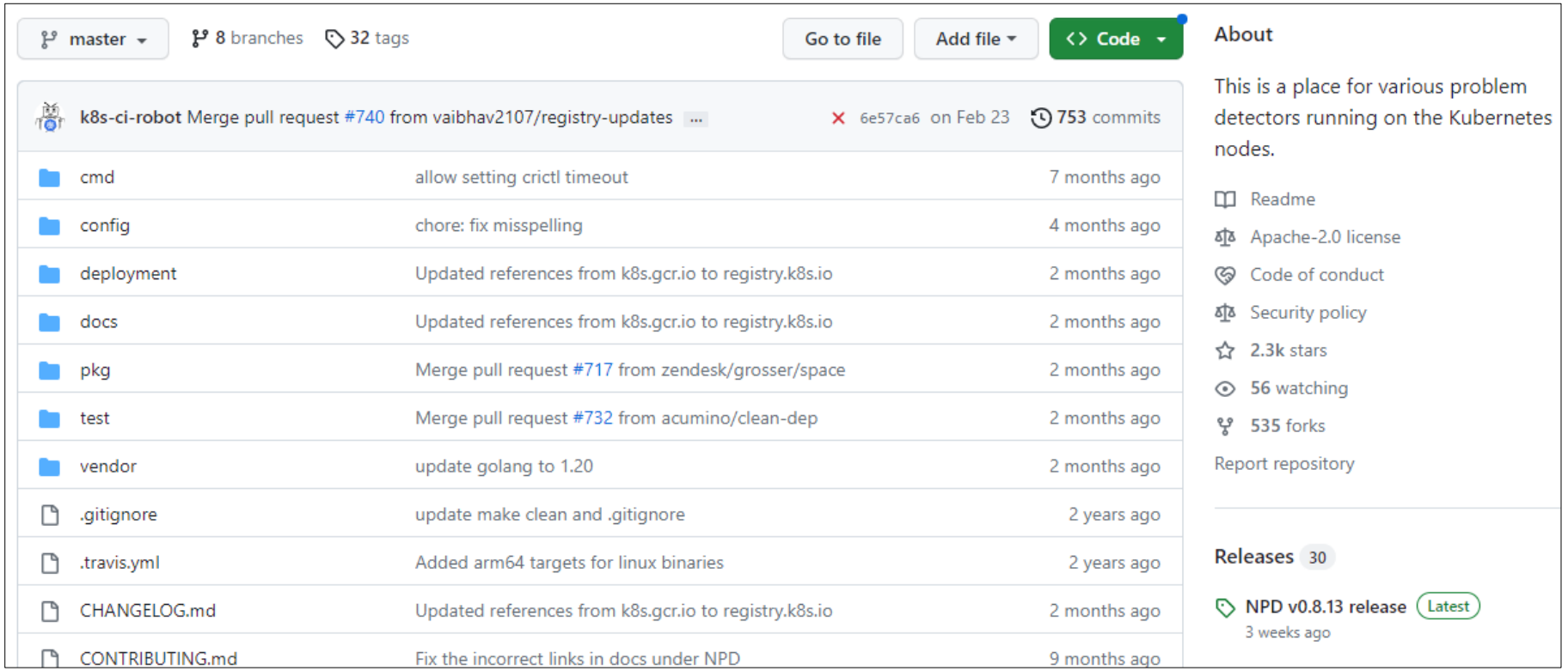

KubeCon
CloudNativeCon
Europe 2023
Intro
Definition
Expectations
Hypothesis
Method
Use case 1

Use case – node-problem-detector

node-problem-detector – TL;DR

KubeCon CloudNativeCon Europe 2023

Deployment yamls
Parent repo



WIZ

node-problem-detector - Versions

- AKS/GKE – starting v1.?? runs as host service, before that a DaemonSet

- EKS – best practices guide recommends the vanilla DaemonSet deployment

```
gke-cluster-1-default-pool-0ff849b1-3vo6 / # ps aux | grep -i node-problem
root        1826  0.0  0.7 1459836 29912 ?       Ssl  Mar03  31:25 /home/kubernetes/bin/node-problem-d
node-problem-detector/config/docker-monitor.json,/home/kubernetes/node-problem-detector/config/systemd
ion-checker-monitor.json --config.system-stats-monitor=/home/kubernetes/node-problem-detector/config/s
ernetes/node-problem-detector/config/systemd-monitor-counter.json,/home/kubernetes/npd-custom-plugins/
20256 --apiserver-overri
root     2034506  0.0
gke-cluster-1-default-p
```

```
root@aks-agentpool-41019565-vmss000002:/# ps aux | grep -i node-problem
root     2973878  0.0  0.0   3468  1780 pts/19   S+   05:31   0:00 grep --color=auto -i node-problem
root     3181062  0.0  0.0   2888  1008 ?        Ss   Apr08   0:00 /bin/sh /usr/local/bin/node-problem-detector-startup.sh
root     3181089  0.1  0.5 2001900 88944 ?       Sl   Apr08   6:24 /usr/local/bin/node-problem-detector --config.system-log-
r.d/system-log-monitor/kernel-monitor.json,/etc/node-problem-detector.d/system-log-monitor/systemd-monitor.json --config.cus
m-detector.d/custom-plugin-monitor/dns-problem-monitor.json,/etc/node-problem-detector.d/custom-plugin-monitor/custom-runtim
detector.d/custom-plugin-monitor/custom-kubelet-monitor.json,/etc/node-problem-detector.d/custom-plugin-monitor/custom-sched
nitor.json,/etc/node-problem-detector.d/custom-plugin-monitor/custom-scheduledevents-consolidated-preempt-plugin-monitor.jso
etc/node-problem-detector.d/custom-plugin-monitor/kernel-monitor-counter.json,/etc/node-problem-detector.d/custom-plugin-mon
nitor/system-stats-monitor.json --prometheus-address 0.0.0.0 --apiserver-override https://shayb-aks-kdr-test-dns-30b522c5.hc
```

node-problem-detector - Versions

On AKS v1.25:

Released on Sep 1 2021

Released on Dec 5 2021

```
root@aks-agentpool-41019565-vmss000002:/# node-problem-detector --version
v0.8.10
root@aks-agentpool-41019565-vmss000002:/# node-exporter --version
node_exporter, version 1.3.1 (branch: HEAD, revision: a2321e7b940ddcff26873612bccdf7cd4c42b6b6)
  build user:       root@bc5e8ad42a2c
  build date:       20220208-21:30:25
  go version:       go1.17.6
  platform:         linux/amd64
```

On GKE v1.25:

```
gke-cluster-1-default-pool-0ff849b1-3vo6 / # node-problem-detector --version
0.8.10
```

node-problem-detector - Exploit

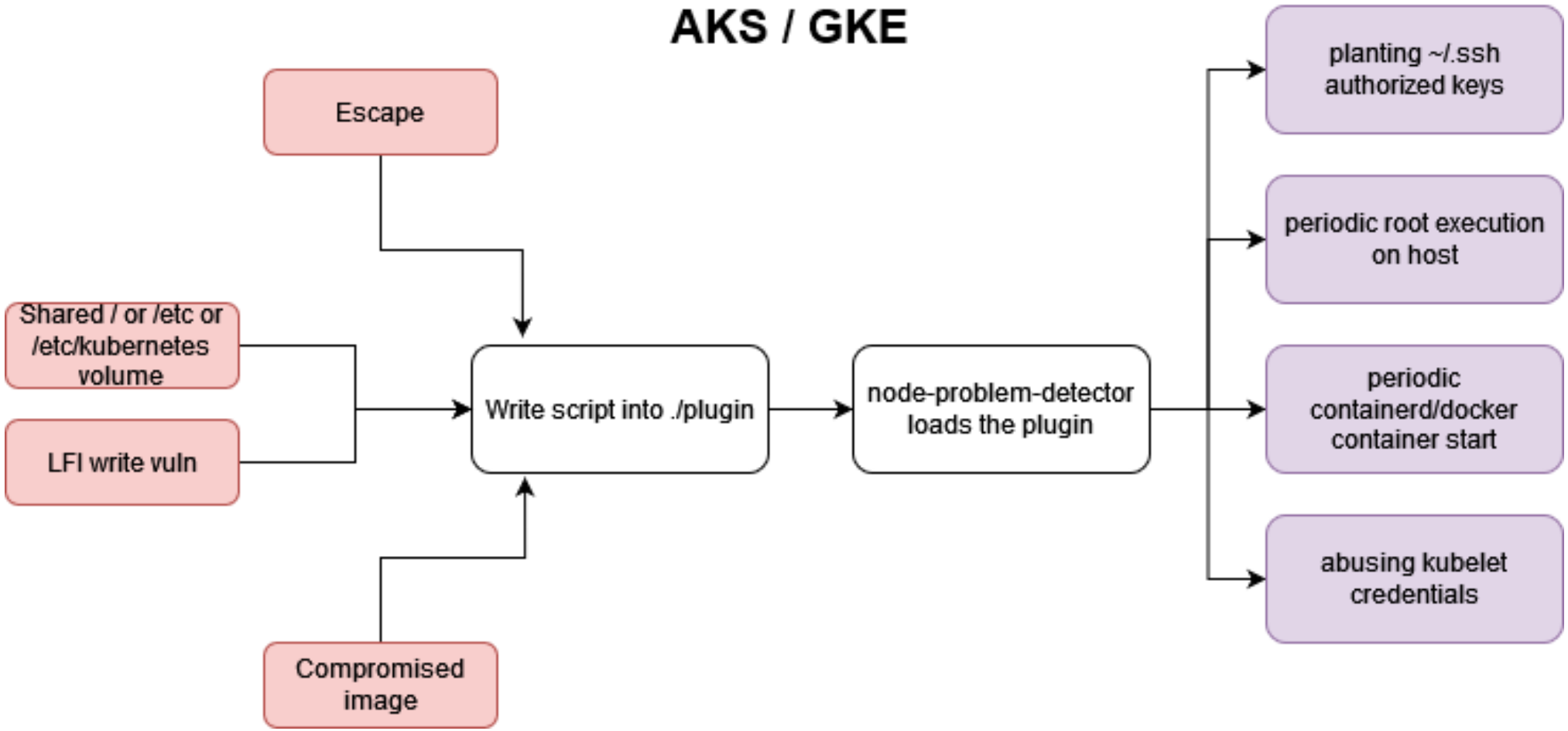
Feature under scope 

custom plugin monitor !!!

Custom Plugin Monitor

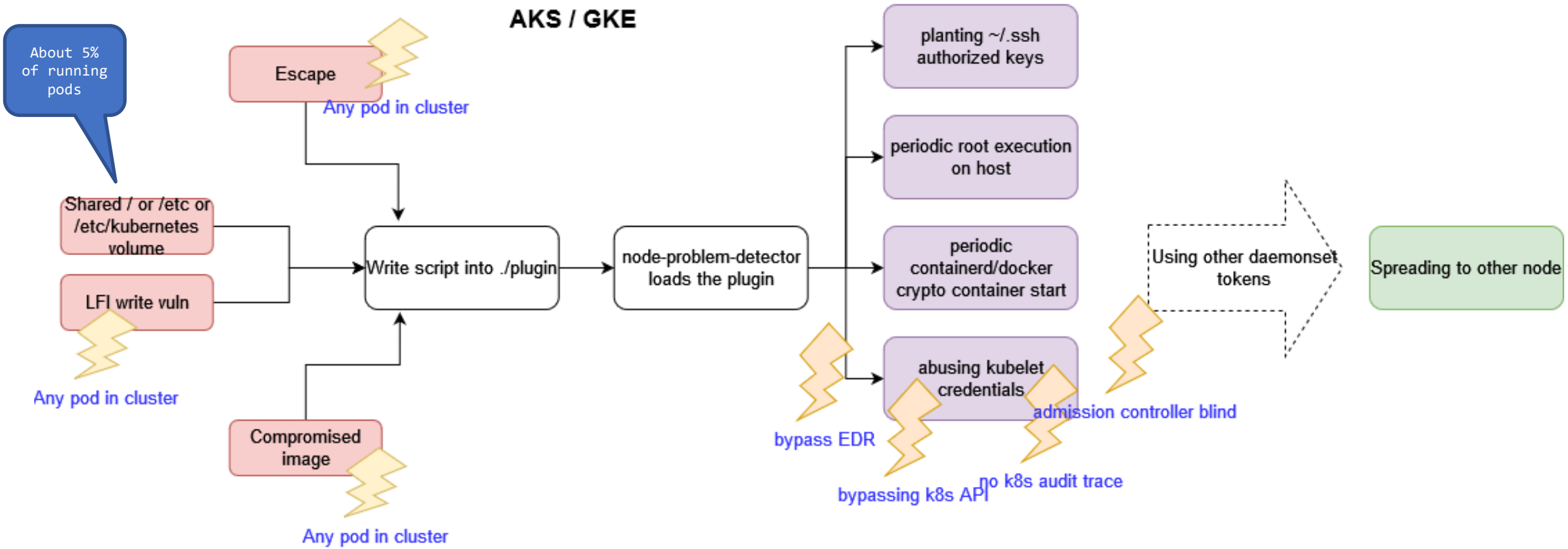
Custom plugin monitor is a plugin mechanism for node-problem-detector. It will extend node-problem-detector to execute any monitor scripts written in any language. The monitor scripts must conform to the plugin protocol in exit code and standard output. For more info about the plugin protocol, please refer to the node-problem-detector plugin interface proposal

node-problem-detector - Attack



Demo Time

node-problem-detector – Attack




KubeCon
CloudNativeCon
Europe 2023
Intro
Definition
Expectations
Hypothesis
Method
Use case 1
Use case 2

Use case - fluentbit

fluentbit – TL;DR

fluentbit – “a super fast, lightweight, and highly scalable logging and metrics processor and forwarder” – from https://fluentbit.io/

Installed on every GKE cluster as a DaemonSet with ConfigMap.

fluentbit - Versions

Latest release version on GitHub v2.0.11, in the image v1.9.9 (in EKS) – from Sep 2022

Latest version in GKE v1.25 is v1.8.12 – from Jan 2022

```
shay_berkovich@cloudshell:~ (shay-junk-cluster)$ kubectl exec -it fluentbit-gke-pjsh5 -n kube-system -- /fluent-bit/bin/fluent-bit --version
Defaulted container "fluentbit" out of: fluentbit, fluentbit-gke, fluentbit-gke-init (init)
Fluent Bit v1.8.12
```

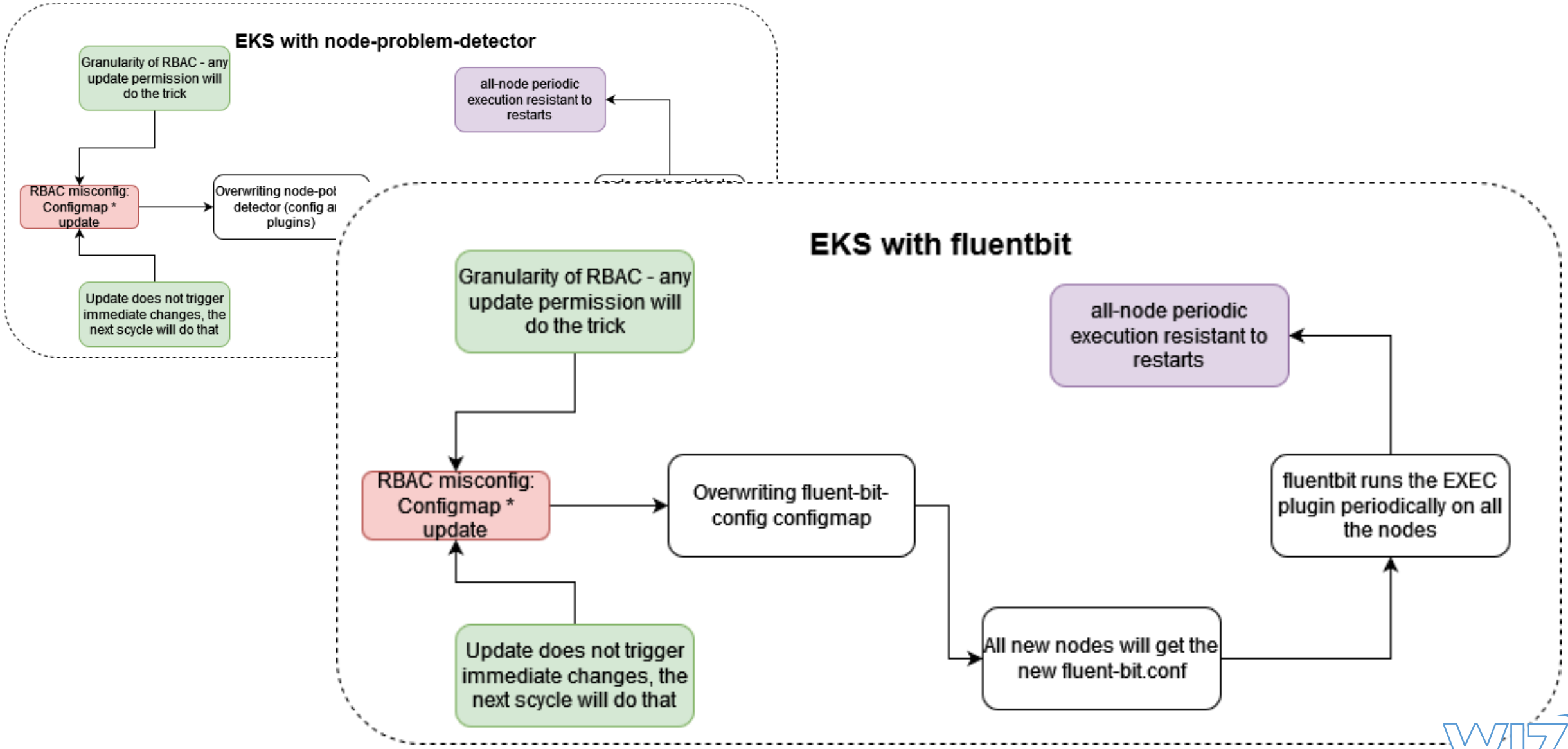
fluentbit - Exploit

Feature under scope 

EXEC input plugin !!!

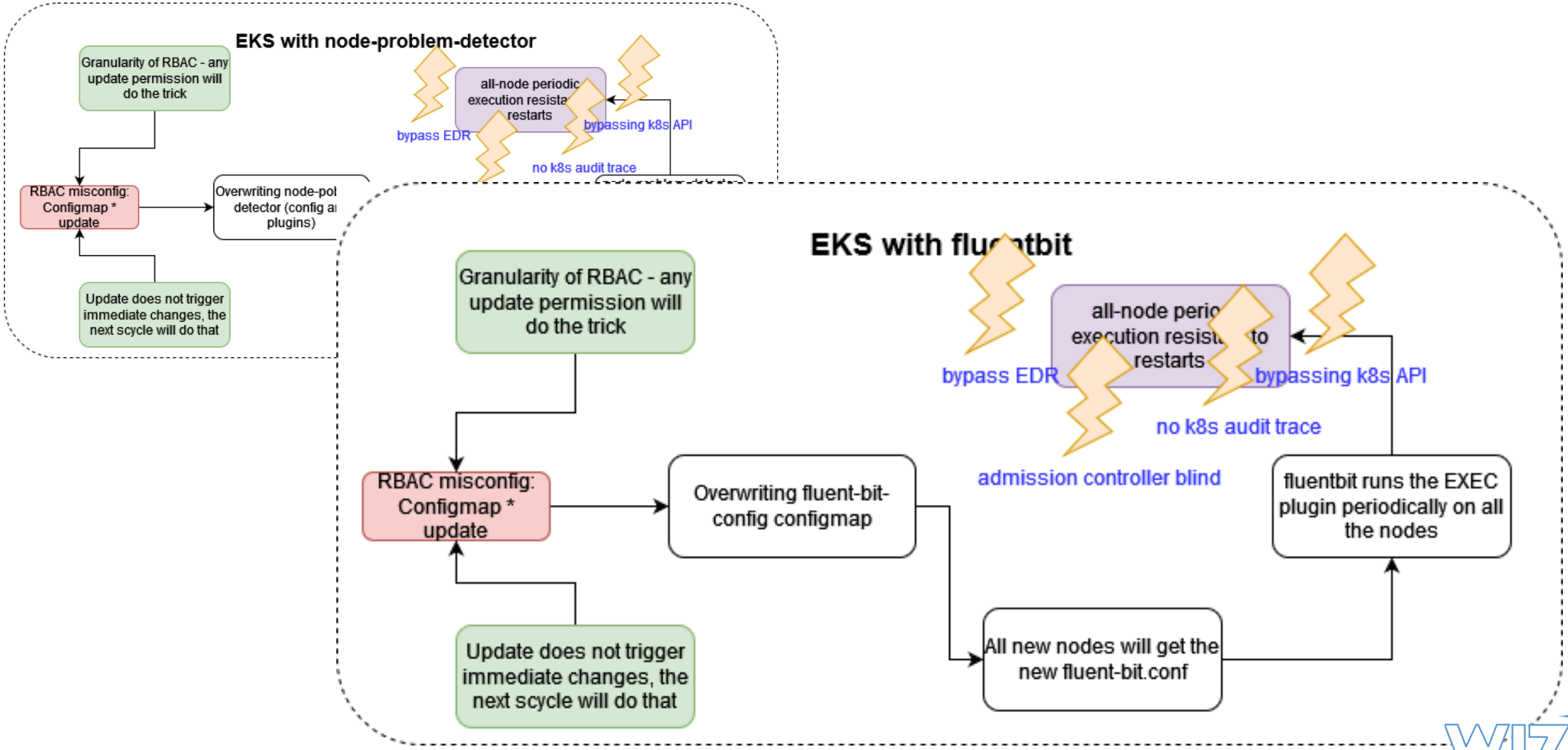
```
[INPUT]
    Name          exec
    Tag           exec_ls
    Command       ls /var/log
    Interval_Sec  1
    Interval_NSec 0

[OUTPUT]
    Name   stdout
    Match  *
```

fluentbit - Attack



Demo Time

fluentbit – Attack Analysis




KubeCon
CloudNativeCon
Europe 2023
Intro
Definition
Expectations
Hypothesis
Method
Use case 1
Use case 2
Reducing the risk

Reducing the Risk - Exceptions

Control plane and middleware traditionally act with privileges

Security vendors often except these components through:

- namespaces (kube-system)
- K8s users and serviceaccounts
- container image names

Exceptions - Examples

ritazh commented on May 16, 2020 — Member

Namespace Exclusion for Gatekeeper design doc:
https://docs.google.com/document/d/1yHuXFs_HQL5N9yT9QVi6AMyflWPtZS4Pg-uXczdqgZ8/edit?usp=sharing

sozercan mentioned this issue on Jun 12, 2020

Config namespace exclusion #678 — Merged

```
28  tracee_match {
29          input.eventName == "security_file_open"
30
31          flags = helpers.get_tracee_argument("flags")
32          helpers.is_file_read(flags)
33
34          pathname := helpers.get_tracee_argument("pathname")
35          contains(pathname, "secrets/kubernetes.io/serviceaccount")
36          endswith(pathname, "token")
37
38          process_names_allowlist := {"flanneld", "kube-proxy", "etcd", "kube-apiserver", "coredns", "kube-controller", "kubectl"}
39          not process_names_allowlist[input.processName]
40  }
```

```
2471    - rule: Contact K8S API Server From Container
2472      desc: Detect attempts to contact the K8S API Server from a container
2473      condition: >
2474        evt.type=connect and evt.dir=< and
2475        (fd.typechar=4 or fd.typechar=6) and
2476        container and
2477        not k8s_containers and
2478        k8s_api_server and
2479        not user_known_contact_k8s_api_server_activities
2480      output: Unexpected connection to K8s API Server from container (command=%proc.cmdline pid=%proc.pid
2481      priority: NOTICE
2482      tags: [network, k8s, container, mitre_discovery, T1565]
```

Reducing the Risk – PSS / PSA

Pod Security Standards / Pod Security Admission (PSP simplified)

Profile	Description
Privileged	Unrestricted policy, providing the widest possible level of permissions. This policy allows for known privilege escalations.
Baseline	Minimally restrictive policy which prevents known privilege escalations. Allows the default (minimally specified) Pod configuration.
Restricted	Heavily restricted policy, following current Pod hardening best practices.

Not so fast:

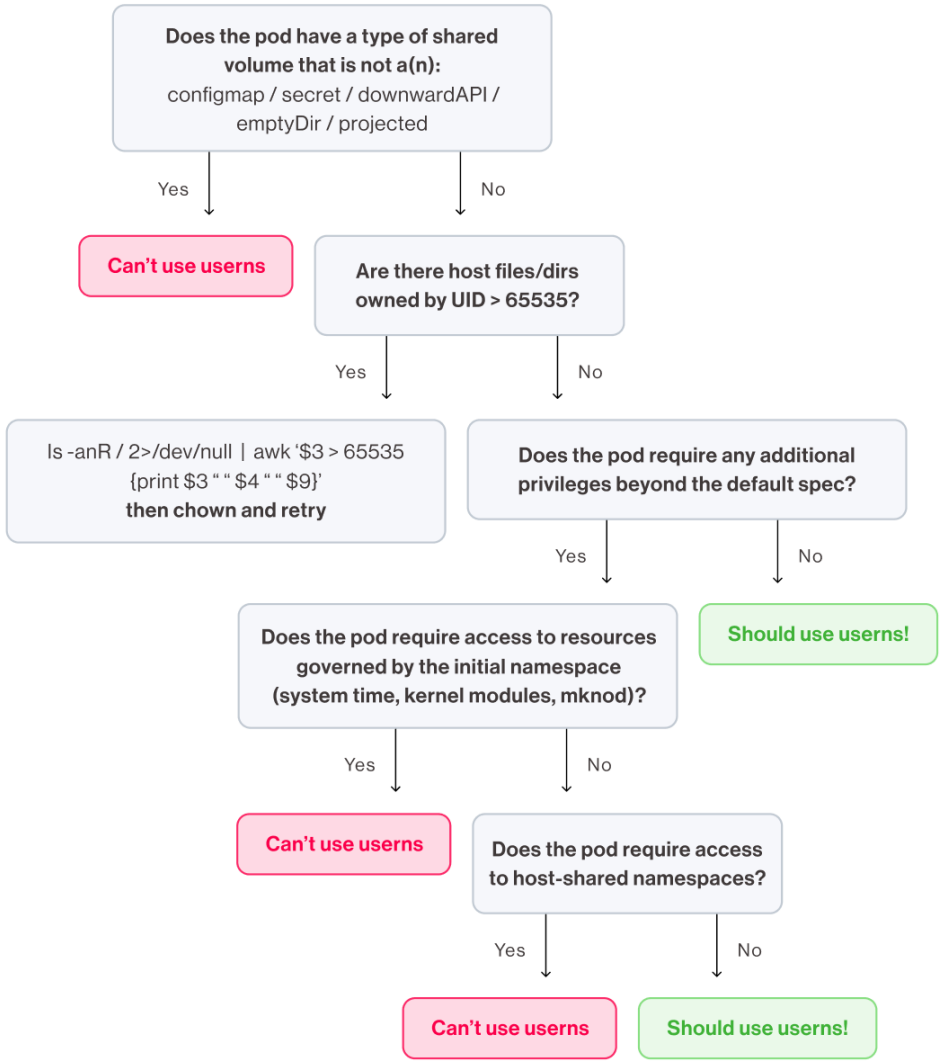
```
shay [ ~ ]$ kubectl label ns kube-system pod-security.kubernetes.io/enforce=restricted --overwrite
Warning: namespace "kube-system" is exempt from Pod Security, and the policy (enforce=restricted:latest) will be ignored
namespace/kube-system labeled
```

Reducing the Risk – userns

User Namespaces:

- Container root != host root
- Isolation between pods
- Separation of resource limits btn pods

Not so fast: can't use userns on 52% (13 out of 25 components



Reducing the Risk – Host Processes

What about non-K8s workloads that run as a host processes?

- Can’t apply K8s-level controls there – all the responsibility is on the CSP
- Very often excluded from EDRs etc.

A word about coredns…

Coredns is a default DNS service in EKS and AKS
https://coredns.io/

Latest version: v1.10.1 (Feb 2013)
Latest version in EKS is v1.8.7 (Jan 2022)
(602401143452.dkr.ecr.us-east-1.amazonaws.com/eks/coredns:v1.8.7-eksbuild.2)
Latest version in AKS is v1.9.3 (May 2022)
(mcr.microsoft.com/oss/kubernetes/coredns:v1.9.3)

A word about coredns…

on

Source Home

Enabled by default

Maintained by CoreDNS

on - executes a command when a specified event is triggered.

```
apiVersion: v1
data:
  Corefile: |
    .:53 {
        on startup touch /tmp/test-startup
        on shutdown touch /tmp/test-shutdown
        errors
        health
        kubernetes cluster.local in-addr.arpa ip6.arpa {
          pods insecure
          fallthrough in-addr.arpa ip6.arpa
        }
        prometheus :9153
        forward . /etc/resolv.conf
```

```
[cloudshell-user@ip-10-2-122-184 ~]$ kubectl logs coredns-d5b9bfc4-2pvcs -n kube-
system.:53[INFO] Blocking Command "touch /tmp/test-startup" with ID a329eeee-73d4-
4fbf-870c-1aa48e19caf2error on 'on-a329eeee-73d4-4fbf-870c-1aa48e19caf2' hook: exec:
"touch": executable file not found in $PATH[INFO] plugin/reload: Running
configuration MD5 = 2fa40b5cf59e6d85ea347bc90cae125dCoreDNS-1.8.7linux/amd64,
go1.17.7, d433a3f2
```

A word about coredns…

Make slim images!!!


KubeCon
CloudNativeCon
Europe 2023
Intro
Definition
Expectations
Hypothesis
Method
Use case 1
Use case 2
Reducing the risk
Conclusions

Conclusions – Inclusive Controls

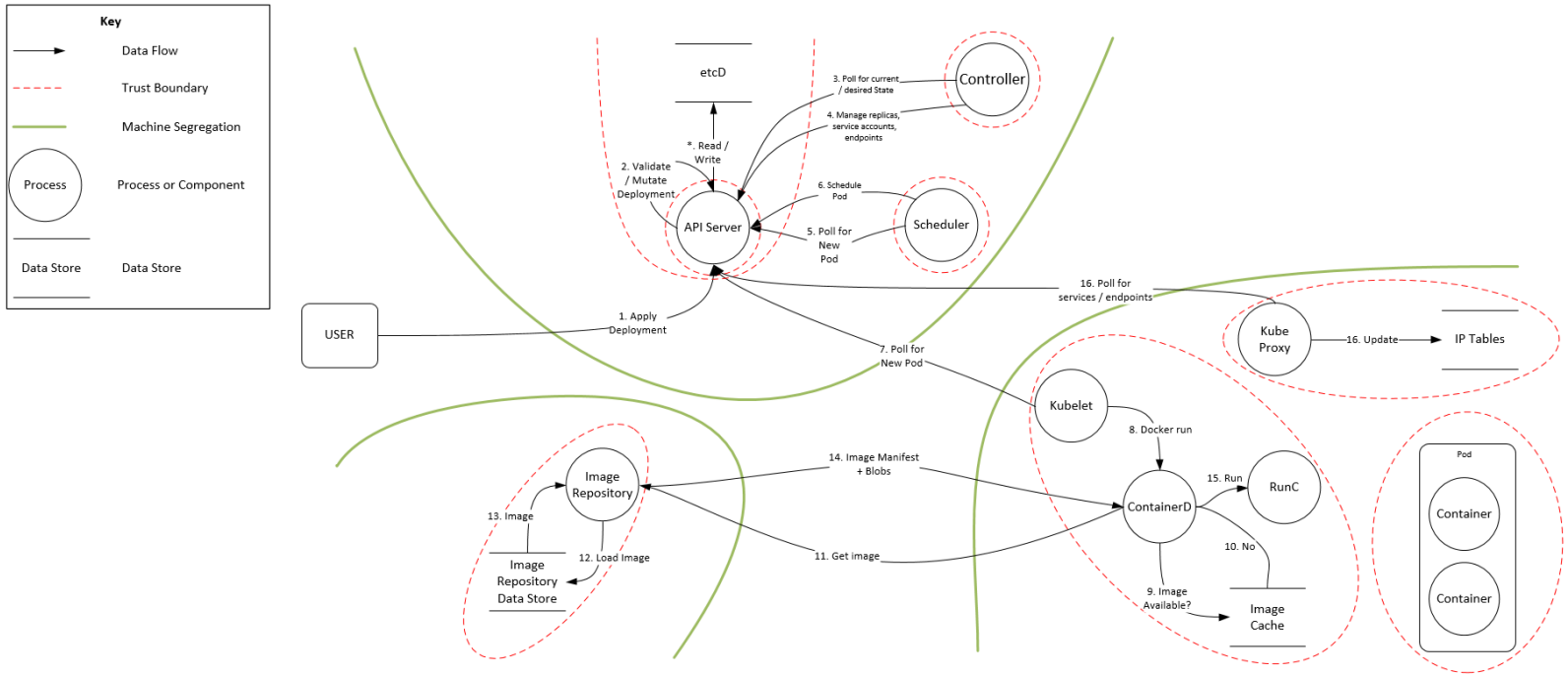
- Workload security (PSP / PSS / PSA)
  - rethinking reliance on kube-system namespace
- Isolation controls for all (User Namespaces)
- Security solutions for all
  - need to be better with FPs

Conclusions – Inclusive TM

KubeCon CloudNativeCon Europe 2023

Existing Threat Models don’t consider MCM

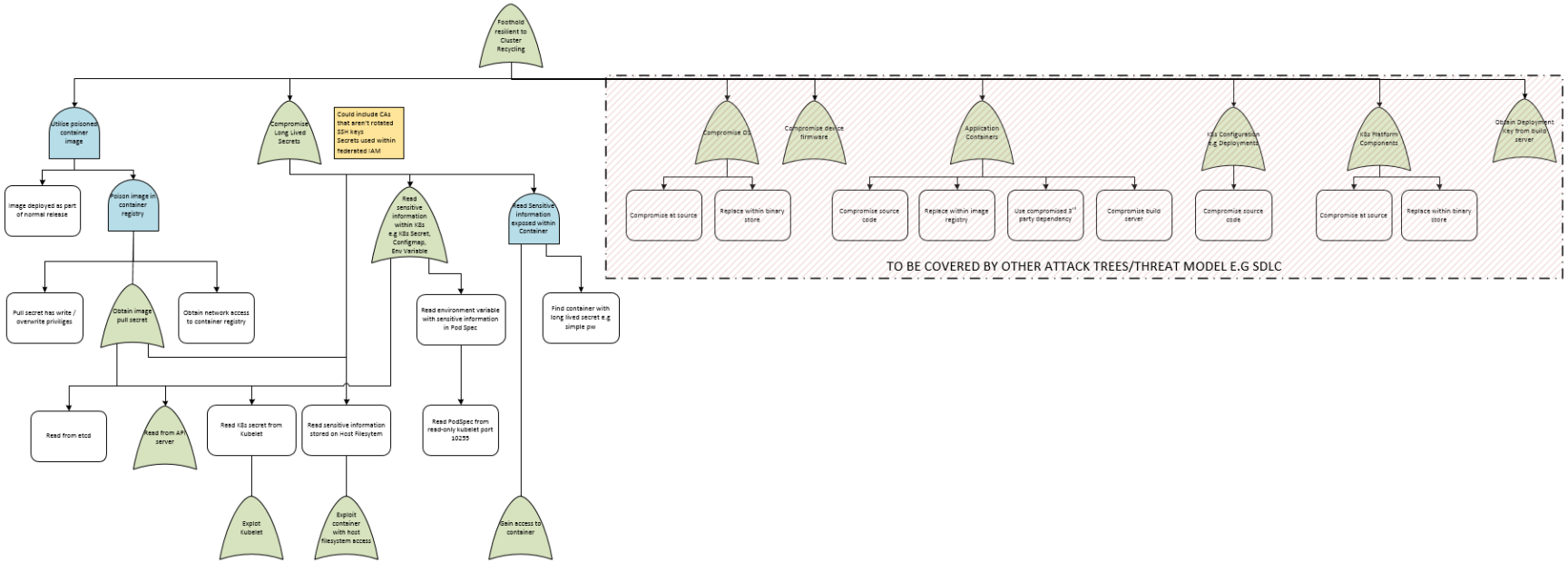
- CNCF Financial User Group – Trust boundaries diagram



Conclusions – Inclusive TM

Existing Threat Models don’t consider MCM

- CNCF Financial User Group – Trust boundaries diagram
- CNCF Financial User Group – Attack trees for establishing persistence



Conclusions – Inclusive TM

KubeCon CloudNativeCon Europe 2023

Existing Threat Models don’t consider MCM

- CNCF Financial User Group – Trust boundaries diagram
- CNCF Financial User Group – Attack trees for establishing persistence
- K8s security audit by Trail of Bits

WIZ

Conclusions – Rethinking TODO

KubeCon CloudNativeCon Europe 2023

Rethinking RBAC permissions

/configmaps:update,patch ~ admin

/namespace:update,patch ~ power user

CSP-based mapping?

Rethinking K8s detections – multi-level approach needed

CSPM + CIEM + Log-based detection + agent-based

Rethinking CSP visibility – what do we really have on our worker nodes?

WIZ


KubeCon
CloudNativeCon
Europe 2023
Intro
Definition
Expectations
Hypothesis
Method
Use case 1
Use case 2
Reducing the risk
Conclusions


KubeCon
CloudNativeCon
Europe 2023
Enjoy the conf!
Rate this talk: