KubeCon | CloudNativeCon

Europe 2022

WELCOME TO VALENCIA

KubeCon CloudNativeCon Europe 2022

Tweezering Kubernetes Resources: Operating on Operators

Kevin Ward, ControlPlane

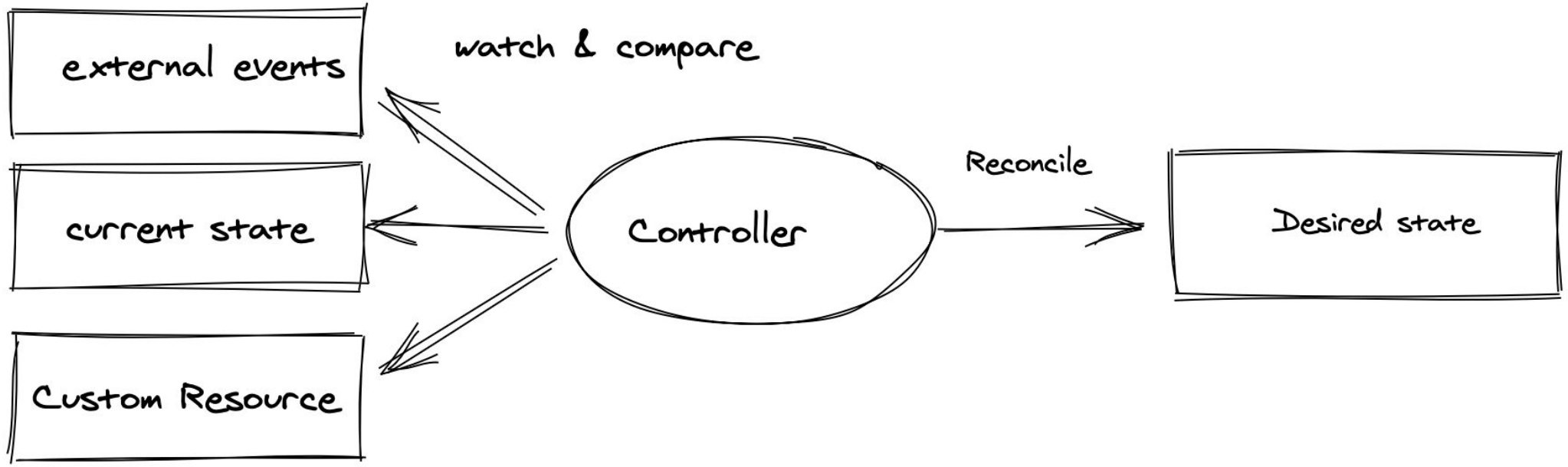
What this talk is about

- Kubernetes Operators & Security
- What does a Operator introduce into Kubernetes?
- How can an Operator be abused by an Attacker?
- How should I perform a security review on an Operator?
- How do I detect Operator Abuse?


Kevin Ward @wakewarduk
Senior Security Engineer @controlplaneio
Matra: Harden by Day, Hack by Night

Kubernetes Operators

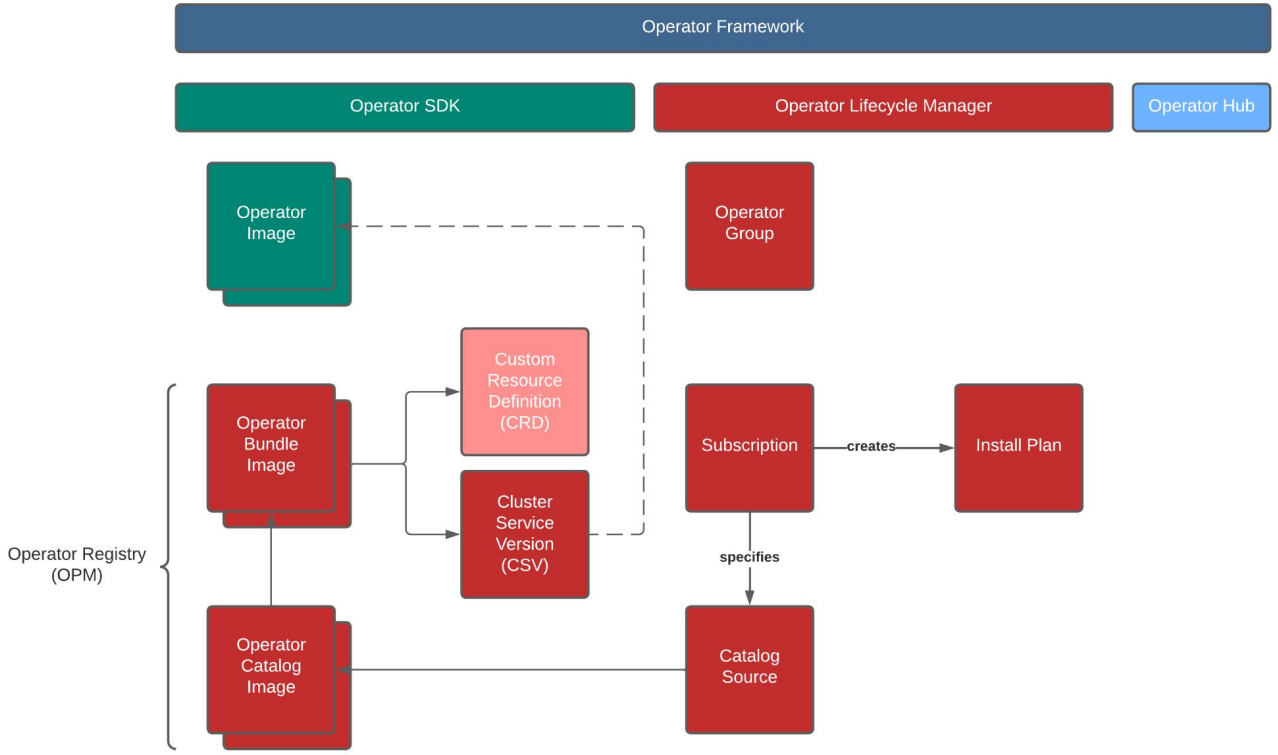
“Operators enable the extension of the Kubernetes API with operational knowledge. This is achieved by combining Kubernetes controllers and watched objects that describe the desired state”



https://github.com/cncf/tag-app-delivery/blob/master/operator-wg/whitepaper/Operator-WhitePaper_v1-0.md

Operator Tools

- Operator Framework is an open source toolkit to develop and manage Kubernetes Operators
  - **Operator SDK** - provides the tools to build, test, and package Operators
  - **Operator Lifecycle Manager (OLM)** - extends Kubernetes to provide a declarative way to install, manage, and upgrade Operators on a cluster
  - **OperatorHub** - Provides a place for the Kubernetes community to share Operators



What does an Operator Introduce?

- Custom Resource Definitions (CRDs)
- Custom Controller
- Operator Namespace*
- Service Account*
- Kubernetes / Cloud Resources
- Logging and Metrics

* Not mandatory and existing cluster resources can be utilised

What can go Wrong?

Key Threats

- Service Account Permissions
- Privileged Container
- Vulnerable Image / Dependencies
- Malicious Operator Code
- Resource Scope
  - Namespace Bound
  - Cluster Bound
  - External Bound (e.g. Infrastructure and Application Configuration)

Operator Threat Matrix

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Impact
Using Cloud Credentials	Exec into container	Malicious Operator	Privileged Container	Clear Container Logs	List K8s Secrets	Access the K8s API server	Kubernetes Service Accounts	***Data from Cloud Storage Object****	***Transfer Data to Cloud Account****	Data Destruction
Compromised Image in Registry	New Container	Backdoor Container	Cluster Admin Binding	Delete K8s Events	Access Operator service account	Access Kubelet API	Writable Host Volume Mounts			***Data Encryption for Impact****
Kubeconfig File	Sidecar Injection	Writable Host Path Mount	Mount Host Path	Use Another Operator	***Access Cloud Credentials****	Network Mapping	Cluster Internal Networking			Resource Hijacking
	OLM Automatic Install	Malicious Admission Controller	***Access Cloud Resources****			***Cloud Infrastructure Discovery****	***Access Cloud Resources****			Denial of Service
	Cloud Instance*	OLM Catalog								
		Access Cloud Resources*								

https://github.com/controlplaneio/operator-threat-matrix

controlplane

Common Attack Path

- Adversary steals Cloud credentials to obtain cluster access
- Enumerate pods
- Exec into Operator container
- Enumerate Service Account permissions
- Leverage ClusterRole binding to deploy malicious container into kube-system
- Takeover cluster resources

Stealth Attack Path

- Compromise an Image in a Registry
- Operator is modified to install a malicious sidecar
- A malicious sidecar is deployed by Operator
- Sidecar intercepts requests
- Data is exfiltrated to Adversary controlled Cloud Account

Operator Related CVEs

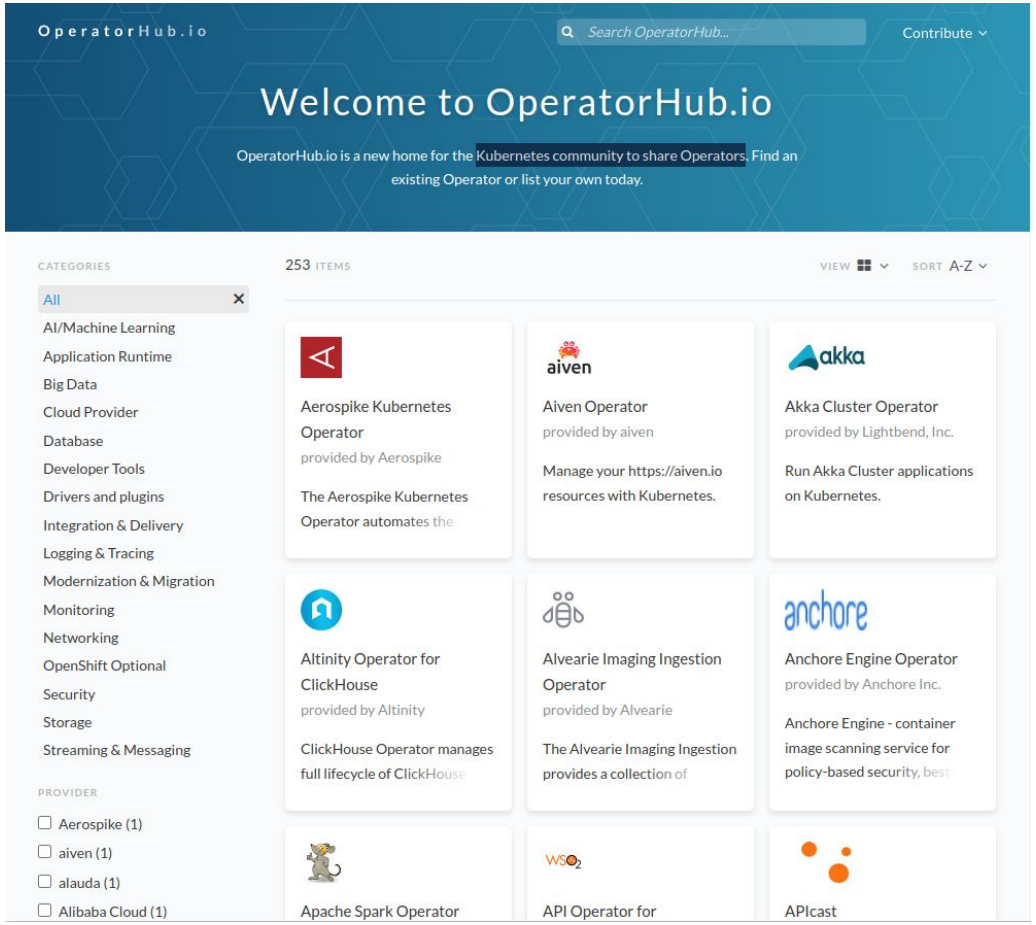
- CVE-2022-26311 - Couchbase Operator 2.2.x before 2.2.3 exposes Sensitive Information to an Unauthorized Actor.
- CVE-2022-23652 - capsule-proxy is a reverse proxy for Capsule Operator which provides multi-tenancy in Kubernetes. This vulnerability allows for an exploit of the `cluster-admin` Role bound to `capsule-proxy`.
- CVE-2021-41266 - Minio console is a graphical user interface for the for MinIO operator. Affected versions are subject to an authentication bypass issue in the Operator Console when an external IDP is enabled.
- CVE-2021-41254 - Users that can create Kubernetes Secrets, Service Accounts and Flux Kustomization objects, could execute commands inside the kustomize-controller container by embedding a shell script in a Kubernetes Secret.
- CVE-2020-7922 - X.509 certificates generated by the MongoDB Enterprise Kubernetes Operator may allow an attacker with access to the Kubernetes cluster improper access to MongoDB instances.

https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=kubernetes+operator

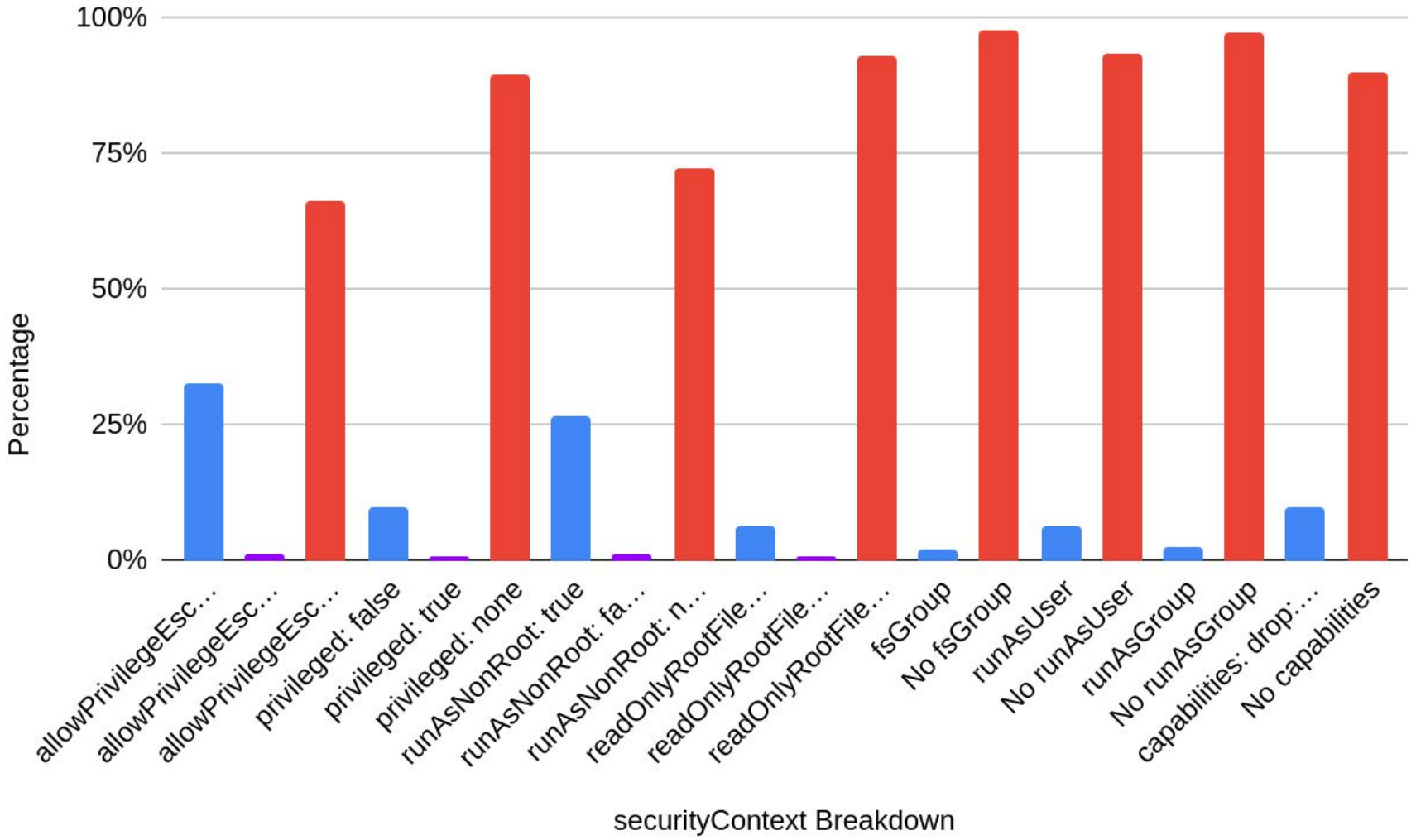
So how **Bad** is it?

OperatorHub Operator Analysis

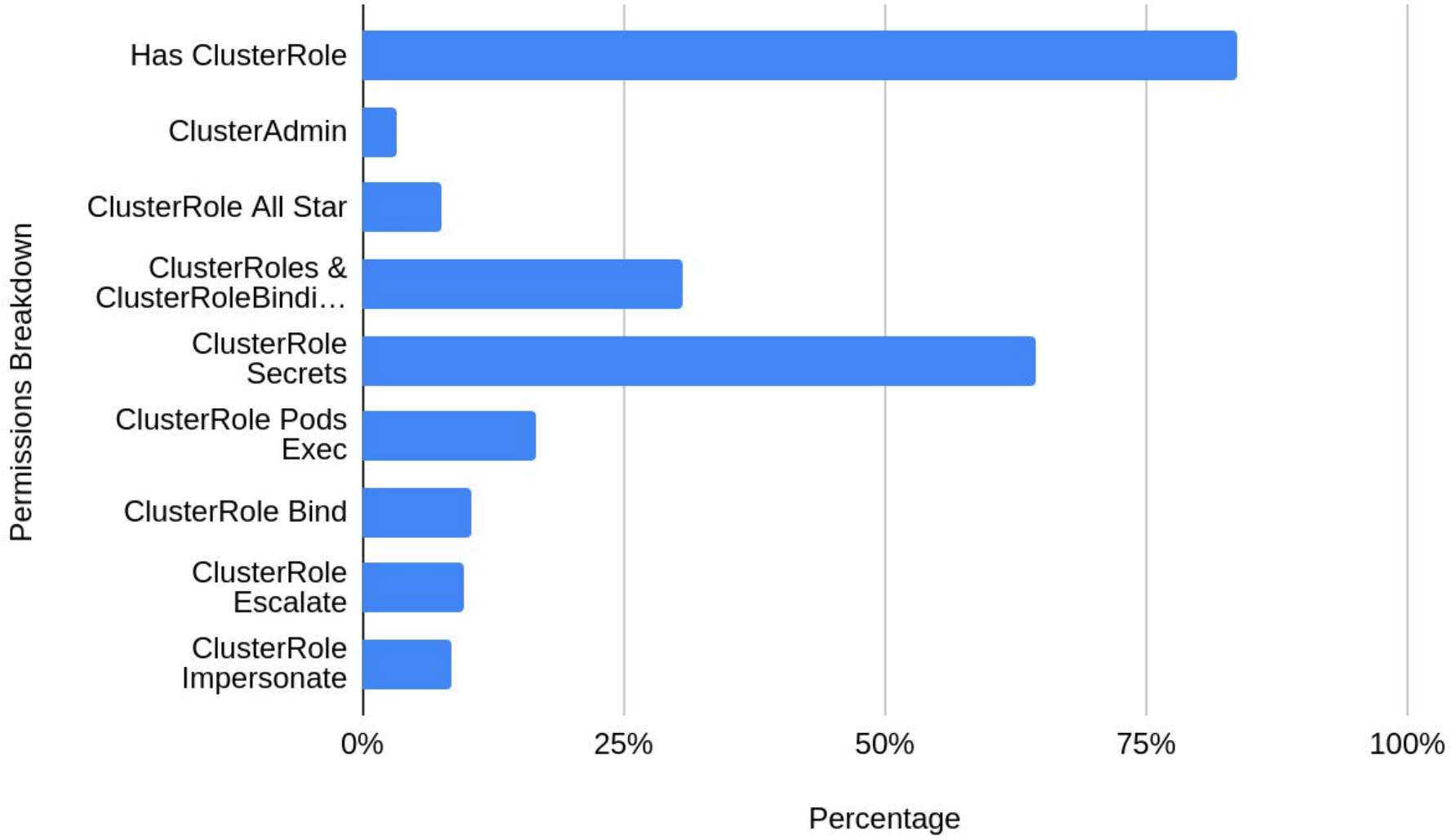
- Reviewed all Operators on https://operatorhub.io/ for key threats
  - Deployed Security Contexts
  - Service Account Permissions
  - Sensitive Cluster Role Bindings
  - Deployed Namespace



OperatorHub Operator securityContext



OperatorHub Operator ClusterRole Permissions



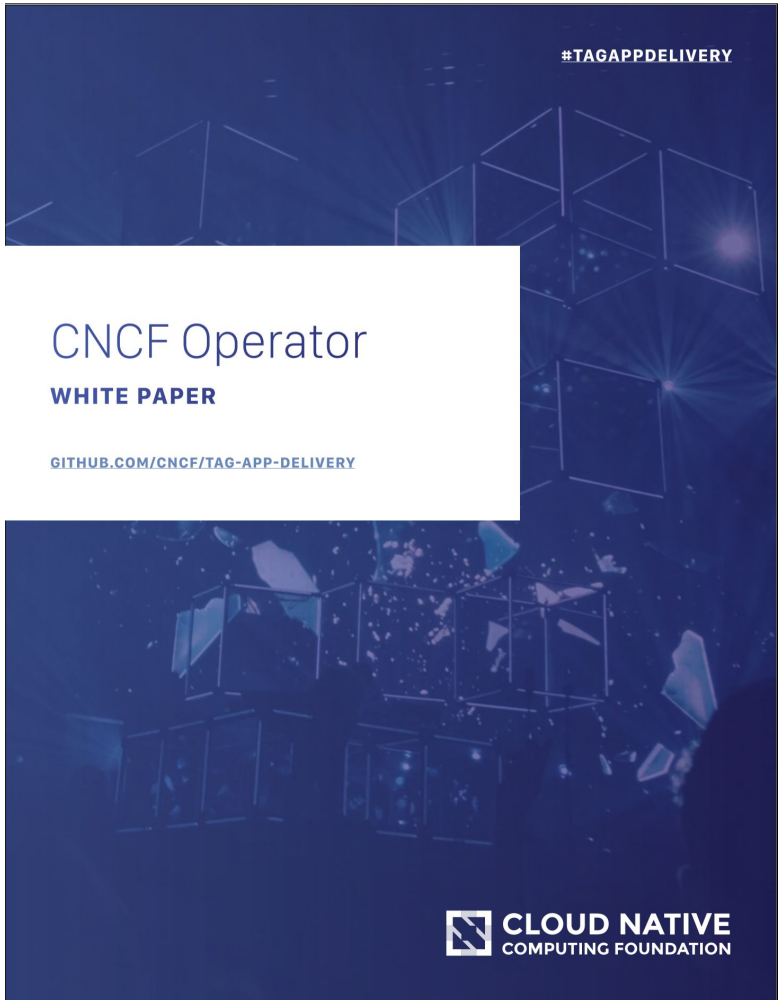
OperatorHub Breakdown

- 90% use a Dedicated Namespace
- 84% use ClusterRoles
- 64% of those ClusterRoles can access secrets
- 17% of those ClusterRoles can exec into pods
- 58% of Operators do not use securityContexts
- Only 10% Drop Linux Capabilities

How do we Secure an Operator?

Operator Best Practices

- CNCF Operator Working Group - Whitepaper
  - https://github.com/cncf/tag-app-delivery/blob/master/operator-wg/whitepaper/Operator-WhitePaper_v1-0.md
- Google Cloud - Operator Best Practices
  - https://cloud.google.com/blog/products/containers-kubernetes/best-practices-for-building-kubernetes-operators-and-stateful-apps



CNCF Operator Security Advice

- Transparency and Documentation
- Define the Operator Scope
  - Cluster-wide Operator
  - External Operator
  - Namespace Operator
- Restrict RBAC Permissions
  - ClusterRoles if absolutely necessary
  - Limit Cloud IAM permissions for External Operators
- Leverage SELinux, AppArmor or Seccomp profiles
- Vulnerabilities & Supply Chain Security

Prevention Strategies

- Operator SDK v1.18.1 sets two security contexts by default
  - `runAsNonRoot: true` and `allowPrivilegeEscalation: false`
- Be explicit with permissions
  - Block deployment of a Operator with * permission set
  - Remember an Operator may require a lot of permissions!
- Work with developers to define the scope of Operator
  - Restrict the Namespace the Operator is deployed
  - Restrict what Namespaces the Operator can watch
  - Cluster-wide - Review ClusterRole permissions
  - External Operators - Review Cloud IAM permissions
  - Namespace Operators - Restrict to only a Role

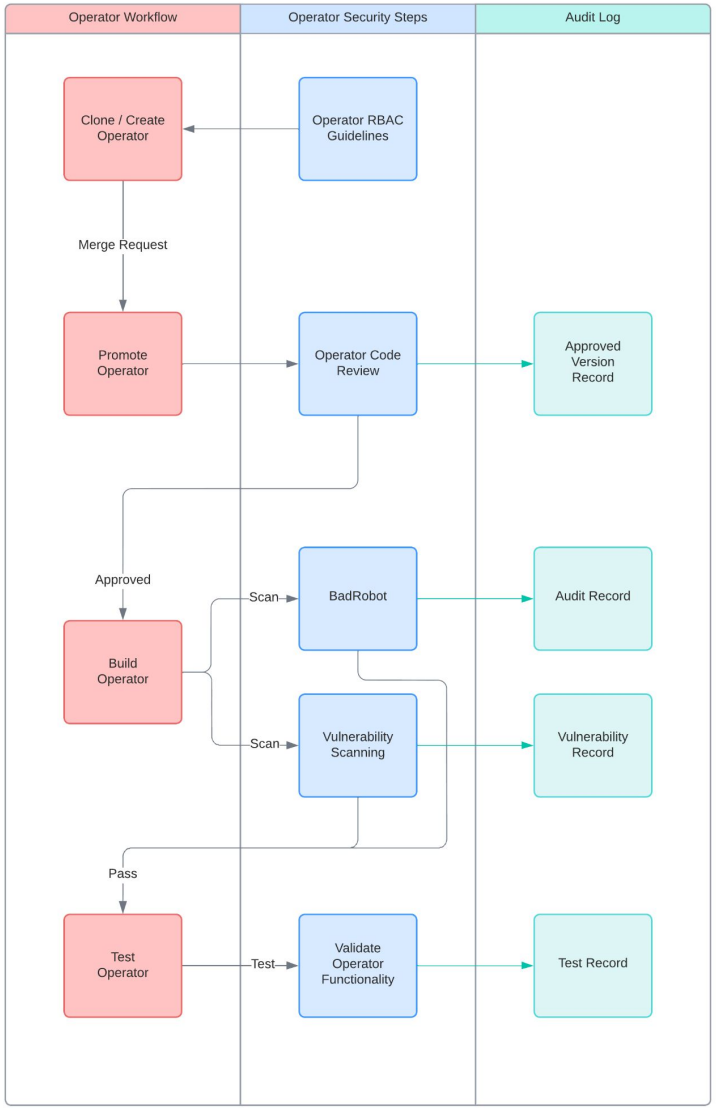
BadRobot - Operator Security Audit Tool

- Static analyser for Operator manifests
- Focussed on compromised Operator obtaining full cluster access
- Highlights risks associated with:
  - Security Contexts
  - Cluster Role Permissions
  - Initial Namespace use
- https://github.com/controlplaneio/badrobot

🙏 Demo - BadRobot 🙏

What about other threats?

- There are several threats which are not covered by BadRobot
  - Malicious Operator code
  - Public Operator overtaken by adversary
  - Non-minimum Operator image
  - Reference malicious Operator image
  - Image and dependency vulnerabilities
  - Public Operator is modified internally
  - OLM misconfigurations
- Use a Operator Pipeline



Detection Strategy

- Operator is an automated runbook
- Capture the logs during the testing
- Determine set events and create an alert when there is a deviation
- Be prepared for edge cases
  - Operator functionality is updated
  - Deployed resources are intentionally vulnerable
  - Operator deploys misconfigured resources

Operator Abuse Detection - Access Operator

- Adversary compromises user account and exec's into Operator

```
kevin@kubecon22eu:~$ kubectl get pods -n op
NAME                                READY   STATUS    RESTARTS   AGE
logging-operator-9d77c48bf-d95v2    1/1     Running   0          7s
kevin@kubecon22eu:~$ kubectl exec -n op -it logging-operator-9d77c48bf-d95v2 -- /bin/bash
[root@logging-operator-9d77c48bf-d95v2 /]#
```

- Exec command is captured via Kubernetes API event logs

```
methodName: io.k8s.core.v1.pods.exec.create, requestMetadata: {…},

resourceName: core/v1/namespaces/op/pods/logging-operator-9d77c48bf-d95v2/exec
```

Operator Abuse Detection - Download Tools

- Adversary installs kubectl on Operator

```
}[root@logging-operator-9d77c48bf-d95v2 /]# cd /usr/bin
[root@logging-operator-9d77c48bf-d95v2 bin]#   curl -LO https://dl.k8s.io/release/v1.24.0/bin/linux/amd64/kubectl
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   154  100   154    0     0   1054      0 --:--:-- --:--:-- --:--:--  1047
100 43.5M  100 43.5M    0     0  68.7M      0 --:--:-- --:--:-- --:--:-- 68.7M
[root@logging-operator-9d77c48bf-d95v2 bin]# chmod +x kubectl
[root@logging-operator-9d77c48bf-d95v2 bin]#
```

- Download is not detected in the Kubernetes Logs

Operator Abuse Detection - Enum Service Account

- Service Account Permissions on kube-system are checked

```
configmaps                                      []                                              []                                              [list get create patch update watch delete]
endpoints                                       []                                              []                                              [list get create patch update watch delete]
events                                          []                                              []                                              [list get create patch update watch delete]
persistentvolumeclaims                          []                                              []                                              [list get create patch update watch delete]
pods/exec                                       []                                              []                                              [list get create patch update watch delete]
pods                                            []                                              []                                              [list get create patch update watch delete]
secrets                                         []                                              []                                              [list get create patch update watch delete]
services/finalizers                             []                                              []                                              [list get create patch update watch delete]
services                                        []                                              []                                              [list get create patch update watch delete]
daemonsets.apps                                 []                                              []                                              [list get create update watch]
deployments.apps                                []                                              []                                              [list get create update watch]
replicasets.apps                                []                                              []                                              [list get create update watch]
statefulsets.apps                               []                                              []                                              [list get create update watch]
cronjobs.batch                                  []                                              []                                              [list get create update watch]
jobs.batch                                      []                                              []                                              [list get create update watch]
```

- The request is not captured in Kubernetes Logs

Operator Abuse Detection - Deploy Malicious Image

- Adversary deploys a malicious container into kube-system namespace

```
[root@logging-operator-9d77c48bf-d95v2 bin]# kubectl run tools -n kube-system --image=
pod/tools created
```

- Pod Deployment is detected via Kubernetes API event logs

```
methodName: io.k8s.core.v1.pods.create,

resourceName: core/v1/namespaces/kube-system/pods/tools,
```

Operator Abuse Detection - Exec Malicious Image

- Adversary pivots to malicious container in kube-system

```
[root@logging-operator-9d77c48bf-d95v2 bin]# kubectl exec -n kube-system -it tools -- /bin/bash
root@tools:~#
```

- Exec command is captured via Kubernetes API event logs

```
methodName: io.k8s.core.v1.pods.exec.create, requestMetadata: {…},

resourceName: core/v1/namespaces/op/pods/logging-operator-9d77c48bf-d95v2/exec
```

Detection Enhancement Options

- Several Options to Enhance Detection
- Cloud Provider Solutions
  - GCP Container Threat Detection
  - Azure Microsoft Defender for Containers
- Third Party Solutions
  - Sysdig
  - AquaSec
  - TwistLock

Operator Future

- Extending Operator SDK Scorecard with security tests
  - Not to be confused with the OpenSSF Scorecard
- Dynamic access for Operators
  - Elevate privileges to perform sensitive operations
- Policy engine to control Operator authorisation
- Anomaly-based Detection
  - Requires full test suite and training the detection engine

Conclusion

- Operators can do as much damage as Kubernetes workloads
- Define the core functionality of the Operator
- Review Operators scope and permissions
- Block the deployment of default Operator permissions
- Apply Linux Security Modules where possible
- Profile the Operator with Logs and Metrics, Alert on Deviations

Thank You

Operator Threat Matrix - https://github.com/controlplaneio/operator-threat-matrix

BadRobot - https://github.com/controlplaneio/badrobot

Website - https://controlplane.io

Twitter - @wakewarduk, @controlplaneio

GitHub - @wakeward