Kubescape Kubernetes security scanner

CONCEPTS

Scan manifests before deploy and the live cluster afterwards: the same controls run on both.

Target cluster, files, Helm, image
↓ load
Framework NSA, MITRE, CIS: sets of controls
↓ evaluate
Controls C-xxxx checks, exceptions applied
↓ score
Report table, JSON, SARIF, HTML

Exceptions show up as exclude, not fail, so accepted risk stays visible in the report.

SCAN TARGETS

`kubescape scan`
Scan the cluster in your kubeconfig

`kubescape scan <path>`
Scan local YAML files or a directory

`kubescape scan <git-url>`
Scan a Git repository

`kubescape scan <chart-dir>`
Scan a Helm chart directory

`--values <file>`
Use a Helm values file when rendering

`kubescape scan <kustomize-dir>`
Scan a Kustomize directory

`kubescape scan image <image>`
Scan an image for vulnerabilities

`--kubeconfig <file>`
Use another kubeconfig

`--include-namespaces <list>`
Scan only these namespaces

`--exclude-namespaces <list>`
Skip these namespaces

FRAMEWORKS AND CONTROLS

`kubescape scan framework <name>`
Run all controls of a framework

`kubescape scan framework nsa`
NSA-CISA hardening guide

FRAMEWORKS AND CONTROLS (cont.)

`kubescape scan framework mitre`
MITRE ATT&CK for Kubernetes

`kubescape scan framework nsa,mitre`
Run several frameworks at once

`kubescape scan control <id>`
Run a single control, e.g. C-0017

`kubescape list controls`
List available controls

`kubescape list frameworks`
List available frameworks

`kubescape download controls-inputs`
Fetch the tunable control parameters

`--controls-config <file>`
Scan with edited control parameters

VIEWS AND THRESHOLDS

`--verbose`
Show each failing resource and the fix

`--view security`
Default view when scanning files

`--view resource`
Group results by resource

`--view control`
Group results by control

`--compliance-threshold <pct>`
Exit 1 if compliance is below pct

`--severity-threshold <level>`
Fail on controls at this severity or above

Gate CI on --compliance-threshold: it is the documented, non-deprecated pass/fail switch.

OUTPUT

`--format json --output <file>`
Machine-readable report

`--format sarif --output <file>`
SARIF for code scanning

OUTPUT (cont.)

`--format junit --output <file>`
JUnit XML for CI test views

`--format html --output <file>`
Shareable HTML report

`--format json,html --output <prefix>`
Several formats in one run

EXCEPTIONS

`--exceptions <file>`
Apply accepted-risk exceptions

`posturePolicies`
Controls or frameworks to exclude

`resources`
Match by kind, name, namespace or labels

Exception fields name, kind, namespace and controlID accept regex; resource labels do not.

Exclude a control in one namespace

```
[{"name": "allow-hostpath",
  "policyType":
"postureExceptionPolicy",
  "actions": ["alertOnly"],
  "resources":
[{"designatorType": "Attributes",
    "attributes": {"namespace":
"monitoring"}}],
  "posturePolicies":
[{"controlID": "C-0060"}]}]
```

FIX, PATCH AND POLICY

`kubescape fix <report.json>`
Auto-fix manifests from a JSON report

`kubescape fix <report.json> --dry-run`
Preview fixes without writing

CONFIG AND OFFLINE

`kubescape config view`
Show stored configuration

`kubescape version`
Print version

`kubectl krew install kubescape`
Install as a kubectl plugin