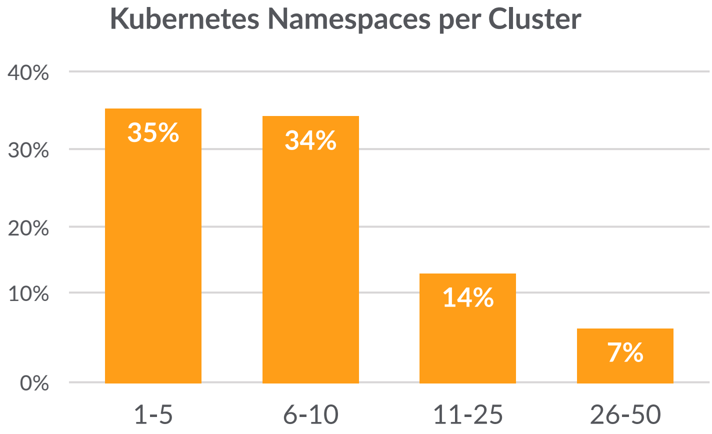
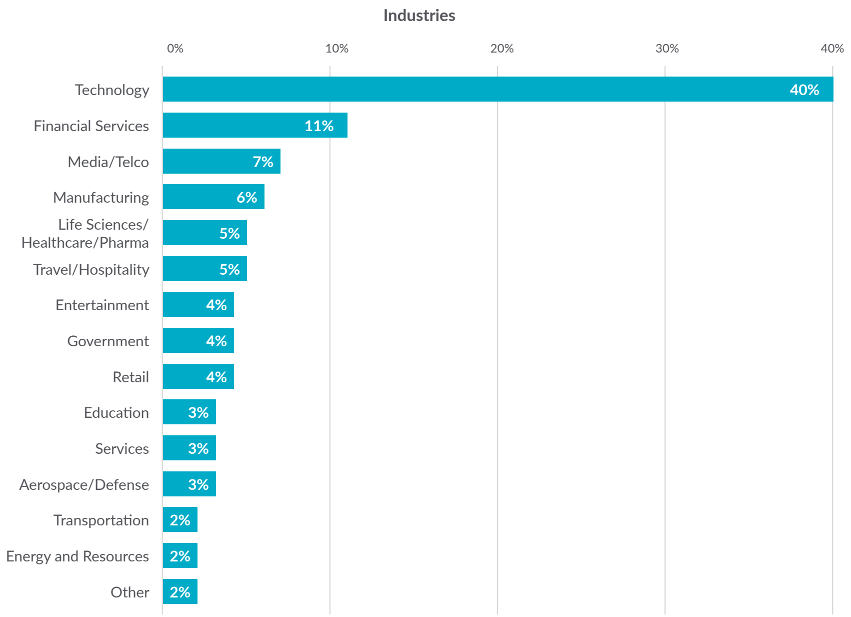
Kubernetes *Configuration Assessment* Operator

Ali Kanso
IBM Research
NY, USA

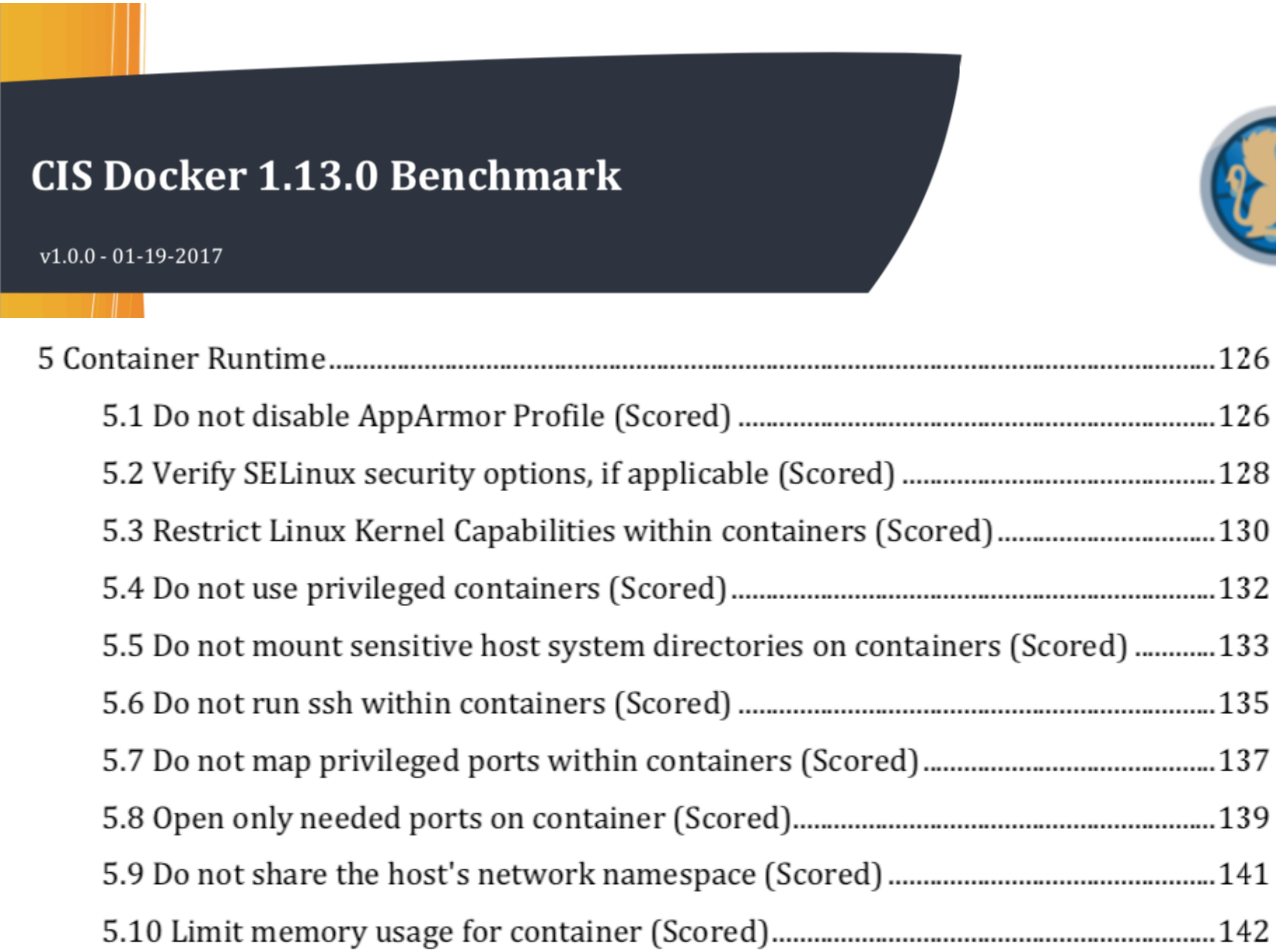
Multiple Namespaces Per K8s Cluster





Sysdig container usage report 2019

Security Standards



CIS Docker 1.13.0 Benchmark

v1.0.0 - 01-19-2017

Center for Internet Security

K8s Controlled Features for the *Container Runtime*

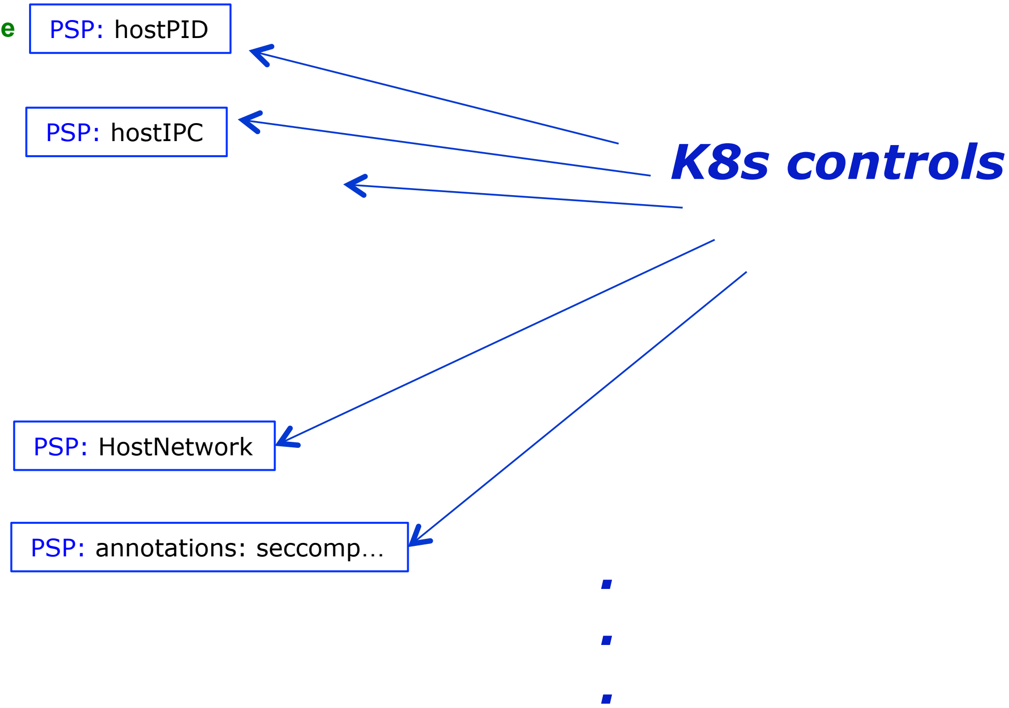
5 Container Runtime

5.1 Do not disable AppArmor Profile (Scored)	PSP: annotations: apparmor…
5.2 Verify SELinux security options, if applicable (Scored)	PSP: seLinuxOptions
5.3 Restrict Linux Kernel Capabilities within containers (Scored)	PSP: AllowedCapabilities
5.4 Do not use privileged containers (Scored).	PSP: Privileged
5.5 Do not mount sensitive host system directories on containers (Scored)	PSP: allowedHostPaths
~~5.6 Do not run ssh within containers (Scored)~~	
5.7 Do not map privileged ports within containers (Scored)	PSP: hostPorts:
~~5.8 Open only needed ports on container (Scored)~~	
5.9 Do not share the host's network namespace (Scored)	PSP: HostNetwork
5.10 Limit memory usage for container (Scored)	LimitRange: default.memory…
5.11 Set container CPU priority appropriately (Scored)	LimitRange: default.CPU…
5.12 Mount container's root filesystem as read only (Scored)	PSP: readOnlyRootFilesystem
~~5.13 Bind incoming container traffic to a specific host interface (Scored)~~	
5.14 Set the 'on-failure' container restart policy to 5 (Scored)	Pod: restartOnFailure

K8s controls

⋮

K8s Controlled Features for the *Container Runtime*



K8s controls

- **5.15 Do not share the host's process namespace (Scored)**

  PSP: hostPID

- **5.16 Do not share the host's IPC namespace (Scored)**

  PSP: hostIPC

- ~~**5.17 Do not directly expose host devices to containers (Not Scored)**~~
- ~~**5.18 Override default ulimit at runtime only if needed (Not Scored) set mount propagation mode to shared (Scored)**~~
- **5.20 Do not share the host's UTS namespace (Scored)**

  PSP: HostNetwork

- **5.21 Do not disable default seccomp profile (Scored)**

  PSP: annotations: seccomp…

- ~~**5.22 Do not docker exec commands with privileged option (Scored)**~~
- ~~**5.23 Do not docker exec commands with user option (Scored)**~~
- **5.24 Confirm cgroup usage (Scored)**

  LimitRange: default…

- **5.25 Restrict container acquiring additional privileges**

  PSP: allowPrivilegeEscalation

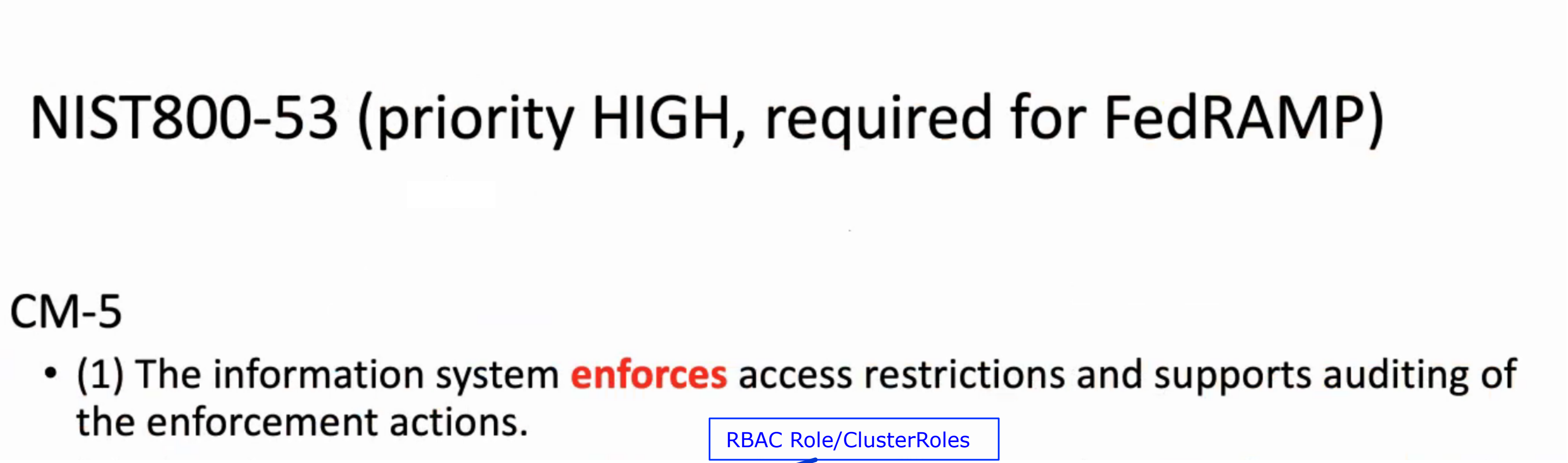
- **5.26 Check container health at runtime (Scored)**

  Pod: livenessProbs

- **5.27 Ensure docker commands always get the latest version of the image (Not Scored)**

  Pod: pullAlways

Regulatory Requirement



NIST800-53 (priority HIGH, required for FedRAMP)

CM-5

- (1) The information system **enforces** access restrictions and supports auditing of the enforcement actions.

RBAC Role/ClusterRoles

K8s controls

K8s Configuration Assessment

- Kubernetes offer several construct to better isolate and secure your cluster and namespaces

ResourceQuota
✓ E.g. limit CPU usage

Network policy
✓ E.g. block certain ingress traffic

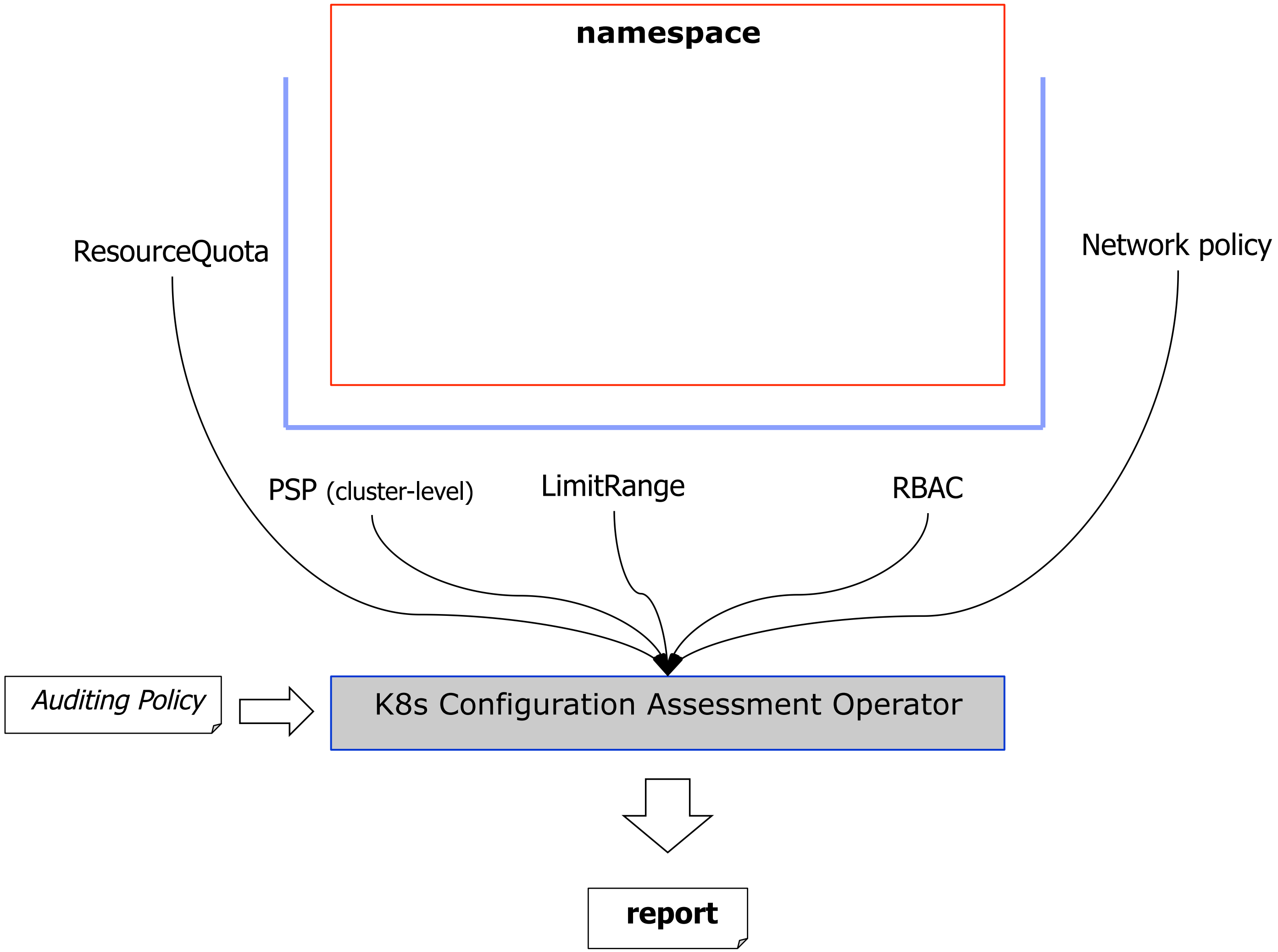
PSP (cluster-level)
✓ E.g. don’t allow privileged containers

LimitRange
✓ E.g. inject default limits when missing

RBAC
✓ E.g. limit access

- However, some of those constructs may be missing or misconfigured.
- Our objective is to offer guidance to SREs and Admins following best practices in security and cluster management

K8s Configuration Assessment at Runtime

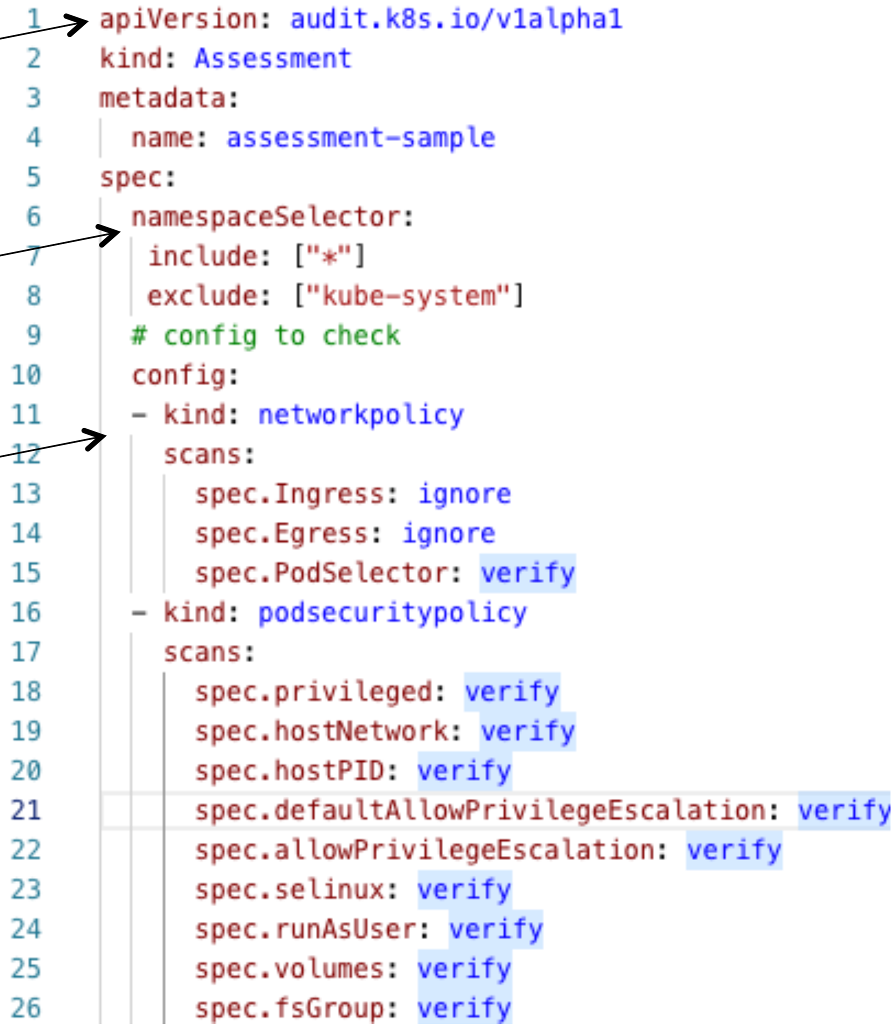


Audit CRD

CRD based

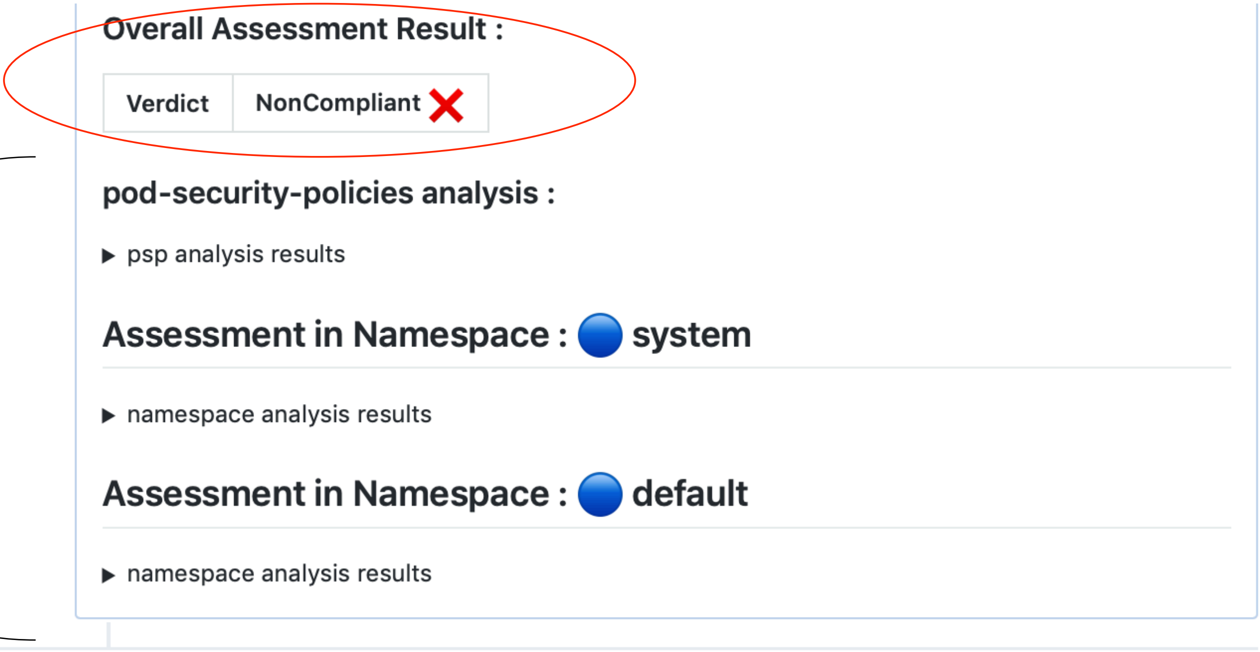
Per namespace selection

Per Kind configuration



```yaml
apiVersion: audit.k8s.io/v1alpha1
kind: Assessment
metadata:
  name: assessment-sample
spec:
  namespaceSelector:
    include: ["*"]
    exclude: ["kube-system"]
  # config to check
  config:
  - kind: networkpolicy
    scans:
      spec.Ingress: ignore
      spec.Egress: ignore
      spec.PodSelector: verify
  - kind: podsecuritypolicy
    scans:
      spec.privileged: verify
      spec.hostNetwork: verify
      spec.hostPID: verify
      spec.defaultAllowPrivilegeEscalation: verify
      spec.allowPrivilegeEscalation: verify
      spec.selinux: verify
      spec.runAsUser: verify
      spec.volumes: verify
      spec.fsGroup: verify
```

Sample Result



Overall Assessment Result :

Verdict	NonCompliant ❌

pod-security-policies analysis :

▶ psp analysis results

Assessment in Namespace : 🔵 system

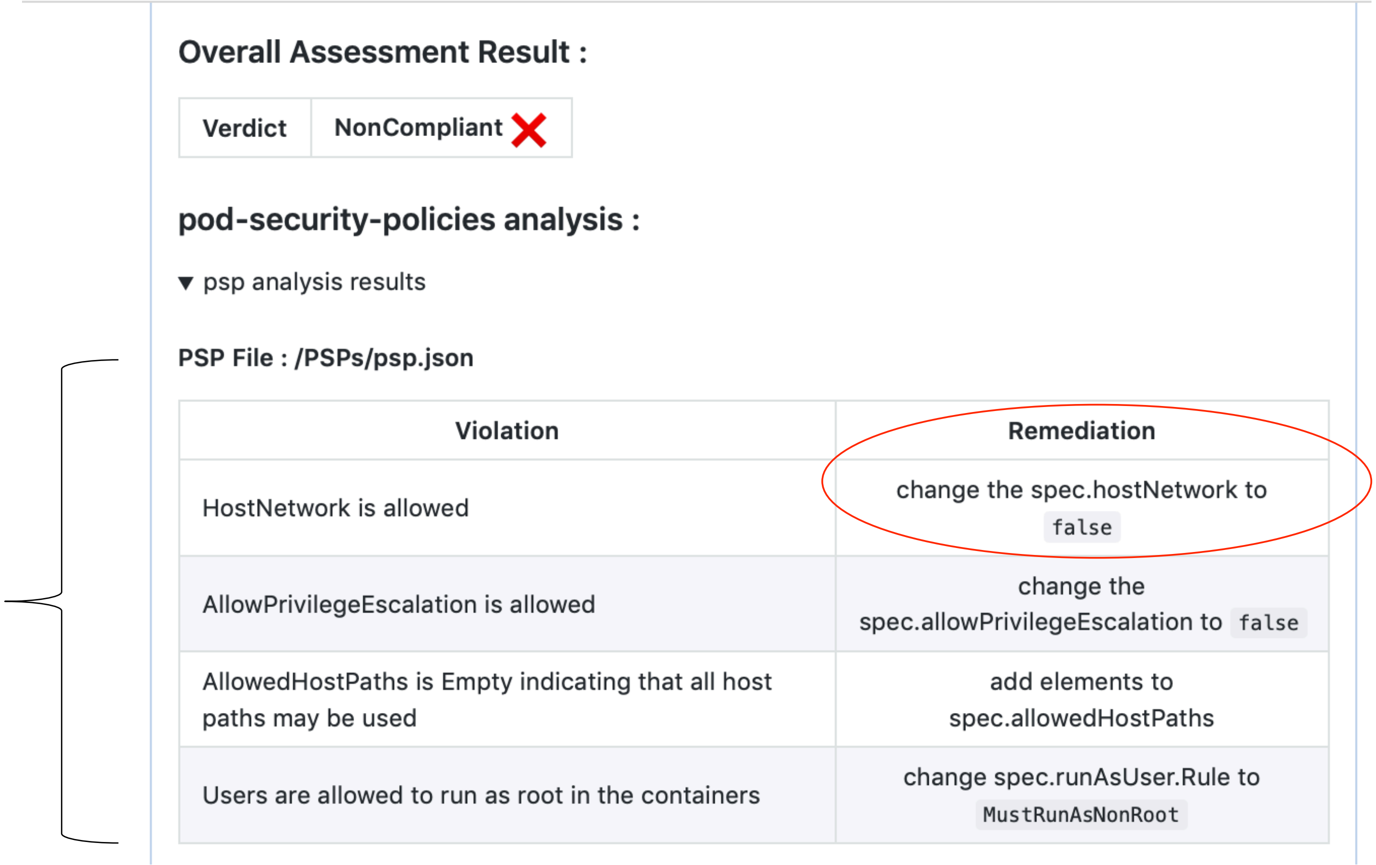
▶ namespace analysis results

Assessment in Namespace : 🔵 default

▶ namespace analysis results

Per **Namespace** assessment

Sample Result



Overall Assessment Result :

Verdict	NonCompliant ❌

pod-security-policies analysis :

▼ psp analysis results

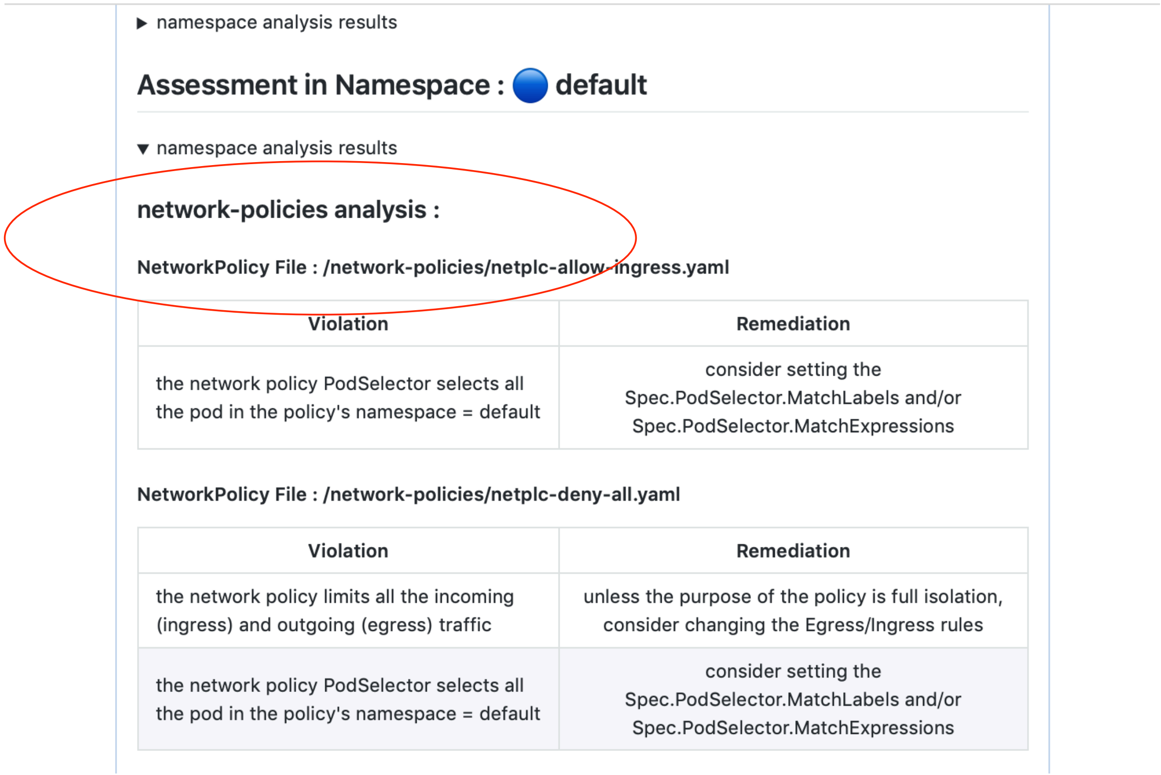
PSP File : /PSPs/psp.json

Violation	Remediation
HostNetwork is allowed	change the spec.hostNetwork to `false`
AllowPrivilegeEscalation is allowed	change the spec.allowPrivilegeEscalation to `false`
AllowedHostPaths is Empty indicating that all host paths may be used	add elements to spec.allowedHostPaths
Users are allowed to run as root in the containers	change spec.runAsUser.Rule to `MustRunAsNonRoot`

Remediation suggestions for each violation

Sample Result

Detailed analysis per object



▶ namespace analysis results

Assessment in Namespace : 🔵 default

▼ namespace analysis results

network-policies analysis :

NetworkPolicy File : /network-policies/netplc-allow-ingress.yaml

Violation	Remediation
the network policy PodSelector selects all the pod in the policy's namespace = default	consider setting the Spec.PodSelector.MatchLabels and/or Spec.PodSelector.MatchExpressions

NetworkPolicy File : /network-policies/netplc-deny-all.yaml

Violation	Remediation
the network policy limits all the incoming (ingress) and outgoing (egress) traffic	unless the purpose of the policy is full isolation, consider changing the Egress/Ingress rules
the network policy PodSelector selects all the pod in the policy's namespace = default	consider setting the Spec.PodSelector.MatchLabels and/or Spec.PodSelector.MatchExpressions