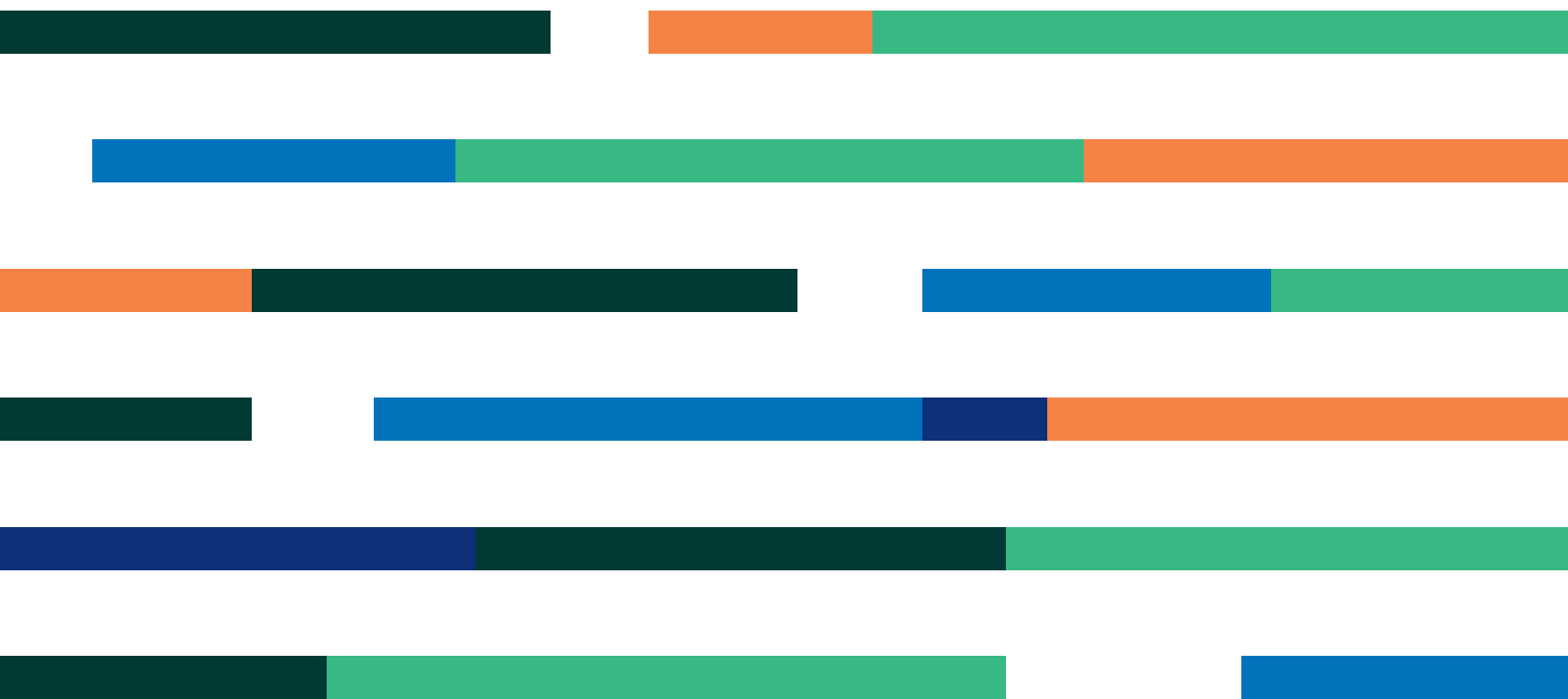


Security Controls for the “OWASP Kubernetes Top 10”

Using NeuVector to Reduce Risk in Kubernetes



Introduction

Kubernetes has become the de-facto standard for container orchestration platforms and is widely used in business critical infrastructure in enterprises of all sizes. With this popularity comes an increase in focus for hackers to exploit vulnerabilities and misconfigurations in Kubernetes clusters. The orchestration layer system resources as well as the application workloads running on it are all prime targets for attackers.

The non-profit organization OWASP, famous for its [OWASP Top 10](#) web application attacks has recently published its initial draft of the [OWASP Kubernetes Top 10](#) outlining the top 10 security risks for Kubernetes.

This guide summarizes those risks and provides insight into how the open source security platform NeuVector can help to mitigate these risks.

Summary Table: NeuVector Security Controls for OWASP K8S Top 10

Kubernetes Risk Vector	Description	NeuVector Security Controls
K01: Insecure workload configurations	Misconfigurations lead to vulnerable workloads	Audit, Admission Controls, and CIS
K02: Supply chain vulnerabilities	Malware, back doors, crypto mining and vulnerabilities introduced in pipeline	Admission Controls, Image Signing, Scanning
K03: Overly permissive RBAC configurations	Unauthorized system resource and console access leads to cluster compromise	Zero-Trust run-time network and process protections
K04: Lack of centralized policy enforcement	Security misconfigurations from lack of centralized, automated policy management	Centralized Admission Controls, Security as Code, Multi-Cluster Federation

Kubernetes Risk Vector	Description	NeuVector Security Controls
K05: Inadequate logging and monitoring	Attack detection and forensics are difficult without security focused event logging	Security Focused Events, Notifications, Packet Captures
K06: Broken authentication mechanisms	Unauthorized access to system resources can lead to lateral movement, corruption, data theft	Zero-Trust Suspicious Activity Detection
K07: Missing network segmentation controls	Lateral movement, network scanning, tunneling, command and control connections can't be stopped	Full Layer7 Firewall, Segmentation, WAF/DLP, Access Control
K08: Secrets management failures	Unprotected secrets could enable an attacker to gain access to resources or workloads	Suspicious System Activity Detection, Secrets Scanning
K09: Misconfigured cluster components	Misconfiguration of system components such as API server, kubelet, and etcd expose risks	Kubernetes and Docker CIS Benchmarks
K10: Outdated and vulnerable Kubernetes components	Critical CVE's in Kubernetes or other system (nginx, Istio) containers leads to exploit	Platform Scanning, CVE Reporting, CIS Benchmarks
Other risks	Zero-day attacks, OWASP Top 10 Web Application Attacks	Zero-Trust Run-Time Security, WAF rules, API Security

K01: Insecure workload configurations

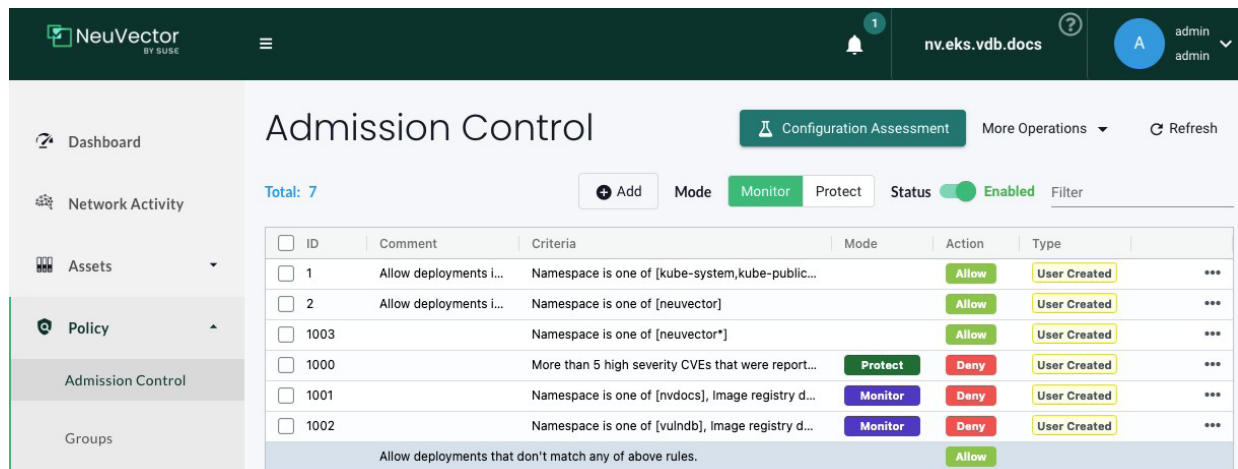
Misconfigurations can lead to deployments of workloads which are susceptible to exploits. Many Kubernetes operators, especially those new to the platform, are not aware of settings in the deployment manifest which can expose an attack surface. Examples include running application containers as root or running as privileged containers.

All deployment manifests for production environments should be audited for potential insecure configurations,

and admission control rules should be created to block (or warn) deployments which don't meet company policy.

NeuVector controls – Audit, admission controls, and CIS

NeuVector audits deployment yaml files by uploading them and analyzing them against rules created in the Policy -> Admission section of NeuVector. In production, NeuVector integrates with the Kubernetes admission controller to block unauthorized and risky deployments.



ID	Comment	Criteria	Mode	Action	Type
1	Allow deployments i...	Namespace is one of [kube-system,kube-public...	Allow	Allow	User Created
2	Allow deployments i...	Namespace is one of [neuvector]	Allow	Allow	User Created
1003		Namespace is one of [neuvector*]	Allow	Allow	User Created
1000		More than 5 high severity CVEs that were report...	Protect	Deny	User Created
1001		Namespace is one of [nvdocs], image registry d...	Monitor	Deny	User Created
1002		Namespace is one of [vulndb], image registry d...	Monitor	Deny	User Created
		Allow deployments that don't match any of above rules.	Allow	Allow	

In addition to checking for root, privileged, and allowPrivilegeEscalation workloads, NeuVector is also able to evaluate other security risks such as vulnerabilities (CVEs), volume mounts, and block unsigned images. When exceptions are granted by the security team for certain applications, rules can be created to allow only those applications by restricting

them to image names, registries, and/or namespaces.

NeuVector also runs the CIS benchmarks for Kubernetes and docker on images as well as on production resources such as kubelets, nodes, and containers to ensure best practices for security Kubernetes workloads are followed.

K02: Supply chain vulnerabilities

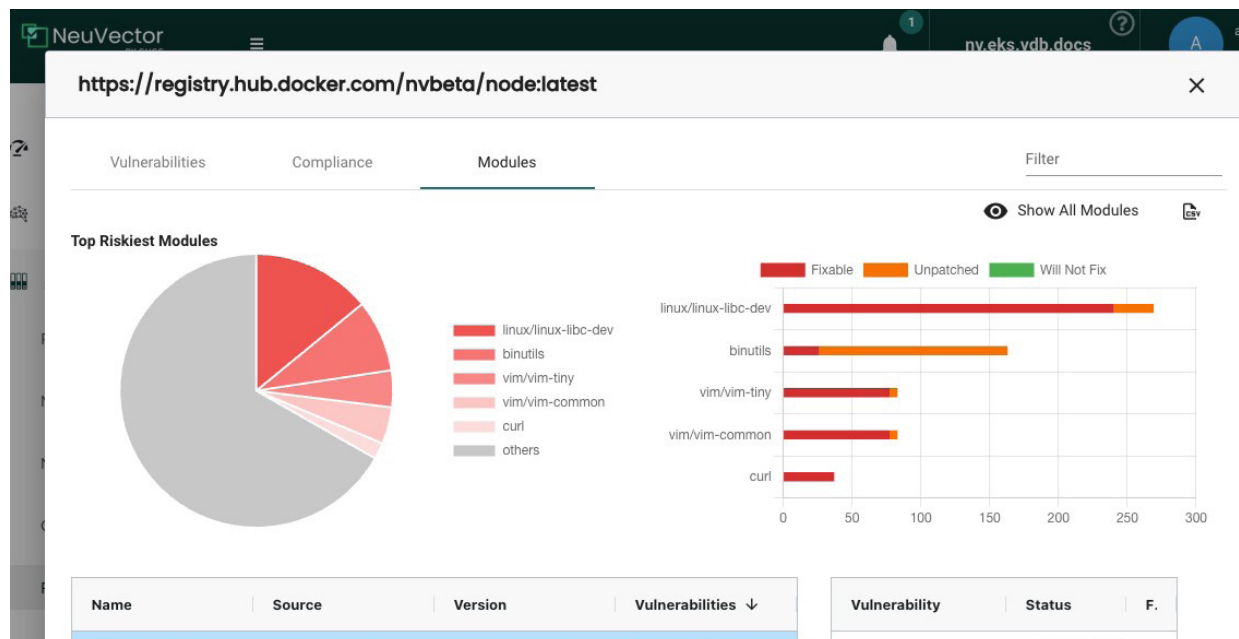
Containers deployed to production have a lifecycle which starts much earlier than the purview of Kubernetes. During this lifecycle, at any point, security risks can be introduced which make them vulnerable once running in production. These include image integrity (e.g. malware introduction), image composition (unauthorized components), and known software vulnerabilities.

With the increased use of open source software, often packaged with customized application code, it is difficult to monitor the supply chain for the entire application lifecycle. Code scanning, image scanning,

and other application security tools can be combined with Kubernetes admission controls to lock down the supply chain. Newer standards around software bill of materials (SBOMs) and SLSA can help to protect the supply chain.

NeuVector controls – admission controls, image signing, scanning

NeuVector scans container images, either in the pipeline (build phase) or continuously in registries to find known vulnerabilities in open source software. It will also scan for embedded secrets, overly permissive file permissions, and compile a summary of the components (modules), as shown below.



Once images are approved for production, they can be signed to protect image integrity. Image signatures can be required by the NeuVector admission controller in order to allow deployments.

Ultimately, should the supply chain become compromised, any attempted exploit should be detected by a properly configured NeuVector run-time security policy based on zero-trust controls for network, process, and file activity of the applications.

K03: Overly permissive RBAC configurations

Kubernetes [Role-Based Access Controls](#) (RBACs) provide a convenient way to control access to critical Kubernetes system resources. However, if compromised, overly permissive RBACs such as the cluster-admin role can lead to attacks on resources themselves, or lateral compromise of application workloads.

RBAC configurations should carefully be created and reviewed, and periodically evaluated to ensure the practice of 'least privileged access' is followed.

NeuVector controls – run-time network and process protections

Although there are separate tools for evaluating RBAC settings, NeuVector does monitor system resources and application workloads. Any unusual behavior (e.g. network connections) in system resources or application workloads (network connections, process or file activity) can be detected. Exploits are often comprised of a 'kill chain' – a series of events as the blast radius of the attack expands to probe further for weaknesses or sensitive applications. In the future, as part of the auditing features in NeuVector, RBAC settings and other Kubernetes resources will be able to be evaluated for security risk.

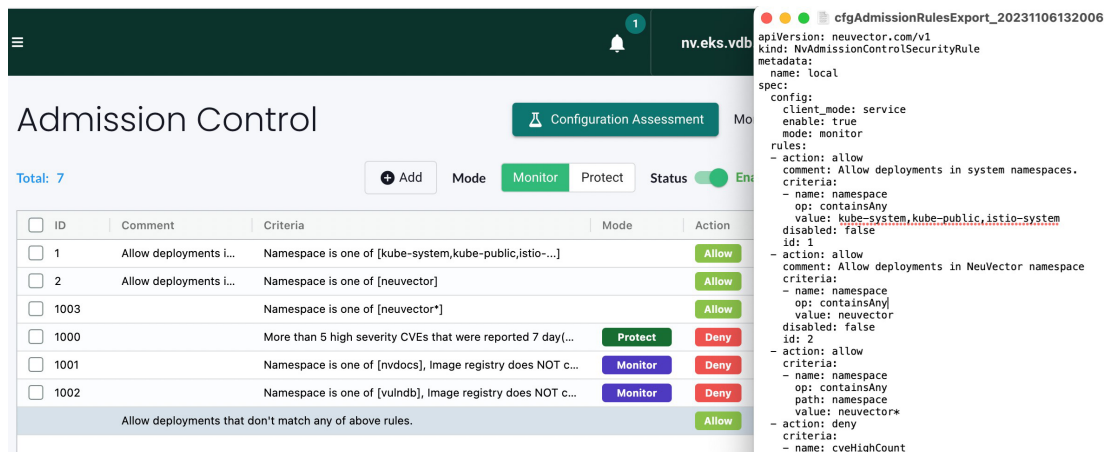
K04: Lack of centralized policy enforcement

While small deployments may have only a few critical security policies, large deployments with multiple application pipelines and clusters distributed across public and private clouds can require hundreds of security rules across many policy categories. Policies must be able to be created and maintained using as much automation as possible to avoid stale or missed application of policies, especially in highly dynamic environments.

Two important Kubernetes resources which support centralized policy enforcement are the admission controller and custom resource definitions (CRDs). Admission control rules should be viewed and managed for each cluster, with rules across clusters maintained in central repositories. CRDs can be used to express admission control rules but can go beyond that to express all critical security rules for clusters and applications. CRDs are typically yaml-based readable files which can declare the desired state of various security policies for a cluster. These files can be managed in a gitops manner, implementing 'security as code.' Managing security policy this way helps prevent misconfigurations through security consoles and policy drift over time.

NeuVector controls – centralized admission controls, security as code, multi-cluster federation

NeuVector admission controls provide a centralized, console-based view into all admission control rules. For example, a rule can be added to disallow deployments from untrusted registries (or conversely only allow from trusted registries). The GUI, console based view provides a convenient way to view all policies together to ensure the correct behavior of security controls. As shown below, the rules can be exported to a CRD yaml based format for gitops security as code, ensuring that this file becomes the single source of security policy for this and other clusters.



The screenshot shows the NeuVector Admission Control console. On the left, a table lists admission control rules with columns for ID, Comment, Criteria, Mode, and Action. The table shows 7 rules in total. On the right, a YAML export view for a rule named 'local' is displayed, showing the configuration for the admission control rule, including the namespace and the action to allow or deny deployments.

ID	Comment	Criteria	Mode	Action
1	Allow deployments i...	Namespace is one of [kube-system,kube-public,istio-...]	Allow	Allow
2	Allow deployments i...	Namespace is one of [neuvevector]	Allow	Allow
1003	Namespace is one of [neuvevector*]		Allow	Allow
1000	More than 5 high severity CVEs that were reported 7 day...		Protect	Deny
1001	Namespace is one of [nvdocs], Image registry does NOT c...		Monitor	Deny
1002	Namespace is one of [vulndb], Image registry does NOT c...		Monitor	Deny
Allow deployments that don't match any of above rules.				Allow

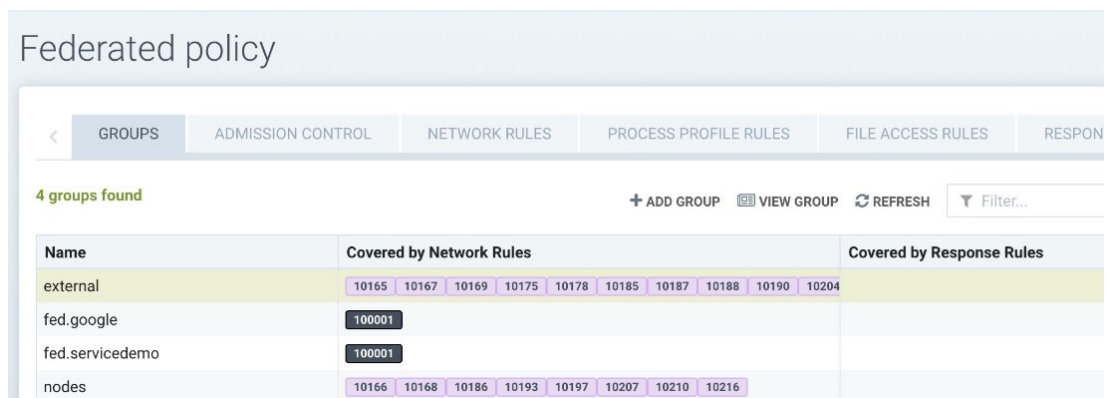
```

apiVersion: neuvector.com/v1
kind: NvAdmissionControlSecurityRule
metadata:
  name: local
spec:
  config:
    client_mode: service
    enable: true
    mode: monitor
  rules:
    - action: allow
      comment: Allow deployments in system namespaces.
      criteria:
        - name: namespace
          op: containsAny
          value: kube-system,kube-public,istio-system
      disabled: false
      id: 1
    - action: allow
      comment: Allow deployments in NeuVector namespace
      criteria:
        - name: namespace
          op: containsAny
          value: neuvector
      disabled: false
      id: 2
    - action: allow
      criteria:
        - name: namespace
          op: containsAny
          path: namespace
          value: neuvector*
      action: deny
      criteria:
        - name: cveHighCount

```

While the example above focuses on admission control policy, other security policy such as run-time security should also be treated similarly. In NeuVector, run-time policy which governs allowed behavior of resources and workloads in the form of network connections and process/file activity are supported in both console and CRD form.

NeuVector also provides Multi-cluster Federation capabilities, where security rules can be automatically propagated from a Primary cluster down to all connected remote clusters.



The screenshot shows the NeuVector Federated policy console. It displays a table of groups and their associated rules. The table has columns for Name, Covered by Network Rules, and Covered by Response Rules. The groups listed are external, fed.google, fed.servicedemo, and nodes. The rules are listed as 10165, 10167, 10169, 10175, 10178, 10185, 10187, 10188, 10190, 10204 for external; 100001 for fed.google and fed.servicedemo; and 10166, 10168, 10186, 10193, 10197, 10207, 10210, 10216 for nodes.

Name	Covered by Network Rules	Covered by Response Rules
external	10165 10167 10169 10175 10178 10185 10187 10188 10190 10204	
fed.google	100001	
fed.servicedemo	100001	
nodes	10166 10168 10186 10193 10197 10207 10210 10216	

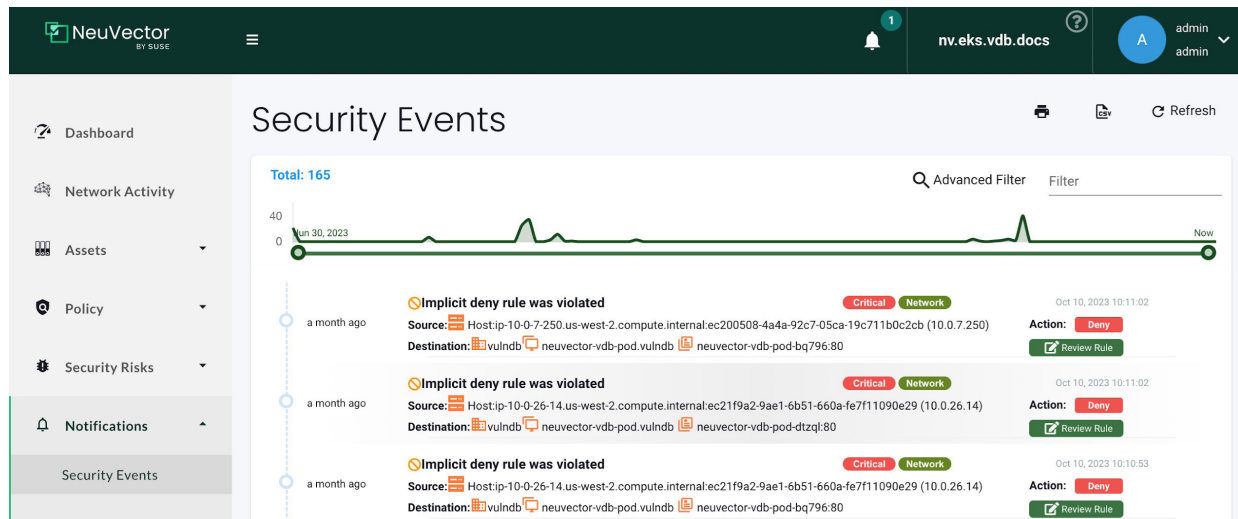
This can make creating and updating rules across multiple clusters efficient and less prone to errors.

K05: Inadequate logging and monitoring

Tools that are not specifically designed for security do not often provide the level of logging and context required for detecting attacks, and Kubernetes is no exception. While it is necessary to enable all the logging and analytics which Kubernetes provides, it is insufficient for exploit detection and prevention.

NeuVector controls – security-focused events, notifications, packet captures

Kubernetes security platforms such as NeuVector provide security event notifications across the container lifecycle, from the pipeline into production. Detailed context logging such as source/destination network connection attempts, images, pods and nodes affected, and even packet captures ensures that proper logging of critical details occurs which can support forensic investigations.



Webhook alerts, SIEM system and SYSLOG integrations, and console-based reporting are provided and are critical for security teams to respond to attacks.

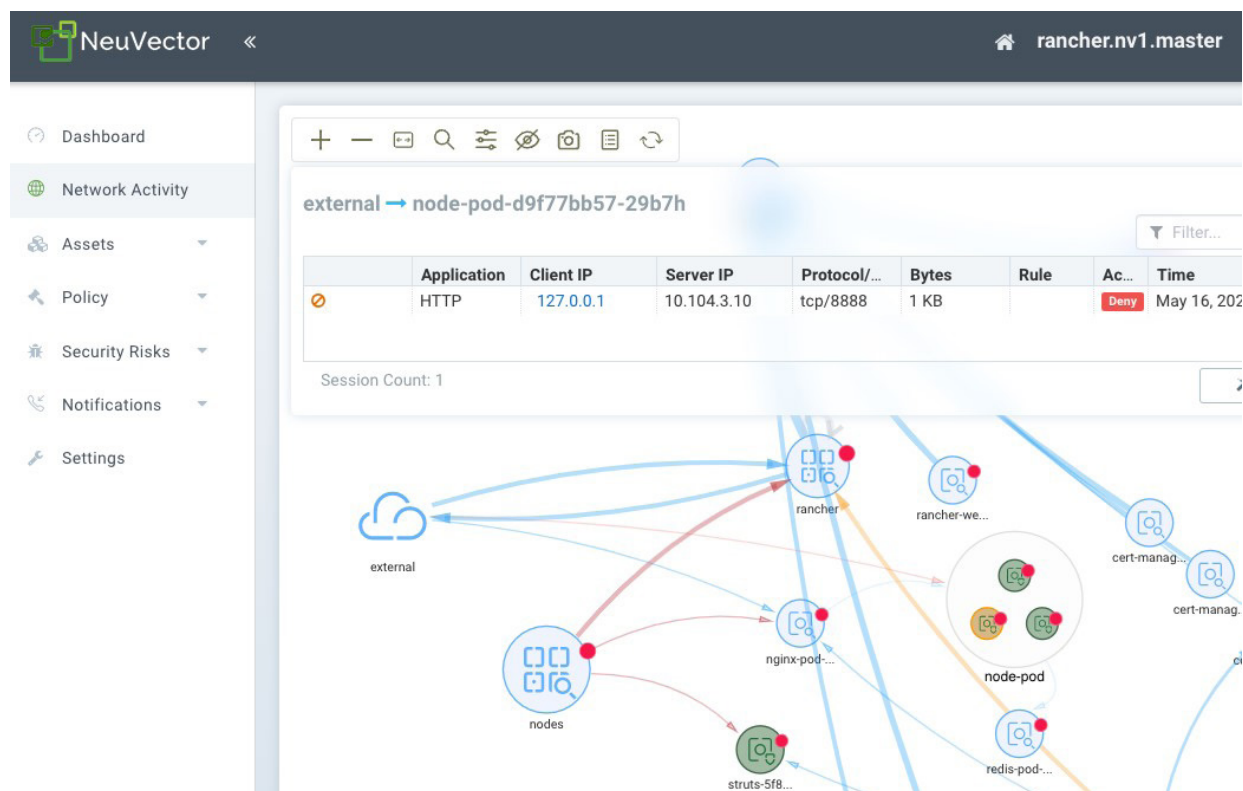
K06: Broken authentication mechanisms

Authentication mechanisms for communication with critical Kubernetes resources such as the API server must be properly configured to protect against compromises. Even with a full lifecycle security platform like NeuVector deployed, if a threat actor gains admin access to the API server, Kubernetes console, or other system resources they can wreak havoc on the cluster. If such a compromise occurs, it is possible that NeuVector will still detect suspicious activity as the hacker attempts to explore and scan workloads and other resources to discover sensitive data or establish

unauthorized external connections. However, corruption of the cluster, including NeuVector workloads, is still possible with such privileged access.

NeuVector controls – suspicious activity detection

As a hacker attempts to access workloads, expand the blast radius of the attack, or establish external connections to command and control servers, NeuVector's zero-trust run-time security policy will detect any suspicious unallowed activity in the form of network connections, suspicious processes or unauthorized file access. Privilege escalations are also detected.



Suspicious activity can be blocked or set to just log and alert the security team for investigation.

K07: Missing network segmentation controls

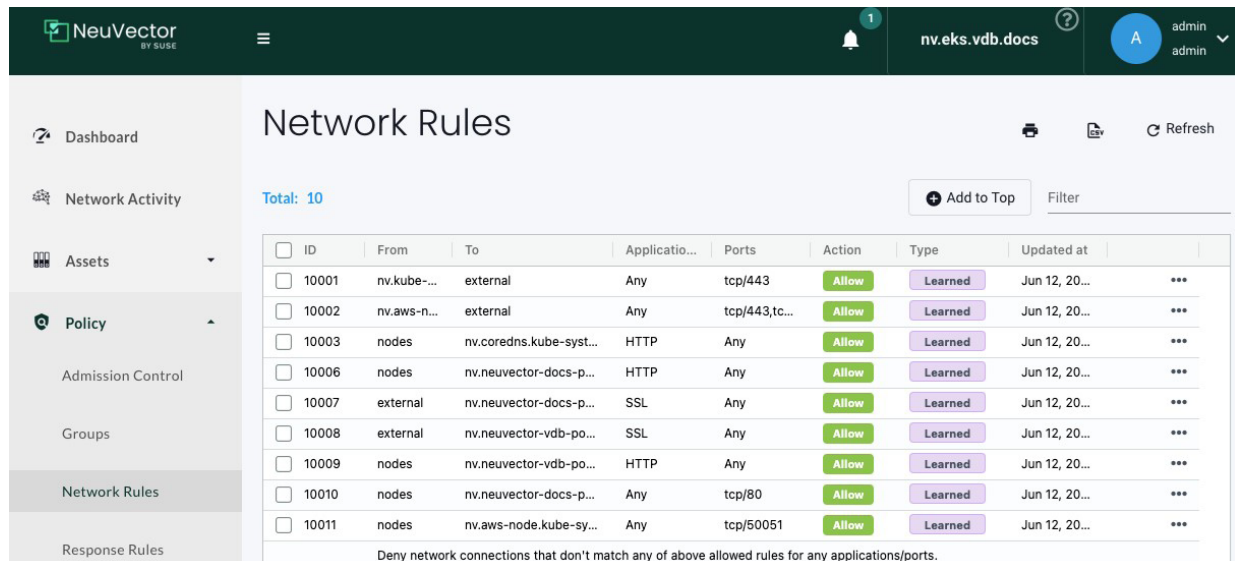
Network segmentation is one of the most critical protections against attacks. As hackers explore a cluster for weaknesses and sensitive data, suspicious network connections should be detected for ingress, lateral movement, port scanning, application probing and other steps in the 'kill chain.' The problem with Kubernetes networking is that current controls such as network policy rules are not so sophisticated and flexible enough and are easily bypassed. Traditional appliance or host-based firewalls lack the Kubernetes context to make any sense of connections and are too slow to process the high volume of east-west network traffic in large microservices-based clusters.

In addition, the high velocity of application workload deployment and updates makes it impossible to create and update network rules using traditional console-

based or manual configurations requiring human intervention.

NeuVector controls – full layer7 firewall, segmentation, WAF/DLP, access control

NeuVector provides unique, patented container layer7 firewalling and network protection designed to work in a high velocity cloud environment. At the core of NeuVector are application segmentation rules which are automatically discovered (learned), created, and can be automated in the form of Kubernetes custom resource definitions (CRDs). This default mode of NeuVector is based on zero-trust declarations of the allowed network connections for containers (applications), pods, and nodes. This includes strong egress controls for allowed connections outside of the cluster to internal networks (defined by IP addresses or ranges) or discoverable services both internal and external (defined by FQDN).



The screenshot shows the NeuVector web interface for managing Network Rules. The interface includes a sidebar with navigation options: Dashboard, Network Activity, Assets, Policy (selected), Admission Control, Groups, Network Rules, and Response Rules. The main content area displays a table of Network Rules with columns: ID, From, To, Application, Ports, Action, Type, and Updated at. There are 10 rules listed, all with an 'Allow' action and 'Learned' type. The rules cover various network connections between Kubernetes components and external services. A 'Total: 10' summary is shown above the table. A 'Filter' button and 'Add to Top' link are also present.

ID	From	To	Application	Ports	Action	Type	Updated at
10001	nv.kube-...	external	Any	tcp/443	Allow	Learned	Jun 12, 20...
10002	nv.aws-n...	external	Any	tcp/443,tc...	Allow	Learned	Jun 12, 20...
10003	nodes	nv.coredns.kube-syst...	HTTP	Any	Allow	Learned	Jun 12, 20...
10006	nodes	nv.neuvector-docs-p...	HTTP	Any	Allow	Learned	Jun 12, 20...
10007	external	nv.neuvector-docs-p...	SSL	Any	Allow	Learned	Jun 12, 20...
10008	external	nv.neuvector-vdb-po...	SSL	Any	Allow	Learned	Jun 12, 20...
10009	nodes	nv.neuvector-vdb-po...	HTTP	Any	Allow	Learned	Jun 12, 20...
10010	nodes	nv.neuvector-docs-p...	Any	tcp/80	Allow	Learned	Jun 12, 20...
10011	nodes	nv.aws-node.kube-sy...	Any	tcp/50051	Allow	Learned	Jun 12, 20...

Deny network connections that don't match any of above allowed rules for any applications/ports.

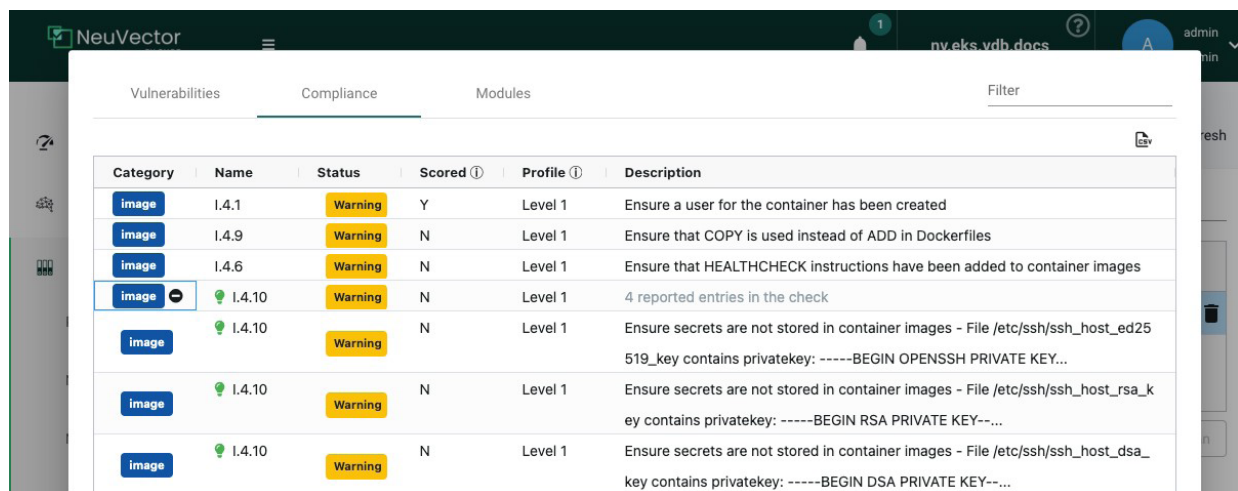
NeuVector also supports other network controls such as deny lists, data loss prevention (DLP inspection of packets for sensitive data), and web application firewall (WAF) rules.

K08: Secrets management failures

Secrets provide valuable information and access to various protected resources, but when compromised can allow an attacker to access these resources as an insider. Kubernetes provides mechanisms to store and protect these secrets, and these should be used and regularly reviewed for proper configuration. Third party secrets management software is also highly recommended to enterprise scale management of secrets.

NeuVector controls – suspicious system activity detection, secrets scanning

While protection of secrets should be addressed outside of the NeuVector platform, NeuVector provides some controls for evaluating and detecting compromises. Image scanning includes scanning for secrets (e.g. authentication keys) embedded in images, which could easily be discovered and compromised. The admission controller in NeuVector can deny deployments which contain secrets in environment variables.



Category	Name	Status	Scored ①	Profile ①	Description
image	I.4.1	Warning	Y	Level 1	Ensure a user for the container has been created
image	I.4.9	Warning	N	Level 1	Ensure that COPY is used instead of ADD in Dockerfiles
image	I.4.6	Warning	N	Level 1	Ensure that HEALTHCHECK instructions have been added to container images
image	I.4.10	Warning	N	Level 1	4 reported entries in the check
image	I.4.10	Warning	N	Level 1	Ensure secrets are not stored in container images - File /etc/ssh/ssh_host_ed25519_key contains privatekey: -----BEGIN OPENSSH PRIVATE KEY...
image	I.4.10	Warning	N	Level 1	Ensure secrets are not stored in container images - File /etc/ssh/ssh_host_rsa_key contains privatekey: -----BEGIN RSA PRIVATE KEY-----
image	I.4.10	Warning	N	Level 1	Ensure secrets are not stored in container images - File /etc/ssh/ssh_host_dsa_key contains privatekey: -----BEGIN DSA PRIVATE KEY-----

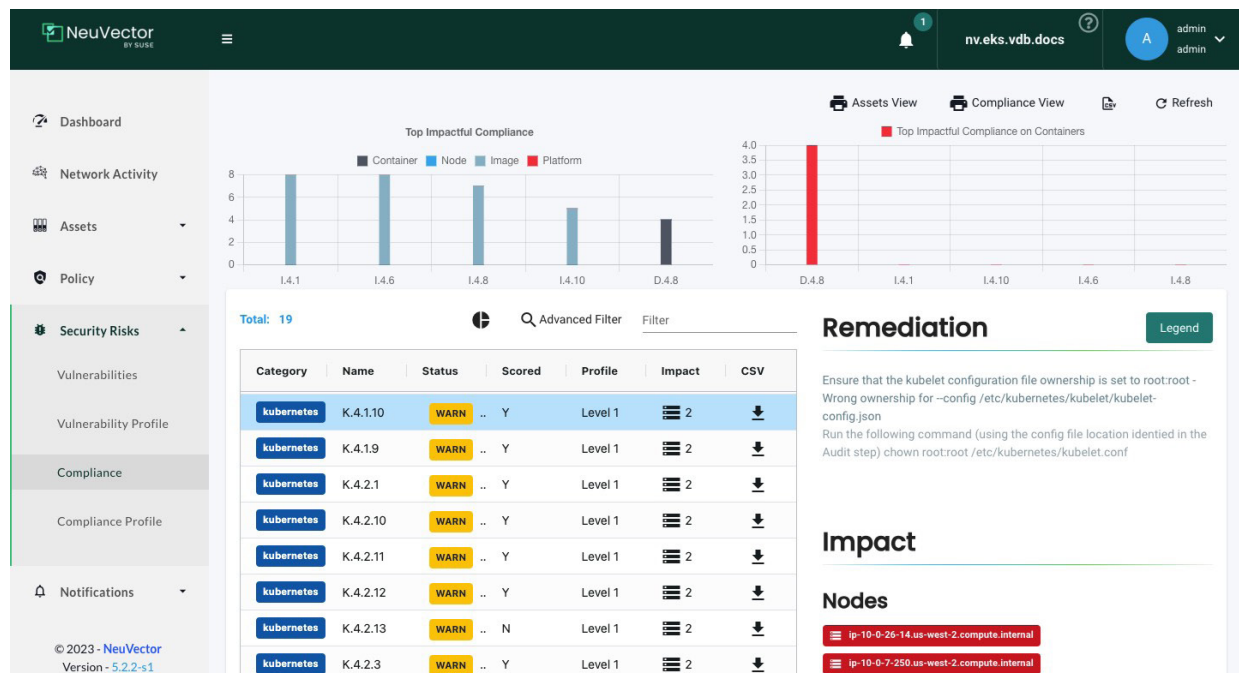
In addition, NeuVector run-time security will monitor all system containers for suspicious activity, which could occur after secrets were compromised.

K09: Misconfigured cluster components

Kubernetes system components such as the API server, kubelet and etcd must be properly configured and protected to reduce the risk of attack on the infrastructure.

NeuVector controls – Kubernetes and docker CIS benchmarks

NeuVector continuously runs the Kubernetes and Docker CIS benchmarks to review configuration and warn if best practices are not followed. These auditing compliance checks are reported through the console and can be customized to include these checks desired. Custom compliance checks can be configured to check containers or host configurations which are not included in the CIS Benchmarks, and these can be added to the compliance reporting.



NeuVector scans images for vulnerabilities, embedded secrets, and other overly permissive file configurations. An auditing function in Admission Controls enables users to scan deployment yaml files for compliance violations. In addition, NeuVector run-time security will monitor all system containers for suspicious activity.

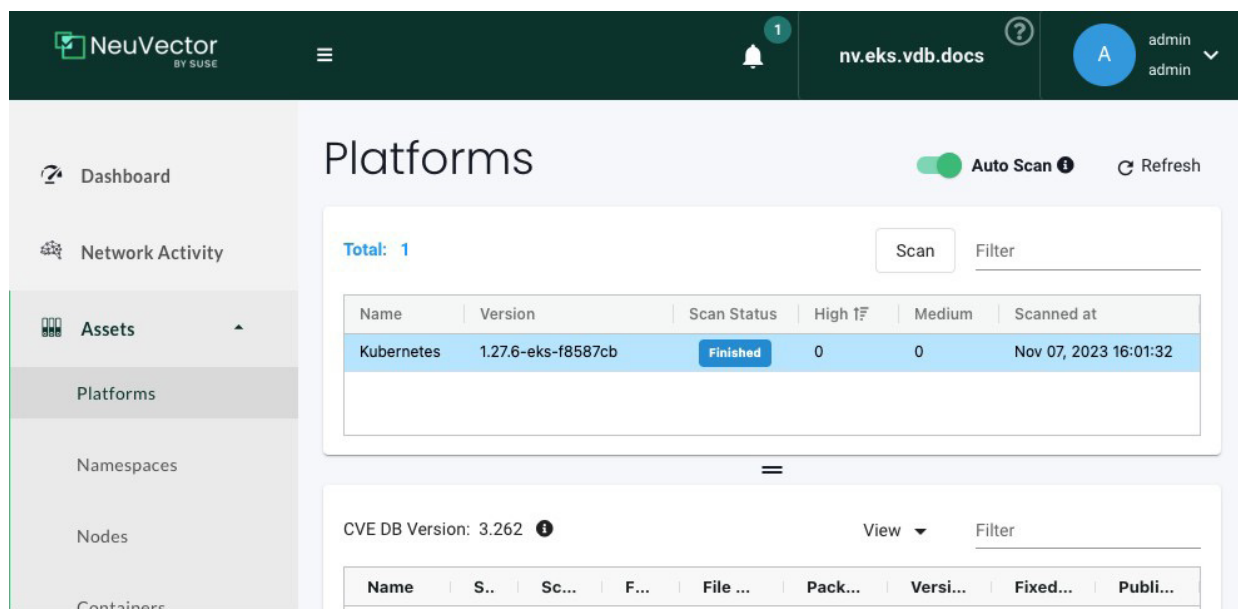
K10: Outdated and vulnerable Kubernetes components

Kubernetes components have experienced critical vulnerabilities in recent years and new ones can be announced at any time for current and past versions. It is critical to update Kubernetes to more recent supported, stable versions in order to avoid past vulnerabilities. Regular scanning of all components for new CVEs is critical to analyzing any threat of exploit to a cluster. Depending on other security controls and configurations

in place, it may or may not require an urgent upgrade or special security action to reduce the risk.

NeuVector controls – platform scanning, CVE reporting, CIS benchmarks

NeuVector automatically scans Kubernetes components and resources to identify vulnerabilities. In addition, when it is possible to deploy NeuVector on nodes with management components such as the API server, NeuVector monitors activity for suspicious behavior and can alert and log events.



NeuVector BY SUSE

nv.eks.vdb.docs

admin admin

Platforms

Auto Scan Refresh

Total: 1

Scan Filter

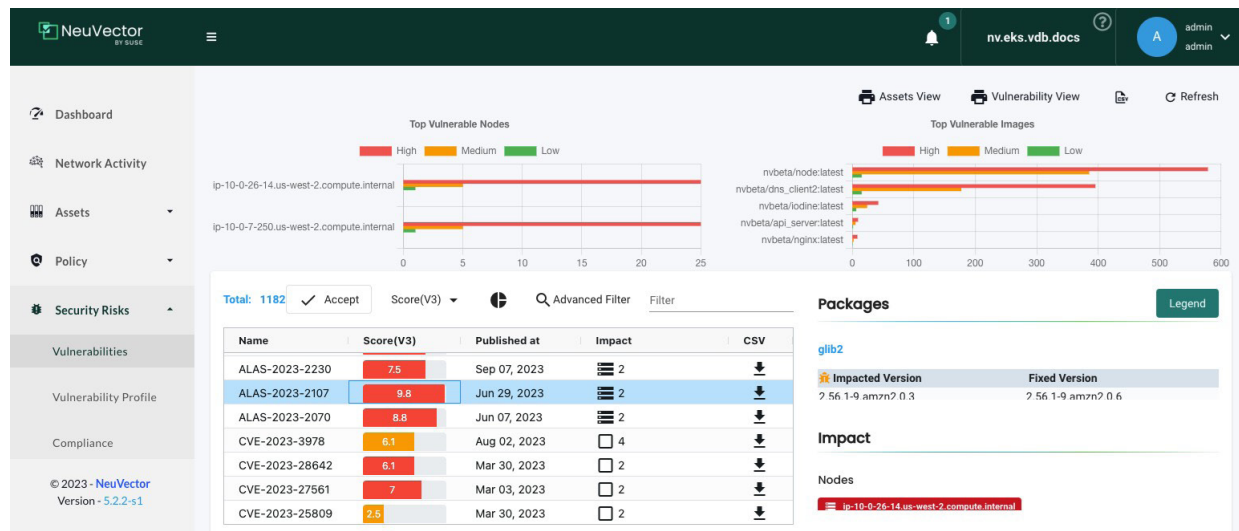
Name	Version	Scan Status	High CVEs	Medium CVEs	Scanned at
Kubernetes	1.27.6-eks-f8587cb	Finished	0	0	Nov 07, 2023 16:01:32

CVE DB Version: 3.262

View Filter

Name	S...	Sc...	F...	File ...	Pack...	Versi...	Fixed...	Publi...
------	------	-------	------	----------	---------	----------	----------	----------

Other workloads deployed on Kubernetes which are generally considered system components such as service meshes (e.g. Istio) and monitoring tools (e.g. Prometheus/Grafana) are also continuously scanned for CVEs (and compliance violations). A ‘vulnerability explorer’ tool in NeuVector combines Kubernetes system, application workload, and node scan results for analysis and reporting.



The ‘impact’ of these vulnerabilities is also displayed, showing which images or running containers and nodes are affected.

Other risks to consider

In addition to attacks on Kubernetes resources and workloads, applications themselves can be vulnerable to exploits. These include critical published vulnerabilities (CVEs) as well as previously unknown zero-day attacks. Even remediating every single CVE from a production environment won’t protect against zero-day attacks. That is why the most critical protection against application exploits is a zero-trust run-time security control. Protection against traditional [OWASP Top 10](#) attacks with WAF rules is an important addition to application security for container workloads.

NeuVector's zero-trust controls are designed to only allow declared behaviors and alert/deny everything else. In this way, new previously unknown attack vectors can be detected and blocked. In addition, traditional WAF rules are supported, and these can be selectively applied to only workloads which expose web applications.

The WAF rules interface above can also be used to create protections for critical API access. NeuVector can inspect the header, URL, body, or full packet and enforce either allow rules or deny rules. These rules can be applied to both outgoing connections from pods as well as incoming connections.

The risk of attackers gaining access to critical resources continues to grow, especially for new cloud technologies such as containers and Kubernetes. In addition to the traditional zero-day application attacks, exploits of misconfigured Kubernetes system or workload configurations are a real threat to business continuity. A layered security strategy is always the best way to mitigate risk. Security should have several layers through which attackers must penetrate before being able to access critical resources and data. No one security tool can or should provide all layers of security, but as seen in the above sections, the NeuVector container security platform provides best in class controls and layers required to detect and prevent exploits.



SUSE Software Solutions
Germany GmbH

Frankenstraße 146
90461 Nürnberg
Germany

www.suse.com

For more information, contact SUSE at:

+1 800 796 3700 (U.S./Canada)

+49 (0)911-740 53-0 (Worldwide)

Thank You

SC000121 | © 2024 SUSE LLC. All Rights Reserved. SUSE and the SUSE logo are registered trademarks of SUSE LLC in the United States and other countries. All third-party trademarks are the property of their respective owners.