EXECUTING ON NIST SP 800-190

Twistlock
Enterprise Security. DevOps Agility.

EXECUTING ON NIST SP 800-190

The National Institute of Standards and Technology (NIST) recently released a draft of Special Publication (SP) 800-190 that provides guidance on securing application containers.

NIST SP 800-190 does an excellent job of describing the security risks and associated countermeasures for safeguarding containerized apps. Organizations, however, are left to fend for themselves on the tactical front. If you're securing containerized apps, then you'll likely want to deploy the controls it describes. Open source offers only a few disparate pieces, and traditional commercial offerings still cannot address the concerns unique to containers.

Enter Twistlock. Twistlock is a security suite that is purpose-built for containers. Since our inception, we've been developing technology to address the types of risks described in NIST SP 800-190.

NIST Special Publications and SP 800-190

NIST develops standards and guidelines for all federal computer systems (except classified computer systems, which falls under the jurisdiction of the National Security Agency). NIST produces the 800 series of Special Publications. These publications draw from industry, government, and academia to provide computer, cyber, and information security guidelines and recommendations.

The impact of SP 800-190 is twofold:

> **For government agencies,** the impact is immediate. Government agencies are expected to comply with NIST security standards and guidelines within a year of a publication's release date. Information systems that are currently under development are expected to be compliant upon deployment. Government agencies need to grok the guidelines in SP 800-190 now, and plan how to roll them out. Twistlock implements most of the countermeasures in SP 800-190 and can help immediately.
>
> **For industry,** there's more time. As companies plan their own rollout, they now have an important new resource from which to glean best practices. NIST special publications are useful to all security practitioners and they are available from NIST for free. To date, there are still very few resources and shared information on securing container environments. The other notable example is the Center for Internet Security (CIS) Docker Benchmark.

Challenges

The considerations and methods for securing containers has morphed with the evolution of infrastructure. Docker streamlines how you package, store, and deploy apps in containers. Because containers encapsulate all their dependencies, they move easily from development, to test, to production, making frequent deployments by way of automation the new reality. Apps built using the microservices architecture consist of many entities, and the number of deployed containers mushroom as orchestrators seamlessly scale your apps up (or down) according to demand.

Containers enable a lot of exciting new efficiencies, but present some challenges:

Scale
Google launches 2 billion containers a week in their infrastructure. Organizations have to think about the controls required to secure such massive deployments. The old method of manually creating and maintaining security rules for each entity in the app is impractical.

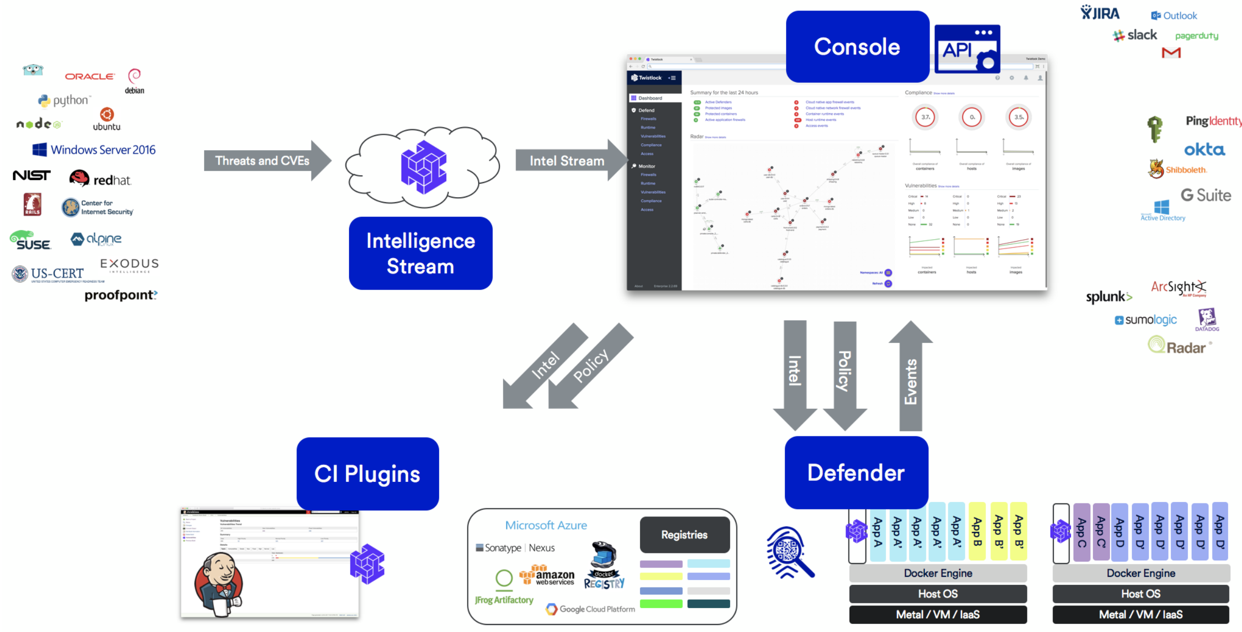
Rate of change
Deployment cycles have been chopped down from one release per quarter to several releases per day. With each release, apps change in subtle ways, and the security rules that protect them must keep up. Otherwise, new attack vectors inadvertently crop up.

Lifecycle
Security is no longer just the domain of a single team or person. Organizations must provide the right tools during each phase of the container lifecycle. For example, engineers now define the environments in which their apps run, and their decisions directly impact the security of the app. Are the right tools available during the development phase to build secure images?

Twistlock

Twistlock is the leading provider of container and cloud native cybersecurity solutions. From precise, actionable vulnerability management to automatically deployed runtime protection and firewalls, Twistlock protects applications across the development lifecycle and into production. Purpose built for containers, serverless, and other leading technologies - Twistlock gives developers the speed they want, and CISOs the controls they need.

A container environment, in general, encompasses your images, containers, hosts, Docker daemons, registries, and orchestrator. The following diagram provides an overview of the major components in a Twistlock deployment:



Twistlock Intelligence Stream

Live feed that delivers the latest threat data to your Twistlock installation.

Console

Management interface for declaring security policies and monitoring the health of a deployment. It also provides an API for controlling the product programmatically.

Defender

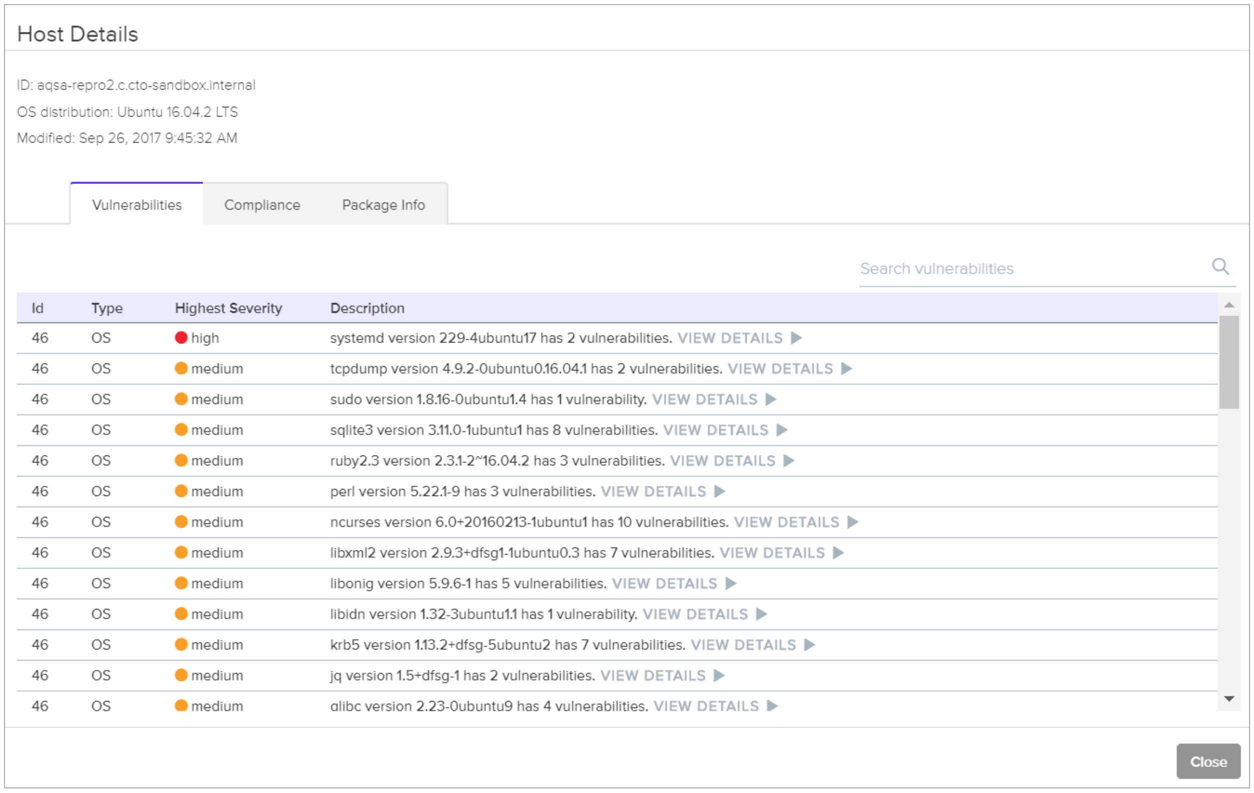
Containerized security agent that enforces the policies defined in Console. Defender is installed on any machine that runs containers.

Twistlock implements the following NIST SP 800-190 countermeasures. These countermeasures are enabled either by default or with minimal extra configuration.

ID	COUNTERMEASURE
4.1	**Image Countermeasures**
4.1.1	Image vulnerabilities
4.1.2	Image configuration defects
4.1.3	Embedded malware
4.1.4	Embedded clear text secrets
4.1.5	Use of untrusted images
4.4	**Container Countermeasures**
4.4.1	Vulnerabilities within the runtime software
4.4.2	Unbounded network access from containers
4.4.3	Insecure container runtime configurations
4.4.4	App vulnerabilities
4.4.5	Rogue containers
4.5	**Host OS Countermeasures**
4.5.1	Large attack surface
4.5.2	Shared kernel
4.5.3	Host OS component vulnerabilities
4.5.4	Improper user access rights
4.5.5	Host file system tampering

Securing The Stack

Securing a container environment requires securing the rest of the underlying stack. Organizations should regularly check for and apply updates to all software components installed on the host OS. Cloud-native operating systems, such as CoreOS, ship with auto-updating enabled by default. Twistlock adds another layer of defense by continually scanning the host OS for vulnerable packages that could be entry points for an attack. (SP 800-190 countermeasure 4.5.3) The following screenshot shows the scan report for a host in the container environment:



ID: aqsa-repro2.c.cto-sandbox.internal
OS distribution: Ubuntu 16.04.2 LTS
Modified: Sep 26, 2017 9:45:32 AM

Id	Type	Highest Severity	Description
46	OS	● high	systemd version 229-4ubuntu17 has 2 vulnerabilities. VIEW DETAILS ▶
46	OS	● medium	tcpdump version 4.9.2-0ubuntu0.16.04.1 has 2 vulnerabilities. VIEW DETAILS ▶
46	OS	● medium	sudo version 1.8.16-0ubuntu1.4 has 1 vulnerability. VIEW DETAILS ▶
46	OS	● medium	sqlite3 version 3.11.0-1ubuntu1 has 8 vulnerabilities. VIEW DETAILS ▶
46	OS	● medium	ruby2.3 version 2.3.1-2~16.04.2 has 3 vulnerabilities. VIEW DETAILS ▶
46	OS	● medium	perl version 5.22.1-9 has 3 vulnerabilities. VIEW DETAILS ▶
46	OS	● medium	ncurses version 6.0+20160213-1ubuntu1 has 10 vulnerabilities. VIEW DETAILS ▶
46	OS	● medium	libxml2 version 2.9.3+dfsg1-1ubuntu0.3 has 7 vulnerabilities. VIEW DETAILS ▶
46	OS	● medium	libonig version 5.9.6-1 has 5 vulnerabilities. VIEW DETAILS ▶
46	OS	● medium	libidn version 1.32-3ubuntu1.1 has 1 vulnerability. VIEW DETAILS ▶
46	OS	● medium	krb5 version 1.13.2+dfsg-5ubuntu2 has 7 vulnerabilities. VIEW DETAILS ▶
46	OS	● medium	jq version 1.5+dfsg-1 has 2 vulnerabilities. VIEW DETAILS ▶
46	OS	● medium	glibc version 2.23-0ubuntu9 has 4 vulnerabilities. VIEW DETAILS ▶

You should also ensure all authentication to the OS is audited, anomalies are monitored, and any escalation to perform privileged operations is logged. Twistlock catches and logs all sudo and sshd events on any host protected by Defender. They show when someone runs commands with elevated privileges or establishes an SSH connection to a host. (SP 800-190 countermeasure 4.5.4)

Vulnerability Management

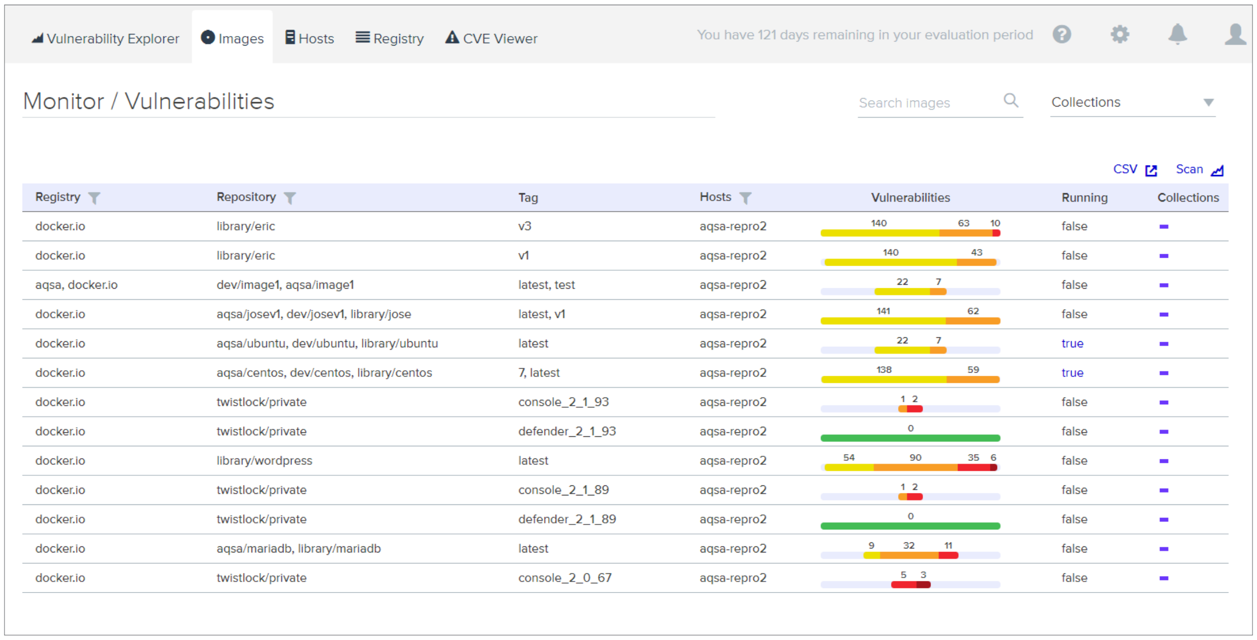
Organizations should employ container-specific vulnerability management tools and processes to look for Common Vulnerabilities and Exposures (CVEs) across all phases of the container lifecycle. You should upgrade any instances at risk, and ensure that orchestrators only permit deployments of properly maintained runtimes.

With containers, engineers now define the environments in which their apps run, and their decisions directly impact security of the environment. Organizations need to arm their engineers with the right tools to find and fix vulnerabilities. Twistlock provides a Jenkins plugin that lets you incorporate vulnerability scanning into the continuous integration pipeline. This plugin enables developers to find and fix security defects before images ever get to the registry (SP 800-190 countermeasure 4.1.1). The plugin also serves as a communications conduit between security and development teams. Twistlock lets the security teams create policies and standards to which the engineering teams must build their images. Non-compliant images fail the build. For example, a security team could define a policy that fails the build of any image with high severity vulnerabilities.

In addition to the Jenkins plugin, the Twistlock command line configuration and control tool, called twistcli, offers many capabilities for users -- one being the ability to scan images for vulnerabilities and compliance issues. The results are displayed locally without external dependencies. Because twistcli runs from the command line, users can easily integrate Twistlock scanning capabilities into custom tooling.

Drift is another issue. After engineers build a clean image, they push it to registry for distribution. An image deemed clean today, may not be clean tomorrow. Twistlock continually monitors the images in your registries and container environment for vulnerabilities. The Twistlock Intelligence Stream (a threat feed) is updated several times a day with new threat data, and that threat data is assessed against all containers in the environment. Scan reports provide intel about which images are vulnerable, and where they live. (SP 800-190 countermeasure 4.4.1).

The following screenshot shows a scan report. Twistlock scans images and containers for CVEs, malware, compliance issues, and zero-day vulnerabilities.



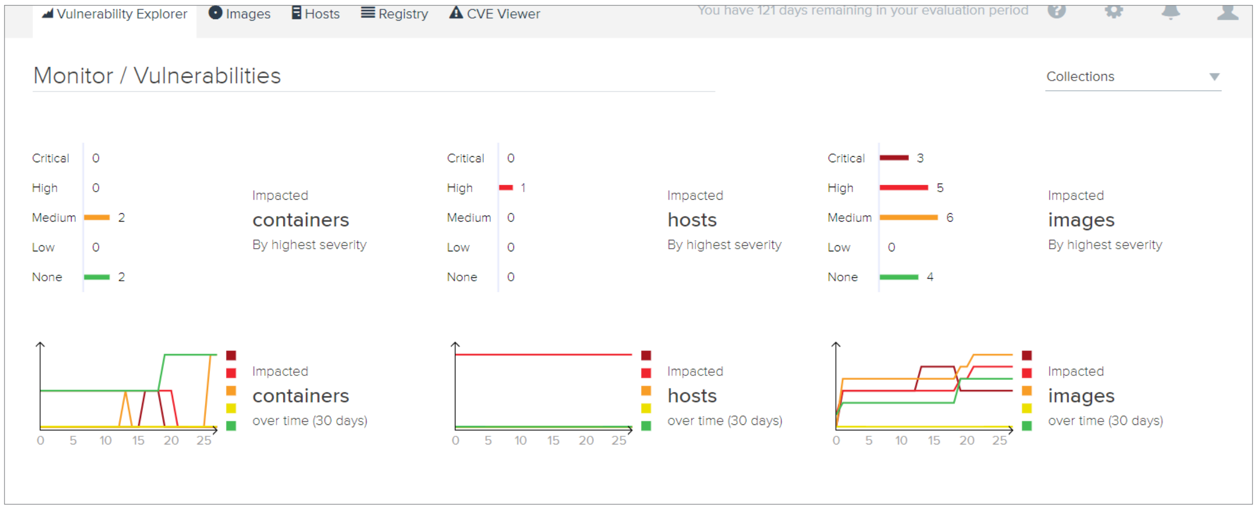
Registry	Repository	Tag	Hosts	Vulnerabilities	Running	Collections
docker.io	library/eric	v3	aqsa-repro2	140 63 10	false	▬
docker.io	library/eric	v1	aqsa-repro2	140 43	false	▬
aqsa, docker.io	dev/image1, aqsa/image1	latest, test	aqsa-repro2	22 7	false	▬
docker.io	aqsa/josev1, dev/josev1, library/jose	latest, v1	aqsa-repro2	141 62	false	▬
docker.io	aqsa/ubuntu, dev/ubuntu, library/ubuntu	latest	aqsa-repro2	22 7	true	▬
docker.io	aqsa/centos, dev/centos, library/centos	7, latest	aqsa-repro2	138 59	true	▬
docker.io	twistlock/private	console_2_1_93	aqsa-repro2	1 2	false	▬
docker.io	twistlock/private	defender_2_1_93	aqsa-repro2	0	false	▬
docker.io	library/wordpress	latest	aqsa-repro2	54 90 35 6	false	▬
docker.io	twistlock/private	console_2_1_89	aqsa-repro2	1 2	false	▬
docker.io	twistlock/private	defender_2_1_89	aqsa-repro2	0	false	▬
docker.io	aqsa/mariadb, library/mariadb	latest	aqsa-repro2	9 32 11	false	▬
docker.io	twistlock/private	console_2_0_67	aqsa-repro2	5 3	false	▬

Policies let you take action on intel from the scan reports. You can create policies that specify how to handle vulnerable images. For example, a policy might block the deployment of a container to the production environment if it has high severity vulnerability.

Finally, organizations need a mechanism to oversee their container environment. The Twistlock dashboard provides several visualization tools, including a timeline view in of Critical, High, Medium, and Low vulnerabilities in Images, Hosts, and Registry, with the option to look further in each individual area.

- The size indicates the number of running containers instantiated from a given image.
- The color indicates the highest severity vulnerability.

Green and orange boxes represent images in relatively good health. Any red boxes have high severity vulnerabilities, and should be immediately addressed.


Vulnerability Explorer
Images
Hosts
Registry
CVE Viewer
Monitor / Vulnerabilities
Collections
Critical 0
High 0
Medium 2
Low 0
None 2
Impacted containers By highest severity
Critical 0
High 1
Medium 0
Low 0
None 0
Impacted hosts By highest severity
Critical 3
High 5
Medium 6
Low 0
None 4
Impacted images By highest severity
Impacted containers over time (30 days)
Impacted hosts over time (30 days)
Impacted images over time (30 days)
0 5 10 15 20 25

You have 121 days remaining in your evaluation period

Compliance

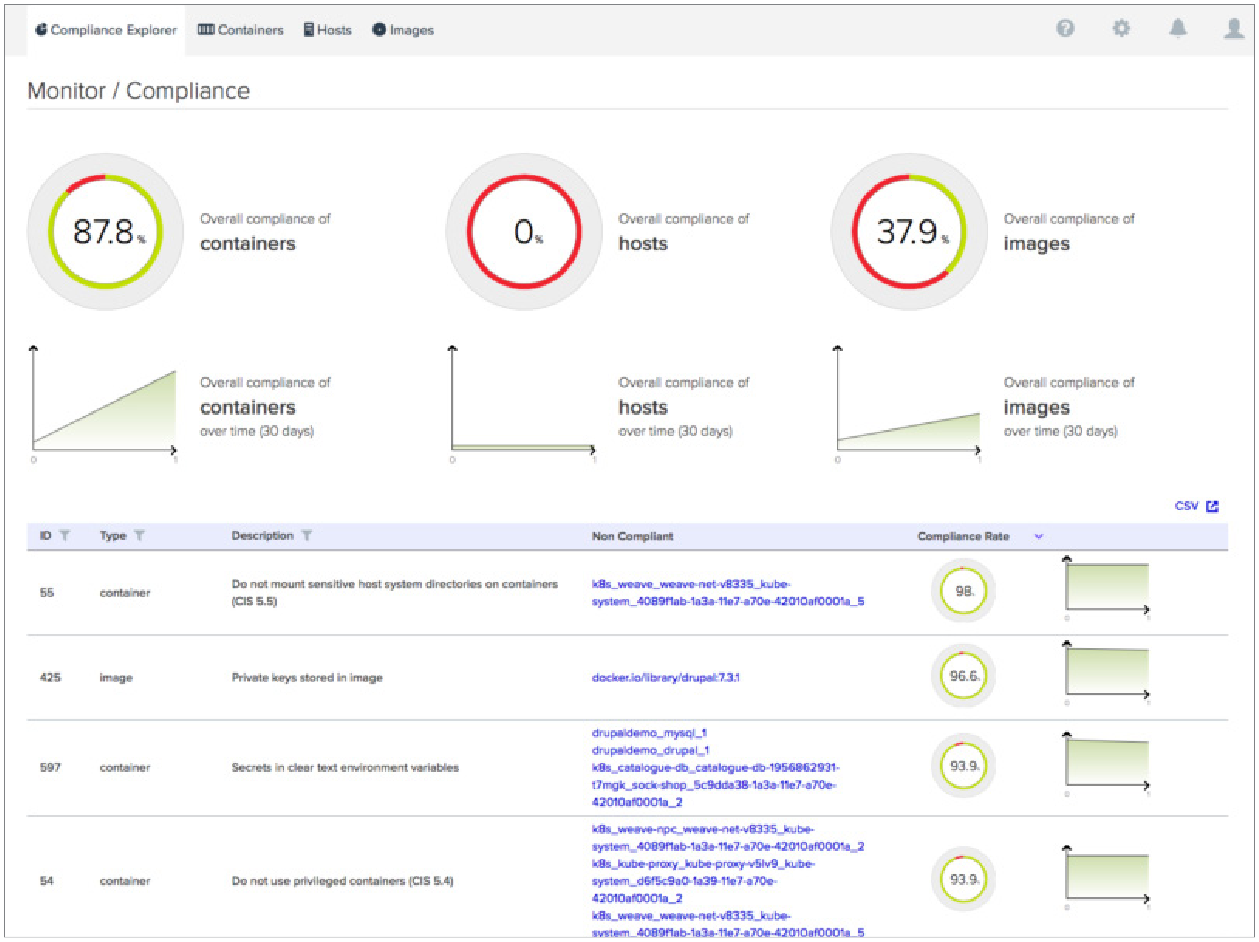
Like many software packages, out-of-the-box Docker has been configured for convenience. Organizations should adopt tools and processes to validate and enforce compliance with secure configuration best practices for container images and runtimes. For best results, these compliance checks should be automated.

The Center for Internet Security (CIS) publishes a document called the Docker Benchmark that defines the security best practices for building Docker images, running containers, and configuring your hosts. Twistlock adapts the Docker Benchmark recommendations into discrete checks (SP 800-190 countermeasure 4.3.3 and 4.4.2).

The discrete checks can be enabled to enforce and maintain compliance to specific requirements from industry and government standards, such as PCI DSS and HIPAA. For example, CIS check 4.1 validates that an image is configured to run as a non-root user. All security standards prescribe least privilege access, and running almost any container as root (which is the default setting) almost certainly yields excessive privileges.

Like vulnerability management, Twistlock lets you set policies around compliance checks. Reports on compliance aren't enough though, so Twistlock lets you protect critical domains by preventing the deployment of non-compliant images.

By default, all compliance checks are set to alert. Most organizations will tailor the list of enabled checks to measure and enforce compliance to a specific industry or government standard. The following screenshot shows Compliance Explorer, which summarizes your environment's compliance to the enabled checks. Compliance officers and auditors can use this dashboard to quickly validate compliance to a predefined checklist.


Compliance Explorer
Containers
Hosts
Images
Monitor / Compliance
87.8%
Overall compliance of containers
0%
Overall compliance of hosts
37.9%
Overall compliance of images
Overall compliance of containers over time (30 days)
Overall compliance of hosts over time (30 days)
Overall compliance of images over time (30 days)
CSV

ID	Type	Description	Non Compliant	Compliance Rate
55	container	Do not mount sensitive host system directories on containers (CIS 5.5)	k8s_weave_weave-net-v8335_kube-system_4089f1ab-1a3a-11e7-a70e-42010af0001a_5	98.
425	image	Private keys stored in image	docker.io/library/drupal:7.3.1	96.6.
597	container	Secrets in clear text environment variables	drupaldemo_mysql_1 drupaldemo_drupal_1 k8s_catalogue-db_catalogue-db-1956862931-t7mgk_sock-shop_5c9dda38-1a3a-11e7-a70e-42010af0001a_2	93.9.
54	container	Do not use privileged containers (CIS 5.4)	k8s_weave-npc_weave-net-v8335_kube-system_4089f1ab-1a3a-11e7-a70e-42010af0001a_2 k8s_kube-proxy_kube-proxy-v5lv9_kube-system_d6f5c9a0-1a39-11e7-a70e-42010af0001a_2 k8s_weave_weave-net-v8335_kube-system_4089f1ab-1a3a-11e7-a70e-42010af0001a_5	93.9.

Besides the out-of-the-box CIS Docker Benchmark checks, Twistlock also supports the Extensible Checklist Configuration and Description Format (XCCDF) so that you can implement your own custom checks.

Runtime Analysis

Organizations should minimize human interaction by automatically profiling containerized apps and building protection profiles for them.

Security tools tend to be reactive, and can end up as relics when new threats emerge. You cannot depend on just tags and identifiers, such as CVEs and malware hashes, to secure your environment. Twistlock runtime defense secures your containers from emerging threats by modeling the intent of each image in your environment and then using those models to detect abnormal activity.

Twistlock builds predictive models for each image in your environment. Models comprise of rules that whitelist specific activities and, conversely, blacklist everything else. Models are built with a combination of static and dynamic (machine learning) analysis.

Rules in the whitelist models span four dimensions:

- **Process control**: Identify a list approved processes that can be started and run in a container. Invalid or unexpected process execution raises an exception.
- **Networking**: Identify ports that should be open and monitor for traffic to malicious endpoints. Creating unexpected network listeners or connecting to known malicious network destinations raises an exception.
- **File system**: Detect changes to the file system. Changes to binaries or protected configuration files, or writes to unexpected locations, raises an exception. Binaries downloaded into containers are assessed against MD5 checksums for known malicious software.
- **System calls**: Profile the container according to the system calls it makes.

After the model is activated, sensors continually monitor running containers on these four dimensions. Policies can be created to specify how anomalies should be handled. For example, if an attacker breaches a container and tries to download malware with curl, the connection would be flagged as an anomaly, and Twistlock can alert or block the container . Exceptions, in the form of additional whitelist or blacklist rules, can be created when the model doesn't fully capture the range of known good activities inside a container.

Nothing is required to activate runtime analysis. Models are automatically generated the first time an image is seen in the environment, and Twistlock automatically enforces the rules in the model. Automatic modeling helps in two ways (SP 800-190 countermeasure 4.4.5):

- It lets you scale your security with apps. No additional manpower is required to create new security rules for new containers.
- It keeps your security rules tight as your app evolves. Twistlock automatically profiles and creates new models for updated images.

Malware

Organizations should adopt tools and processes to monitor images for malware, both at rest and runtime.

Twistlock scans all images in your environment for malware (SP 800-190 countermeasure 4.1.3). Twistlock runtime defense monitors running containers for malware being downloaded into the container file system. Hashes for known malware is delivered to your environment through the Twistlock Intelligence Stream. You can augment data from Twistlock's feed with your own custom data.

Embedded Secrets

Organizations should never store sensitive data in image files.

Twistlock provides a control that detects secrets embedded in images (SP 800-190 countermeasure 4.1.4):. You can then create policies that take action by alerting or actively blocking the deployment of images that contain embedded secrets. This compliance check forces developers to utilize safer methods for injecting secrets, to do things such as connect to a database, for example, into running containers.

Twistlock detects:

- Private keys in the container file system.
- Environment variables that expose sensitive data.

By default, Twistlock raises an alert when it detects secrets in an image. You can refine your compliance policy to block deployment of images that contain secrets, and apply the policy to specific images, containers, or hosts (such as your production environment hosts).

Image trust

Organizations should designate images and registries that have been fully vetted as trusted, then ensure that only these images are allowed to run in your environment. Twistlock lets you define an explicit list of trusted repositories and/or images, and then create rules that only allows only those images to run in your environment. Twistlock's image integrity feature reduces the risk of running sabotaged or unauthorized images in your environment (SP 800-190 countermeasure 4.1.5).

For example, you could create a policy that limits the production environment to running just approved images from your internal (trusted) registry, while blocking anything from external public repositories, such as Docker Hub. You should run Docker 1.10 or later in your environment. Docker 1.10 improves the way image IDs work. Previously image IDs were randomly generated UUIDs. Starting with Docker 1.10, image IDs represent the content inside the image. Now image IDs are a secure hash of the image and its configuration (the image manifest). This new method guarantees the integrity of images that are pulled, pushed, or loaded.

Conclusion

There is good reason that there is tight alignment between the recommendations in SP 800-190 and Twistlock's offering. Since our inception, we've been thinking about the right way to solve the problems related to container security. We're not repurposing old tools to solve new problems. We're building cloud-native tools to address the challenges and opportunities unique to containers.

NIST SP 800-190 contains the current best thinking on securing containers, and Twistlock implements most of the countermeasures described in it. When apps scale out, the old method of manually creating and maintaining security rules becomes impractical. Twistlock is purpose-built for securing container environments at scale. Automation plays a key role. Runtime analysis and machine learning automatically create and enforce policies to secure container workloads across the environment.

ABOUT TWISTLOCK

ENTERPRISE SECURITY. DEVOPS AGILITY.

Twistlock protects today's applications from tomorrow's threats with advanced intelligence and machine learning capabilities. Automated policy creation and enforcement along with native integration to leading CI/CD tools provide security that enables innovation by not slowing development. Robust compliance checks and extensibility allow full control over your environment from developer workstations through to production. As the first end-to-end container security solution, Twistlock is purpose-built to deliver modern security.

LEARN MORE AT

Twistlock.com