Twistlock

Enterprise Security. DevOps Agility.

GUIDE TO **HIPAA** COMPLIANCE FOR CONTAINERS

TABLE OF CONTENTS

EXECUTIVE SUMMARY

With the advent of microservices, it is likely that many of your new apps are built with containers. This guide will help you to achieve compliance with the HIPAA Security Rule for containerized workloads. It shows you how Twistlock's capabilities can be deployed to realize and maintain ongoing compliance with the HIPAA Security Rule. Note that this guide is not a comprehensive treatment of the Security Rule. It will not, by itself, get you fully compliant.

In health care, there is a class of sensitive data known as electronic personal health information, or ePHI. Any covered entity, business associate, or subcontractor that creates, receives, maintains, or transmits ePHI is subject to a piece of legislation known as the HIPAA Security Rule. The Security Rule requires you take the appropriate measures to safeguard ePHI.

ePHI is tightly woven into the fabric of the typical health care app. Apps such as virtual doctor visits, glucose monitors, and billing systems are built around ePHI. It's the ePHI itself — collecting it, storing it, analyzing it — and the exchange of it — between doctor and patient, provider and insurer, covered entity and business associate — that creates value, engages patients, and improves efficiencies.

Safeguarding ePHI in compliance with HIPAA is a challenge because the Security Rule only provides high-level guidance. It doesn't prescribe specific countermeasures to the threats that make ePHI most susceptible to breaches. This guide takes an alternative approach to HIPAA compliance by employing a risk-based framework created by the National Institute of Standards and Technology (NIST), known as the Cybersecurity Framework, to meet the standards and specifications of the Security Rule.

This guide is written for members of both the technical and compliance teams that work with containers.

INTRODUCTION TO THE HIPAA COMPLIANCE BY WAY OF THE NIST CYBERSECURITY FRAMEWORK

The HIPAA Security Rule is flexible, and it can handle any number of approaches to compliance. This guide approaches HIPAA compliance by way of the NIST Cybersecurity Framework.
The challenge with HIPAA is that it doesn't define, at a detailed level, the countermeasures you must put in place to comply with the Security Rule. HIPAA calls for organizations to conduct their own risk analyses to identify the complete set of security controls that can address all reasonably anticipated threats. Risk analysis is an involved process that requires time, budget, and experience. Without a thorough analysis, there could be significant weaknesses in your security posture.

The Cybersecurity Framework, in contrast, is a comprehensive framework of security controls built on a rigorous analysis of the types of threats faced by organizations that use similar information technologies. The Cybersecurity Framework is useful because it's prescriptive. It tells you how to secure sensitive data using existing standards and best practices.

The "glue" between the HIPAA Security Rule and the NIST Cybersecurity is a "Crosswalk" document, published by NIST. It conveniently maps the standards and implementation specifications in the HIPAA Security Rule to subcategories in the NIST Cybersecurity Framework.

For each relevant HIPAA compliance requirement in the Cybersecurity Framework, this guide describes the applicable Twistlock feature or technology, shows how it can be configured to satisfy the HIPPA compliance requirement, and provides links to the official Twistlock documentation for more detailed information.

NOTE: In order to access links to our official documentation, you must have a Twistlock subscription. Subscriptions to the Developer Edition are free. For more information, or to sign up, go to www.twistlock.com.

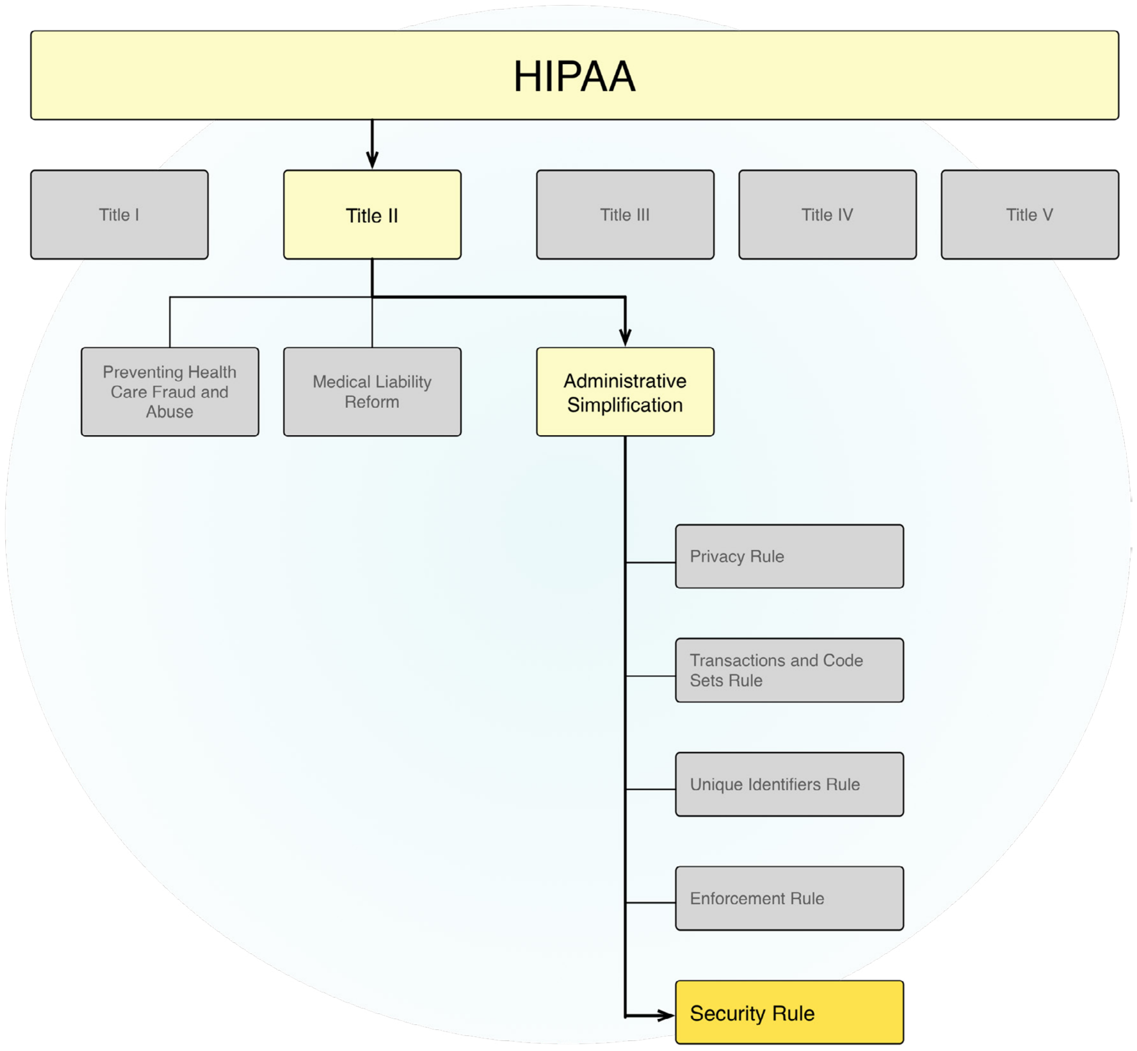
TWISTLOCK

Twistlock is a security suite that is purpose-built for containers. Twistlock lets you centrally declare security policies, and then enforces your policies across your container environment. In general, a container environment includes all your images, containers, hosts, registries, and Docker daemons.
With Twistlock, your data never leaves your jurisdiction. All scanning, analysis, and detection is executed inside your environment only.

HIPAA

The Health Insurance Portability and Accountability Act, or HIPAA, was enacted in 1996. It covers a large number of topics, including health care access when changing or losing a job, efficiency improvements, and fraud. Title II was added in 2003, and it contains the Security Rule. The Security Rule sets standards for safeguarding protected health information stored in digital format. Anyone that accesses, stores, transmits, or processes electronic protected health information (ePHI) must meet the requirements set out in the Security Rule.

The following diagram shows how HIPAA is organized, and where the Security Rule fits into place:



The Security Rule contains three sections: Administrative Safeguards, Physical Safeguards, and Technical Safeguards. The section on Technical Safeguards is most relevant to app developers because it lays out the technology controls that should be put in place to secure ePHI.

Electronic protected health information (ePHI)
Protected health information (PHI) consists of the data collected during the course of providing and paying for health care. Electronic protected health information (ePHI) is defined as individually identifiable health information that is transmitted or stored electronically. It is important to determine if your app deals with ePHI. If it does, you might be subject to HIPAA. If it doesn't, you are probably not subject to HIPAA.

There are two key terms in the definition of ePHI: Health information and individually identifiable. HIPAA defines both terms.

Health information is defined in 1171 of Part C of Subtitle F of Public Law 104-191:

> **HEALTH INFORMATION.**
> The term 'health information' means any information, whether oral or recorded in any form or medium, that...
>
> **A.** is created or received by a health care provider, health plan, public health authority, employer, life insurer, school or university, or health care clearinghouse;
>
> **B.** relates to the past, present, or future physical or mental health or condition of an individual, the provision of health care to an individual, or the past, present, or future payment for the provision of health care to an individual.

Health information is individually identifiable when it is associated with any of the following identifiers. These identifiers are defined in the de-identification standard in the Privacy Rule. For more information, see the Code of Federal Regulations (CFR) Title 45 Section 164.514.

A. Names.

B. Address (all geographic subdivisions smaller than state, including street address, city, county, zip code).

C. All elements of dates (except year) for dates directly related to an individual, including birth date, admission date, discharge date, date of death; and all ages over 89.

D. Telephone numbers.

E. Fax numbers.

F. Electronic mail addresses.

G. Social security numbers.

H. Medical record numbers;

I. Health plan beneficiary numbers.

J. Account numbers.

K. Certificate/license numbers.

L. Vehicle identifiers and serial numbers, including license plate numbers.

M. Device identifiers and serial numbers.

N. Web Universal Resource Locators (URLs).

O. Internet Protocol (IP) address numbers.

P. Biometric identifiers, including finger and voice prints.

Q. Full face photographic images and any comparable images.

R. Any other characteristic that could uniquely identify the individual.

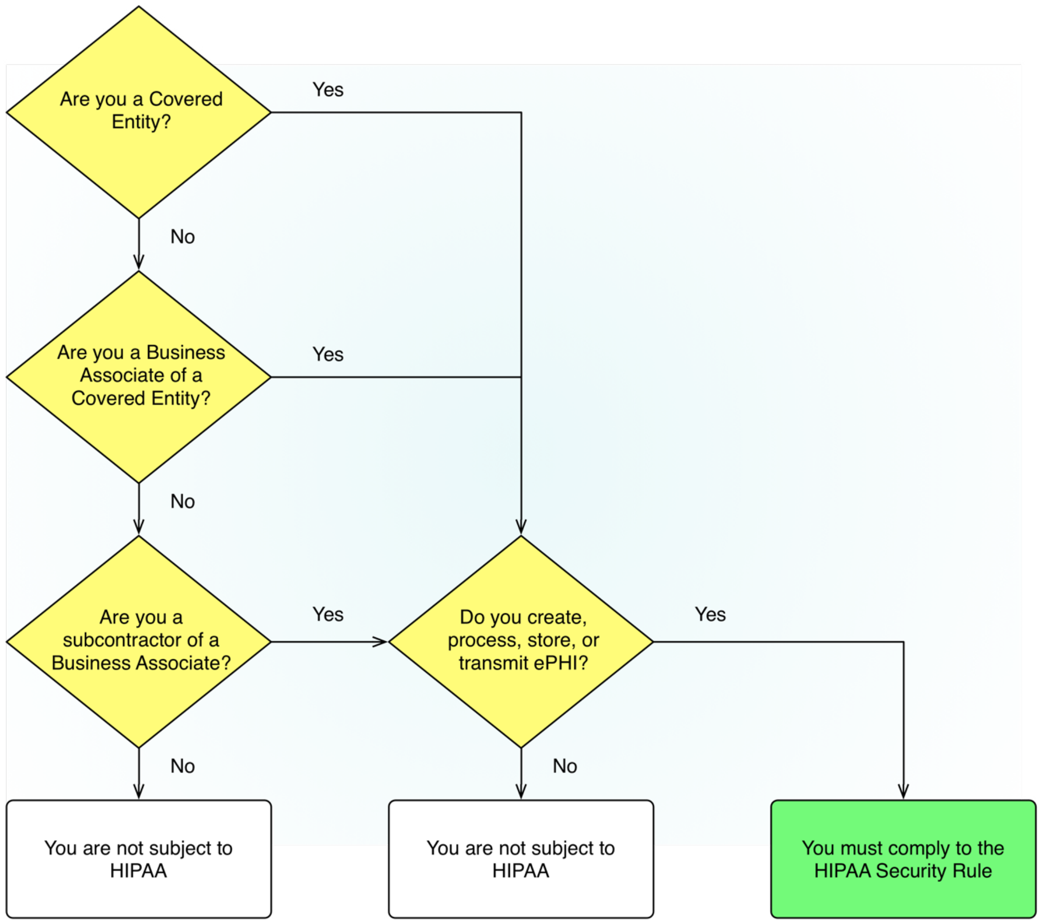
In ecommerce apps, it's pretty clear what is, and what is not, credit card data. The Payment Card Industry (PCI) Data Security Standard (DSS) defines it in no uncertain terms. In health care, it's not always so clear what is considered ePHI, and some cases require careful analysis. Take something as innocuous as email appointment reminders. Is that considered ePHI? It is created by the health care provider, it relates to the provisioning of health care, and it contains personally identifiable information, such as an email address. It most certainly is ePHI!

Who must comply with HIPAA?
Just because your app handles health information, doesn't mean you are required to comply with HIPAA. The HIPAA Security Rule only applies to covered entities, their business associates, and subcontractors that deal with ePHI.

Covered entities are health care providers, health plans, and health care clearing houses. Health providers furnish medical services in exchange for payment. Examples include doctors, dentists, and hospitals. Health plans pay for the cost of medical care. They include insurance companies, Medicare, and Medicaid. Health care clearing houses transform health information into standardized formats so that it can be relayed between covered entities.

Business associates are organizations contracted by covered entities to provide services. They provide services such as storage, billing, and data analysis. Subcontractors provide services to business associates. If a business associate or subcontractor deals with ePHI, they are also required to comply to HIPAA.

The following flow chart shows how to determine if you are required to comply to HIPAA:


Are you a Covered Entity?
Yes
No
Are you a Business Associate of a Covered Entity?
Yes
No
Are you a subcontractor of a Business Associate?
Yes
No
Do you create, process, store, or transmit ePHI?
Yes
No
You are not subject to HIPAA
You are not subject to HIPAA
You must comply to the HIPAA Security Rule

For example, if you develop a fitness app that tracks weight loss, but you do not share the data with a Covered Entity, then you do not need to comply with HIPAA.

NIST CYBERSECURITY FRAMEWORK

The Cybersecurity Framework is a set of industry standards and best practices designed to help organizations reduce and better manage cybersecurity risks. It was created by the National Institute of Standards and Technology (NIST) as the result of an Executive Order from the President of the United States. The Executive Order established a policy to "enhance the security and resilience of the Nation's critical infrastructure and to maintain a cyber environment that encourages efficiency, innovation, and economic prosperity while promoting safety, security, business confidentiality, privacy, and civil liberties."

The Framework Core defines a set of outcomes, where an outcome is a desired state that can be achieved when the appropriate security controls are applied. For each outcome, the Framework Core provides a list of informative references that point to specific sections in standards, such as ISO/IEC 27001:2013 and NIST SP 800-53 Rev 4. These sections illustrate the methods that can be employed to achieve an outcome.

The Cybersecurity Framework is technology neutral. It addresses threats common across all critical infrastructure sectors, and it is voluntary. Voluntary means that there are no regulatory requirements to comply with it.

Configuring Twistlock to comply with the HIPAA Security Rule
This section shows you how to configure Twistlock and install security policies to bring your container environment into compliance with the HIPAA Security Rule. The container environment consists of any host, Docker daemon, image, or container that handles ePHI.

Compliance to the HIPAA Security Rule is addressed with the NIST Cybersecurity Framework. NIST publishes a crosswalk document that maps the Cybersecurity Framework requirements to HIPAA Security Rule requirements. Compliance to standards and implementation specifications in the HIPAA Security Rule are achieved by satisfying subcategories in the NIST Cybersecurity Framework.

The following table summarizes the requirements from the Cybersecurity Framework that can be addressed by Twistlock. Each row lists a Cybersecurity Framework (CSF) requirement and a corresponding HIPAA Security Rule standard (or implementation specification) that can be satisfied by a Twistlock feature/function.

NIST CSF Function	NIST CSF Category	NIST CSF Subcategory	HIPAA Security Rule
Identify	Risk Assessment	ID.RA-2: Threat and vulnerability information is received from information sharing forums and sources.	No direct mapping*
Protect	Access Control	PR.AC-3: Remote access is managed.	164.312(e)(1) 164.312(e)(2)(ii)
Protect	Access Control	PR.AC-4: Access permissions are managed, incorporating the principles of least privilege and separation of duties.	164.312(a)(1)
Protect	Access Control	PR.AC-5: Network integrity is protected, incorporating network segregation where appropriate.	164.312(e)
Protect	Data Security	PR.DS-2: Data in transit is protected.	164.312(e)(1)
Protect	Data Security	PR.DS-5: Protections against data leaks are implemented.	164.312(a)(1) 164.312(e)(1)
Protect	Data Security	PR.DS-6: Integrity checking mechanisms are used to verify software, firmware, and information integrity.	164.312(c)(1) 164.312(c)(2) 164.312(e)(2)(i)
Protect	Data Security	PR.DS-7: The development and testing environment(s) are separate from the production environment.	164.308(a)(4) 164.502
Protect	Info Protection	PR.IP-1: A baseline configuration of information technology/ industrial control systems is created and maintained.	164.308(a)(7)(ii)(A)
Protect	Info Protection	PR.IP-2: A System Development Life Cycle to manage systems is implemented.	164.308(a)(1)(i)
Protect	Info Protection	PR.IP-4: Backups of information are conducted, maintained, and tested periodically.	164.308(a)(7)(ii)(A) 164.308(a)(7)(ii)(B)
Detect	Anomalies and Events	DE.AE-3: Event data are aggregated and correlated from multiple sources and sensors.	164.312(b)
Detect	Security Continuous Monitoring	DE.CM-1: The network is monitored to detect potential cybersecurity events.	164.312(b)
Detect	Security Continuous Monitoring	DE.CM-4: Malicious code is detected.	164.308(a)(5)(ii)(B)
Detect	Security Continuous Monitoring	DE.CM-8: Vulnerability scans are performed.	164.308(a)(1)(i)

*Your risk analysis might conclude that participating in information sharing forums is a reasonable and appropriate measure for reducing your security risk.

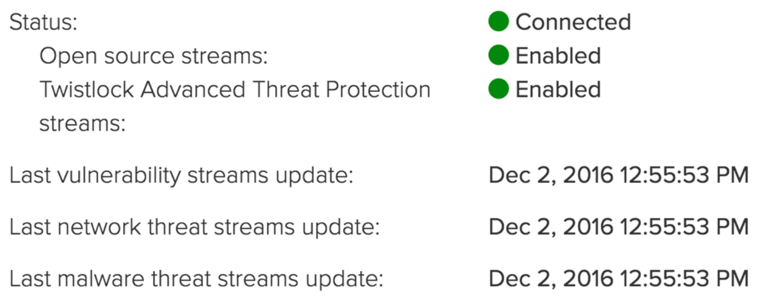
ID.RA-2

Threat and vulnerability information is received from information sharing forums and sources.

DISCUSSION
Twistlock threat data is procured from open source lists, commercial vendors, and Twistlock Labs. Threat data is delivered to Console via a real-time feed known as the Twistlock Intelligence Stream. When Twistlock updates the feed, your installation is automatically updated with the latest threat data.

Console shows the status of the Twistlock Intelligence Stream connection, along with a timestamp for the latest update.

To see the status page, open **Console**, and then go to **Settings > SYSTEM > Intelligence**. The Settings menu is the gear icon in the top right side of the user interface.



Status:	● Connected
Open source streams:	● Enabled
Twistlock Advanced Threat Protection streams:	● Enabled
Last vulnerability streams update:	Dec 2, 2016 12:55:53 PM
Last network threat streams update:	Dec 2, 2016 12:55:53 PM
Last malware threat streams update:	Dec 2, 2016 12:55:53 PM

TWISTLOCK CONFIGURATION GUIDE
No action is required. By default, Twistlock keeps threat data up to date with the latest threat data.

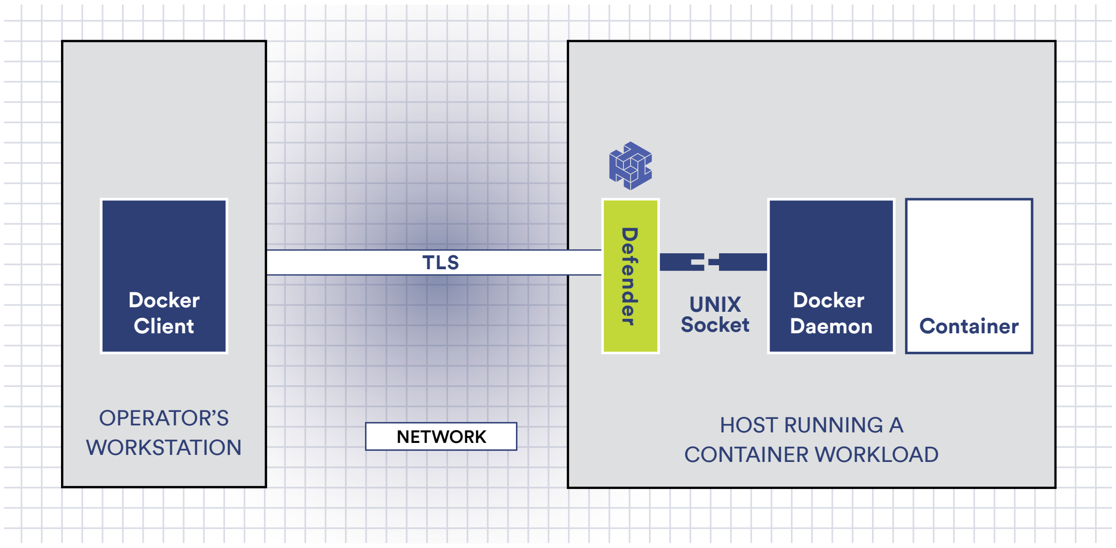
PR-AC-3

Remote access is managed.

DISCUSSION

All access to your container environment is brokered through Defender. Defender acts as a proxy to the Docker daemon. Defender should be deployed to every host that runs containers, so that it can control access to your containers according to the policies you have configured in Console.

The following diagram shows how Docker commands are routed from an operator's workstation over the network to a host protected by Defender:



The Docker client securely transmits the command over the network to Defender using the Transport Layer Security (TLS) protocol. If the installed policy permits the command to be executed, it is forwarded to the Docker daemon over the UNIX socket. The UNIX socket is created when the Docker daemon first starts, and it exposes a REST API through which Docker commands can be run.

Admin access to hosts running your containerized workload should be restricted so that no one can log directly into a host and bypass Defender by running Docker commands locally.

TWISTLOCK CONFIGURATION GUIDE

Install Defender on every host that runs containers.

On the client side, anyone accessing the container environment must log into Console to download their certificates. Certificates are bound to the user's identity, and are used to enforce your access control policies.

MORE INFORMATION

[Install Defender]

[Configure Docker client variables]

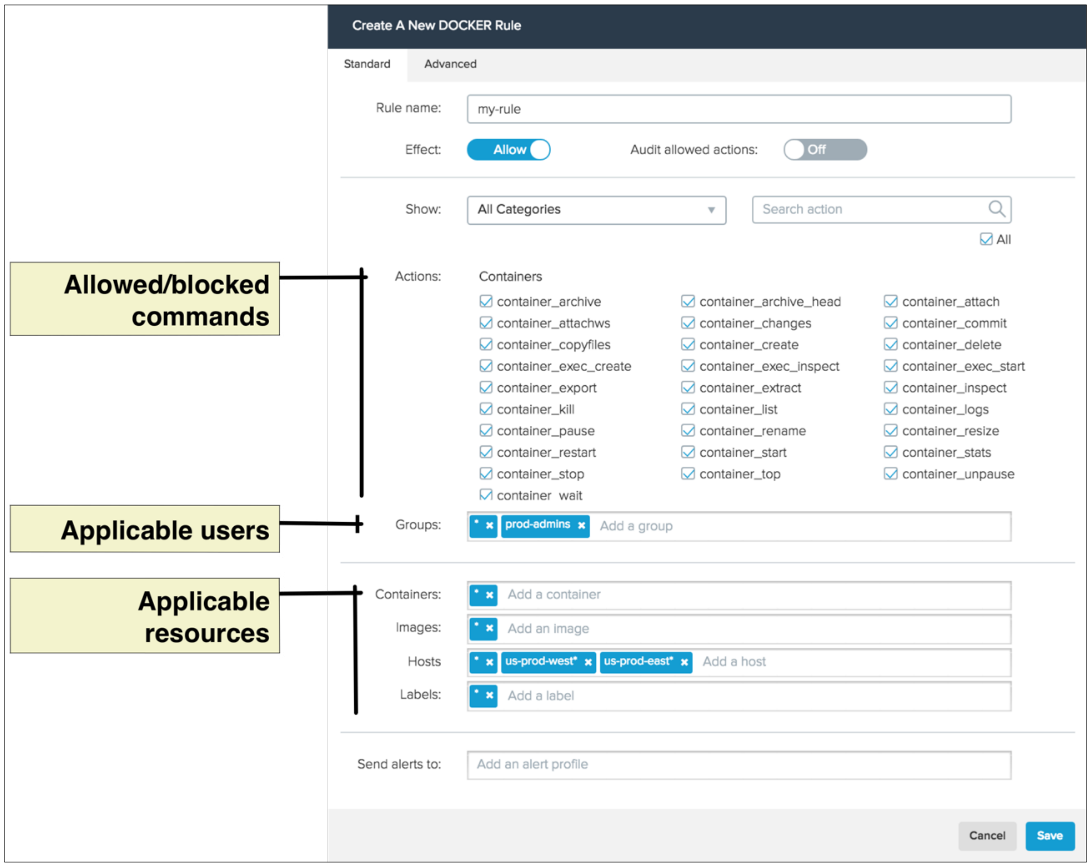
PR.AC-4

Access permissions are managed, incorporating the principles of least privilege and separation of duties.

DISCUSSION
Twistlock lets you limit access to the images and containers in your environment with access control rules. Each access control rule lets you target:

- Specific users (using groups).
- Specific commands: Docker Engine, Docker Swarm, and Kubernetes commands.
- Specific resources: Containers, images, and hosts.

The following screenshot shows the dialog for a creating new access control rule:



Rule name: my-rule
Effect: Allow
Audit allowed actions: Off
Show: All Categories
Search action
☑ All

Actions: Containers

☑ container_archive	☑ container_archive_head	☑ container_attach
☑ container_attachws	☑ container_changes	☑ container_commit
☑ container_copyfiles	☑ container_create	☑ container_delete
☑ container_exec_create	☑ container_exec_inspect	☑ container_exec_start
☑ container_export	☑ container_extract	☑ container_inspect
☑ container_kill	☑ container_list	☑ container_logs
☑ container_pause	☑ container_rename	☑ container_resize
☑ container_restart	☑ container_start	☑ container_stats
☑ container_stop	☑ container_top	☑ container_unpause
☑ container_wait		

Groups: * prod-admins Add a group
Containers: * Add a container
Images: * Add an image
Hosts: * us-prod-west* us-prod-east* Add a host
Labels: * Add a label
Send alerts to: Add an alert profile

WHERE:

Allowed/blocked commands
Selects the commands that are allowed to run, or that should be blocked.

Applicable users
Specifies the groups for which this rule applies. Groups provide an easy way to manage privileges for large numbers of users. In this example, a group called prod-admins contains all users that require privileges to administer the pro duction environment.

Applicable resources
Specifies where the commands in this rule can be run.

TWISTLOCK CONFIGURATION GUIDE

Before you can start creating access control policies, every user in the system must have a unique ID.

1 | Integrate Twistlock with your organization's directory service so that you can re-use the identities and groups already set up in your directory service. Twistlock supports Active Directory, OpenLDAP, and SAML.

If you do not have a directory service, Twistlock provides a mechanism to create local users and groups.

2 | Create the rules required to realize your access control policy. By default, Twistlock blocks all actions by all users on all resources. You must build out your policies by creating rules that explicitly whitelist approved actions.

MORE INFORMATION

Access control screencast

Integration with directory services:

- Active Directory
- OpenLDAP
- SAML

Access control rules:

- Docker Engine
- Docker Swarm
- Kubernetes

PR-AC-5

Network integrity is protected, incorporating network segregation where appropriate.

DISCUSSION

Every container should have its own network namespace. Network namespaces enforce isolation between containers, and between containers and your host. Containers can be configured to share the host's network stack, this setup is insecure because the container would have full access to the host's network interfaces.

TWISTLOCK CONFIGURATION GUIDE

Create a compliance rule that raises an alert when a container is started with the **--net=host** option.

You can either create this compliance rule manually, or you can import this rule, along with all other recommended rules from this guide, using the Twistlock API.

- To install this rule manually, see Managing compliance. Select compliance check ID 59 (CIS 5.9).
- To install this rule with the Twistlock API, see Sample policy file.

MORE INFORMATION

Managing compliance policies with the API

PR.DS-2

Data-in-transit is protected.

DISCUSSION
Network traffic between containers on a host is unrestricted by default. A breached container could intercept packets on the default container network and steal sensitive information. For containers that must communicate with each other, you should explicitly link them together. All other communication between containers should be blocked.

TWISTLOCK CONFIGURATION GUIDE
Create a compliance rule that checks that the Docker daemon on your host is started with the **`--icc=false`** option. This option disables inter-container communication unless explicitly specified.

When you want two containers to be able to communicate, start the container **`(docker run)`** with the **`--link=`** option, Docker inserts a pair of iptables ACCEPT rules so that the new container can connect to the ports exposed by the other container.

You can either create this compliance rule manually, or you can import this rule, along with all other recommended rules from this guide, using the Twistlock API.

- To install this rule manually, see Managing compliance. Select compliance check ID 21 (CIS 2.1).
- To install this rule with the Twistlock API, see Sample policy file.

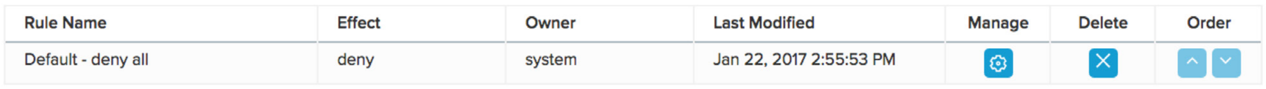
MORE INFORMATION
Managing compliance policies with the API

PR.DS-5

Protections against data leaks are implemented.

DISCUSSION
Twistlock ships with a default deny-all rule that blocks all actions by all users (including the default admin user) on all resources. You must create explicit rules that grant specific users access to specific resources.

To see the default deny-all rule, log into Console, then go to **Defend > Access > Docker.** Click on the **Manage** button to see how the rule is configured.



Rule Name	Effect	Owner	Last Modified	Manage	Delete	Order
Default - deny all	deny	system	Jan 22, 2017 2:55:53 PM	⚙	✕	˄ ˅

TWISTLOCK CONFIGURATION GUIDE
No action required. Twistlock ships with a deny-all rule installed by default.

PR.DS-6

Integrity checking mechanisms are used to verify software, firmware, and information integrity.

DISCUSSION
Twistlock lets you define an explicit list of trusted repositories and images, and then create rules that only allows those images to run in your environment. 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 also install 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.

Twistlock's image integrity feature works together with Docker 1.10 to reduce the risk of running sabotaged or unauthorized images in your environment.

TWISTLOCK CONFIGURATION GUIDE
Specify a list of trusted repositories and trusted images. Then create compliance rules that regulate how untrusted images are handled in your environment.

MORE INFORMATION
Trusted images

PR.DS-7

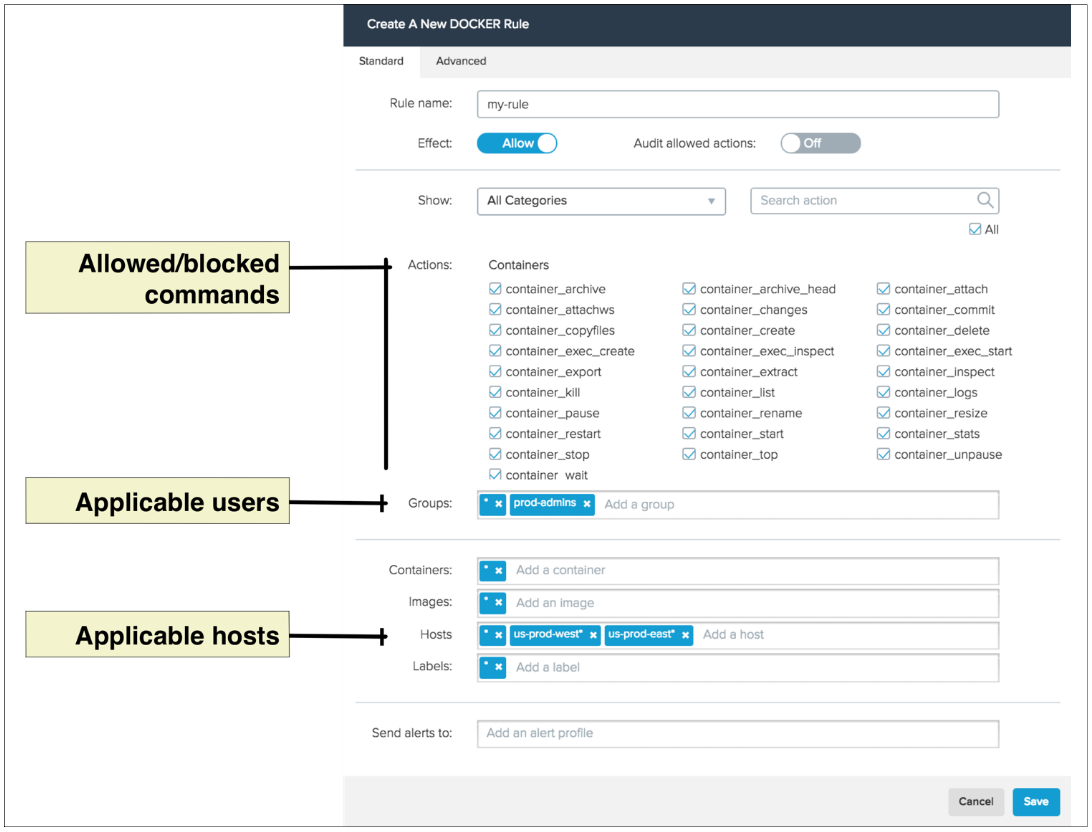
The development and testing environment(s) are separate from the production environment.

DISCUSSION

You can use Twistlock access control rules to enforce the separation of development and testing environments from the production environment.

Twistlock access control rules specify what users can take what actions against what resources. If you have established a standardized naming scheme for the hosts in your environment, you can separate and target access control rules for test, development, and production environments.

The following screenshot shows the dialog for creating an access control rule for Docker Engine commands. A group called prod-admins contains all users that require privileges to administer the production environment. This rule specifies that prod-admins can run all Docker commands on any container or image on hosts named us-prod-west* and us-prod-east*.



Rule name: my-rule

Effect: Allow

Audit allowed actions: Off

Show: All Categories

Search action

✓ All

Actions: Containers

✓ container_archive	✓ container_archive_head	✓ container_attach
✓ container_attachws	✓ container_changes	✓ container_commit
✓ container_copyfiles	✓ container_create	✓ container_delete
✓ container_exec_create	✓ container_exec_inspect	✓ container_exec_start
✓ container_export	✓ container_extract	✓ container_inspect
✓ container_kill	✓ container_list	✓ container_logs
✓ container_pause	✓ container_rename	✓ container_resize
✓ container_restart	✓ container_start	✓ container_stats
✓ container_stop	✓ container_top	✓ container_unpause
✓ container_wait		

Groups: * prod-admins Add a group

Containers: * Add a container

Images: * Add an image

Hosts: * us-prod-west* us-prod-east* Add a host

Labels: * Add a label

Send alerts to: Add an alert profile

For example, a developer named Bruce is a member of the engineering group. You could create a rule that grants full access to all commands to users in the engineering group on all hosts in the development environment. When coupled with the default deny-all rule, Bruce would be able to do his work in the development environment, but he would be blocked from running any Docker command in any other environment, including the production environment.

TWISTLOCK CONFIGURATION GUIDE

Before you can start creating access policies, every user in the system must have a unique ID.

1 | Integrate Twistlock with your organization's directory service so that you can re-use the identities and groups already set up in your directory service. Twistlock supports Active Directory, OpenLDAP, and SAML.

If you do not have a directory service, Twistlock provides a mechanism to create local users and groups.

2 | Create the rules required to realize your access control policy. By default, Twistlock blocks all actions by all users on all resources. You must build out your policies by creating rules that explicitly whitelist approved actions.

MORE INFORMATION

Access control screencast

Integration with directory services:

- Active Directory
- OpenLDAP
- SAML

Access control rules:

- Docker Engine
- Docker Swarm
- Kubernetes

PR.IP-1

A baseline configuration of information technology/industrial control systems is created and maintained.

DISCUSSION
The default installation of Twistlock is reasonably secure. However, the overall posture is weakened by a handful default settings designed to optimize the out-of-the-box experience. Appendix A documents how to fortify the security posture of a default Twistlock installation. The contents of Appendix A can be added to your documentation library for operational procedures, or adapted for configuration management tools such as Chef or Puppet.

Besides fortifying your default installation, you can use the Twistlock API to install and deploy policies. The API gives you a way to formalize your policies in a file so that they can be reused, audited, and version controlled. Formalizing your policies this way ensures consistent configuration across your various environments.

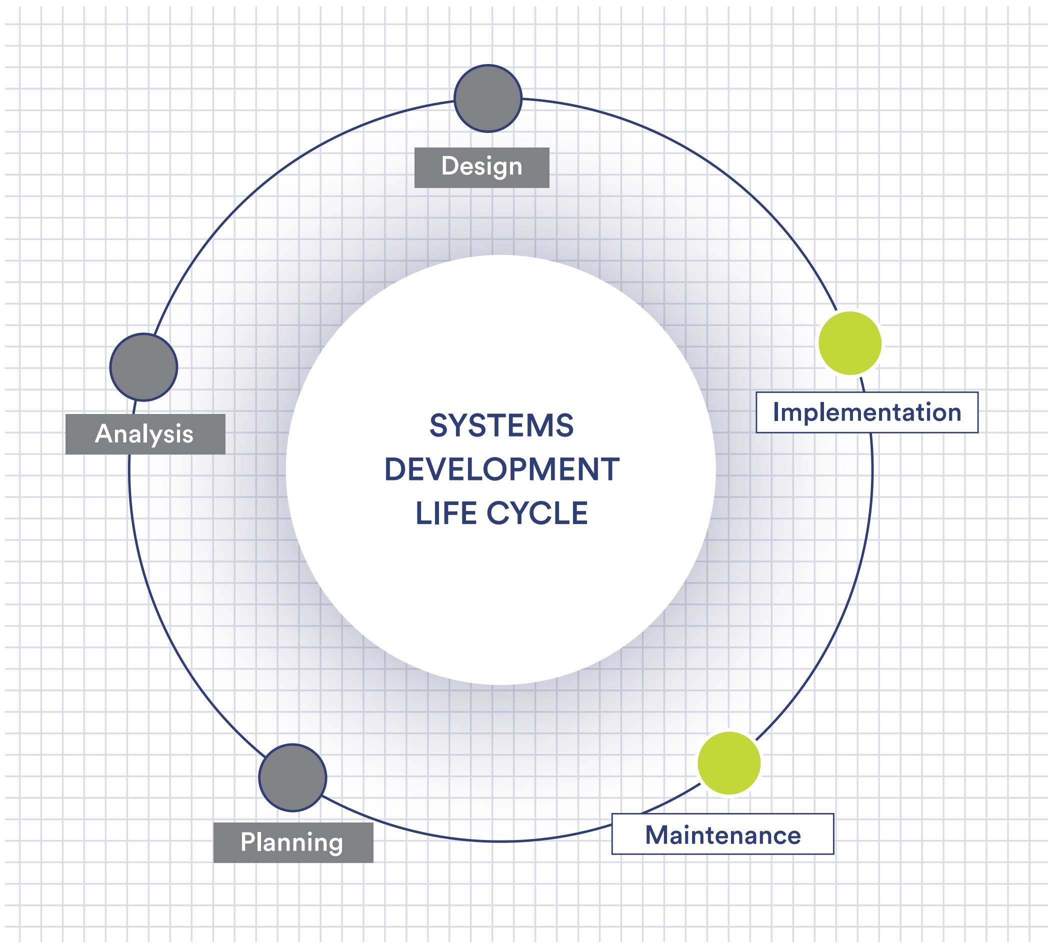
MORE INFORMATION
Appendix A

PR.IP-2

A System Development Life Cycle to manage systems is implemented.

DISCUSSION
Twistlock lets you integrate vulnerability scanning into key phases of the container development lifecycle.

The following diagram shows the container development life cycle, and the places where Twistlock can play a role.



Implementation

Twistlock provides Jenkins and TeamCity plugins that let you incorporate vulnerability scanning into your continuous integration (CI) pipeline so that developers can identify and fix critical security defects before images ever get to the registry. TwistScan, a stand-alone program, lets you build vulnerability scanning into custom tooling.

Maintenance

Twistlock continually scans your environment for containers with critical vulnerabilities. Vulnerabilities discovered in in live production images can be addressed in subsequent development cycles in the planning and analysis phases of the development life cycle.

TWISTLOCK CONFIGURATION GUIDE

Install Defender on any host that runs containers.

Install the Jenkins or TeamCity plugin. If you do not use these CI tools, you can integrate TwistScan into your continuous integration processes instead.

MORE INFORMATION

Jenkins plugin
TeamCity plugin
TwistScan

PR.IP-4

Backups of information are conducted, maintained, and tested periodically.

DISCUSSION

Twistlock automatically backs up all data and configuration files. By default, data files are archived every 24 hours, and the backup file is copied to a location you specify.

TWISTLOCK CONFIGURATION GUIDE

By default, Twistlock saves its backup files to ***/var/lib/twistlock-backup*** on the same host where Console runs. You should configure Twistlock to save your backup files to a durable location outside your production environment.

To specify a backup volume, edit ***twistlock.cfg*** before installing Twistlock:

1 | Open ***twistlock.cfg*** for editing.

2 | Specify the directory where the backup volume is mounted:

```
DATA_RECOVERY_VOLUME=<PATH/TO/BACKUP/VOLUME>
```

3 | Install Twistlock using the normal procedure.

NOTE: If you have already installed Twistlock, and you want to specify a different backup volume, rerun the installer with an updated configuration file.

MORE INFORMATION

Disaster recovery
Install Twistlock
Reconfigure Twistlock

DE.AE-3

Event data are aggregated and correlated from multiple sources and sensors.

DISCUSSION
Audit events from all Defenders deployed across your environment are aggregated in Console. These include access events, scan events, alerts from the runtime protection sensors, and compliance issues.

Twistlock can be configured to send all audit events to syslog in RFC5424-compliant format. If you already have a centralized syslog collector, you can quickly integrate Twistlock into your existing infrastructure.

TWISTLOCK CONFIGURATION GUIDE
Configure Twistlock to send audit events to syslog.

MORE INFORMATION
Syslog integration

DE.CM-1

The network is monitored to detect potential cybersecurity events.

DISCUSSION

Twistlock monitors all containers for connections to malicious destinations. A list of banned IP addresses is delivered to your installation through the Twistlock Intelligence Stream, and it can be augmented with your own custom data.

Twistlock runtime protection offers a second, more sophisticated layer of security. Runtime protection builds predictive models for every image in your environment, and then uses the models, which are essentially "allow lists", to detect suspicious activity. The network sensor can then spot malicious software that tries to create sockets, listen on unexpected ports, and so on.

Models are built from both static and dynamic analysis of the underlying images. For example, static analysis would whitelist ports exposed with the **`EXPOSE`** instruction from an image's Dockerfile. Dynamic analysis would whitelist a port opened by a legitimate app using a command line parameter. After the learning algorithm completes the model, any attempt to open a port not described in the model would raise an alert.

TWISTLOCK CONFIGURATION GUIDE

No action required. Defender continuously monitors all containers for connections to malicious endpoints. Twistlock runtime protection automatically creates models and raises alerts when containers do something that deviates from their models.

MORE INFORMATION

How Twistlock Runtime Defense works
Runtime defense screencast
App-specific network intelligence
Import banned IP addresses

DE.CM-4

Malicious code is detected.

DISCUSSION

Twistlock runtime defense monitors running containers for malware being downloaded into the container file system. Hashes for known malware is delivered to your installation through the Twistlock Intelligence Stream. You can augment data from Twistlock's feed with your own custom data.

TWISTLOCK CONFIGURATION GUIDE

Install Defender on every host that runs containers. Twistlock automatically monitors containers for malware.

MORE INFORMATION

Malware scanning
Import custom malware data

DE.CM-8

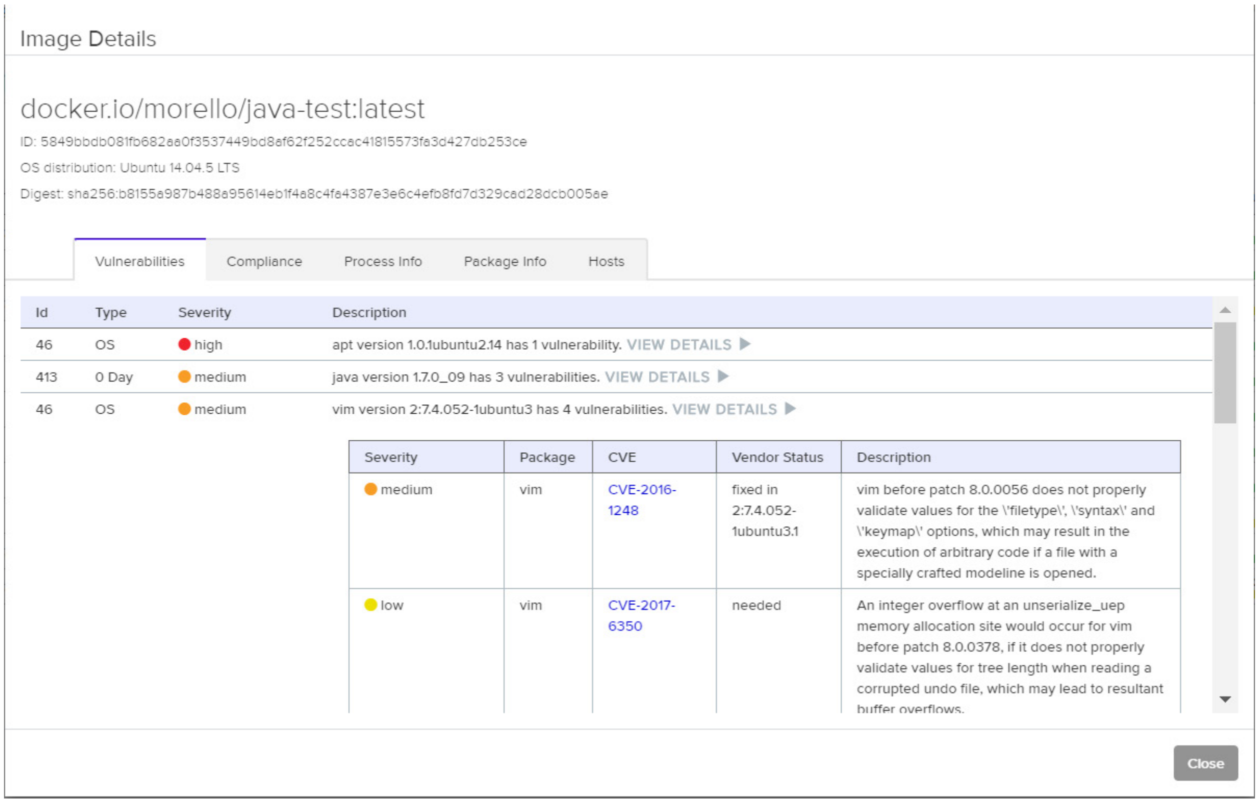
Vulnerability scans are performed.

DISCUSSION

Twistlock periodically scans all images and containers in your environment for CVEs and zero-day vulnerabilities. The default scan interval is 24 hours, and the size of this interval is configurable. Scans are also immediately triggered when images or containers first appear on a host (for example, after you run docker pull). Alerts are raised for vulnerabilities with a CVSS score of 4.0 or higher, and this threshold is configurable.

Twistlock can also be set up to scan your registries. You can use webhooks to notify Defender when new images are pushed to the registry so that they can be scanned immediately.

Scan reports are published in Console. Every reported vulnerability comes with a link to the information source. The following screenshot shows a scan report:



docker.io/morello/java-test:latest

ID: 5849bbdb081fb682aa0f3537449bd8af62f252ccac41815573fa3d427db253ce

OS distribution: Ubuntu 14.04.5 LTS

Digest: sha256:b8155a987b488a95614eb1f4a8c4fa4387e3e6c4efb8fd7d329cad28dcb005ae

Id	Type	Severity	Description
46	OS	● high	apt version 1.0.1ubuntu2.14 has 1 vulnerability. VIEW DETAILS ▶
413	0 Day	● medium	java version 1.7.0_09 has 3 vulnerabilities. VIEW DETAILS ▶
46	OS	● medium	vim version 2:7.4.052-1ubuntu3 has 4 vulnerabilities. VIEW DETAILS ▶

Severity	Package	CVE	Vendor Status	Description
● medium	vim	CVE-2016-1248	fixed in 2:7.4.052-1ubuntu3.1	vim before patch 8.0.0056 does not properly validate values for the \'filetype\', \'syntax\' and \'keymap\' options, which may result in the execution of arbitrary code if a file with a specially crafted modeline is opened.
● low	vim	CVE-2017-6350	needed	An integer overflow at an unserialize_uep memory allocation site would occur for vim before patch 8.0.0378, if it does not properly validate values for tree length when reading a corrupted undo file, which may lead to resultant buffer overflows.

Vulnerabilities in the scan report are assigned a severity of high, medium, or low. Severity is based on the Common Vulnerability Scoring System (CVSS) score assigned to the CVE. When possible, Twistlock uses the vendor's score in order to capture the severity accurately.

High — The vulnerability has a CVSS base score that ranges from 8.0 to 10.0.

Medium — The vulnerability has a CVSS base score that ranges from 4.0-7.9.

Low — The vulnerability has a CVSS base score that ranges from 0.0-3.9.

TWISTLOCK CONFIGURATION GUIDE

Install Defender on any host that runs containers. Defender automatically scans all images and containers on the host for vulnerabilities.

MORE INFORMATION

View image scan reports
View container scan reports
Configure scan intervals
Configure registry scans

SAMPLE POLICY FILE

This guide comes with a sample policy file that can be used to deploy the recommended compliance checks using the Twistlock API.

The Twistlock API provides complete control over the product, and includes endpoints to install and deploy policies. The API gives you a way to formalize your policies in a file so that they can be reused, audited, and version controlled. Formalizing your policies this way ensures a consistent configuration across your various environments.

The sample policy file, policy_hipaa.txt, installs a rule that blocks any container that violates compliance check 21, 41, 59, or 599. The compliance checks recommended in this guide are summarized in the following table:

NIST CSF subcategory	Twistlock compliance check ID	Action
PR.AC-5	59 (CIS 5.9)	Block
PR.DS-2	21 (CIS 2.1)	Block

Alerting and Blocking
Twistlock ships with a default policy that raises an alert when any of the compliance checks fail. Alerting provides a benign way to validate and tune your policies on production systems. It lets you analyze your logs and reduce the number of false positives that could disrupt your service. After you are confident that your policies have the intended effect, you can flip the action from alert to block.

The block action in the sample policy file is for illustrative purposes only. It is used to differentiate the checks in the sample file (action=block) from the compliance rules in Twistlock's default rule (action=-block). Apply blocking only after you have validated your policies with alerts.

Deploying the policies in policies_hipaa.txt
This section shows you how to upload *policy_hipaa.txt* using the Twistlock API.

For each of the following steps, replace **`<CONSOLE>`** with the hostname or IP address for Console.

1 | Get your auth token.

The following command specifies the default admin username and password. Specify the appropriate admin user for your environment.

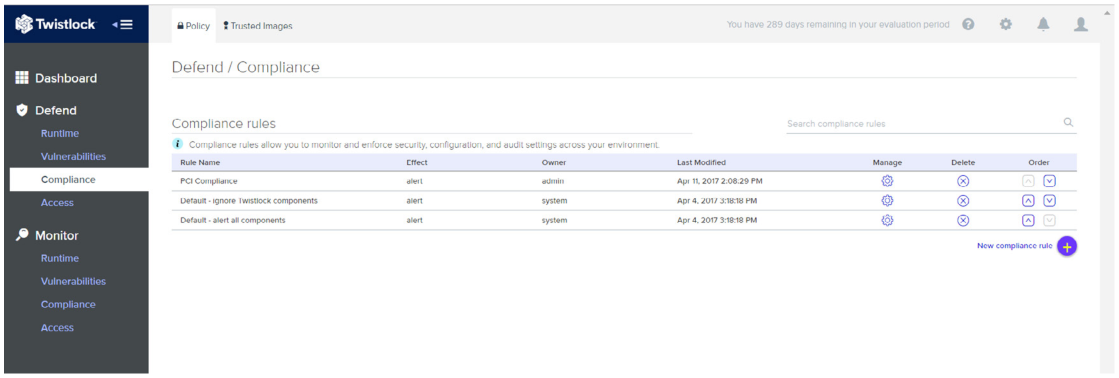
```
$ curl -k -H "Content-Type: application/json" \
-d '{"username":"admin", "password":"admin"}' \
https://<CONSOLE>:8083/api/v1/authenticate
{"token":"eyJ0eXAiOiJKV1QiLC..."}
```

2 | Update the installed policies by pushing the new JSON payload in policy_hipaa.txt.

```
$ curl -k -H "Authorization: Bearer eyJ0eXAiOiJKV1QiLC ..." \
-H "Content-Type:application/json" \
-X PUT \
https://<CONSOLE>:8083/api/v1/policies/compliance \
--data-binary "@policy_hipaa.txt"
policy updated
```

3 | Log into Console, and go to **Defend > Compliance > Compliance.**

4 | Validate your new rule has been installed.



Defend / Compliance

Compliance rules

Compliance rules allow you to monitor and enforce security, configuration, and audit settings across your environment.

Rule Name	Effect	Owner	Last Modified	Manage	Delete	Order
PCI Compliance	alert	admin	Apr 11, 2017 2:08:29 PM			
Default - ignore Twistlock components	alert	system	Apr 4, 2017 3:18:18 PM			
Default - alert all components	alert	system	Apr 4, 2017 3:18:18 PM			

Notes about the JSON in *policy_hipaa.txt*:

- Rules are processed in priority order. The first rule in the **`statement`** array has the highest priority. Therefore, any container that violates compliance check 59 is blocked, and the alert action specified in the default rule is overridden. The endpoint updates all compliance rules in a single shot to retain a strict order between rules.
- The name of the rule, as it appears in Console, is **HIPAA compliance** rules is specified with the **name** key.
- The value for last_modified is displayed in Console. It is a user defined value that can be used to track the version of the installed rule. It does not designate the time that the policy file was uploaded to Console.
- To create your own policy file from scratch, get the default policy from the ***/api/v1/policy/compliance*** endpoint, edit the JSON, then push the updated JSON back to the same end point. For more information, see the Twistlock API.

Appendix A

This section shows you how to fortify the security posture of a default Twistlock installation.

Remove the default admin account
The default Twistlock installation comes with a default user with administrator-level privileges (admin/admin). It should be removed before Twistlock is installed.

Remove the default admin account before installing Twistlock on the network.

TWISTLOCK CONFIGURATION
To ensure that Twistlock never appears on the network with this default account, specify the admin username and password in twistlock.cfg before installing Twistlock:

1 | Open ***twistlock.cfg*** for editing.

2 | Set values for the following parameters:

```
ADMIN_USERNAME=
ADMIN_PASSWORD=
```

3 | Install Twistlock using the normal procedure.

4 | Delete ***twistlock.cfg*** so that the admin user account credentials cannot be compromised.

NOTE: A copy of your ***twistlock.cfg*** file is kept in ***/var/lib/twistlock/scripts*** for future reference. Because this file contains sensitive information, it is secured with the following permissions:

- Ownership and group ownership for the ***/var/lib/twistlock*** directory is set to root, and the permissions are set to 750 (writable by owner, readable by owner and group).
- Ownership and group ownership for ***twistlock.cfg*** is set to root, and permissions are set to 640 (writable by owner, readable by group).

MORE INFORMATION
Install Twistlock

Enforce encryption for all connections to Console
For convenience, the default Twistlock installation lets you connect to Console (web portal and API) over HTTP on port 8081. HTTP is a clear-text, unencrypted protocol. Eavesdroppers can easily intercept sensitive data transmitted over HTTP. You should disable the HTTP management port, and force users to connect to Console over HTTPS on port 8083.

TWISTLOCK CONFIGURATION
To disable the HTTP management port, edit ***twistlock.cfg*** before installing Twistlock:

1. Open ***twistlock.cfg*** for editing.

2. Disable the HTTP management port by leaving the value empty:
   ```
   MANAGEMENT_PORT_HTTP=
   ```

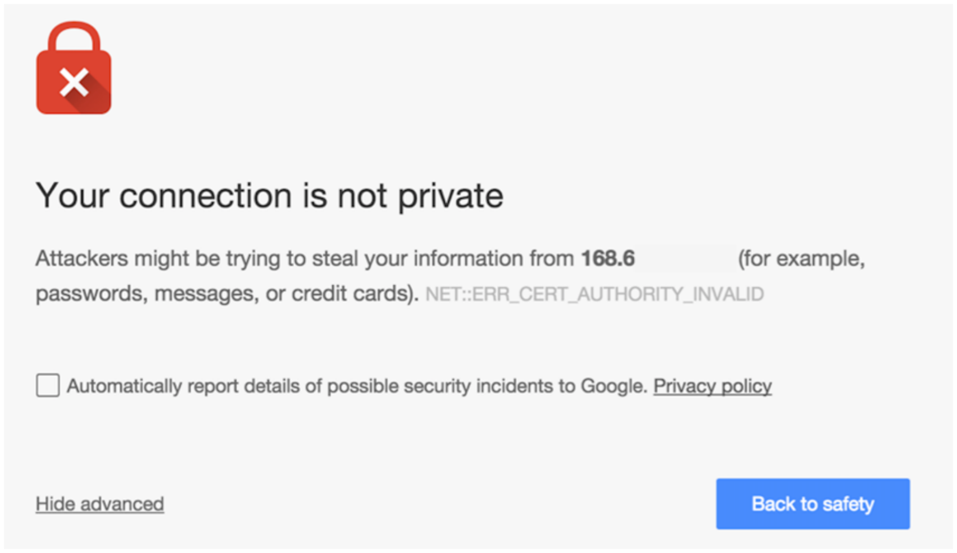
3. Install Twistlock using the normal procedure.

NOTE: If you have already installed Twistlock, and you want to disable the HTTP management port, re-run the installer with an updated configuration file.

MORE INFORMATION
Install Twistlock
Reconfigure Twistlock

Install your own certificates
By default, Twistlock secures access to Console's web portal and API with a self-signed certificate. When you access Console's web portal with this setup, the browser flags the portal as untrusted with a warning message.

For example, the Chrome browser shows the following message when you access Console over HTTPS with Twistlock's self-signed certificate:



Your connection is not private

Attackers might be trying to steal your information from **168.6** (for example, passwords, messages, or credit cards). NET::ERR_CERT_AUTHORITY_INVALID

☐ Automatically report details of possible security incidents to Google. Privacy policy

You can resolve these warnings by installing your own certificate that proves your server's identity to the client.

MORE INFORMATION
[Custom certs for Console access]

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