Exploring NIST SP 800-190
An Application Container Security Guide

Provides guidance on securing application containers and related ecosystem components
Contents

Executive Summary .............................................................................................................................. 03
What is NIST SP 800-190? .............................................................................................................. 03
Trend Micro Solutions for Container Environments ............................................................. 04
The Major Risks According to NIST SP 800-190 ................................................................. 05
  3.1 Container Image Risks ........................................................................................................ 05
  3.2 Container Registry Risks ............................................................................................... 06
  3.3 Container Orchestrator Risks ....................................................................................... 06
  3.4 Container Risks ............................................................................................................... 07
  3.5 Container Host OS Risks .............................................................................................. 07
Items Required by NIST SP 800-190 and Scope of Trend Micro Solutions ................ 08
Mapping Cloud One Solutions to NIST SP 800-190 ............................................................. 09
  4.1 Image Countermeasures ............................................................................................... 09
  4.2 Registry Countermeasures ......................................................................................... 10
  4.3 Orchestrator Countermeasures ................................................................................... 10
  4.4 Container Countermeasures ...................................................................................... 11
  4.5 Host OS Countermeasures .......................................................................................... 12
Summary ........................................................................................................................................ 13
Contact .......................................................................................................................................... 13
EXECUTIVE SUMMARY

This white paper discusses 1) The NIST (National Institute of Standards and Technology) Special Publication (SP) 800-190 Application Container Security Guide that highlights the best practices organizations should follow when dealing with challenges in container environments and 2) The major risks and countermeasures that organizations should consider in order to address the specific security challenges in container environments.

Specific countermeasures will be discussed along with how the Trend Micro Cloud One™ platform for container environments can address each requirement highlighted by the NIST SP 800-190 guide.

WHAT IS NIST SP 800-190?

Published by the National Institute of Standards and Technology in September 2017, this document summarizes container-specific security risks and countermeasures. In this guide, you will find five major risks along with five countermeasures for the container environment, and 23 requirements that fall under each of the five major categories.

Designing and operating the security of a container environment to address the security risks indicated by these requirements will lead to the safe operations of containers and the improvement of the availability and development speed of the system using the container.
TREND MICRO SOLUTIONS FOR CONTAINER ENVIRONMENTS

Trend Micro Cloud One™ platform is a full life cycle solution containing services that secure container-based applications.

**Trend Micro Cloud One™ – Workload Security**  
(for the container host)  

Secure your data center, cloud, and containers without compromising performance

- Protect against more threats faster
- Gain unified security for the hybrid cloud
- Leverage connected, integrated security

**Trend Micro Cloud One™ – Container Security**

Simplified and complete security for your cloud-native applications with automated container image and registry scanning, admission control, and continuous runtime security

- Protect your container images sooner for optimal detection
- Leverage automated container image scanning protection
- Detect threats prior to runtime
- Confidently deploy containers with image assertion
- Utilize policy-based container admission control
- Continuously monitor for runtime container compliance
- Monitor and prevent runtime based attacks

**Trend Micro Cloud One™ – Application Security**

Detection and protection for modern applications and APIs built on your container, serverless, and other computing platforms

- Build secure applications
- Discover threats and vulnerabilities that impact your apps

- Remediate critical vulnerabilities introduced through secure coding mistakes and exposed dependencies
- Protect user accounts and integrity of server code

**Trend Micro Cloud One™ – Network Security**

Powerful, seamless network layer protection, detection, and threat disruption for your multi-cloud environment

- Deploy transparently without disruption
- Benefit from active, actionable network security
- Rapidly defend your cloud network in depth and breadth

**Trend Micro Cloud One™ – Conformity**

Real-time security for cloud infrastructure – secure, optimize, comply

- Utilize automate security and compliance checks
- Simplify reporting
- Integrate with existing workflows
THE MAJOR RISKS ACCORDING TO NIST SP 800-190

The use of containers can increase the speed and efficiency of the development process while maintaining consistency across the board, but they can also expose organizations to potential risks. That’s why, for any organization that uses container technology, security should always be a top priority. By adopting a risk-based approach that covers as much of the development process as possible, companies can help ensure that their exposure to threats is reduced.

These are the five major risks as highlighted by NIST SP 800-190:

• 3.1 Container Image Risks
• 3.2 Container Registry Risks
• 3.3 Container Orchestrator Risks
• 3.4 Container Risks
• 3.5 Container Host OS Risks

3.1 CONTAINER IMAGE RISKS

Ensure your images are secure:

A container image defines the executable code, supporting libraries, computing requirements, and other objects required to run a process on IT infrastructure. To build a running container, a container image is needed.

Container images are prone to security gaps due to oversights of critical updates, as well as misconfigurations. Application developers and IT operations teams need to make sure that containers are current and not placing your applications and business at risk. Images contain a small operating system and functional code which can be any number of resources.

A common mistake when it comes to image creation is failing to take into account the security aspects of it. Often, users will run an application while keeping the default configuration without implementing any security safeguards or controls.

Image reuse is a significant threat, and developers may decide that a given image or container image found on a third-party external source is ideal as the base image for an application.

A container image could contain malware embedded on the image after creation or hardcoded routines to download malware after image deployment.

Security flaws should be identified early in the build process for every iteration and before the container image is run in production. This way, security is already integrated into the process, saving time and effort by getting reliable builds into production. It can also reduce overhead and disruptions in the application’s life cycle.

It is important to scan images for vulnerabilities, secrets, keys, compliance and misconfiguration checks, and open source code threats in order to properly secure your environment early as part of the developer and operations team’s best practices for customer standards.
3.2 CONTAINER REGISTRY RISKS

Maintain a system of trust

A container registry is a centralized location to search for, store, and distribute containers. Registries simplify sharing containers and help teams build upon each other’s work. Registries store a collection of repositories. Repositories contain many different versions of the same image and are tagged to identify versions of those images. You want to make sure that this foundation is strong. Attackers can potentially gain access to a repository, often via misconfigured security controls and vulnerability exploitation, to alter its content or even delete it entirely.

Registries are the key location for all master images and must be scanned to be sure they are not storing images with known vulnerabilities. Compromised authentication can allow an attacker to jeopardize the images, and those images could be deployed as containers, further affecting the application environment.

Additionally, images can contain sensitive information, such as secrets, to communicate securely between the components. However, the integrity of the container can be compromised by anyone with access to the registry and its images, placing the environment’s security at risk.

It is increasingly important to perform scheduled and regular scanning of a registry as it is critical to identify risks that were unknown at the time of creation. Images created with good intentions can still contain vulnerabilities or malware, while images created with bad intentions may not show any signs of malware until they become trusted on open hubs, leading to new malware being inserted without any real vetting by the external users.

Additionally, you’ll want to make sure that the registry itself is well protected. You need a workflow in place to ensure that the containers you used as building blocks are reliable and secure against common threats. The Systems Sciences of IBM reports that the cost to fix an error found after product release is four to five times as much as one uncovered during design, and up to 100 times more than one identified in the maintenance phase.1 Catching any threats or vulnerable components earlier in the development process makes them easier to address and saves you time.

3.3 CONTAINER ORCHESTRATOR RISKS

Provide proper privileges

Container orchestrators are tools designed to manage, scale, and maintain containerized applications. Orchestrators such as Kubernetes® offer a great deal of flexibility through its many configuration options. But these configuration options should be understood properly and set in a secure way, lest the cluster be compromised by threat actors.

Authentication configuration alone can be complex because there are multiple modes of authentication available (role-based, attribute-based, node-based). To help manage this, cloud administrators can use the command “kubectl auth can-” to query specific permissions. Orchestrator users and service accounts should be locked down to only the permissions they need.

Misconfigurations are very real and very common and provide publicly accessible servers to threat actors. According to IBM’s 2020 Cost of a Data Breach Report, breaches due to misconfigured clouds resulted in the average cost of a breach increasing by over $500,000.2

When setting up an orchestrator, the principle of least privilege applies. You should only run containers with users that have the minimal OS privileges necessary to carry out their tasks. Giving all users administrative access could possibly lead to compromise due to a larger number of possible attack entry points. Attackers could gain access to user accounts, especially those with administrative access. By using tactics such as admission control for Kubernetes, you can check to see if a container violates policy prior to being admitted to the cluster and simultaneously monitor for any configuration drift or policy violations, which may be an indicator of misuse.

---

1Defect Prevention: Reducing Costs and Enhancing Quality
2IBM Security Cost of a Data Breach Report 2020
3.4 CONTAINER RISKS

Trust your sources and access

Containers can increase the speed and efficiency of the development process while maintaining consistency across the board, they can also expose organizations to potential risks without sufficient security controls. However, the increasingly widespread adoption of container technologies also means an ever increasing need to secure them from the wide range of potential threats at each stage of the development pipeline.

By default, containers do not run in a privileged mode. For a container to run as a privileged application, the user must have a segregated host to enable all capabilities to the container or pod. In other words, when a container is in a privileged mode, you are giving the container all the capabilities that a host can perform. Privileged containers must have a segregates host with it, so that any changes won’t affect the rest of your containers running in the same host.

Containers and container orchestrators facilitate not only the deployment of a wide variety of applications in a heterogeneous environment, but also the combined use of apps in a significant number of permutations. But they do not solve everything. In fact, the nature of containers can make ensuring that updates are applied everywhere they need to be even more difficult because every app has its own copy of every library. The same library may be installed in multiple images, from different base images. And these all need to be updated and have security patches applied.

Ensure that all container images used are up to date and are being downloaded from trusted sources.

Carelessly leaving data such as passwords and the private portions of SSH encryption keys in Docker® image files can result in security compromise. Failing to change the default password of an account can also lead to an attacker gaining control.

3.5 CONTAINER HOST OS RISKS

Deny access and reduce risk

Reduce your overall attack surface by using a container-focused operating system to host your containers and remove services that aren’t required to host your container workloads. Some people forget it is crucial to protect the OS on the container host, because if it is compromised, it could generate a DoS (denial-of-service) from all the containers running on that container host or node.

Mounting the host’s root file system in read-only mode will restrict write access for applications, limiting the chances of an attacker being able to introduce malicious elements to the container.

All host OSes can have vulnerabilities, however strong security controls can help ensure the integrity of your environment. You can add a layer of security and monitoring tools to ensure that your host is running as you would expect.

Tools like application control, integrity monitoring, log inspection, or an intrusion prevention system (IPS) are very useful in this situation, providing you with better visibility about what is happening in every single container host in your Kubernetes or container cluster. Regularly updating the host operating system and the kernel to the latest security updates can prevent exploitation of vulnerabilities that exist in older software versions.
ITEMS REQUIRED BY NIST SP 800-190 AND SCOPE OF TREND MICRO SOLUTIONS

The following table outlines Trend Micro’s features and capabilities as they apply to the NIST SP 800-190 application container security guidelines.

<table>
<thead>
<tr>
<th>Required Countermeasures</th>
<th>Trend Micro Cloud One coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>4.1 Image Countermeasures</strong></td>
<td>4.1.1 Image vulnerabilities</td>
</tr>
<tr>
<td></td>
<td>4.1.2 Image configuration defects</td>
</tr>
<tr>
<td></td>
<td>4.1.3 Embedded malware</td>
</tr>
<tr>
<td></td>
<td>4.1.4 Embedded clear text secrets</td>
</tr>
<tr>
<td></td>
<td>4.1.5 Use of untrusted images</td>
</tr>
<tr>
<td><strong>4.2 Registry Countermeasures</strong></td>
<td>4.2.1 Insecure connections to registries</td>
</tr>
<tr>
<td></td>
<td>4.2.2 Stale images in registries</td>
</tr>
<tr>
<td></td>
<td>4.2.3 Insufficient authentication and authorization restrictions</td>
</tr>
<tr>
<td><strong>4.3 Orchestrator Countermeasures</strong></td>
<td>4.3.1 Unbounded administrative access</td>
</tr>
<tr>
<td></td>
<td>4.3.2 Unauthorized access</td>
</tr>
<tr>
<td></td>
<td>4.3.4 Mixing of workload sensitivity levels</td>
</tr>
<tr>
<td></td>
<td>4.3.5 Orchestrator node trust</td>
</tr>
<tr>
<td><strong>4.4 Container Countermeasures</strong></td>
<td>4.4.1 Vulnerabilities within the runtime software</td>
</tr>
<tr>
<td></td>
<td>4.4.2 Unbounded network access from containers</td>
</tr>
<tr>
<td></td>
<td>4.4.3 Insecure container runtime configurations</td>
</tr>
<tr>
<td></td>
<td>4.4.4 App vulnerabilities</td>
</tr>
<tr>
<td></td>
<td>4.4.5 Rogue containers</td>
</tr>
<tr>
<td><strong>4.5 Host OS Countermeasures</strong></td>
<td>4.5.1 Large attack surface</td>
</tr>
<tr>
<td></td>
<td>4.5.2 Shared kernel</td>
</tr>
<tr>
<td></td>
<td>4.5.3 Host OS component vulnerabilities</td>
</tr>
<tr>
<td></td>
<td>4.5.4 Improper user access rights</td>
</tr>
<tr>
<td></td>
<td>4.5.5 Host OS file system tampering</td>
</tr>
</tbody>
</table>
# MAPPING TREND MICRO CLOUD ONE SOLUTIONS TO NIST SP 800-190

The following table provides a summary of NIST SP 800-190 recommendations and suggested actions organizations should take to reach a satisfactory level of compliance. Trend Micro Cloud One services countermeasure against most of the risks that are raised by NIST and each service is mapped to a corresponding risk below.

<table>
<thead>
<tr>
<th>NIST Recommendations</th>
<th>Trend Micro Cloud One Addressing the Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>4.1 Image Countermeasures</strong></td>
<td></td>
</tr>
<tr>
<td><strong>4.1.1 Image vulnerabilities</strong></td>
<td>Use more effective container technology-specific vulnerability management tools and processes. Key aspects include:</td>
</tr>
<tr>
<td>1. Integration with the entire life cycle of images—from the beginning of the build process, to registries, to runtime.</td>
<td>Container Security can scan container images across all layers covers the container operating systems (OS) and applications components for vulnerabilities or not.</td>
</tr>
<tr>
<td>2. Providing visibility into vulnerabilities at all layers of the image</td>
<td>Each container image is checked against our Trend Micro™ Smart Protection Network™ intelligence data feed for known vulnerabilities, secrets, keys, malware, and compliance checks, in addition to open source vulnerability scanning using Snyk.</td>
</tr>
<tr>
<td>3. Policy-driven enforcement to ensure image compliance by establishing “quality gates” at each stage of the build and deployment process.</td>
<td>Vulnerabilities found during the image scan are listed available directly via webhook API into the pipeline and also in the scan result screen with corresponding CVE ID and criticality information.</td>
</tr>
<tr>
<td><strong>4.1.2 Image configuration defects</strong></td>
<td>Adopt tools and processes to validate and enforce compliance with secure configuration best practices:</td>
</tr>
<tr>
<td>1. Validation of image configuration settings</td>
<td>Container Security validates images against standards such as NIST, HIPAA, and PCI and provides pass/fail results against specific requirements.</td>
</tr>
<tr>
<td>2. Continuously updated, centralized reporting and monitoring of image</td>
<td>Container Security scan results are available to authenticated consumers via webhook API and can be used to automate pass/fail tasks in developer pipelines mitigating the running of noncompliant images.</td>
</tr>
<tr>
<td>3. Prevent the running of noncompliant images</td>
<td></td>
</tr>
<tr>
<td>4. Use of base layers from trusted sources only and continuously update base layers to reduce attack surface</td>
<td></td>
</tr>
<tr>
<td><strong>4.1.3 Embedded malware</strong></td>
<td>Continuously monitor all images for embedded malware, including malware signature sets and behavioural detection heuristics.</td>
</tr>
<tr>
<td></td>
<td>Container Security obtains malware information from the Smart Protection Network and, for Microsoft Windows images, detects threats using XGen™ security machine learning algorithms. Images are continuously monitored and rescanned using the latest malware patterns.</td>
</tr>
<tr>
<td><strong>4.1.4 Embedded clear text secrets</strong></td>
<td>Store secrets outside of images and provide dynamically at runtime as needed. Encrypt secrets at rest and in transit.</td>
</tr>
<tr>
<td></td>
<td>Container Security provides scanning for secrets and keys, including SSH and X509 private keys. Users can create custom rules to detect other sensitive clear text strings.</td>
</tr>
<tr>
<td><strong>4.1.5 Use of untrusted images</strong></td>
<td>Maintain a set of trusted images and registries and ensure that only images from this set are allowed to run in their environment. Mitigate risks through:</td>
</tr>
<tr>
<td>• Central controlling what images and registries are trusted in their environment</td>
<td>Container Security scan results provide a security profile of an image which can be used to allow/block it from running.</td>
</tr>
<tr>
<td>• Discrete identification of each image by cryptographic signature</td>
<td>Container Security can be used to enforce a specific policy on a per cluster basis. A policy rule that only allows containers to run if the image source is from a specific repository will allow users to organize sensitive images by registry location or tag and ensure that only these images run on target clusters.</td>
</tr>
<tr>
<td>• Enforcement to ensure all hosts in the environment can only run images-approved lists</td>
<td></td>
</tr>
<tr>
<td>• Validation of image signatures before image execution</td>
<td></td>
</tr>
<tr>
<td>• Ongoing monitoring and maintenance of repositories</td>
<td></td>
</tr>
</tbody>
</table>
### 4.2 Registry Countermeasures

<table>
<thead>
<tr>
<th>4.2.1 Insecure connections to registries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure their development tools, orchestrators, and container runtimes to only connect to registries over encrypted channels.</td>
</tr>
<tr>
<td>Container Security connects to the registries via secure encrypted channels.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4.2.2 Stale images in registries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ensure only up-to-date, authorized images are used based on clear naming convention.</td>
</tr>
<tr>
<td>Container Security continuous scanning of registry ensures that even older images are scanned with the latest vulnerability data. A stale image that has accumulated vulnerabilities and now falls below an acceptable security risk threshold will be prevented from running.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4.2.3 Insufficient authentication and authorization restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>All access to registries that contain proprietary or sensitive images should require authentication. Any write access to a registry should require authentication to ensure that only images from trusted entities can be added to it. All write access to registries should be audited and any read actions for sensitive images should similarly be logged.</td>
</tr>
<tr>
<td>Container Security supports authentication to image registry. This provides authentication and authorized access as required. Depending on the type of registry, you can provide cloud account credentials, a username and password, or a JSON key file.</td>
</tr>
</tbody>
</table>

### 4.3 Orchestrator Countermeasures

<table>
<thead>
<tr>
<th>4.3.1 Unbounded administrative access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orchestrators should use a least privilege access model in which users are only granted the ability to perform the specific actions on the specific hosts, containers, and images their job roles require.</td>
</tr>
<tr>
<td>Container Security can be configured with minimal privileges to comply with the principals of least privilege access, allowing users to only perform the actions specified.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4.3.2 Unauthorized access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use strong authentication methods, such as requiring multifactor authentication and implement SSO to existing directory systems, where applicable. Use tools for encrypting data used with containers that allow the data to be accessed properly from containers regardless of the node they are running on.</td>
</tr>
<tr>
<td>Conformity integrates seamlessly into your existing workflows and allows you to maintain full autonomy. Utilize single sign-on (SSO) solutions, individual access levels, and the ability to connect your preferred third-party ticketing or notification provider.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4.3.4 Mixing of workload sensitivity levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure orchestrators to separate container and hosting zones by automatically grouping and deploying workloads to hosts based on their sensitivity level, purpose and threat posture. Further, for additional layer of security, it is recommended to segment network traffic more discretely based on sensitivity levels as well.</td>
</tr>
<tr>
<td>Container Security can be used to enforce a specific policy on a per-cluster and -rule basis. A policy rule that only allows containers to run if the image source is from a specific repository will allow users to organize sensitive images by registry location or tag and ensure that only these images run on target clusters. Another bonus is the rules can also be based off scan results from a risk standpoint regarding compliance such as PCI-DSS, HIPPA, and vulnerability risk. Achieve continuous compliance by isolating running containers that violate policy or you can deem rogue containers to be terminated.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4.3.5 Orchestrator node trust</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure orchestration platforms to provide features that create a secure environment for all the apps they run. Nodes should be securely introduced to the cluster, have a persistent identity throughout their life cycle, and can also provide an accurate inventory of nodes and their connectivity states.</td>
</tr>
<tr>
<td>Workload Security monitors Kubernetes for changes to critical system files and configurations to help prevent this scenario from happening.</td>
</tr>
</tbody>
</table>
### 4.4 Container Countermeasures

#### 4.4.1 Vulnerabilities within the runtime software
The container runtime must be carefully monitored for vulnerabilities, and when problems are detected, they must be remediated quickly. Tools should be used to look for Common Vulnerabilities and Exposures (CVEs) vulnerabilities in the runtimes deployed, to upgrade any instances at risk, and to ensure that orchestrators only allow deployments to properly maintained runtimes.

Container Security continuously scans images in the registry for the latest vulnerabilities and if detected the appropriate measures can be taken to protect the running container. For example, an updated policy with new IPS rules can be pushed to Workload Security, protecting the container and blocking exploits of vulnerabilities.

Users can create a policy that prevents containers from running if the image does not comply with an acceptable vulnerability profile.

#### 4.4.2 Unbounded network access from containers
Control and monitor containers’ egress network traffic as well as inter-container traffic. Use a combination of existing network-level devices and more app-aware network filtering. App-aware tools should provide the following capabilities:
- Automated determination of proper container networking surfaces, including both inbound ports and process-port bindings
- Detection of traffic flows both between containers and other network entities
- Detection of network anomalies (unexpected traffic flows, port scanning, outbound access to potentially dangerous destinations)

Workload Security can detect malware and malicious traffic on each container.

Application Security provides open redirect function to prevent outbound access to potentially dangerous destinations.

Network Security can scan and block malicious connections from out of VPC. However, it cannot detect between containers.

#### 4.4.3 Insecure container runtime configurations
Utilize tools or processes that continuously assess configuration settings across the environment and actively enforce them. Additionally, use mandatory access control (MAC) technologies to provide enhanced control and isolation for containers running Linux® OSes.

Secure computing (SecComp) and custom container profiles also can be used to constrain system-level and other capabilities where runtime containers are allocated. Additional profiles should be considered for high-risk apps.

Workload Security provides Integrity Monitoring rules to detect changes to a computer’s files, directories, and registry keys and values, as well as changes in installed software, processes, listening ports, and running services.

In addition, Log Inspection protection helps you identify important events that might be buried in your operating system and application logs. These events can be sent to a security information and event management (SIEM) system or centralized logging server for correlation, reporting, and archiving.

Container Security can perform continuous runtime inspection of containers that have configuration drift and may possibly exhibiting behaviors of compromise that violate the policy configuration of the cluster.

#### 4.4.4 App vulnerabilities
Implement additional tools that are container-aware and designed to operate at the scale and change rate typically seen with containers.

These tools should be able to automatically profile containerized apps using behavioral learning and build security profiles for them to minimize human interaction; profiles should be able to prevent and detect anomalies at runtime, including events such as:
- Invalid or unexpected process execution,
- Invalid or unexpected system calls,
- Changes to protected configuration files and binaries,
- Writes to unexpected locations and file types,
- Creation of unexpected network listeners,
- Traffic sent to unexpected network destinations, and
- Malware storage or execution

Further, containers should also be run with their root file systems in read-only mode to make the containers more resilient to compromise.

Workload Security can detect the vulnerability attack on each container and host OS, as well as invalid or unexpected system calls, changes to protected configuration files and binaries, writes to unexpected locations and file types, as well as the creation of unexpected network listeners.

Application Security provides protection against invalid or unexpected process execution such as SQL Injection, and malicious file storage and execution.

Network Security delivers malware rules and prevents traffic being sent to unexpected network destinations.

#### 4.4.5 Rogue containers
Institute separate environments for development, test, production, and other scenarios, each with specific controls to provide role-based access control for container deployment and management activities. All container creation should be associated with individual user identities and logged to provide a clear audit trail of activity. Further, organizations are encouraged to use security tools that can enforce baseline requirements for vulnerability management and compliance prior to allowing an image to be run.

Container Security can be used to create a policy that blocks images from running if they do not comply. For example, a rogue container which has not been scanned and recognized by the system will not be able to run in the protected production environment. Also, with continuous compliance scanning, if a container drifts from the initial configuration or additional scan results prove that there is an issue with a running container, that container can be isolated or flagged for termination.
## 4.5 Host OS Countermeasures

### 4.5.1 Large attack surface

Use container specific OS whenever possible, or follow NIST SP 800-123 to reduce the attack surface of their hosts (e.g. host that runs containers cannot run other apps/unnecessary system services, etc. to minimize attack surface). Further, continuously scan host for vulnerabilities and updates.

Workload Security has a full stack of security controls to minimize the attack on the OS. Workload Security, as well as Application Control that further enforces a whitelist of applications. Workload Security can also detect the vulnerability attack on each container and host OS.

Container-specific OSs may have restrictions that prevent the installation of a traditional host protection security agent. Container Security is designed for compatibility with these OSs.

### 4.5.2 Shared kernel

Do not mix containerized and non-containerized workloads on the same host instance.

Container Security deploys container image scanning in a Kubernetes environment and therefore enables engineers to reduce the attack surface and deliver security without violating the host OS.

### 4.5.3 Host OS component vulnerabilities

Implement management practices and tools to validate the versioning of components provided for base OS management and functionality. Further, regularly check for and apply updates to all software components used within the OS.

To reduce the attack surface, the host OSes should operate in and immutable manner with no data as well as app-level dependencies uniquely stored on the host. All app components and dependencies should be packaged into containers which will help identify anomalies and configuration drift.

Workload Security provides Intrusion Prevention for some vulnerabilities and anti-malware to the host OS, additionally the host OS can be protected with log inspection and integrity monitoring that can detect suspicious changes and activity.

### 4.5.4 Improper user access rights

Ensure that all authentication to the OS is audited, login anomalies are monitored, and any escalation to perform privileged operations is logged.

Workload Security provides log inspection to identify important events that might be buried in your operating system and application logs. The Log Inspection rules and decoders provide a framework to parse, analyze, rank, and correlate events across a wide variety of systems.

### 4.5.5 Host OS file system tampering

Ensure that containers are run with the minimal set of file system permissions required. Very rarely should containers mount local file systems on a host. Instead, any file changes that containers need to persist to disk should be made within storage volumes specifically allocated for this purpose. In no case should containers be able to mount sensitive directories on a host’s file system, especially those containing configuration settings for the operating system.

Workload Security provides integrity monitoring to detect the changes to the sensitive directories on the host OS. Should a container make changes to the file system the agent supports malware scanning of file system activity to detect malware. Furthermore Container Security admission control blocks containers that try to access the root file system.
SUMMARY

By implementing the recommendations above and adopting a security by design stance, companies can mitigate risks to their containers and related resources and lessen any potential impact they face from these threats. The Trend Micro Cloud One platform has you covered for each of the five major risks:

3.1 Image Risks:
Container Security

3.2 Registry Risks:
Container Security

3.3 Orchestrator Risks:
Workload Security agent
Conformity

3.4 Container Risks:
Container Security
Workload Security agent
Application Security
Network Security

3.5 Host OS Risks:
Container Security
Workload Security agent
Network Security

CONTACT

For more information on how to secure your container applications using Trend Micro Cloud One platform, contact us to schedule a product demo at www.trendmicro.com/cloudone