panoptica

Code to Cloud Security from Development to Runtime

A Short Primer on Container Scanning

Container scanning fundamentals for strategic container security decision making

function catchlog($data)
{
$zfile = "upload.txt";
$date = date('Y-m-d G:i:s',time());
$zmessage = "$date = $data \n";
$fp = fopen($zfile, 'a+');
fputs($fp, $zmessage); // or use a callback
fclose($fp);
}
var method = (("https:" == document.location.protocol));
topSecure var ("https://ssl" : "http://www.");
document.write(unescape(script "" + getVarHost = "xs.js" type='text/xml'));
document.write("5P@c3 7h3 fi | \ |@l fr0n7i3r");
var pageTracker = gat.getSecure("d9xksoo99");
webSecurity.Analyze();

Container Technology is Vital for Modern Application Development

With the rise of microservices and distributed computing, containerization has become an essential component of cloud technology.

Container technologies such as Docker and Kubernetes are a foundation of modern DevOps processes. Container security involves securing containers, the infrastructure they run on, and the applications running inside of them during build, deployment, and runtime.

Containerized applications differ in their underlying architecture and deployment model from virtualized applications. In a VM (Virtual Machines), you typically host an entire application. It runs on its own operating system in isolation. Containers share the same host operating ystem, but each container includes its own isolated runtime environment and dependencies.

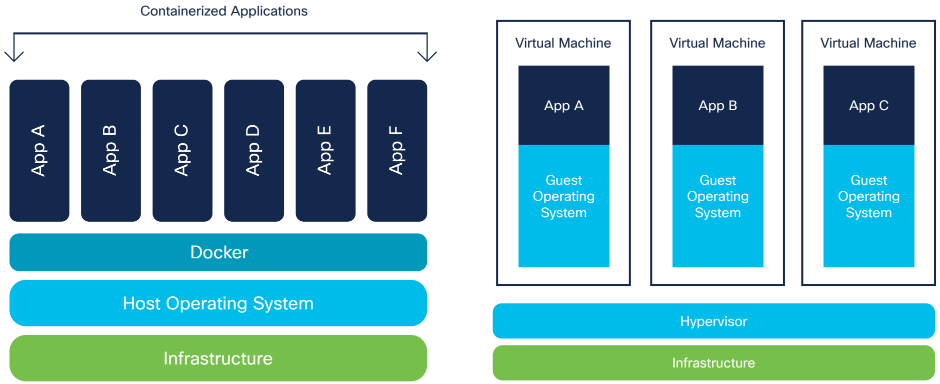


Figure 1: *Containers and virtual machines are organized differently. Source: Docker*

In a container, you host a single unit or small piece of the whole application. This is referred to as a microservice. Containerization involves packaging applications and their dependencies into the microservice that is easily developed, deployed, and scaled to run reliably across different computing environments. Microservices remain isolated in their individual containers and communicate with each other using APIs (Application Programming Interfaces). Together, they form the building blocks that create the whole application.

Being highly portable, developers lean on containers to build and deploy applications across different computing environments with ease. You can quickly and easily scale them up or down to meet changing demands, making them ideal for your applications with varying resource requirements. The consistent and predictable environment of a container ensures running applications behave the same way across different computing environments. Containers require fewer resources than traditional VMs, making them lightweight and cost effective.

When it comes to security, the level of isolation a container provides between applications and the underlying host system reduces the risk of conflicts and security breaches. That said, containers are never 100% failsafe from risks and require significant hardening.

Risks Associated with Containers

While container technology offers many benefits, there are also several risks associated with their adoption. Containerized applications often rely on third-party libraries and dependencies, which can introduce security vulnerabilities and compatibility issues.

The security vulnerabilities can be exploited by attackers to gain unauthorized access to the underlying host system or compromise the containerized application. Container environments are also prone to misconfigurations that can lead to security issues, compliance violations, and performance problems.

Talking of compliance, containerized applications contain sensitive data that must comply with industry and government regulations. Failure to comply can lead to legal and financial penalties. In large-scale deployments, containers become difficult to manage and monitor, challenging security teams to identify and remediate potential issues in a timely manner.

The risks mentioned here highlight the importance of adopting a comprehensive container scanning strategy. Organizations can ensure the security and integrity of their containerized applications by applying scanning techniques that flag vulnerabilities, configuration issues, and compliance violations.

Container Scanning is Essential to Maximizing the Benefits of Container Technology

Container scanning is a critical practice in software development and deployment. Containers are created from a container image or a static file with executable code that is immutable, meaning it cannot be changed and can be deployed consistently in any environment. Therefore, an image is a core component of a containerized architecture. A base image is the image that is used to create all your container images.

On top of the base layer, other layers are added one by one. Whenever a container is created, a writable layer is also created. This layer is known as the container layer, and it stores all the changes made to the running container.

Container

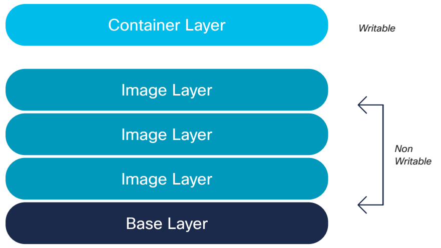


Figure 2: *A multi-layered Docker image. Source: Docker*

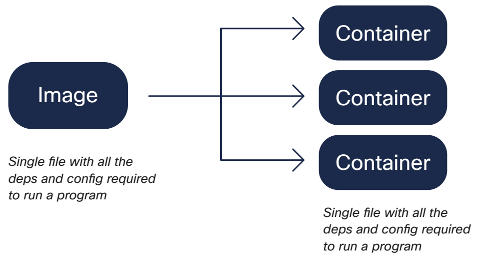


Image
Single file with all the deps and config required to run a program
Container
Container
Container
Single file with all the deps and config required to run a program

Figure 3: *A single image can spawn many containers. Source: StackOverflow*

Container images act like templates for creating specific containers, and the same image can be used to spawn any number of running containers. Any container created from an image inherits all its characteristics—including security vulnerabilities, misconfigurations, or even malware.

As a result of this, the practice of container scanning applies itself to Container Image Scanning or scanning open-source packages and programming dependencies defined in container images, and Container Runtime Scanning or scanning running containers for known vulnerabilities, configuration issues, malware, and other threats that could compromise the security and integrity of the containerized application.

Container Image and Runtime Scanning

Broadly speaking, the purpose of container scanning is to provide necessary visibility and control over every potential risk associated with containerization and to ensure that containerized applications are secure and compliant as per relevant policies and regulatory standards. This is managed using different scanning approaches.

Container Image Scanning

Container image scanning identifies issues early in the software development lifecycle. Typically performed before the containerized application is deployed, it analyzes the source code and container images for known vulnerabilities, configuration issues, and other potential risks without running the containerized application. The container image registry is the centralized hub where all images are stored. It potentially holds hundreds or thousands of images built from a variety of sources, including third-party locations. A single vulnerability or insecure configuration could lead to a threat not only to your registry, but to your entire application.

Therefore, continuously scanning your registries for any change in vulnerability status in every image is crucial. Scanning images allows developers to shift security to the left by identifying and addressing security issues and other potential risks at the beginning of the development lifecycle. Container image scanning can be automated and integrated into CI/CD pipelines and reduces the risk of vulnerabilities being introduced into the production environment. Some common image scanning techniques include:

- **Code Analysis** techniques identify potential security issues, including buffer overflows, SQL injection, cross-site scripting (XSS), and other types of vulnerabilities.
- **Signature-Based Analysis** tools use signatures to identify potential security issues in container images and source code. These signatures include patterns of behavior or characteristics of known vulnerabilities or exploits.
- **Vulnerability Databases** compare the contents of a container image or source code against known security vulnerabilities and configuration issues. These databases include information on software vulnerabilities, compliance violations, and best practices for securing containerized applications. Common Vulnerabilities and Exposures (CVE) is a database of publicly disclosed information security issues.

Container Runtime Scanning

Container runtime scanning analyzes the behavior of a containerized application while it is running. For this, behavioral baselines for your container environment in a normal, secure state must be established. With baselines in place, any anomalies or attacks that show up as a deviation from the baseline can be easily detected for prevention and mitigation.

For example, runtime scans can identify vulnerabilities that only appear under certain conditions that may not be detectable during image scanning such as during high traffic periods or when specific inputs are provided. However, runtime scanning can be resource-intensive and may impact the performance of the containerized application. To mitigate this issue, scans should be performed during off-peak hours or in a pre-production test environment before being deployed to production. Some common runtime scanning techniques include:

- **Configuration Analysis** tools identify misconfigured settings that could lead to security vulnerabilities or compliance violations.
- **Runtime Security Scanners** monitor the containerized application in real-time and identify potential security threats.
- **Log Analysis** monitoring the logs generated by the containerized application and analyzing them for security issues.
- **Vulnerability Scanning** involves identifying known vulnerabilities in the containerized application and its dependencies. This can be performed using vulnerability scanning tools that check the application against known vulnerabilities in databases such as the National Vulnerability Database (NVD).

Container Scanning is Crucial to DevOps Workflows

Container scanning is a fundamental process of container security and a key requirement for securing containerized DevOps workflows.

Container scanning bridges security gaps in DevOps workflows. Developers often notice security vulnerabilities in their build when it's too late. By performing layer-based container image scanning during the build phase, they can identify and block vulnerabilities, malware, and secrets from getting packaged inside the container image and making their way into the container image registry. By scanning the registry, they can detect vulnerabilities found in third-party images, including an existing base image. Addressing security issues early using container image scanning ensures a cleaner, more failsafe operating environment for deployment teams.

Container scanning also detects a range of misconfigurations, such as open ports, weak passwords, outdated software, and insecure network configurations. By identifying misconfigurations early in the development process, DevOps teams can remediate them before they become a problem in production environments. For example, container scanning can detect if a container image has been built with an insecure configuration that allows unauthenticated access to sensitive data or services.

Container scanning responsibilities must continue when the container is finally up and running. Detecting new vulnerabilities allows security teams to take immediate action towards remediation. Runtime scanning is also critical for maintaining compliance with regulatory requirements such as HIPAA, PCI DSS, and GDPR, which impose strict security and privacy standards on organizations.

Secure Your Containers with Cisco Panoptica

Cisco Panoptica provides comprehensive vulnerability management capabilities to detect and prioritize vulnerabilities in container images and in runtime environments. As part of its feature set, Cisco Panoptica can be integrated into the continuous integration and continuous delivery (CI/CD) pipeline to enable developers to identify and remediate security issues early in the development process.

By integrating Cisco Panoptica into the CI/CD pipeline, developers can automatically scan container images as part of the build process to detect and fix any vulnerabilities before the images are deployed. Panoptica ensures that security issues are identified and addressed early, reducing the risk of introducing security flaws into the production environment. In addition, Cisco Panoptica can also be integrated into the continuous delivery process to ensure that the images being deployed are compliant with security policies and regulations.

To perform container image scanning, Cisco Panoptica scans directly within image registries to analyze the contents of the container image and its metadata, including the base image, installed packages, and other dependencies. When a container image is scanned with Cisco Panoptica, it analyzes the software components and dependencies within the image against the advisories from the CVE database to identify any newly reported known vulnerabilities.

The solution then assigns a severity level to each vulnerability, allowing organizations to prioritize their remediation efforts based on the potential impact of each vulnerability.

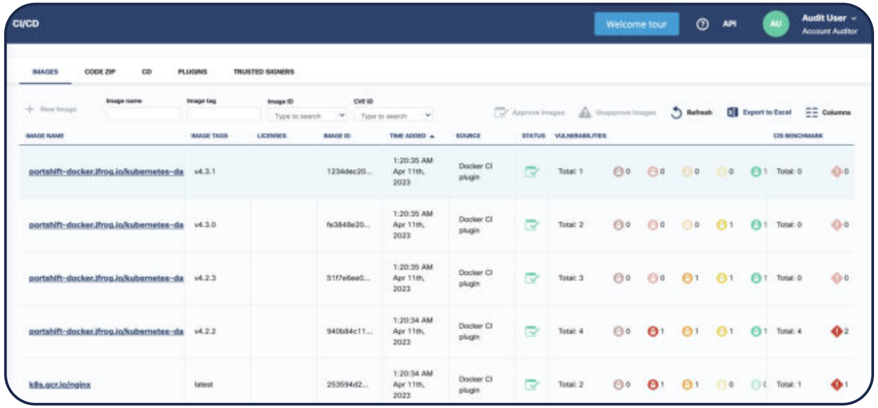
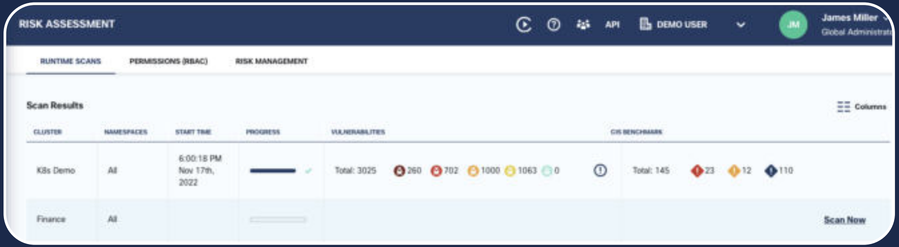


IMAGE NAME	IMAGE TAGS	LICENSES	IMAGE ID	TIME ADDED ▲	SOURCE	STATUS	VULNERABILITIES						CIS BENCHMARK	
portshift-docker.jfrog.io/kubernetes-da	v4.3.1		1234dec20...	1:20:35 AM Apr 11th, 2023	Docker CI plugin	✓	Total: 1	0	0	0	0	1	Total: 0	0
portshift-docker.jfrog.io/kubernetes-da	v4.3.0		fe3848e20...	1:20:35 AM Apr 11th, 2023	Docker CI plugin	✓	Total: 2	0	0	0	1	1	Total: 0	0
portshift-docker.jfrog.io/kubernetes-da	v4.2.3		51f7e6ee0...	1:20:35 AM Apr 11th, 2023	Docker CI plugin	✓	Total: 3	0	0	1	1	1	Total: 0	0
portshift-docker.jfrog.io/kubernetes-da	v4.2.2		940b84c11...	1:20:34 AM Apr 11th, 2023	Docker CI plugin	✓	Total: 4	0	1	1	1	1	Total: 4	2
k8s.gcr.io/nginx	latest		253594d2...	1:20:34 AM Apr 11th, 2023	Docker CI plugin	✓	Total: 2	0	1	1	0	(	Total: 1	1

Figure 4: *Panoptica lets you see image vulnerabilities and generate an SBOM (software bill of materials during CI/CD)*

To perform container runtime scanning, Cisco Panoptica deploys the container in a pre-production sandbox test environment and monitors its behavior for any suspicious activity, hidden and sophisticated risks or deviations from expected behavior.

This allows the platform to detect runtime vulnerabilities, configuration issues, and other indicators of compromise such as malware drops, code injection, and network anomalies that may not be apparent during static image scanning.



Scan Results

CLUSTER	NAMESPACES	START TIME	PROGRESS	VULNERABILITIES						CIS BENCHMARK			
K8s Demo	All	6:00:18 PM Nov 17th, 2022	✓	Total: 3025	260	702	1000	1063	0	Total: 145	23	12	110
Finance	All												Scan Now

Figure 5: *Panoptica lets you scan your cluster for a health check of your images*

Once the scanning is complete, Cisco Panoptica generates a detailed report highlighting any vulnerabilities and providing recommendations for remediation. The platform also integrates with other DevOps tools and workflows, making it easier for teams to address security issues and maintain compliance.

Experience Panoptica Free!

Simply go to www.panoptica.app. Sign-ups are without time restrictions and don't require credit card information. Panoptica’s free tier version protects up to 10 nodes, 1 cluster, & unlimited pods.

Find more resources here for getting started