CONTAINER SECURITY BEST PRACTICES: A HOW-TO GUIDE

CONTENTS

Containers are ruling the world

Application containers, like Docker, are standalone executable software packages that contain everything required to run an application: application code, dependencies, libraries, binaries and configuration files (see Figure 1).

With containers, DevOps is finally living the dream – "write once, deploy anywhere." Containers offer seamless portability across different computing environments, making it much easier for developers to build and deploy applications.

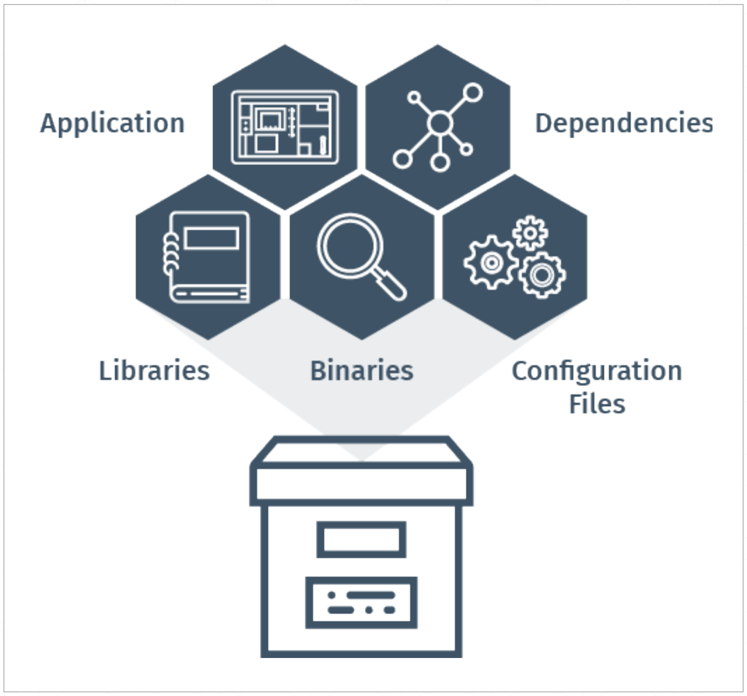


Figure 1. Containers have everything needed to run an application

Containers speed application development and deployment in DevOps

Developers are the driving force for container adoption because it dramatically simplifies application development and deployment. Containers allow developers to focus on what they do best — writing code — instead of mucking around with configurations and set-up during quality assurance testing, regardless whether the code is running on a laptop, a pre-production staging environment or a production environment.

IT operations prefer containers, too. They can pack more workloads on existing infrastructure because containers are lightweight and efficient, which means less hardware to run business services.

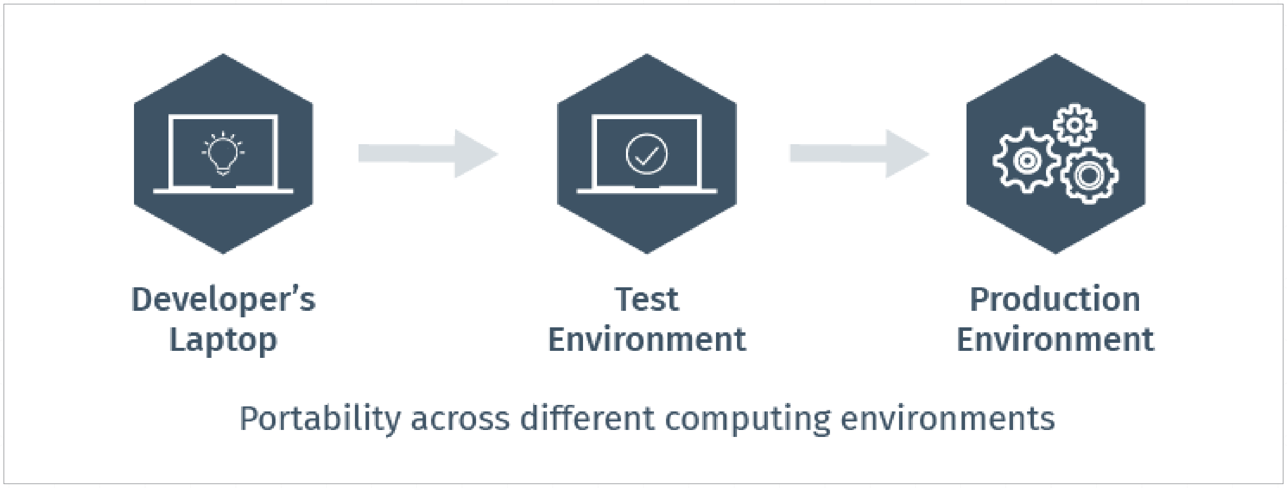


Figure 2. Portability across different computing environments

Docker is the leading container platform

Docker has been synonymous with containers since the release of Docker 1.0 in 2014. Docker has become the de facto container technology standard because of its portable format and easy-to-use developer tooling.

Fittingly, nearly three-quarters of organizations are using or evaluating Docker today[1].

75%

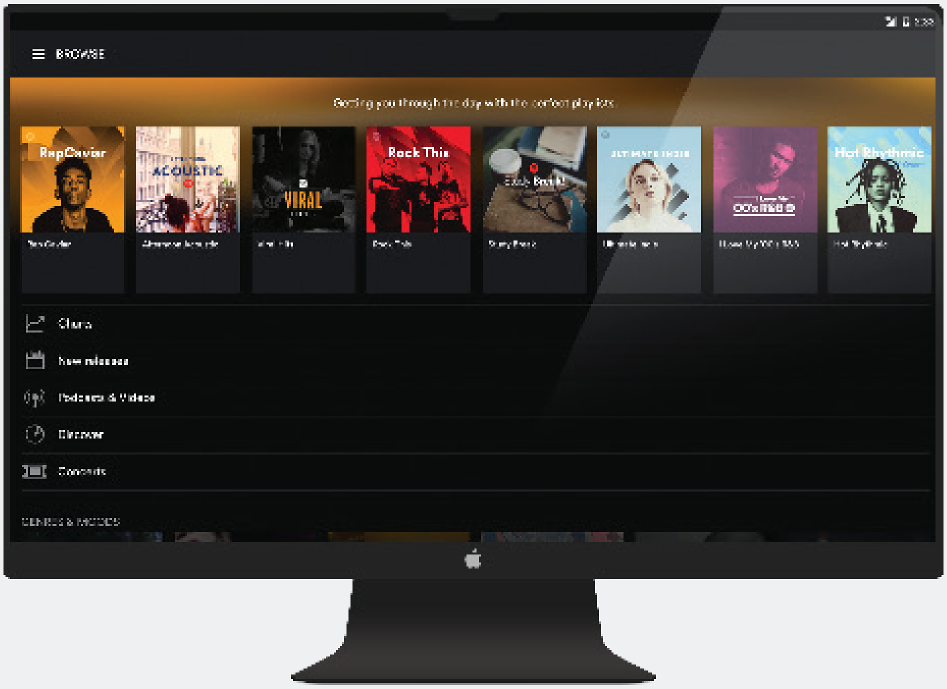
The average size of a 10-month-old Docker deployment has soared 75% over the past year.[2]

3.5M

There are now over 3.5 million applications in Docker containers and over 37 billion Docker container downloads[3]. What started as a transformative technology in the lab has blossomed into a production-ready platform for velocity and efficiency.

We live in a world where business is written in code, where all companies are software companies. This means DevOps teams are wielding more influence on key business and technology decisions. Rapid container adoption is an outgrowth of this phenomenon.

Did you know Spotify uses containerized apps?



Here is an example of how organizations are using containers. In this scenario, Spotify relies on containerized applications to spin up new computing resources more quickly and efficiently to accommodate additional customer demand. Containers help to ensure high levels of service when Spotify's customers add new product features and simultaneously access streaming services. Containers are then turned off when customer traffic wanes.

Has the DevOps dream spawned a security nightmare?

So, with DevOps and IT blissfully on board with containers, where does that leave security?

Traditional vulnerability management is keeping security in the dark

The truth is, containers enable DevOps and quicken time-to-market, but they also create a major Cyber Exposure gap. Traditional vulnerability management approaches can’t easily secure containers. Here’s why:

Containers have short lifespans. The average lifespan of containers is only 2.5 days (with many deployed for mere minutes!)[4]. Unless security teams are constantly scanning their environment using extremely tight scan ranges, they won’t be able to detect containers until they’re long gone – which is too late.

Containers are hard to assess. Containers usually lack an IP address and a login for credentialed scans. So, it’s difficult to use traditional vulnerability management techniques to identify threats or misconfigurations. Alternative approaches such as placing agents in the host OS or relying on privileged containers running inside the host have downsides, too. They drive up compute costs and create larger security issues that require additional controls.

Container remediation is different. Containers are part of the immutable infrastructure revolution, which describes a new mindset of replacing, not changing, infrastructure during operations. To address container vulnerabilities, security and DevOps teams must stop the running container, fix all issues directly in the container image, and re-deploy. Immutable infrastructure has turned vulnerability management on its head.

The takeaway? There’s an increasing gap between what legacy security products provide and what’s actually needed to protect containers, cloud-native IT architectures and the rest of the new digital infrastructure in today’s modern attack surface.

And guess what? That DevOps' sandbox is teeming with known vulns

When it comes to container security, traditional vulnerability management approaches just aren't doing the job. To make matters worse, there's another pesky problem.

To shorten time to market, developers often assemble (versus develop) applications using open-source components and frameworks. But, it turns out these open-source building blocks are bursting with known vulnerabilities.

Let's look at Docker Hub, a popular cloud-based container registry hosting 2 million Dockerized applications, for sharing and uploading images[6]. According to Tenable Research, there are an average of 40 vulnerabilities per container image shared by the community[7]. Even among official Docker images maintained by the company, the average number of vulnerabilities per image is 16[8]. That's one treacherous sandbox!

> "Modern applications are assembled, not developed. Every piece of code produced by the development organization should be analyzed to build a detailed list of every element used to build the application. Most cloud-native application teams use a large amount of open-source software, and understanding exactly what components and versions are within an application is critical."[5] — Gartner

OK, it's been a nightmare, but it's all about to change

Container invisibility, their unchanging nature and vuln-laden open-source building blocks – we've examined many reasons why container security is no easy feat. In fact, it's so daunting, it's hard to know where to start.

Ask yourself: Would any sane cybersecurity leader pick up a random, unverified USB flash drive and use it in their data center without scanning first for vulnerabilities and malware? Zero chance, right? But, this type of behavior happens in the world of containers and DevOps all the time. Recent research suggests that of organizations with containers in production, fewer than 20% perform any image scanning[9]. Yikes!

Maybe it's because we're so often stuck in an endless game of Whac-A-Mole: Spot issue. Fix issue. Spot issue. Fix issue. Constantly identifying and remediating threats to assets in production? This reactive behavior is no match for the high-velocity world of containers, DevOps and continuous innovation.

So, what can we do to wake from this nightmarish ordeal? Begin by shifting left.

> James Ford, chief strategic architect for ADP, summarizes the container threat very simply: "Even if Docker certifies an app as being safe and effective, I'm not **risking $11 billion** on Docker telling me it's safe. We need **extra assurance** and to prove it to ourselves."

Shifting left: How security teams gain visibility

Shifting left is a key DevOps principle – borne out of the need to find and fix software defects as early in the software development lifecycle (SDLC) as possible to reduce operational costs and increase code velocity. The thing is, developers focus on finding bugs and getting their code to run better, but security is not yet part of that process. To them, it's a side issue best handled by another team. As a result, security is not treated as a code quality issue today, and most vulnerabilities are remediated after software has already been shipped to production.

Shifting left with security is a significant change in mindset. For example: rather than conducting gate reviews and penetration testing at the end of a release cycle, integrate security testing programmatically from within the DevOps toolchain. Broaden the focus of existing vulnerability management programs by assessing assets and applications in development in addition to scanning those in production.

In short, security needs to live where developers live – namely the Continuous Integration and Continuous Delivery (CI/CD) build systems for compiling and testing code.

Don't let security operate as a silo outside the development process. This is an opportunity for security to plug into the innovation cycle as applications and services are being created to **proactively prevent** vulnerabilities prior to deployment.

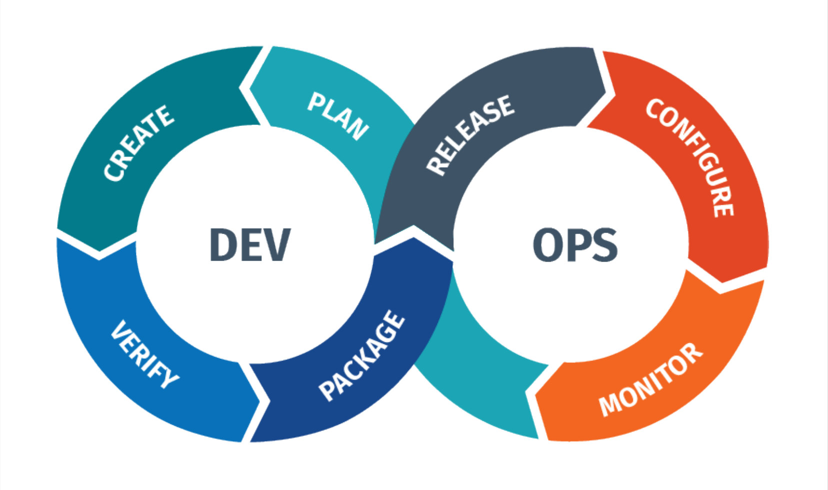


Figure 3. DevOps' CI/CD innovation cycle

> Mannie Romero, executive director, Office of the CISO for Optiv, explains: "If a container has a vulnerable piece of software and the security team is in a security DevOps model, they can work quickly, fix the container image and, likely, the next time that container gets spun up, it is secure. That is a reaction time measured in days, and you don't have to deal with thousands and thousands of systems. **You only have to deal with the golden image of that container.** So there are a lot of opportunities that security teams can take if they embrace the DevOps model."

With containers and other short-lived cloud assets, shifting left is the only way security can manage cyber risk. Protect the asset image, not only the actual asset itself, and ensure all base images used by developers are secure and compliant according to policies.

When new threats are discovered in a running container, the container must first be stopped in order to remediate the issue from within the image Dockerfile. Then the container can be redeployed.

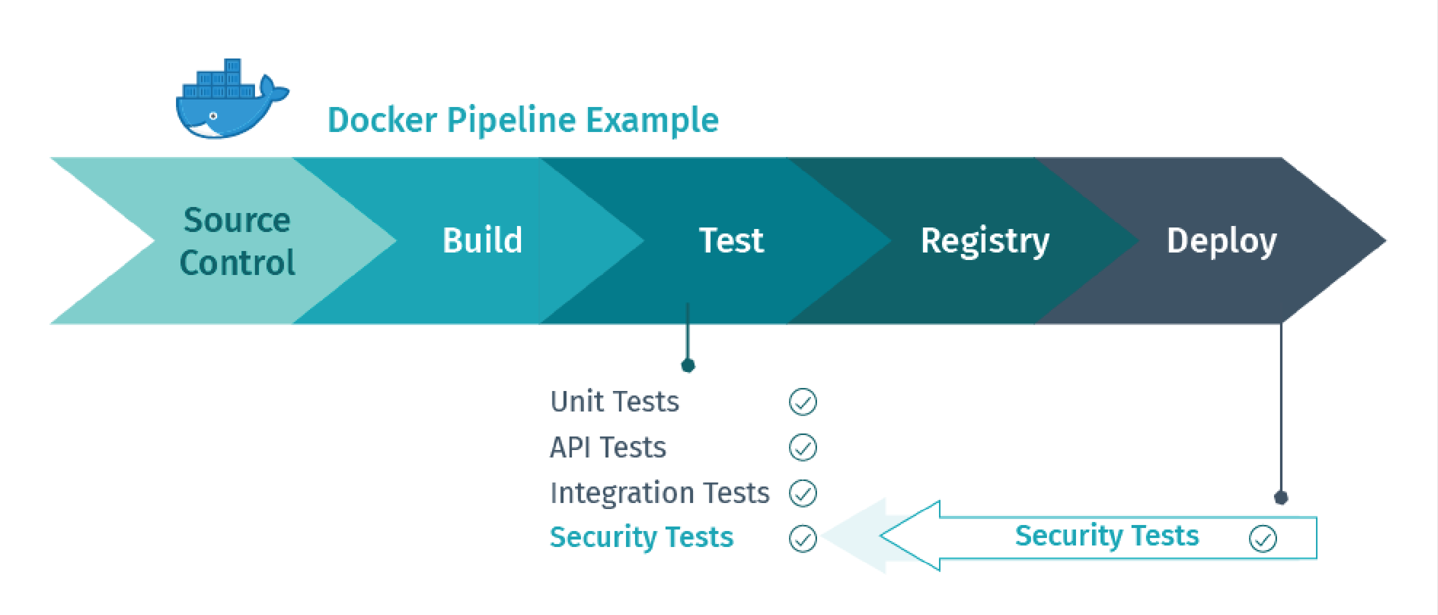


Figure 4. Docker pipeline example

How to master container security in three steps

While shifting left with security is a game-changer for cybersecurity leaders, there are other security considerations as well. Follow these three steps to master container security and manage your Cyber Exposure:

STEP 1 Discover and secure container infrastructure

Before you can secure containers, you need to be able to detect and find them. Gaining visibility into your container adoption is the first step in determining your security posture. To find and secure container infrastructure, you need to be able to:

Detect Docker

Docker detection services, like those available from Tenable® (Nessus® plugins #8595 and 93561), make it easy to detect Docker installs on host systems, and, if detected, enumerate all the running containers on those hosts. Many Docker deployments are operating unbeknownst to cybersecurity, creating a significant blind spot. Docker detection services is an essential plugin and first step in container security.

Patch Docker hosts

Once you detect Docker in your environment, you need to patch Docker host vulnerabilities. Docker containers share the kernel with the host OS, which means kernel-level vulnerabilities now gain a whole-new level of significance on Docker hosts. It's critical to run a comprehensive credentialed patch audit against Docker hosts to ensure they're up-to-date with the latest patches and security fixes. Nessus supports local security checks for a variety of Linux distributions, so regardless of which base operating system you select from a Docker host, there's a good chance Tenable already has support for it.

Harden Docker hosts and Kubernetes systems

Finally, harden your container infrastructure to reduce its attack surface. You can do this by:

- Limiting the number of services running other than the Docker daemon
- Limiting user access to the Docker daemon
- Securely configuring core components of Kubernetes

Take advantage of industry best practices, such as the CIS Docker Benchmark and CIS Kubernetes benchmark, which cover configuration, patching, permissions, access and sprawl. Tenable offers support for both CIS benchmark audits in Nessus, Tenable.sc™ and Tenable.io®.

STEP 2 Shift left with security

After you find running Docker containers, the next step is to integrate security into the DevOps pipeline to provide comprehensive insight into container security risks to address them prior to deployment. Enable shifting left with security via:

DevOps integration

As mentioned, security needs to be an important part of the DevOps process. Just as developers run unit, API and integration testing on software builds, security is another critical QA test before pushing the container image to a registry. The security test needs to be fast – less than a minute – to avoid blocking or disrupting software development workflows. So, how do you do it?

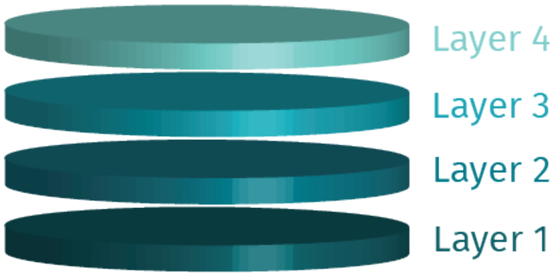
First, take advantage of fully documented APIs to integrate security testing programmatically within your CI/CD build systems, such as Jenkins, Bamboo and Travis CI. Next, be sure you can import and connect to a wide range of container image registries such as Docker Trusted Registry, JFrog Artifactory and Amazon ECR to enable continuous protection of images. We'll discuss the importance of connecting to registries in Step 3 under continuous vulnerability assessment.

Automated inspection

Once security testing is integrated into CI/CD build systems, conduct a complete bill of materials covering all container image layers and components. Gaining visibility into what's inside a container allows you to perform in-depth vulnerability assessments on each container image and assess container image source code for malware. Make sure this inspection happens automatically – without manual intervention from security — each time there's a new build.

PRO TIP Container images often consist of multiple layers of different underlying images. Each layer of an image represents new software functionality, and it's common that higher layers remediate vulnerabilities found in lower layers. Make sure security tests provide layer hierarchy intelligence to identify when security issues in lower layers are automatically mitigated in a higher layer to avoid frequent and costly false positives.

Container Image Layers



Policy assurance

Enterprise policy assurance helps to certify containers are compliant with organizational risk thresholds before accepting the container image. Create container security policies that align to corporate goals and objectives based on overall risk scores and presence of malware. If a container image exceeds the risk threshold, developers must be notified immediately with layer-specific information to help them take direct action to remediate. Policy violations can trigger alert notifications in bug tracking tools or emails or can optionally block specific images from being deployed depending on organization preferences.

STEP 3 Incorporate into a comprehensive Cyber Exposure program

Finally, ensure container security operates as part of a larger Cyber Exposure program protecting your entire attack surface, including traditional IT, public cloud, mobile and IoT. These are the three key capabilities you need:

Integrated security platform

Containers are but one of an emerging category of assets across the modern attack surface that's disrupting current security approaches and techniques. The last thing cybersecurity leaders want is to manage disparate point solutions and multiple tools for different asset classes to protect traditional IT and modern assets. This only creates isolated visibility, excess management overhead and constant reactive firefighting of new threats. Establish a Cyber Exposure platform that provides the most comprehensive visibility and insight and the broadest coverage of assets and vulnerabilities to protect your organization.

Continuous vulnerability assessment

In our evolving technology landscape, new vulnerabilities are identified daily. Respond to new risks quickly by continuously monitoring a wide range of external vulnerability databases looking for new threats. Automatically retest all your container images in your registries against any new vulnerabilities. If the newly identified vulnerability is present, provide the vulnerability and remediation details developers need, so they can push new container images and secure their applications.

Container runtime protection

Scan running containers to gain visibility into which container images are running in production and view important container information related to metadata, image changes and runtime vulnerabilities. Also, automatically detect and assess new containers running in production that have not yet been tested for vulnerabilites and malware.

As previously mentioned, containers are designed to be immutable – or unchanged – from development into operation. And while immutable infrastructure gives us some assurance that deployed containers are just as secure as when they were tested in development, you need to verify immutability and be notified immediately when containers change during runtime.

Container runtime protection helps you understand the Cyber Exposure of containers in production and extends security controls integrated into the development process.

Container security benefits cybersecurity and DevOps

Securing containers across the entire SDLC is a huge win for both cybersecurity teams and DevOps.

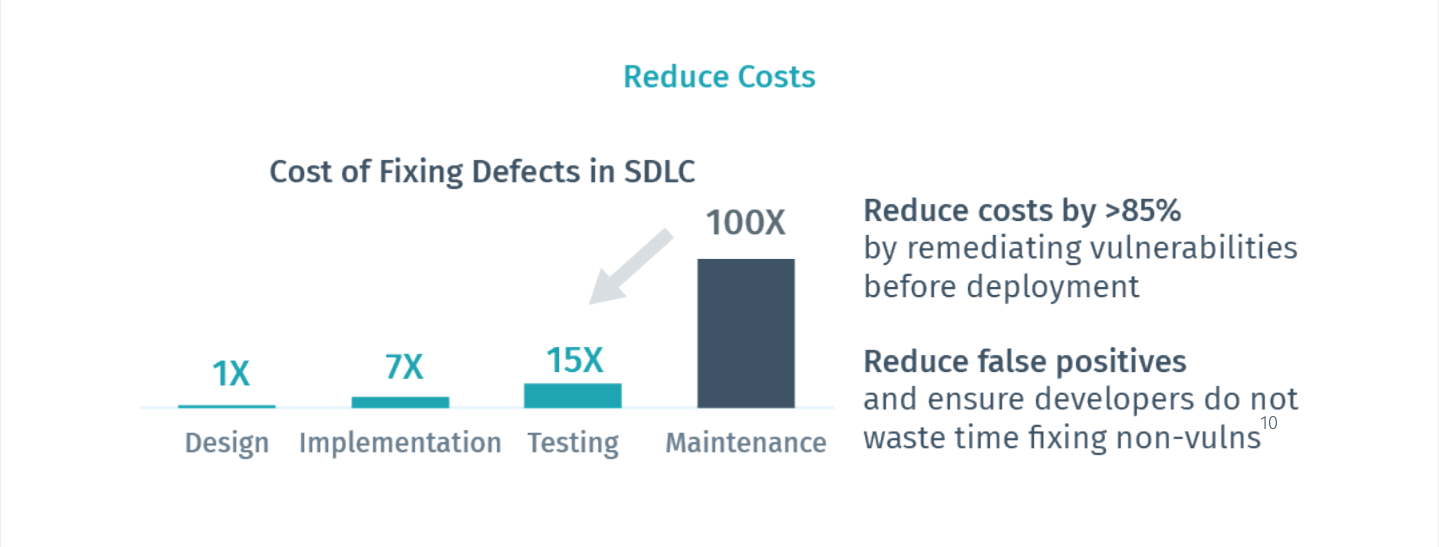
For cybersecurity teams:

Drastically reduce operational costs

Research indicates it costs **85% less** to fix software defects before production compared to after the software has been implemented[10]. In fact, costs to fix software bugs increase exponentially as they are discovered later in the SDLC as software moves from design through maintenance. The difference in remediation costs is due to the increased complexity of implementing changes during production, from identifying application owners to issuing counter-changes to offset functionality modifications. Remediating vulnerabilities before deployment is critical to lowering overall security administration and labor costs.

Eliminate blind spots and excessive cyber risk

You can't secure what you can't see, and poor visibility is a major challenge with containers. Container visibility is essential because of the widespread proliferation of known vulnerabilities in the Docker ecosystem and within open-source software used to build container images[11]. Gain insight into vulnerability, malware and policy compliance for all container images used in production and get peace of mind with your security posture. Imagine seeing and preventing potential container risks before they're deployed.



Reduce Costs

Cost of Fixing Defects in SDLC

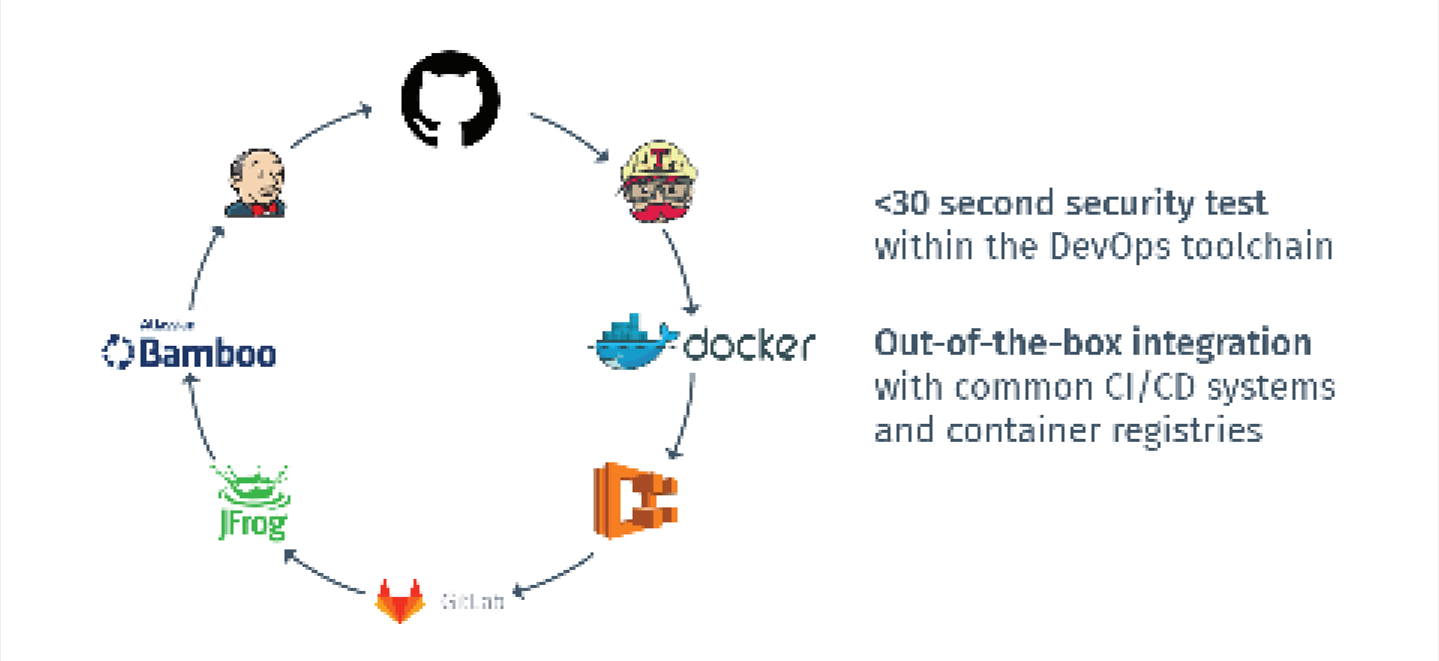
Reduce costs by >85% by remediating vulnerabilities before deployment

Reduce false positives and ensure developers do not waste time fixing non-vulns[10]

Figure 5. Reduce operational costs by catching container vulnerabilities earlier

Accelerate DevOps to maintain velocity

Tired of being seen as too slow, too inflexible and too late? Container security using secure DevOps principles can help flip this mindset among line of business leaders, IT operations and developers and show that cybersecurity can operate at DevOps speed. Container security tests integrated into CI/CD systems can take as little as 30 seconds to complete.



<30 second security test within the DevOps toolchain

Out-of-the-box integration with common CI/CD systems and container registries

Figure 6. Accelerate DevOps with fast container security testing and easy integration with CI/CD systems

For DevOps teams:

Generate higher-quality code

Integrating vulnerability and malware testing into CI/CD systems gives developers confidence their code has fewer defects. Developers can pinpoint security risks and fix vulnerabilities more easily with specific remediation advice.

Release software faster

At sub-30 seconds per test, security can operate at the speed of DevOps without blocking or disrupting current workflows. Solutions with container layer hierarchy intelligence capabilities increase code velocity even further by drastically reducing time-intensive false positive results.

Gain personalized visibility

DevOps teams can each get their own tailored dashboard that provides vulnerability metrics for their specific images and repositories. Developers can also be notified about new threats via email or issue tracking and ticketing systems for streamlined communication.

Get started on your container security journey with Tenable

Fortunately, you don't need to manage separate security tools just for your container assets. As the Cyber Exposure platform, Tenable helps you manage and measure cyber risks of containers and the rest of your attack surface – essentially, any asset on any computing platform.

Ready to secure your container environment, spanning container hosts, images, development pipelines and even the web applications increasingly running inside Docker containers?

Tenable offers multiple applications to meet your organizational requirements, such as:

Tenable.io

Detect containers running in your environment and assess the underlying container infrastructure for cyber risks, including vulnerabilities and misconfigurations. Bring clarity to your security and compliance posture across traditional IT, cloud infrastructure, mobile devices, IoT assets and containers. Learn more about Tenable.io.

Tenable.io Container Security

Deliver end-to-end visibility of Docker container images, providing vulnerability assessment, malware detection and policy enforcement across the SDLC – from development through operations. Provide proactive visibility to solve the security challenges of containers at the speed of DevOps. Address new cyber risks that emerge after container deployment, such as securing rogue containers that have not yet been assessed and alerting security teams immediately when containers are modified during runtime. Learn more about Tenable.io Container Security.

1. Cloud Foundry, "Where PaaS, Containers and Serverless Stand in a Multi-Platform World," June 2018
2. Datadog, "8 Surprising Facts About Real Docker Adoption," April 2017
3. ZDNet, "What is Docker and Why Is It So Darn Popular," March 2018
4. Datadog, "8 Surprising Facts About Real Docker Adoption," April 2017
5. Gartner, Reimagining Security and IT Resilience for a Cloud-Native DevSecOps World, Neil MacDonald, May 24, 2018
6. Docker, "About Docker," September 2018
7. Tenable, "2017 Sourcing Container Images from Docker Hosts," July 2017
8. Ibid
9. Anchore, "Snapshot of the Container Ecosystem," April 2017
10. Computer Business Review, "The cost of fixing bugs throughout the SDLC," March 2017
11. Tenable, "2017 Sourcing Container Images from Docker Hosts," July 2017

7021 Columbia Gateway Drive
Suite 500
Columbia, MD 21046

North America +1 (410) 872-0555

www.tenable.com

11/06/18 V01