Twistlock

Guide to

The Proactive Security Paradigm

How Containers Can Revamp Your Approach to Security

OVERVIEW

Ask most DevOps engineers about the benefits of containers and microservices, and their responses will center on the agility, modularity and scalability that microservices-based architectures provide.

But there is another crucial and oft-overlooked advantage of migrating to containers and microservices: Security. Containers offer a number of opportunities for building and deploying more secure applications and environments.

That is because, when you migrate from legacy environments to a containerized environment, you gain the opportunity to establish a new security paradigm—one that is based on a proactive rather than reactive approach to preventing intrusions. Containers allow you to secure applications and environments at a more granular and nuanced level. They also empower you to identify and resolve potential security threats before they disrupt your workflows, rather than having to adopt a defensive stance oriented around addressing attacks once they are fully underway. And they make it possible to combine static analysis with machine learning in order to automate runtime defense and enforce policies across your environment.

This whitepaper explains what a proactive approach to security for a microservices-based environment looks like, what makes it radically different from traditional security strategies, and how your organization can adopt a proactive security stance when it makes the migration to containers.

THE TRADITIONAL WORLD OF SECURITY

Security threats have evolved significantly over the past two decades. The viruses and worms of yesteryear, whose consequences in many cases amounted only to annoyances, have been replaced by a new generation of threats that target critical infrastructure and can truly disrupt lives.

Security tools have changed, too. Passive antivirus scanners have been replaced by real-time intrusion detection tools. Simplistic firewalls have evolved into fine-tuned perimeter defenses with granular access control definitions.

Yet despite these changes, the fundamentals of security practices have not evolved at many companies. No matter which tools they use or which types of threats they face, organizations today tend to remain mired in a set of inefficient, conventional approaches to security that deliver subpar results in the face of today's threats.

"SILOED" SECURITY TEAMS

At many organizations, security remains the province of a team of security experts. They review code after it is written—or worse, already in production. They work in silos, isolated from the rest of the software delivery team. This isolation (which results in part from the difficulty of integrating security review into monolithic application development) leads to security lapses.

PERIMETER-LEVEL SECURITY

Most security tools still focus on protecting the perimeter of software environments. They harden the network using firewall rules. They lock down servers using access control policies. While practices like these are useful for keeping intruders out, they do not help in the event that an attacker is able to defeat perimeter-level defenses and gain access to the interior of an environment.

ROT-PRONE CONFIGURATION

Another crucial weakness that arises from a reliance on manual configuration is a susceptibility to configuration "rot." As a software environment changes, configurations that are manually updated become outdated—or in other words, they rot.

MANUAL CONFIGURATION & MANAGEMENT

Today's security tools are often capable in theory of real-time threat detection. But because they require manual configuration, their ability to identify and react to threats in real time is limited. If you have to configure security definitions manually to find threats, you will not be able to detect threats quickly.

That's what the old world of software security looked like. Now, let's take a look at the brave new world of microservices and containers—and the security practices it has in it…

THE SECURITY BENEFITS OF CONTAINERS

At first glance, containers may not seem especially secure. Greater degrees of abstraction and complexity often run contrary to security goals. Because a microservices application has more moving parts and greater complexity than a monolith, configuring security policies and tools effectively may seem that much more difficult.

Yet containers actually create opportunities to build much more secure environments and applications. By taking advantage of properties unique to containers, you can harden your containerized environment even further in order to build a truly proactive defense.

Consider the following ways in which microservices and containers empower organizations to define a new, proactive security paradigm, and keep applications and the environments in which they live more secure than ever:

- **Containers have a small surface area.** In a traditional environment, you would host your applications on bare-metal servers or in virtual machines. In either case, attackers could take advantage of vulnerabilities in the operating system or other high-level parts of the environment in order to control your application. With containers, however, the attack vector is much narrower. Because a well-designed container image includes only the minimum amount of code and utilities required to run a particular microservice, the number of potential security holes is much smaller.
- **Containers are transparent.** Virtual machines are built with large, unwieldy disk images. Inspecting them requires digging through reams of components or manually auditing running environments. With containers, however, you get a much higher degree of transparency. With a quick look at a Dockerfile, any DevOps engineer can identify which components exist inside a Docker image. Images can be scanned automatically inside registries in order to detect vulnerabilities, and changes to images are constantly recorded. Sockets, exposed ports, and so on are simple to identify by examining Dockerfiles or using simple tools like the Docker inspect utility.
- **Containerized environments are consistent and predictable.** Containers provide environment parity by creating a homogenous environment for writing, testing and deploying applications. As a result, security teams can review applications before production deployment with the confidence that the application will behave the same way once it is in production.
- **Containers eliminate silos.** The environment parity that containers offer also makes it easy to eliminate the silos that separate security from the rest of your DevOps team. Your security pros can run tests and review code at any point in the software delivery chain using the same container images that developers and IT Ops are working with. No special coordination is required. With containers, security becomes simple to integrate directly into the rest of your software delivery chain.

- **Containers enable immutable infrastructure.** When you want to update a containerized application, you simply rebuild the image for the microservice you wish to change, then redeploy it. You don't have to redeploy the entire application, and your outdated containers disappear forever. This makes your infrastructure immutable. From a security perspective, immutable infrastructure is tremendously advantageous, especially when combined with machine learning. Because it is not possible to patch or update containers when they are running, it is easier to detect anomalies. Any behavior by a container that was not part of the original profile can safely be assumed to be an aberration and a possible emerging threat.
- **Containers enable automation.** Another benefit of immutable infrastructure is its ability to enable a high degree of automation. With immutable infrastructure, you can make changes quickly and seamlessly, without relying on human operators to make complex judgements about what to change when modifying a running application.

Machine learning also allows you to profile applications automatically and model what containers should and should not do—and in this way, it helps you to detect anomalies.

In all of these ways, properly managed containers make applications much more secure, not less.

THE NEW WORLD OF SECURITY:
USING CONTAINERS TO REDEFINE THE SECURITY PARADIGM

4 KEYS TO THE NEW WORLD OF PROACTIVE SECURITY

Perimeter-Level Security

Rot-Prone Configuration

"Siloed" Security Teams

Manual Configuration And Management

By extension, containers make it possible to adopt a fundamentally new approach to security. Rather than relying solely on perimeter-level defenses and waiting for intrusion attempts to start before reacting, you can minimize security vulnerabilities within your production environment and prevent intrusions before they even begin.

The small surface area of containers minimizes the opportunity for attackers to find a vulnerability to exploit. The transparency of containerized applications, combined with environment consistency, makes it much easier for your security experts to become part and parcel of the software delivery process and identify potential security gaps at any level (within application code itself, in environment parameters, or in the host server) before software is released to production.

More critically, this integration enables automated quality control that allows security teams to set up rules that prevent vulnerable or misconfigured images from ever progressing through the dev pipeline. Environment parity also helps to ensure that software that is deemed secure during pre-production testing can be trusted to remain secure once it is in production.

That's not all. In addition to helping to prevent opportunities for attack, microservices architectures and containers also enable organizations to be proactive about responding to attacks once they are underway. In the old, reactive world of security, a response to an intrusion would involve assessing how the attack occurred, then plugging up whichever hole facilitated it.

Under the new, proactive paradigm, organizations can mitigate attacks instantly. Using immutable infrastructure and machine learning, security policies can be updated constantly so that attacks are detected in real time. With the same resources, responses can also be automated in order to stop an intruder in his tracks, rather than waiting until the damage is done to shut him out.

Last but not least, by leveraging static analysis and machine learning to automate security policy creation, containers obviate the need for developers and security analysts to collaborate manually on security policy. For this reason, containers enable a much greater degree of scalability by removing dependencies on human actors. They also help to prevent configuration rot by ensuring that security policies are updated automatically, in real time.

All of the above means that when you migrate to microservices and containers, your security strategy no longer boils down to securing your software environment's perimeter and hoping for the best. You can instead adopt a proactive, continuous approach to security that takes advantage of special characteristics that are available only inside a containerized environment.

TAKE THE NEXT STEP TO PROACTIVE SECURITY

While containers and microservices enable new opportunities for adopting a proactive approach to security, organizations must choose to leverage those opportunities if they wish to escape the old world. While containers provide some great foundational security capabilities, extra steps are necessary for ensuring compliance in containerized environments and integrating methodologies like machine learning into the proactive defenses that containers make possible.

Twistlock, a container-first security platform, can help organizations make the jump from traditional to new-world security. Designed as an end-to-end security solution for containers, it hardens containerized environments at all levels—from the host operating system to the container registry to live containers—and integrates seamlessly with continuous delivery pipelines so that security operations can be bricked into the rest of the DevOps workflow.

To learn more, check out Twistlock's Blog and Resource Center, which is full of practical advice for making the most of security in a containerized world.

Ready to try Twistlock? Visit www.Twistlock.com/Get-Twistlock and get a Twistlock Developer or Enterprise license today.

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