LEARNING MADE EASY

Container Security for dummies

A Wiley Brand

As organizations accelerate modern app deployment, the adoption of containers and container orchestration platforms, such as Docker and Kubernetes, is on the rise. The container ecosystem can be difficult to understand given the plethora of new, esoteric tools and the unique problems they solve when compared to traditional platforms. At the same time, the rapid adoption of container technologies creates a unique opportunity to shift security left and build bridges between development and security teams.

A Brief History of Modern Computing

Modern computing history can be characterized in three distinct waves (see Figure 1). The first wave, during the 1990s, was defined by client–server architectures running on bare metal servers running a single operating system (OS) and typically a single application. The second wave began with virtualization disrupting the server market in 2001 and ushering in the age of virtualized computing. We're now in the third wave of modern computing, defined by cloud and container technologies. Container adoption is a result of two factors: a demand for accelerated time-to-market enabled by DevOps, and a desire for application portability across clouds.

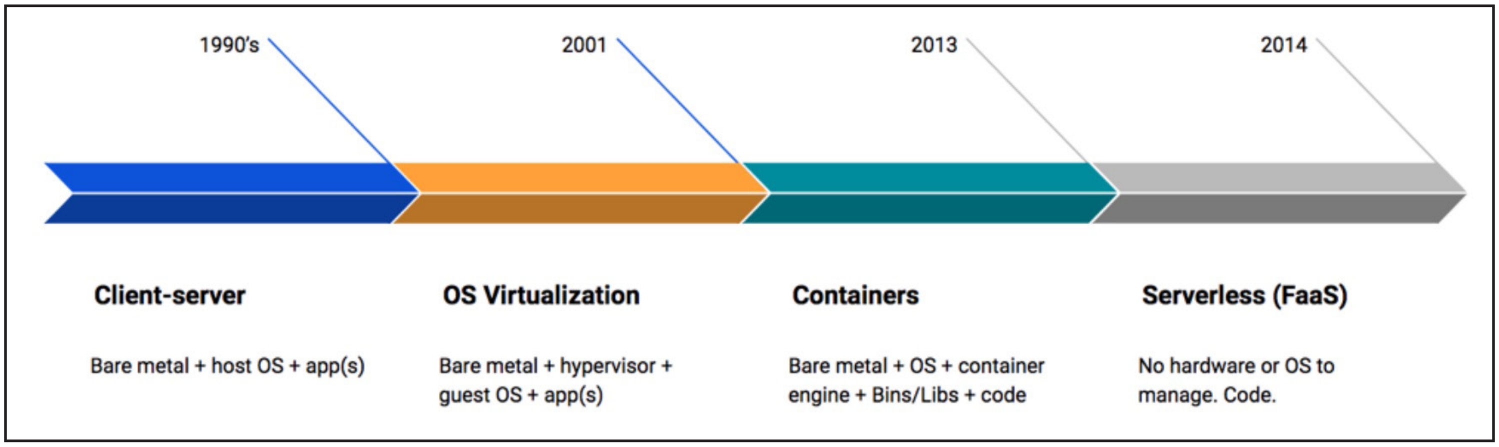
1990's
2001
2013
2014
Client-server
Bare metal + host OS + app(s)
OS Virtualization
Bare metal + hypervisor + guest OS + app(s)
Containers
Bare metal + OS + container engine + Bins/Libs + code
Serverless (FaaS)
No hardware or OS to manage. Code.

Figure 1: The evolution of modern computing.

TIP

A fourth wave is now forming but not yet widely adopted in the enterprise: serverless or Function as a Service (FaaS). This fourth wave completely abstracts compute through a hardware and OS-agnostic model. The most recognizable implementations of FaaS today are Google Cloud Functions, Azure Functions, and Amazon Web Services (AWS) Lambda.

Cloud and Containers Go Together Like Peanut Butter and Jelly

The public cloud and containers are intrinsically linked, yet many security teams mistakenly attempt to address container security separately from cloud.

Over the years, developers have grown tired of dealing with OS and application dependencies. Containers address this issue but create a whole new set of security challenges. Market demand has grown quickly and point security products — from commercial to open source — have sprung up. These point products narrowly address some container security challenges, but they completely miss the bigger picture. Most apps developed on containers utilize a mix of Platform-as-a-Service (PaaS) resources like AWS Redshift, Google Cloud Platform (GCP) Cloud Datastore, and Azure SQL (Figure 2 illustrates this point). Without complete visibility into your cloud's application programming interface (API) layer, which knows exactly which

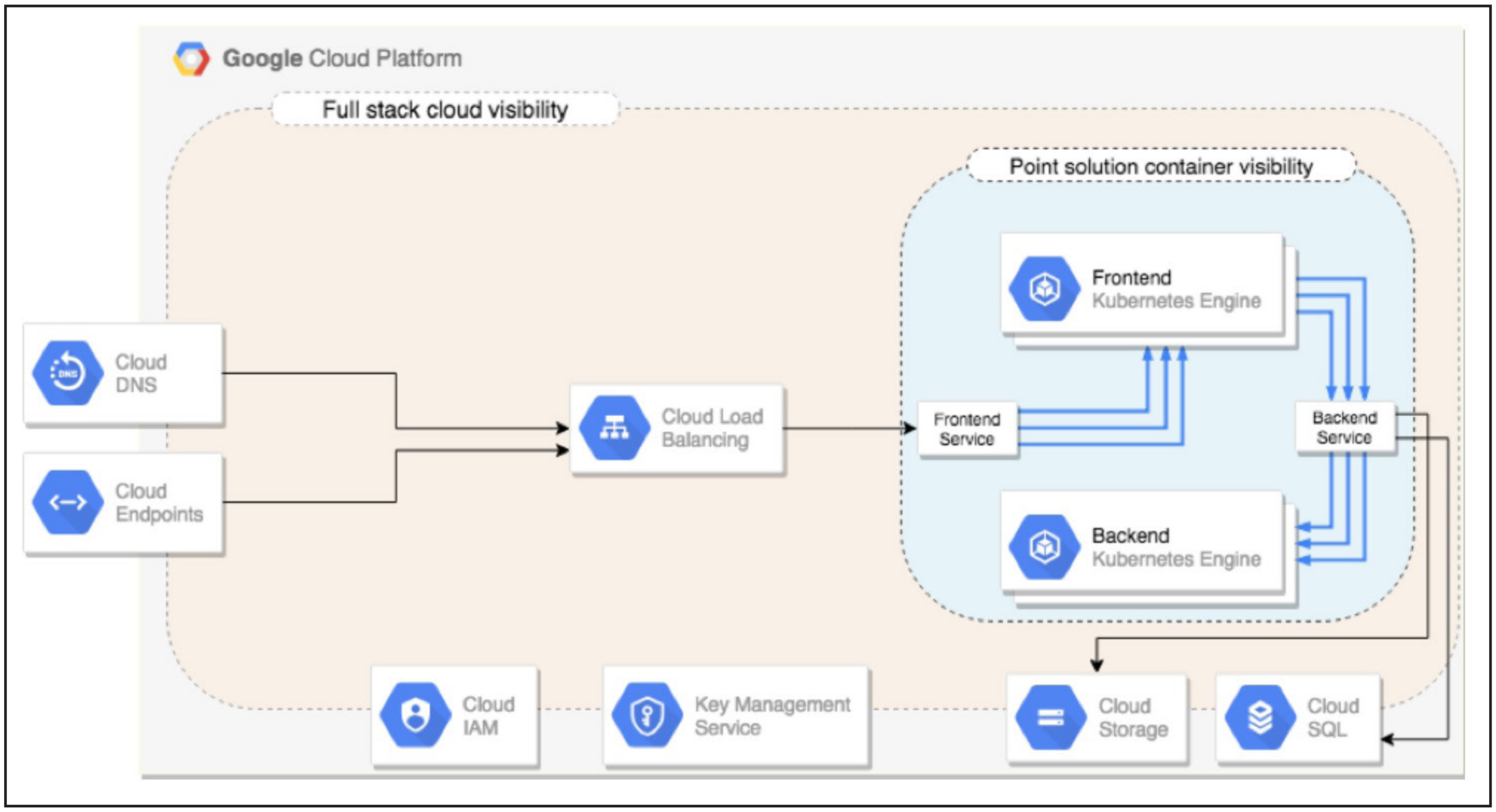

Figure 2: Full stack cloud visibility versus siloed container visibility.

cloud-native services are in use, how can you accurately assess and effectively mitigate the risk that containers introduce in your enterprise?

Consider a scenario in which the latest Common Vulnerabilities and Exposures (CVE) bulletin identifies a vulnerability in your containers, but your AWS security group is not open on the port required to exploit the vulnerability. Having this full stack security knowledge dramatically changes the risk equation but would otherwise be missed by container security point products. Why? Because they often do not have visibility into the cloud provider's all-important API layer. Full stack knowledge allows you to address this vulnerability later and instead focus on other vulnerabilities that are publicly exposed and, thus, more likely to be exploited.

Embrace DevOps and Embed Security In It

Given the speed and velocity at which containers and cloud operate, DevSecOps is the only viable path forward for security teams. DevSecOps brings DevOps and security teams together and introduces

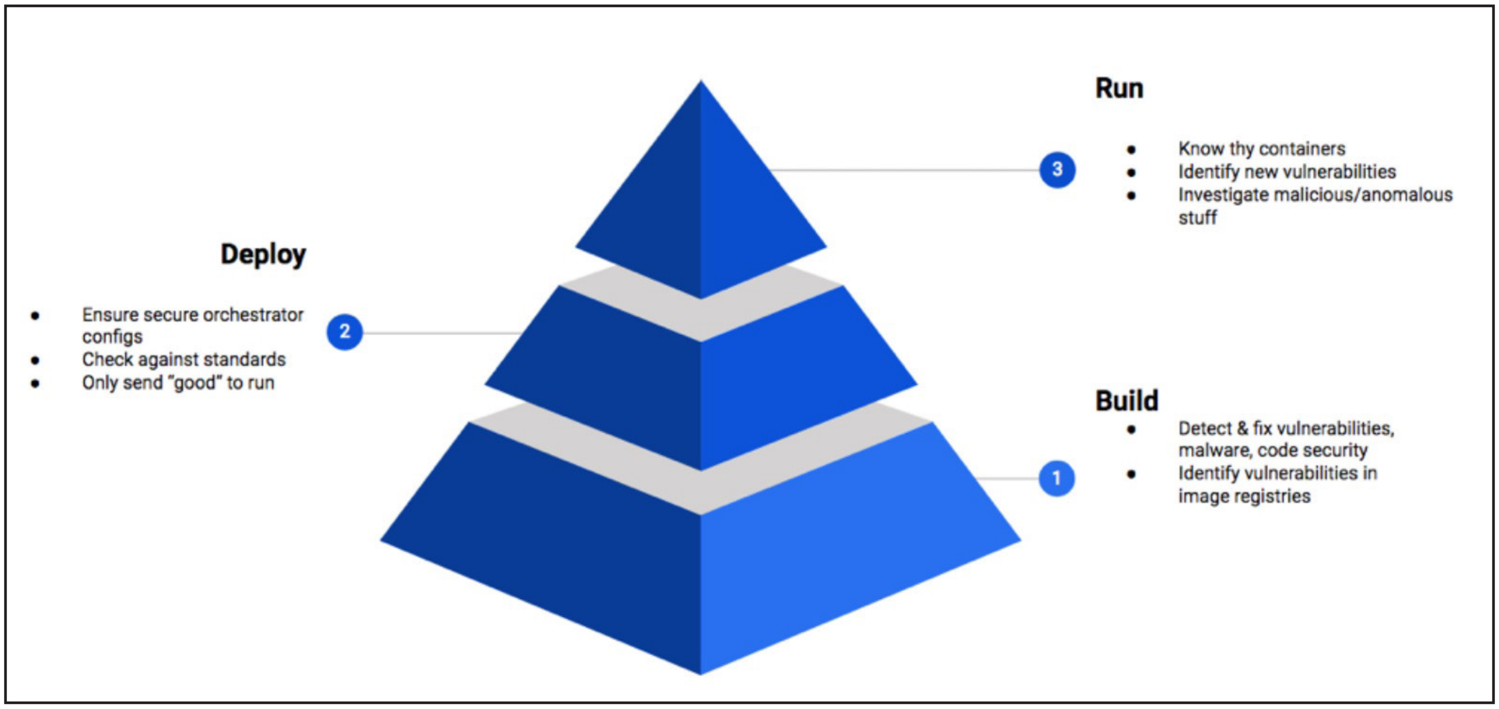
Run
3
Know thy containers
Identify new vulnerabilities
Investigate malicious/anomalous stuff
Deploy
2
Ensure secure orchestrator configs
Check against standards
Only send "good" to run
Build
1
Detect & fix vulnerabilities, malware, code security
Identify vulnerabilities in image registries

Figure 3: The container security triad — embedding security across the entire container life cycle.

security as early as possible in the container life cycle. This approach embeds security across the entire container life cycle: build, deploy, and run (see Figure 3).

Build security

Integrating security in the container build phase means you're introducing security checks early on during the build phase instead of reactively introducing security at runtime. Build phase security should focus on removing vulnerabilities, malware, and insecure code. Because containers are made up of libraries, binaries, and application code, establishing an official container registry for your organization is critical. Undoubtedly, one or more such registries already exist. It's the security team's job to find the registries and quickly gain buy-in around getting access and setting security standards. The main goal of identifying and creating a standard container registry is to create trusted images. A process needs to be agreed upon and automatically enforced in which no container can be deployed from an untrusted registry.

Deploy security

In the deploy phase, the focus shifts to making sure your teams are putting things together correctly. You can have an image that is free of vulnerabilities, but if it's deployed to an insecurely configured Kubernetes pod, you haven't sufficiently managed your container risk. Deploying to a secure configuration can be achieved by adopting a security standard for both your orchestration and container engine of choice. Don't forget to put the necessary processes and tools (that is, guardrails) in place that will enable you to automate and monitor. The Center for Internet Security (CIS) has done an excellent job creating security benchmarks for both Docker and Kubernetes. They should be your starting point. If you get deploy security right, only the "good" should be making it into runtime.

WARNING

Palo Alto Networks' Unit 42 threat research found that 46 percent of organizations accept traffic to Kubernetes pods from any source. In the on-premises world, this is equivalent to deploying a server and then leaving it open "any any" to the Internet. You wouldn't do this on-premises, so why are almost half of all organizations doing it in the public cloud?

Runtime security

Runtime security is about identifying new vulnerabilities in running containers and knowing what normal looks like. It also involves investigating anomalous and suspicious activities that could indicate zero-day attacks. If your security team is involved from the beginning (during the build phase), getting runtime security right is much less complex. If you're coming late to the game and you're reactively focusing on runtime, its best to work backward. Yes, it's important to make sure that the end state is secure, but if your focus is narrowly on runtime, the same issues will likely be repeated.

WARNING

The IBM Systems Sciences Institute found that the cost to fix a bug during the maintenance phase (that is, runtime) was 100 times more costly than a bug fixed during design.

Taking a Holistic Approach to Cloud and Container Security

The most recent Portworx Annual Container Adoption Survey shows that 87 percent of enterprise IT professionals are using container technologies and 90 percent of those containers are deployed in production. Thus, container security must be addressed as part of a holistic enterprise cloud security strategy. It's always tempting to bolt on yet another point security product, but the most mature organizations see containers as an essential component of their cloud infrastructure. Addressing container and cloud security separately will leave organizations blind to risks that an otherwise integrated strategy would address. The most successful enterprises treat cloud and containers as one and the same when it comes to security.

TIP

Check out the following resources to learn more about container security:

- Secure DevOps
- *Cloud Security & Compliance For Dummies*
- Palo Alto Networks Vulnerability Scanner: Take advantage of this open-source tool to start scanning and identifying vulnerabilities in your containers.

For Dummies is a trademark of John Wiley & Sons, Inc. ISBN: 978-1-000-00000-0