aqua

Debunking the

Top Seven Cloud Native Security Myths

The Promise, And Real Challenge, Of A Cloud Native Journey

McKinsey published an article this last October, aiming to debunk the 7 top myths about cloud. We were inspired by the concept and the content, as there is no shortage of myths in **cloud native security** – a world characterized by auspicious beginnings, early adopters, and fast-moving trends. A growing number of organizations are successfully using cloud native methodologies for business-critical applications, while others are just getting started or thinking about it.

Adoption is buoyed by the promise of business benefits like reduced deployment times, scalability, portability and storage utilization, as in the following examples:

Speed to market allows businesses to react faster to change. “The retailer Gap Inc. used cloud-native application architectures for price optimization and can now handle 6,000 price adjustments every four hours.”

Less was spent on infrastructure maintenance and more was invested in innovation when The US Air Force was able to completely inverse the ratio of budget spend on R&D versus maintenance of existing hardware from 30/70 to 70/30.

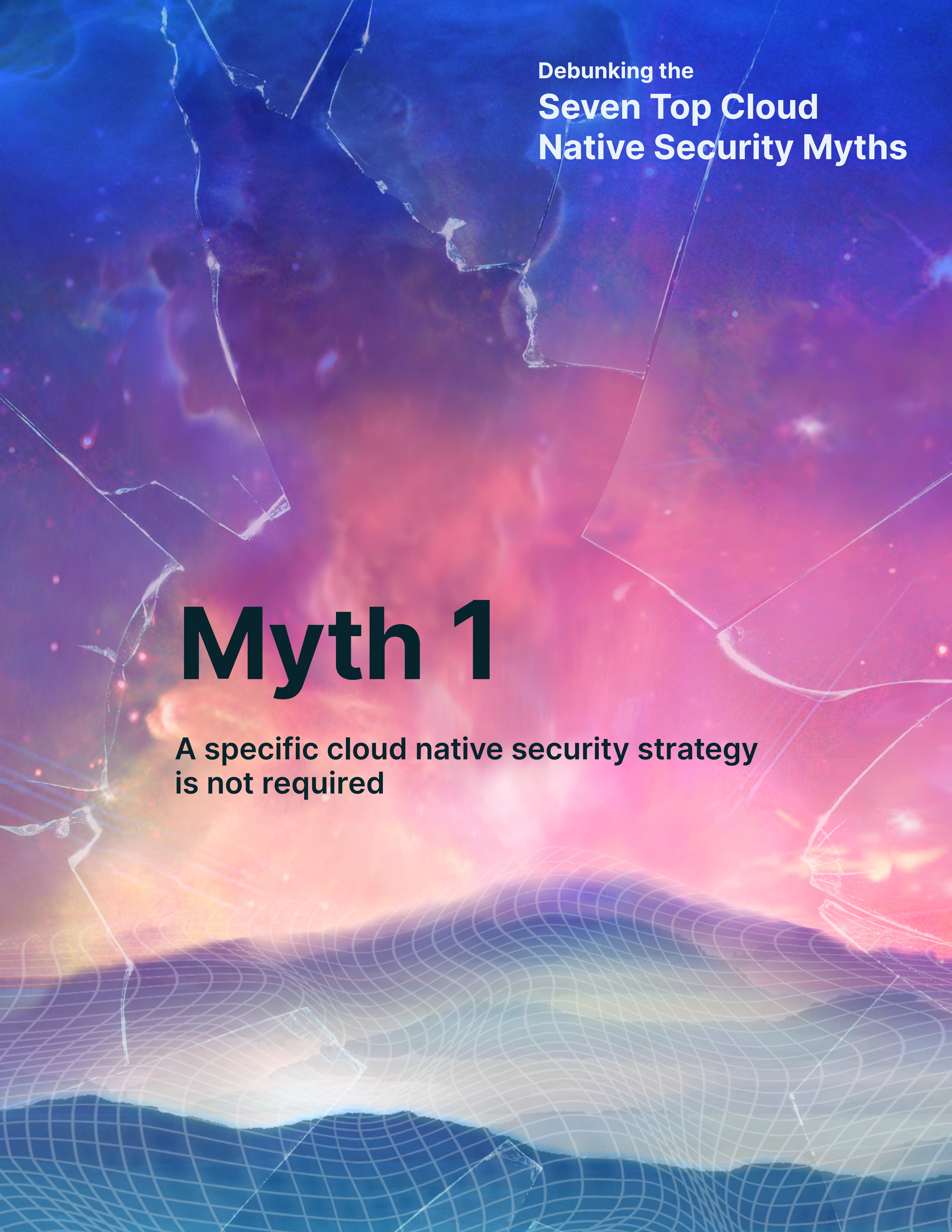
Beginning With the End in Mind

Marcus Aurelius famously stated, “Even the smallest things should be done with a reference to the end.” To make quicker gains toward the mindset shift required for a successful cloud native journey, cloud native security - which is generally thought of after or at the end of a cloud native deployment - should form a critical part of early exploration and research into cloud native requirements.

Even though security is generally not the first element put in place in a cloud native deployment, it is one of, if not the top, concern for teams embarking on a cloud native journey. Effective cloud native security is part of the architecture, design and functionality of the environment itself. So early understanding of what effective cloud native security entails can illuminate the cross-team collaboration challenges required in a ‘cloud native mindset’ and architecture, helping all teams draw realistic expectations and boundaries around the journey ahead.

> Confucius famously said, “By three methods we may learn wisdom: First, by reflection, which is noblest; Second, by imitation, which is easiest; and third by experience, which is the bitterest.” Maybe the bitterest mistakes of others can be avoided.

Where mindset matters and the technology is constantly changing, learning about others’ false cloud native security assumptions provides critical context around what could otherwise appear to be an insurmountable challenge.



Myth 1

A specific cloud native security strategy is not required

A Specific Cloud Native Security Strategy is Not Required

Improved security is one of the most highly touted benefits of a bona fide cloud native strategy, because of concepts like immutability and the separation of discrete computing components in containers, serverless functions, microservices, and cloud infrastructure. Unfortunately, there is a basic misconception that cloud native applications are more secure by default, without additional security controls in place.

For example, it is convenient to assume that containers 'contain' and it's impossible for them to access other containers and the OS on which they run. But this is a general misunderstanding of the concept of root versus privileged. Running a container with root privileges potentially gives access to all the resources on the host, so an attacker could essentially take over the host. To protect against the dangers of an attacker having access to the host itself, privileged access must be controlled before the container is deployed, and re-checked using cloud native runtime enforcement capabilities (See Myth #4 for more on Runtime protection).

Getting security right is absolutely crucial to the business, and the misconception that a cloud native approach can simply be secure by default or by virtue of traditional security tools (see Myth #2) confuses the very positive and optimistic truth. Embedding cloud native security into cloud native initiatives can indeed make applications and infrastructure more secure than ever before, but a cloud native deployment without a security strategy does not necessarily enjoy additional protection.

Myth 2

Traditional security products are effective for a cloud native environment

Traditional Security Products are Effective for a Cloud Native Environment

Traditional security tools are not adequately nuanced – and sometimes entirely unsuitable - for the specific needs of a cloud native environment. For example, host-based intrusion prevention systems (HIPS) monitor the OS, or Operating System, for exploits and vulnerabilities, while they lack both the visibility into cloud native workload activity as well as the ability to monitor and control the same OS/host workload's traffic.

Firewalls are accustomed to viewing the host as an unchanging IP address, but in a cloud native architecture, IP addresses are dynamically assigned to Pods within a cluster in an orchestrated Kubernetes environment. This means that a classic firewall, or even a VM-based one, would not "know" where those Pods are running, or to which application they belong. And once a Pod is taken down in place and restarted elsewhere, the firewall would not know whether it is the same Pod. If firewall capabilities are required to automatically deny outbound connectivity from, say, a database container (encapsulated in a Pod), a classic firewall will be ineffective.

Why traditional security concepts are not effective in a cloud native environment

Traditional Security	Cloud Native Reality	Cloud Native Security
Discrete, infrequent releases	∞ CI/CD	Shifting left with automated testing
Very little open source	Open source everywhere	Software composition analysis and vulnerability mgmt.
Persistent workloads	Ephemeral workloads	Contextual runtime controls that follow the workload
Permanent address	Orchestrated pods	Identity-based segmentation
Hypervisor or hardware isolation	Shared kernel, obscured OS	Enforce least privilege on each workload

key concept

While the same security concepts (like firewalls or hardening) exist in a cloud native deployment, implementation must match the cloud native architecture, design and process

Debunking the

Seven Top Cloud Native Security Myths

Myth 3

The same methods can be used to achieve compliance in a cloud native environment

The Same Methods can be Used to Achieve Compliance in a Cloud Native Environment

As of the writing of this article, many compliance regulators do not include specific guidelines that relate to the processes or artifacts specific to cloud native environments. Ironically, despite this lack of guidance (or possibly as a result), teams tend to believe, as in Myth #2, that classic controls satisfying compliance requirements in other environments will satisfy the same requirements for a cloud native architecture. We already established the ineffectiveness of classic controls for a cloud native environment. But this does not mean our next step is panic or despair. It means that an informed demonstration of “best effort,” aligned to the concepts laid out by the regulators, is the desired goal.

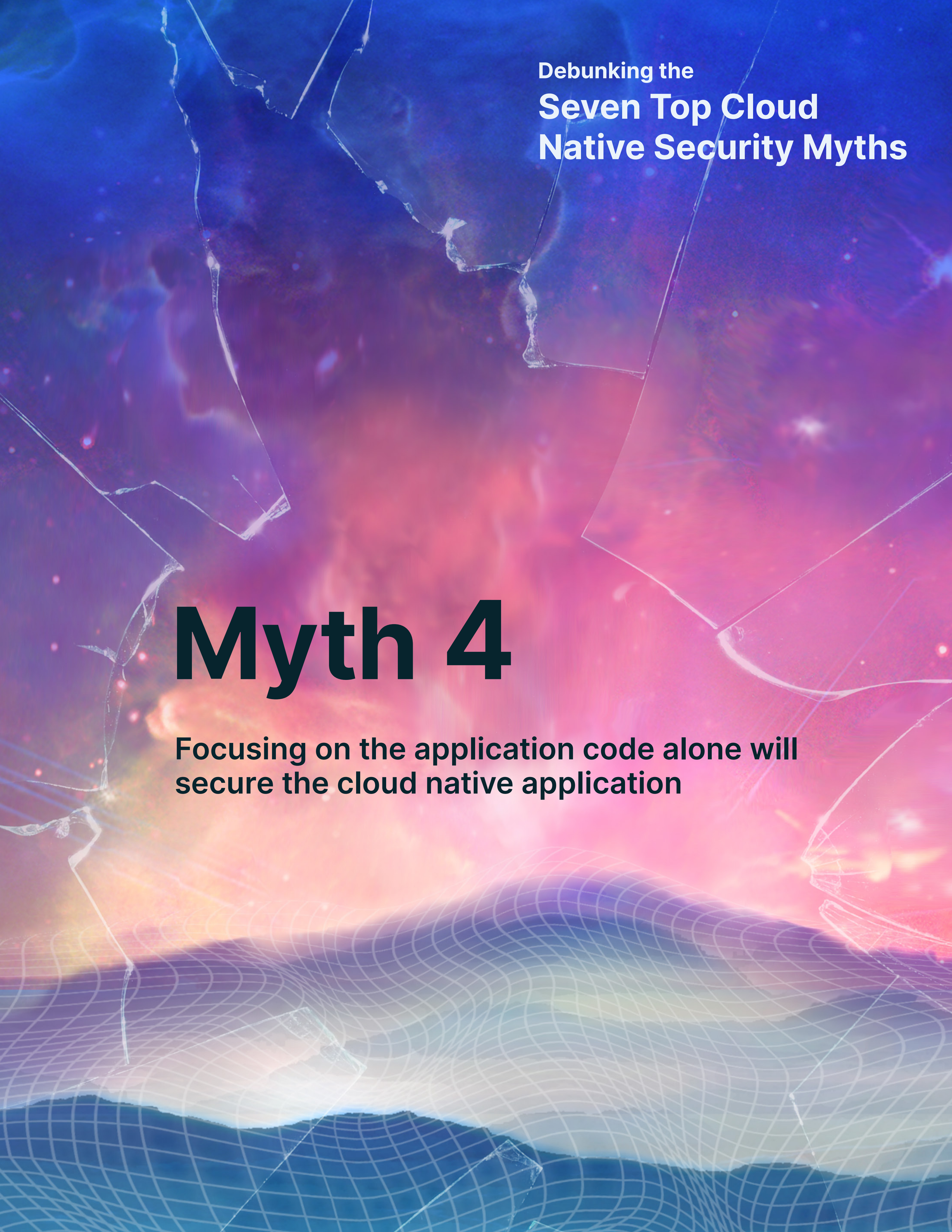
Cloud native environments will generally involve controls for components and layers that are updated separately of one another. For example, where before we might have just hardened a VM and scanned for malware, now it’s important to scan container images, scan and harden the VM, and also scan and harden the orchestrator. And we must monitor and log events, to show proof that the controls are in place.

For a practical example, PCI DSS guidelines require the separation of PCI from non-PCI systems. The guidelines name firewalls, physical access controls, Multi-Factor Authentication, active monitoring and the restriction of administrative access as methods to “provide reasonable assurance that the out-of-scope system cannot be used to compromise an in-scope system component.”

To achieve the level of separation required by PCI DSS in a multi-tenant Kubernetes cluster, we would need to have separate registries and pipelines, divide resources across the separated entities and approved administrators via K8s namespaces, then ensure proper labeling and tagging within those namespaces to further enforce segregation, and RBAC would be required to control access to the runtime and firewall policies used by the security tool to prevent violations of the desired segmentation. It’s all open to interpretation, but if we do that and show proof that these controls are in place, most auditors would be wholly satisfied.

Key concept

Achieving compliance for a cloud native environment, in the absence of specific guidelines, requires demonstration of an informed ‘best effort’ that shows all the appropriate ‘layers’ of a cloud native architecture are being taken into account



Debunking the

Seven Top Cloud Native Security Myths

Myth 4

Focusing on the application code alone will secure the cloud native application

Focusing on the Application Code Alone Will Secure the Cloud Native Application

Checking application code for errors is important, whether it is custom code in a proprietary language or code that is assembled from multiple open-source libraries. Technologies like SAST, DAST, SCA, and IAST all have a purpose and exist for a good reason. But there is more to securing a cloud native application than shifting left and implementing secure code (Myths #5 and 6 will describe the importance of 'shift-left').

For example, the best code in the world will not protect against a scenario where images, which should be immutable and should never change in production, begin executing commands in runtime that were not included in the base image (See Myth #5). Code will also not prevent orchestrator misconfigurations, like failing to disable anonymous access to the K8s API server, which could allow an attacker to take control over an entire cluster and set of nodes. And code will not protect against cloud account misconfigurations that can leave entire cloud services open to attackers, like when Tesla's AWS account was being used to mine cryptocurrency because its Kubernetes administrative console was not password-protected.

Shift-left, as we will see in Myth #5, is a critical concept and mindset shift for Security teams securing cloud native applications. But Security teams must still set policies across the Build, Infrastructure and Workloads.

key concept

Runtime protection and secure cloud account configurations are essential controls for cloud native security, alongside secure code

Myth 5

We will continue fixing development issues in production

Myth #5

We Will Continue Fixing Development Issues in Production

Immutability is a cloud native benefit that receives a lot of attention, in large part because of its purported benefits to security and efficiency. Immutability means that workloads will not – and should not – change while running in production. Instead, in a cloud native environment, if the configuration of a VM needs to change or a container needs to be updated, a new version will be hatched in the pipeline, and it will completely replace the old workload instead of being 'patched.'

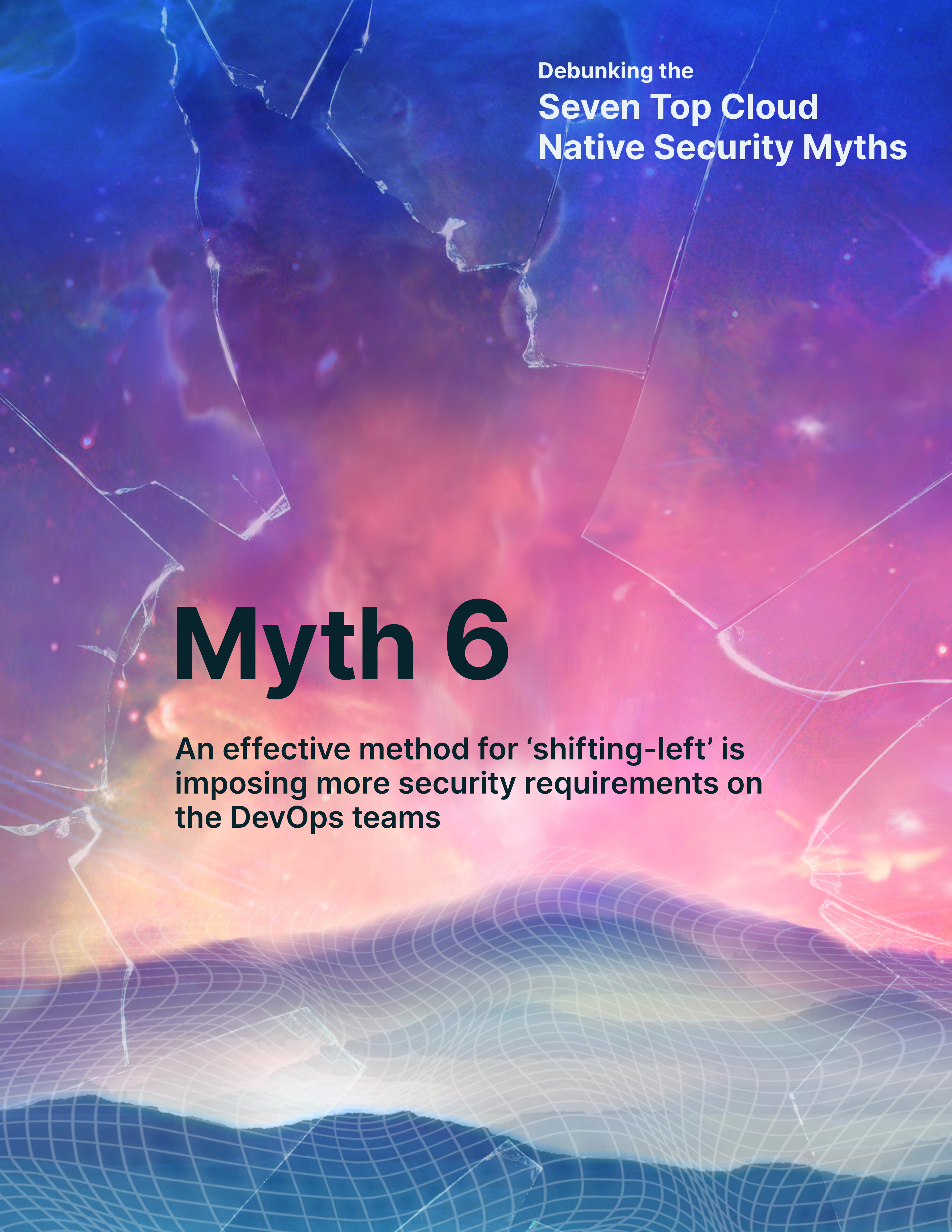
Bringing fixes earlier into the pipeline, and then automating the roll-out of new versions, implies reduced downtime for patching in runtime, which in turn has huge implications for how security teams plan to fix vulnerabilities. Unfortunately, if security teams are still believing that fixes will be made in runtime, they are at a severe disadvantage when the real security issues are 'shifted left' to the DevOps teams who build and test the app in the CI/CD workflow.

According to Gartner's 2020 Market Guide for CWPP, "Workloads need to be 'born secure' from the moment they are instantiated. This places a critical need on development scanning and modelling/simulation in the CI/CD pipeline." In other words, the security team still needs to participate in protecting against vulnerabilities and exploits in the application – but this must now happen in partnership with the DevSecOps team, earlier in the pipeline.

Some Forrester Total Impact studies have shown that, with a cloud native deployment model, operations teams reduced the time spent deploying patches by more than 75%. Instead, fixes were made early on and workloads were simply re-deployed on the same hardware. That shift implies an enormous change in mindset for both the DevOps and Security teams.

key concept

Security teams should expect to make fixes in the build pipeline in partnership with DevOps, and not in production



Myth 6

An effective method for 'shifting-left' is imposing more security requirements on the DevOps teams

An Effective Method for 'Shifting-Left' is Imposing More Security Requirements on The Devops Teams

The previous point poses the question – how can we effectively incorporate security into a DevOps process? The security team will always have a purview of security over and above the DevOps teams' responsibilities and the DevOps teams will not accept any processes that slow them down. The key is for the security team to involve DevOps in improving security and adapting it to their existing methods, using the tools the DevOps teams already use.

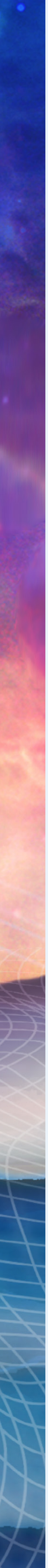
For example, if the DevOps teams use Jenkins for CI/CD, the security team can set up a policy that will warn when an image has failed to meet the security policy and explain why. The DevOps team should see the failed image from within Jenkins and see, within Jenkins, how the issue can be fixed and an approved image created. In this illustration, security has been able to educate and shorten time-to-fix without imposing its own processes and knowledge on the DevOps teams, through integration of a cloud native security solution into a DevOps tool. The same method can also be used to block such images from progressing through the pipeline, but at least initially warning is better than blocking.

key concept

Integrate cloud native security solutions into the build pipeline, so DevOps teams will use them

Myth 7

The cloud provider will secure both account configurations and what is run in the cloud



The Cloud Provider Will Secure Both Account Configurations and What is Run in the Cloud

It is critical to understand the responsibilities that a cloud provider will – and will not – assume, and where the gray areas exist. AWS, for example, briefly describes its responsibility as protection ‘of’ the cloud, whereas customers are responsible for protection ‘in’ the cloud. This simplification appears to belie two critical responsibilities for customers:

1. The cloud provider is not responsible for the safe configuration of its customers’ accounts and services. While cloud providers offer many default security configurations, it is the customers’ responsibility to check the configurations and add additional protection as needed for the security context of their applications. A false sense of security here can lead to a drastic underestimation of the time and effort required to properly configure a set of services. Gartner states, in its report on ‘How to Respond to the 2020 Threat Landscape’ that, “Through 2023, at least 99% of cloud security failures will be the customer’s fault.”

   Simply with an EC2 instance, an S3 bucket, Lambda for functions and CloudTrail for auditing, already these services require dozens of key configurations to prevent potential data leaks and security breaches. The good news is that mistakes can easily be prevented with a Cloud Security Posture Management solution, which can be set up by whoever has access to those cloud accounts in an organization. Preventing mistakes in simple deployments could be as simple as a quick, free trial.

2. There is no one formula to protecting ‘in’ the cloud, and the customer is required to learn and understand the nuances involved. For example, Kubernetes in the cloud could be accomplished via EKS or Open Source with AWS’s default EC2 Amazon Linux 2 OS, or the customer’s own Linux OS. The shared responsibility model is vastly different between these options and requires homework on the part of the customer to understand the full set of security responsibilities. For instance, in the Linux example above, customers must patch any guest OS and applications, whereas AWS would take care of patching when its default OS is used.

key concept

The customer is responsible for properly configuring cloud services, and understanding how the shared model of responsibility with the cloud providers varies, depending on the cloud services chosen

Conclusion

Considering how to implement cloud native security capabilities at the beginning of a cloud native journey brings key cloud native concepts like immutability, shifting left, and isolation into focus. With even a basic understanding of why the most commonly held myths do not apply, teams can begin to gain a more accurate sense of how to begin planning an effective cloud native security strategy, and cloud native journey overall. And planning responsibilities, resources, controls, and how teams need to interface with one another is only possible through an understanding of the realities at hand.

Most importantly, learning about cloud native security early on in a cloud native journey can move teams a step closer to the ultimate promise, which is to make cloud native environments more secure than ever before. CISOs would do well to immerse themselves in this cloud native world sooner rather than later, because security done right will accelerate adoption, improve efficiency, and make security teams the organization's cloud native heroes.

Aqua Security is the largest pure-play cloud native security company, providing customers the freedom to innovate and run their businesses with minimal friction. The Aqua Cloud Native Security Platform provides prevention, detection, and response automation across the entire application lifecycle to secure the build, secure cloud infrastructure and secure running workloads wherever they are deployed.
Aqua customers are among the world's largest enterprises in financial services, software, media, manufacturing and retail, with implementations across a broad range of cloud providers and modern technology stacks spanning containers, serverless functions, and cloud VMs.

aquasec.com

contact@aquasec.com

@Aqua Security

@AquaSecTeam

in/Aqua Security

@AquaSecTeam