RSAC Innovation Sandbox 2021 WINNER

Gartner COOL VENDOR 2021

Break Down the Barriers Between App & Infrastructure as Code Security

by apiiro

Welcome!

Cloud security is everywhere. Numerous security solutions integrate with cloud infrastructure platforms to identify vulnerabilities, but they all have one thing in common: they detect issues that are already in production. With the speed that modern attackers exploit exposed weaknesses, remediating issues in production systems is no longer acceptable - or necessary!

Analyzing Infrastructure as Code (IaC) to detect cloud misconfigurations is one of the hottest areas of cybersecurity. An entire industry has popped up around this single topic - and for good reason. By looking at Infra as Code, we can identify security issues before they ever reach production systems. However, today's Infra as Code tools analyze security configurations in a limited, myopic way that generates noisy alerts and does not help organizations understand their risk.

Evaluating cloud infrastructure configurations without context is meaningless. The risks of cloud infrastructure and the applications that run on it are inherently connected. An effective understanding of risk requires connecting the dots between the applications and their infrastructure, from cloud to code.

The rise of cloud-native applications has only made the need to combine application security with cloud security context more pressing. According to Gartner, "Cloud-native applications arise from the combination of microservices applications (typically using Linux containers), built using rapid DevOps-style development and automatically deployed onto programmatic cloud infrastructure." Applications and infrastructure are now intrinsically intertwined and so security must be as well - in a cohesive, complementary, and contextual way.

[1] Gartner, "Innovation Insight for Cloud-Native Application Protection Platforms", Neil MacDonald and Charlie Winkless, August 23, 2021

Idan Plotnik,
Co-Founder & CEO

Table of Contents

Break Down the Barriers Between App & Infrastructure as Code Security

Siloed Approaches to Detect Cloud Misconfigurations Lack the Context to Understand Risk

Introduction

Cloud misconfigurations are one of the greatest risks when it comes to cloud computing - not the underlying infrastructure from Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), and others. The issue comes from the speed of cloud adoption and cloud-native development. Virtual machines and containers are created and destroyed in near real-time. Entire environments can be spun up and existing environments duplicated, including complex architectures and configuration settings. But traditional means of securing these environments cannot keep up.

Cloud environments can be as secure - and even more secure - than traditional on-premises environments, but agility, speed, and absolute consistency are required for Security and Privacy. This means automation and Infrastructure as Code, which help prevent manual configuration errors before they reach production. The idea of Infrastructure as Code has been around for quite some time but it isn't until now that security is beginning to catch up and make use of its full potential to reduce risk continuously and automatically.

The current Cloud Security industry, such as Cloud Security Posture Management (CSPM) tools, focuses on only one aspect of the problem and lacks the context needed to help organizations make risk-based decisions. A cloud environment never exists in true isolation from a security perspective.

> "Understanding and addressing the real risk of cloud-native applications requires advanced analytics combining siloed views of application risk, open-source component risk, cloud infrastructure risk and runtime workload risk."

Gartner,
"Innovation Insight for Cloud-Native Application Protection Platforms"
Neil MacDonald and Charlie Winkless, August 23, 2021

Gartner

Examples of Breaches Due to Cloud Misconfigurations

Many large and public breaches have occurred due to cloud misconfigurations - many of them due to a single setting! A few examples include:

Marriott

In 2020, information on over 5.2 million guests was stolen from Marriott systems using compromised employee accounts. Multi-Factor Authentication (MFA) was not required for system access.

MageCart

In 2019, websites for over 17,000 domains were compromised due to misconfigured Amazon S3 buckets that allowed anyone with an Amazon Web Services account to read or write content.

Capital One

In 2019, an attacker gained access to over 100 Million Capital One credit card accounts and applications. According to Capital One, "We believe that a highly sophisticated individual was able to exploit a specific configuration vulnerability in our infrastructure." The issue is believed to be due to misconfigured permissions on an open source Web Application Firewall (WAF) that was running on top of Amazon Web Services (AWS). Capital One was later fined $80m by U.S. regulators.

Dow Jones

In 2019, security researcher Bob Diachenko discovered an unprotected database with 2,418,862 on a public AWS Elasticsearch cluster. The exposed data included the identities of government officials and politicians, including details on government sanctions and criminal activity.

Booz Allen Hamilton

In 2017, more than 60,000 sensitive files on a U.S. military project were discovered on a publicly-accessible AWS system that did not have a password set. Among the 28GB of exposed data were passwords to accounts held by government contractors with Top Secret Facility Clearance.

06

The Top 5 Infrastructure as Code Misconfigurations

Cloud misconfigurations are the cause of many highly-publicized, damaging, and expensive security breaches, but there are many types of misconfigurations that can result in a security breach!

1 Unencrypted & Unprotected Storage Buckets

2 Hardcoded Secrets

3 Overly-Permissive Access Controls

4 Weak or Missing Server Hardening

5 Unencrypted Launch Configurations

Here are the top 5:

1 **Unencrypted & Unprotected Storage Buckets**
Many highly-publicized breaches are the result of the simple act of forgetting to turn on password protection for Amazon S3 and other storage buckets. In addition, all sensitive data must be encrypted at rest, in transit, and in use.

2 **Hardcoded Secrets**
Developers often leave secrets exposed in source code, including user passwords, API keys, authentication tokens, private encryption keys, and more. These can be found in many places, from source code to Infra-as-Code, test code, scripts, and documentation.

3 **Overly-Permissive Access Controls**
Users are given excessive access permissions that are not required for their use of the system. This violates the fundamental security principle of least privilege.

4 **Weak or Missing Server Hardening**
Individual cloud systems are not appropriately hardened, leading to unnecessarily exposed ports and default services.

5 **Unencrypted Launch Configurations**
Cloud configuration tools such as Amazon Elastic Block Store (EBS) allow for encrypted launch configurations that are used for auto-scaling. Not using such tools often leads to unencrypted data.

The Problems with Today’s Siloed Cloud & IaC Security (1/2)

There are numerous factors that may impact the business risk of a breach, including:

1. The type of applications that are deployed in the cloud, which can have a high, medium, or low impact on the business
2. The type of data that is stored and/or processed in the application (e.g., PII, PHI, etc.)
3. Compliance requirements (e.g., HIPAA, PCI, and GDPR)
4. Upcoming features that may change any of the above (such as a user story that requires a new PII field)
5. Additional services that function as part of the application’s architecture, such as a cloud-based firewall, an API gateway, or identity provider
6. Whether the application is back-end, for internal users, or Internet-facing

The Problems with Today's Siloed Cloud & IaC Security (2/2)

Current solutions of detecting cloud misconfigurations using Infrastructure as Code are single-dimensional. They look at individual factors, such as a storage bucket missing encryption, connections being unencrypted, etc. But without context, these alerts are just noise. In an ideal world, all data-at-rest and data-in-transit would be encrypted using the latest algorithms and frameworks. In the real-world, where Security Engineers are overwhelmed with alerts, prioritizing based on risk isn't a "nice to have" - it's required. Some data are more sensitive and business critical than other data and any security tool that can't effectively make that distinction is not useful for modern AppSec professionals. For example:

Example:

1. If a connector is unencrypted but isn't exposed to the outside world, the priority of fixing this "misconfiguration" may be low. A single-dimensional scanner won't be able to correlate those two facts to prioritize appropriately.

2. Data on internal catering orders or past events may not need to be encrypted while encrypting customer data and financial information is critical. At an even more granular level, it is critical to encrypt a storage bucket that does not contain sensitive information but did in the past!

Risk is multidimensional and existing cloud security tools can't keep up.

09

Combining App & Infra Security (1/2)

It’s time to stop thinking about application security and infrastructure security separately. Individuals and teams cannot be assigned to one area or the other and be expected to succeed. The barriers between App and infra must be broken down and security people, processes, and tools need to work together seamlessly in order to build an accurate view of risk:

- To improve visibility, inventories of application and infrastructure components need to be combined into a single source.

- Having separate lists of alerts for code vulnerabilities and cloud misconfigurations - from separate tools - leads to siloed processes, thinking, and activities. The wrong things get remediated at the wrong times, while critical risks get missed. Alerts from across the entire SSDLC need to be analyzed and prioritized in a single place.

- Risk assessments and change management processes need to include both application and infrastructure data.

- When evaluating probable attack paths and assessing impact (the two most common factors within any risk calculation), a combined view of the app and infra realms is the only way to have a valid view of those factors. Only after they are viewed together can the true-to-life and extended impact be properly understood.

Combining App & Infra Security (2/2)

Watch any experienced Security expert investigate a potential security risk and it will soon be obvious that context is everything! But an understanding of that context shouldn't be left to manual reviews. Contextual risk assessments of both application and infrastructure code changes need to be performed both continuously and automatically. This is the only way to provide consistency, efficiency, and ultimately, remediate the app and infra risks that matter.

Example:

If an API Gateway is found to not properly enforce authentication or authorization controls on a particular API, that does not provide enough information for a cloud security professional to understand the risk. They must also understand:

- Which application is the API tied to?
- Is that application business-critical?
- Does the application store sensitive information?
- Is that information subject to regulatory and compliance requirements?

The combination of these questions highly impacts your prioritization and approach to if and how to remediate different security issues.

apiiro

About Apiiro

Proactively remediate risk before releasing to the cloud.

Apiiro helps you detect & fix critical risks such as design flaws, secrets, Infra-as-Code, API & OSS vulnerabilities across the software supply chain.

Apiiro is re-inventing risk remediation for cloud-native applications.

Discover

All application assets to build an inventory, from cloud to code

Remediate

Critical risks in cloud-native applications, before releasing

Measure

DevSecOps maturity and remediation KPIs across the entire SDLC