

Although infrastructure as code (IaC) is an important part of DevOps and has many benefits, it also introduces significant risks. Many successful attacks on cloud services result from misconfigurations buried in a plethora of infrastructure code and configuration files.

Cloud Infrastructure Configuration Is a Critical Component of a Holistic, Secure Software Supply Chain Strategy

January 2022

Written by: Jim Mercer, Research Director, DevOps and DevSecOps, and Katie Norton, Senior Research Analyst, DevOps

Introduction

With the explosion of digital transformation, digital end users are placing a higher priority on speed and convenience via applications. To meet these consumer expectations, organizations are embracing DevOps and cloud platforms for fast application development and updates leveraging virtual software-defined infrastructure rather than physical servers stacked in racks within a company datacenter. As a result, there has been significant growth in new applications running on the cloud. Unfortunately, the speed of application development using DevOps combined with the proliferation of application changes and the accessibility of the cloud offers attackers an attractive attack surface, increasing the need for DevSecOps to improve the security of vulnerable applications.

DevSecOps, although important, focuses on securing the applications themselves, but it does not typically secure the software supply chain (i.e., the software factory) used to manufacture and deploy those applications. The contemporary software supply chain is made up of the multiple components that go into developing an application, such as people, processes, dependencies, and tools. The security of the software supply chain is fundamental to ensuring the fortification of modern applications. Without the proper protection, supply chain attacks can go undetected and create exponential harm to both application producers and consumers.

Software supply chain security has garnered major attention after President Biden issued an Executive Order aimed at improving the nation's cybersecurity in May 2021. The order mandates several key changes to how both the federal government and the private sector companies that do business with the government will handle software supply chain security. Included in the order is the development of baseline security standards by the National Institute of Standards

AT A GLANCE

KEY TAKEAWAYS

- » DevSecOps often overlooks the security of the software supply chain.
- » Modern applications rely on IaC to programmatically define and configure application infrastructure.
- » Many successful attacks on cloud services result from misconfigurations buried in a plethora of infrastructure code and configuration files.
- » IaC security scanning can mitigate risks associated with software-defined cloud infrastructure.
- » IaC security scanning should be used as part of a larger software supply chain security approach.

and Technology (NIST) that require software development organizations to maintain a software bill of materials (SBOM). For holistic security, organizations must ensure that the independent components of their applications are protected and that all digital entry points to the software factory are locked and guarded.

Modern distributed applications use cloud-native architectures that rely upon software-defined infrastructure through infrastructure as code (IaC). IaC automates infrastructure provisioning by defining it programmatically rather than leaning on tedious, error-prone manual configurations of application resources. Defining infrastructure provisioning in code enables the application of DevOps principles such as version control and continuous integration/continuous delivery (CI/CD) to the testing and deployment process. Additionally, IaC allows identical workloads to be deployed in various locations and/or across projects and permits the creation of separate but identical (or near-identical) environments for development, testing, staging, and production. Through IaC, DevOps teams can automate cloud resource deployments and replicate environments more quickly and efficiently, resulting in improved speed and consistency as well as reduced application development costs.

Benefits Versus Risks

Although IaC is important in DevOps and has many benefits, it can also introduce significant risks. Many successful attacks on cloud services result from misconfigurations buried in a plethora of infrastructure code and configuration files. IaC usage has evolved across hybrid cloud environments with deployments in both on-premises and multicloud environments. These complex deployments combined with human error make it easy to overlook coding misconfigurations. Further, when misconfigurations remain undetected in infrastructure code or YAML files, they frequently are replicated and shared among application environments, resulting in minor errors becoming larger runtime attack surface vulnerabilities. Additionally, because IaC is written in declarative languages, it requires less skill to reverse engineer. Hence, attackers can more easily determine where assets are and how to access them, which can be devastating when the IaC contains a security misconfiguration. Therefore, undetected IaC code tampering can be just as destructive as application code tampering.

IaC security scanning, which examines infrastructure code and configuration files to uncover vulnerable cloud infrastructure resources, such as misconfigurations that can expose confidential company intellectual property (IP) and/or customer data, is essential to mitigate the inherent risks of using IaC. Developers and DevOps engineers can scan IaC code pre-commit during Git merge requests and before the potentially misconfigured code makes its way into the master branch. Integrating IaC scans into the CI/CD process ensures that infrastructure code and configurations are audited early and often throughout the DevOps pipeline. Shifting security to the left-hand side of the process and identifying problems when and where they occur allow developers to fix misconfigurations before they reach production. When scanning happens within the developer's workflow, it helps teach secure infrastructure coding practices and can foster a culture of security awareness.

DevSecOps, although important, focuses on securing the applications themselves, but it does not typically secure the software supply chain (i.e., the software factory) used to manufacture and deploy those applications.

A critical part of an IaC security program is checking for hard-coded secrets, such as SSH keys, API tokens, and passwords in the infrastructure code and configuration files. If these secrets are inadvertently embedded in the code and exposed, then attackers are essentially handed the keys to the software factory and aided in lateral movement. For example, when credentials are exposed, bad actors can use those keys to shut down services, steal IP, or upload malware. The best way to protect against these hard-coded secrets is to prevent them from ever getting into the version control system by scanning commits before they are merged into the main branch. The magnitude of leaked secrets is significant; thus IaC scanning cannot be considered complete without addressing hard-coded secrets.

IaC security should also ensure proper checks occur when changes are made to IaC code; in general, IaC code isn't changed as frequently as application code. However, cloud infrastructure environments are highly dynamic and subject to unapproved production changes post-deployment. These unexpected changes can spawn configuration drift. Drifts in configuration can make test and production environments fall out of sync, causing configurations to deviate from their secure state over time. Identifying the configuration changes causing drift, whether rogue, unintended, or ad hoc, can help ensure there are no open doors for bad actors.

An IaC security program can ensure an organization adheres to industry best practices, custom rules, or internal policies. Standardization bodies such as NIST and the Center for Internet Security (CIS) help guide organizations with consensus-based standards and frameworks. Additionally, each cloud and resource may have its own set of security best practices and corresponding compliance benchmarks. Organizations may also enforce policies unique to their infrastructure and business compliance guidelines.

Considerations

IaC is critical for modern DevOps teams deploying to the cloud. However, it is equally essential to implement real-time feedback for developers before building cloud environments containing security or compliance flaws that could potentially create an unexpected attack vector. The ability to catch security misconfigurations before they reach the production environment is a key benefit of modern AppSec programs.

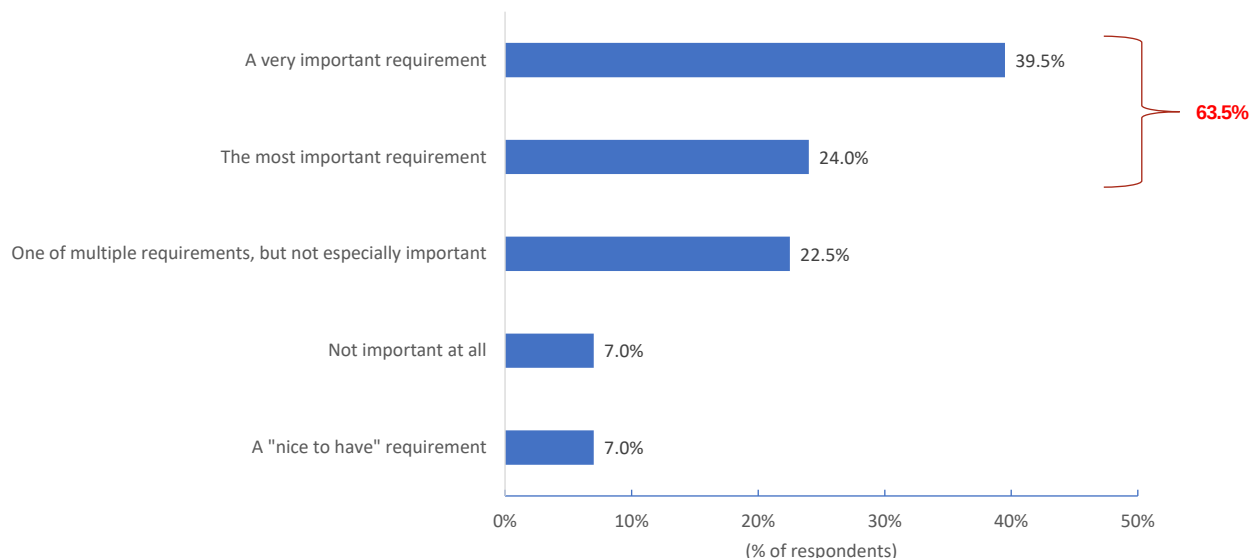
When organizations implement IaC security, it is important for DevOps teams to consider the scope of infrastructure artifacts being scanned. Scanning just the actual scripting code or looking at just infrastructure files in a curated repository of source code is likely insufficient. There are most assuredly multiple types of IaC code artifacts, such as HashiCorp Configuration Language (HCL), AWS CloudFormation, Kubernetes, and other configuration files that will need to be scanned to get a complete picture of IaC exposure; thus the IaC security solution will need to be capable of scanning multiple artifact types throughout the DevOps pipeline. Further, the IaC security solution must be able to reconcile the desired infrastructure configuration in the source code with the actual configuration running in production to be able to identify and resolve precarious drifts in infrastructure configurations.

In large enterprises, roles for writing application code and IaC may be distinct, yet both types of code often reside in the same code repository. Therefore, the most effective way to improve the governance of IaC is to store it in dedicated repositories to which least privileged policies are applied. Additional controls may include multifactor authentication, branch protection rules, and separation of duties.

IaC security scans should run seamlessly, and the scanning process cannot be disruptive to developers and the DevOps pipeline. Figure 1 shows that developer acceptance of security tools is critical for scaling DevSecOps.

FIGURE 1: **Developer Acceptance of Security Tools**

Q How important has developer acceptance of using security tooling been as part of your adoption of DevSecOps?



n = 200

Source: IDC's U.S. Survey of DevSecOps Adopters, December 2020

Almost 64% of the survey respondents acknowledged that to scale DevSecOps, the security tools must be accepted by developers as tools they want to use. Modern DevOps and development teams have a lot on their plates, so the security tools must be easy to use and provide accurate results without many false positives that can waste valuable time. Therefore, the ideal IaC solution should integrate natively with the source control management system so that workflows become a seamless part of the developer experience, providing accurate results that can be both relied on and acted upon.

It is important to note that there are multiple software supply chain attack areas to consider, and IaC security scanning is only one part of a larger secure software supply chain. Some fundamental supply chain issues to keep in mind include:

- » **DevOps and developer tooling visibility.** Security teams often have little to no visibility into many of the tools that make up the software supply chain. Many times, these tools have been deployed by and are administered by the developers themselves. Developers commonly see the security of their tools as too restrictive and tend to remove security constraints over time.
- » **Lateral movement.** A consequence of increasing DevOps automation along with the overall complexity of the modern software supply chain is that once bad actors get a foothold, they can conceal their activity and move laterally across the supply chain to infiltrate critical data, systems, and resources. From here, an attacker could use a compromised developer account or even create a fake account that resembles a legitimate developer account to tamper with code that compromises the application software packages.

Failure to secure one attack entry point is the equivalent of locking the front door to your house while leaving the back door wide open. Modern DevOps teams need to be cognizant of and diligent about securing the software supply chain from multiple dimensions. When combined with ease of lateral movement, the many entry points into the software supply chain constitute a significant risk unaccounted for in many organizations. All the software supply chain attack vectors are interrelated and interdependent, so looking at IaC alone is simply not sufficient.

Key Trends

In IDC's December 2020 *U.S. Survey of DevSecOps Adopters*, when DevOps teams were asked for their top DevSecOps security challenges, 42% called out misconfigurations. As such, we expect greater adoption of IaC scanning as organizations continue to scale their use of DevOps and cloud automation. This is playing out in the marketplace as well; in 2021, IDC tracked five acquisitions related to IaC security as larger security vendors try to enhance their ability to address these challenging misconfigurations. However, IaC scanning alone is too narrowly focused. Security vendors need to consider providing comprehensive capabilities that can reduce the IaC security risks more holistically by inspecting areas such as hard-coded secrets, IaC leakage, weak governance, and code tampering.

Bad actors now recognize that the software supply chain is a soft target. As a result, they are increasingly looking for new ways to infiltrate software supply chains to gain access to critical business systems and data. Unfortunately, infiltrating a software supply chain is easier than ever because hacking tools and techniques are readily available on the web and because advanced programming skills are no longer required. In the future, these actors will continue to become more sophisticated in how they attack the software supply chain as well as more resourceful in trying every potential entry point. As a result, organizations will be forced to look more holistically at software supply chain security because manual security processes and a one-dimensional approach, such as focusing solely on IaC security, are both unscalable and inadequate.

Conclusion

Organizations are embracing DevOps and cloud platforms, leveraging virtual software-defined infrastructure for fast application development and updates to meet consumer demand. The pace of application development using DevOps combined with the growth of new application changes and cloud platforms has made modern applications an attractive attack surface, increasing the need for DevSecOps to improve the security of unsafe applications.

Despite the importance of DevSecOps, it does not protect the software supply chain (i.e., the software factory) used to build and deploy those applications. This insufficiency drives the adoption of new software supply chain security solutions, such as IaC security that secures the provisioning and configuration of the infrastructure running modern applications. However, there are multiple vectors to consider, and IaC security is only one part of a larger secure software supply chain. Therefore, DevOps teams need to be cognizant of and diligent about securing the software supply chain from multiple dimensions. Because bad actors are opportunistic and the risk of a software supply chain cyberattack is increasing, they unfortunately need to win only once for a business to be impaired.

About the Analysts



Jim Mercer, Research Director, DevOps and DevSecOps

Jim Mercer is responsible for researching, writing, and advising clients on the fast-evolving DevOps market. Jim's core research includes topics such as rapid enterprise application development, modern microservice-based packaging, application security, and automated deployment and life-cycle/management strategies as applied to a DevOps practice. He looks at how organizations are prioritizing DevSecOps and using automation to insert security assessments at the beginning of the DevOps delivery pipeline (i.e., shift left).



Katie Norton, Senior Research Analyst, DevOps

Katie Norton is responsible for researching, writing, and advising clients on the fast-evolving DevOps market. Katie's core research includes topics such as rapid enterprise application development, modern microservice-based packaging, application security, and automated deployment and life-cycle/management strategies as applied to a DevOps practice.

MESSAGE FROM THE SPONSOR

About Cyscale

Cyscale is a complete software supply chain security solution that provides visibility, security, and integrity across all phases of the SDLC. Cyscale integrates with DevOps tools and infrastructure providers, hardens their security postures by implementing consistent governance, and reduces the risk of breaches with a series of scanning engines that look for issues like hardcoded secrets, infrastructure as code misconfigurations, code leaks and more. Cyscale's knowledge graph tracks code integrity, user activity, and events across the SDLC to prioritize risk, find anomalies, and prevent code tampering. To learn more visit www.cyscale.com.



The content in this paper was adapted from existing IDC research published on www.idc.com.

This publication was produced by IDC Custom Solutions. The opinion, analysis, and research results presented herein are drawn from more detailed research and analysis independently conducted and published by IDC, unless specific vendor sponsorship is noted. IDC Custom Solutions makes IDC content available in a wide range of formats for distribution by various companies. A license to distribute IDC content does not imply endorsement of or opinion about the licensee.

External Publication of IDC Information and Data — Any IDC information that is to be used in advertising, press releases, or promotional materials requires prior written approval from the appropriate IDC Vice President or Country Manager. A draft of the proposed document should accompany any such request. IDC reserves the right to deny approval of external usage for any reason.

Copyright 2022 IDC. Reproduction without written permission is completely forbidden.

IDC Research, Inc.
140 Kendrick Street
Building B
Needham, MA 02494, USA
T 508.872.8200
F 508.935.4015
Twitter @IDC
idc-insights-community.com
www.idc.com