La sécurité dans l'Infrastructure as Code (IaC) : une nécessité pour les entreprises

La sécurité dans l'Infrastructure as Code (IaC) : une nécessité pour les entreprises

Par Jim Amstrong, Directeur Produit Sécurité et Containers

Les applications natives du cloud ne se limitent pas au code créé par les développeurs : elles incluent aujourd'hui l'Infrastructure as Code (IaC) qui dicte la façon dont les applications sont configurées sur l'infrastructure du cloud et la manière dont les applications conteneurisées fonctionnent sur Kubernetes.

L'utilisation de l'IaC permet des déploiements plus rapides et automatisés, mais son utilisation augmente également la charge de travail des développeurs pour sécuriser, non seulement leur code, mais aussi la configuration de l'infrastructure, en plus des dépendances du code et des containers.

Bien que les organisations ne se soient pas encore mises d'accord sur la meilleure façon d'utiliser l'IaC ou sur la personne responsable de son écriture et de sa maintenance, celles qui profitent des tests de sécurité automatisés détectent et corrigent les mauvaises configurations plus rapidement que leurs pairs.

A ce jour, de nombreuses entreprises n'en sont qu'au début de leur aventure avec l'IaC.
63 % d'entre elles commencent tout juste à explorer la technologie quand seulement 7 % déclarent avoir mis en œuvre l'IaC au mieux des capacités actuelles du secteur. Bien qu'il existe de nombreux outils utilisés ou envisagés, 71% préféreraient standardiser un ensemble d'outils et un flux de travail communs pour tous les types et formats de configuration IaC.

Une détection plus efficace…

Actuellement, les applications modernes se déploient automatiquement sur une infrastructure créée et configurée par le code. En conséquence, la sécurité est souvent reléguée au second plan au profit d'un déploiement rapide, ce qui signifie que les problèmes de configuration ne sont découverts qu'après le déploiement de ces applications.
Gartner déclare d'ailleurs que, « d'ici 2025, 70 % des attaques contre les containers proviendront de vulnérabilités connues et de mauvaises configurations qui auraient pu être corrigées. »

Pourtant, tout cela ne signifie pas nécessairement que la rapidité est intrinsèquement risquée lorsque nous parlons d'IaC. En fait, les tests automatisés et les contrôles de sécurité qui sont en place pour d'autres formes de code peuvent être utilisés avec l'IaC et contribuer à intégrer les meilleures pratiques de sécurité dans le processus de développement et de déploiement.

Toutefois, ce type de test automatisé est relativement nouveau. Près des deux tiers des entreprises déclarent que leur flux de travail actuel pour l'IaC et le code de configuration passe par des tests d'intégration continue (IC)*, mais les contrôles de sécurité ne font pas toujours partie de ces tests.

Seules 32 % d'entre elles incluent des contrôles de sécurité dans leurs pipelines.

En fait, la plupart des problèmes de sécurité sont encore découverts après le déploiement, par le biais de tests d'intrusion, d'audits et d'enquêtes sur les incidents de sécurité. Pour ceux qui n'utilisent que ces contrôles post-déploiement, cela représente potentiellement 9 jours de fonctionnement avec une faille de sécurité.

Pour les entreprises dont le code IaC et de configuration est soumis à des tests, le principal obstacle à l'intégration des contrôles de sécurité est le manque de bonnes pratiques standardisées sur les éléments à vérifier, et le manque de clarté des critères de référence en matière de sécurité. Avec des outils de détection automatisés, les organisations peuvent aujourd'hui améliorer la sécurité de l'IaC, tout en laissant aux équipes le temps de déterminer ce qui est le plus important pour leurs besoins.

… pour une résolution plus rapide

Pourtant, trouver le problème ne représente que la moitié du chemin. Une fois le problème découvert, il est nécessaire de le résoudre. Plus d'une personne sur deux affirment qu'elles remédient généralement à un problème de sécurité en modifiant directement l'infrastructure plutôt qu'en modifiant le code source de l'IaC. Ce qui peut se révéler problématique à long terme. Pour ceux qui choisissent cette voie de remédiation manuelle, leur raisonnement est partagé entre un manque de normalisation, de connaissances et de communication, et un désir d'accélérer les corrections autant que possible.

L'un des obstacles à l'abandon de la sécurité des IaC vient du fait que les équipes ont eu du mal à normaliser les pratiques dans l'ensemble de leur organisation, laissant chaque équipe auditer les IaC comme elle l'entend. Ce qui pose des problèmes évidents de sécurité, et témoigne d'une problématique plus large en matière de responsabilités.

Il semble qu'il n'y ait pas de consensus actuel sur qui est responsable de la sécurité de l'IaC.
Les développeurs et les DevOps ont un rôle légèrement plus important que les autres équipes individuelles et un bon nombre d'entre eux s'accordent sur le fait qu'il s'agit d'une responsabilité partagée, ce qui correspond à l'approche DevSecOps.

Une solution claire pour répondre aux problèmes de sécurité de l'IaC consiste à investir dans les outils et la formation nécessaires pour renforcer la confiance et la bande passante de ces équipes, afin de leur permettre de déployer le code rapidement, et ce, en toute sécurité. Dans le même rapport cité plus haut,

Gartner voit également le potentiel de ces outils automatisés et prédit que, d'ici 2025, les entreprises amélioreront de 30 % la correction des vulnérabilités de codage grâce aux suggestions de code appliquées par des solutions automatisées, réduisant ainsi de 50 % le temps consacré à la correction des bugs et des failles de sécurité.