GitHub

Le guide DevSecOps pour Architecte d’entreprise

Introduction

Les architectes d'entreprise ont compris depuis bien longtemps ce que les leaders DevOps découvrent tout juste : que la sécurité, et pas seulement l'exécution et la livraison, est une responsabilité partagée.

Aujourd'hui, les équipes Opérations utilisent la collaboration, l'automatisation et les conteneurs afin d'accélérer la livraison de logiciels. Alors que les pratiques DevOps les ont aidés à trouver de nouvelles façons de développer plus rapidement, les pratiques de sécurité obsolètes continuent de ralentir de nombreuses entreprises.

Entrez dans le monde de DevSecOps.
DevSecOps apporte une sécurité informatique au sein des équipes de développement et opérations afin d'assurer que la sécurité est une priorité à chaque étape du cycle de développement du logiciel. Avec seulement quelques changements, votre organisation peut délivrer des logiciels de meilleure qualité et plus sécurisés, sans retards ni augmentation de coûts.

Pourquoi DevSecOps ?

Des coûts de sécurité réduits

DevSecOps inclut toutes les meilleures pratiques DevOps qu'appliquent en permanence les équipes les plus performantes, avec la sécurité requise par les grandes organisations. En incorporant la sécurité dans votre pipeline DevOps, il est possible de trouver des failles avant même qu'elles ne soient publiées, et il est plus simple et moins onéreux d'y remédier.

Un travail d'équipe plus efficace

Tout comme les développeurs et les équipes Opérations sont responsables de la fiabilité et de la qualité dans le principe DevOps, DevSecOps fait de la sécurité un effort collectif et non une étape en fin de cycle.

Les équipes de développeurs, d'Opérations et de sécurité travaillent ensemble afin de sécuriser les applications de la première ligne de code jusqu'au déploiement en production.

Une automatisation guidée par les directives

Un bon programme DevSecOps augmente également la confiance dans tout le processus de livraison de logiciels de votre organisation. La sécurité est implémentée par des tests automatiques de façon à ce qu'elle soit guidée par les directives, plutôt que par un enchaînement d'outils manuels, portant à confusion et ralentissant le développement pour tout le monde.

AVANT :

Une sécurité compartimentée

Des tests juste avant le déploiement

Des tests statiques et dynamiques ont lieu à la fin du cycle de livraison, juste avant le déploiement.

Une expertise en sécurité compartimentée

Les équipes de développement, d'opérations informatiques et de sécurité travaillent de manière indépendante.

Des tests de sécurité manuels

Les organisations déploient moins souvent et effectuent des tests de sécurité ponctuellement, en fonction des besoins.

MAINTENANT :

DevSecOps

Des tests de la conception au déploiement

Des tests statiques et dynamiques ont lieu de pair avec des pratiques de développement sécurisé, des contrôles qualité réguliers, et des corrections des failles de sécurité.

Une expertise en sécurité partagée

Les équipes de développement, d'opérations informatiques et de sécurité suivent des consignes de sécurité communes au quotidien.

Des tests de sécurité automatiques

Les organisations déploient plus souvent et effectuent des tests de sécurité automatiquement au sein de leur pipeline CI/CD.

Débuter avec DevSecOps : Trois conseils

1 Utilisez une plateforme partagée et sécurisée pour la collaboration

Tout comme DevOps, DevSecOps dépend de la collaboration et se conclut par la collaboration. Une plateforme partagée aide les équipes de développement, d'opérations informatiques et de sécurité à construire ensemble et à standardiser leur manière de travailler. Privilégiez les plateformes intégrant nativement la sécurité afin que l'ensemble de votre organisation puisse partager les bonnes pratiques, trouver et réutiliser du code, et collaborer dès le départ.

CONSEIL GITHUB

Une bonne sécurité commence au moment de l'authentification. Lorsque vous trouvez la bonne plateforme de collaboration, elle doit également prendre en charge les fonctionnalités de gestion des identités comme l'authentification à deux facteurs, l'identification unique ("single sign-on"), les synchronisations automatiques de l'organisation, etc.

2 Sécurisez votre SDLC (software development lifecycle) de bout en bout

Jusqu'à 99 % des applications récentes contiennent du code open source. Cela signifie que des dépendances open source font déjà partie de votre patrimoine.* Intégrez des outils de sécurisation du code dans votre pipeline CI/CD qui peuvent identifier proactivement des failles de sécurité à la fois dans le code open source et le code interne.

*2019 Open Source Security and Risk Analysis Report

CONSEIL GITHUB

Les logiciels open source sont partout. Des outils de sécurité automatisés comme l'analyse de variantes LGTM, WhiteSource, et Snyk peuvent faciliter la recherche et l'élimination des bugs et failles que votre équipe ne peut pas identifier manuellement.

3 Suivez la sécurité après la mise en production

La sécurité ne s'arrête pas après la publication du code, et il en va de même pour votre pipeline DevSecOps. Après le déploiement, maintenez la sécurité du code et des clients en contrôlant en permanence les éventuelles failles. Recherchez des outils qui peuvent suivre et mettre à jour de potentielles failles après le lancement, et ce avant que des hackers n'en profitent.

CONSEIL GITHUB

Alors même que les alertes de failles de sécurité rendent les projets plus sécurisés, les données de l'industrie montrent que plus de 70 % des failles restent non corrigées après 30 jours, voire jusqu'à un an pour beaucoup d'entre elles.

Utilisez des solutions intégrées qui ne se contentent pas d'identifier les failles, mais qui y remédient automatiquement.

Des questions au sujet du développement sécurisé de logiciels ?

Nous pouvons vous aider.

Plus d’informations sur
fr.github.com/enterprise