EBOOK

Contrast SECURITY

Guide d'Achat DevSecOps : Sécurité des Applications

Synthèse

Alors que les applications évoluent et deviennent de plus en plus complexes et distribuées, les méthodes de sécurisation traditionnelles montrent leurs limites sur plusieurs plans critiques, notamment l'élimination des vulnérabilités pendant le développement du logiciel, le suivi des risques liés aux composants open source (SCA) et la protection des applications une fois lancées en production. Le résultat ? L'usine logicielle des entreprises n'a jamais été aussi vulnérable. Il n'est donc pas étonnant que les attaques d'applications web aient augmenté de 56 % en un an. En outre, nous assistons à une recrudescence des attaques contre les éditeurs de logiciels et la chaîne d'approvisionnement, ainsi qu'à une progression des menaces ciblant des applications cloud natives.

Pour être efficace, le test de sécurité des applications doit tenir compte des exigences spécifiques des applications modernes, fonctionner à n'importe quel moment du cycle de vie du développement logiciel (SDLC) et se concentrer sur la découverte et les tests des vulnérabilités en plus de leur correction. Son fonctionnement doit s'articuler autour de critères permettant d'identifier les vulnérabilités, d'analyser la composition logicielle, de protéger l'exécution et d'assurer la conformité, entre autres. Ce document est destiné à servir de modèle pour les appels d'offres (RFP) ou les projets de sélection de fournisseurs de solutions de sécurité pour les applications.

Table des matières

01

Outils Disparates, Contrôles Peu Pertinents et Indicateurs Mal Choisis

L'usine logicielle n'a jamais été aussi vulnérable. Les tests de sécurité des applications traditionnels n'ont pas réussi à s'adapter à l'ampleur et à la complexité des applications modernes, ce qui n'est pas sans poser aujourd'hui de graves problèmes dans les environnements DevOps :

LES DÉVELOPPEURS N'ONT QUE PEU DE TEMPS POUR CORRIGER LES VULNÉRABILITÉS

Contraints de raccourcir en permanence les cycles de livraison, les développeurs n'ont guère le temps d'identifier et de corriger les vulnérabilités des applications. Les volumes de code augmentent au fil du temps, et les vulnérabilités s'accumulent. Les applications en production se retrouvent donc exposées – presque toutes les entreprises (99 % et plus) avouent avoir décelé au moins quatre vulnérabilités dans leurs applications en production.[1]

> ***Des efforts concertés pour corriger les vulnérabilités qui menacent les entreprises et gérer la dette de sécurité constituent le meilleur moyen de réduire le risque d'intrusion.[2]***

[1] « The State of DevSecOps Report », Contrast Security, décembre 2020.
[2] Yaniv Bar-Yadan, « How To Get Out Of Security Debt », Forbes, 3 septembre 2020.

DES OUTILS DISPARATES ET DÉCONNECTÉS

Les entreprises se doivent d'utiliser toute une gamme d'outils différents pour optimiser leur modèle DevSecOps : il leur faut en effet analyser les composants open source (SCA), tester les applications en cours de développement (tests statiques [SAST], tests dynamiques [DAST], tests interactifs [IAST]) et se protéger contre les attaques en phase de production (pare-feu applicatif [WAF]). Il en résulte une courbe d'apprentissage longue et fastidieuse et une faible capacité d'action pour les équipes de développement. Et pour aggraver encore les choses, la sécurité des applications s'intègre rarement aux différents outils de déploiement, de compilation et dans les tests utilisés par les développeurs. Par ailleurs, les développeurs sont généralement notifiés de problèmes de sécurité n'étant pas ou n'étant plus de leur ressort et sans aucun contexte. Ils doivent alors suspendre leur projet en cours pour revenir au code précédent afin de résoudre le problème – c'est ce que l'on appelle un « changement de contexte ».[3]

> ***Plus de la moitié des entreprises déclarent que leur équipe de sécurité a atteint un point de bascule, les outils de sécurité étant si nombreux qu'ils finissent par compromettre l'état de leur cybersécurité et augmenter les risques.[4]***

[3] « Modernize your CI/CD », GitLab, consulté le 25 novembre 2020.

[4] « The rise of cyber security product sprawl », Security Boulevard, 10 mars 2020.

DES INDICATEURS INUTILES

Les indicateurs utilisés aujourd'hui pour évaluer la sécurité des applications n'ont aucune utilité, car ils ne permettent pas de réellement réduire les risques. Il peut s'agir du nombre d'analyses effectuées, d'applications testées et/ou de vulnérabilités découvertes. Cependant, la plupart de ces outils traditionnels échouent à atteindre leur objectif primordial : la correction des vulnérabilités.

Le nombre de vulnérabilités corrigées et le temps moyen de correction sont deux indicateurs illustrant véritablement la sécurité des applications, et qui en disent long sur la maturité d'un programme de sécurité des applications ; des délais de correction plus courts se traduisent par une diminution des risques et de la dette de sécurité. Les entreprises ayant une dette de sécurité considérable (autrement dit des vulnérabilités non corrigées) cumulent les retards et enregistrent un nombre croissant de vulnérabilités – 1,7 fois plus que celles dont la dette de sécurité est inférieure à la moyenne.[5] Une spirale coûteuse et sans fin.

> ***Plus une vulnérabilité est identifiée tardivement dans le développement, plus elle sera coûteuse à corriger – pour un grand fournisseur de solutions de sécurité, le temps moyen de correction s'élève actuellement même à 171 jours.*** [6]

[5] Katharine Watson, « Application Risk is 1.7x Higher for Organizations That Fail To Manage Security Debt », Contrast Security, 24 juillet 2020.
[6] Jeff Williams, « How To Start Decluttering Application Security », Forbes, 27 janvier 2021.

02

Mesurer La Sécurité des Applications Modernes

Pour être efficace, une solution de test de sécurité adaptée aux environnements de développement modernes (par exemple, DevOps, Agile) doit :

- posséder des capacités de correction optimales (tant en termes de nombre de vulnérabilités corrigées que de temps moyen de correction),
- évaluer les risques et la réduction des risques au niveau d'une application, d'un portefeuille d'applications, d'un département ou d'une entreprise,
- couvrir l'ensemble du SDLC,
- pouvoir s'adapter aux technologies modernes et anciennes,
- être utilisable et exploitable par une ressource technologique, quelle qu'elle soit, à toutes les phases du cycle de développement.

Alors que les entreprises cherchent à remplacer les outils de sécurité obsolètes et commencent à examiner leurs programmes au travers du prisme de la réduction des risques plutôt que de la couverture, elles peuvent évaluer les solutions modernes à l'aune de critères des catégories suivantes (voir annexe) :

- Identification et analyse des vulnérabilités
- Analyse de la composition logicielle (SCA)
- Protection de l'exécution
- État de cybersécurité
- Rapports et conformité
- Intégration à grande échelle
- Expérience des développeurs
- Exigences de la plateforme

ASSURER LA SÉCURITÉ DE LA CONCEPTION À LA PRODUCTION

Une plateforme DevSecOps moderne et efficace peut aider les organisations à fluidifier leur pipeline d'intégration et de distribution continues (CI/CD) en identifiant davantage de défauts avérés en temps réel. Elle peut également transformer les développeurs en spécialistes de la sécurité grâce à des conseils de correction faciles à comprendre et à des outils d'interfaces en ligne de commande (CLI). Enfin, elle peut garantir une livraison des applications en toute sécurité, même avec des vulnérabilités ouvertes ou inconnues en production.

> ***Pour être complète, l'approche DevSecOps nécessite une intégration étroite entre la sécurité des applications et les outils DevOps. Mais 45 % des entreprises indiquent avoir des difficultés à assurer la sécurité dans l'ensemble de leur chaîne d'outils DevOps.[7]***

[7] « Modernize your CI/CD », GitLab, consulté le 21 décembre 2020.

03

Appendice : Évaluer et Acheter une Plateforme DevSecOps Moderne

Pour que la plateforme DevSecOps soit efficace dans les environnements de développement modernes, sa couverture doit s'étendre à toutes les parties d'une application. Cela inclut le code personnalisé, les bibliothèques open source, les composants tiers, ainsi que les interfaces de programmation d'applications (API).

La liste de contrôle suivante peut aider les entreprises à élaborer des demandes de proposition (RFP) en vue d'évaluer et d'acheter une solution DevSecOps.

IDENTIFICATION ET ANALYSE DES VULNÉRABILITÉS

- Rendre les vulnérabilités visibles pour les développeurs au moment du développement et des tests.
- Donner la possibilité d'effectuer des tests depuis l'IDE des développeurs, en mode Security Gate et dans les phases de Build.
- Donner l'option de tester le code côté client et côté serveur.
- Fournir des conseils spécifiques pour corriger les vulnérabilités durant le SDLC.
- Mesurer le nombre de chemins parcourus pour identifier les lacunes de couverture.

ANALYSE DE LA COMPOSITION LOGICIELLE (SCA)

- Apporter une visibilité sur les vulnérabilités présentes dans les bibliothèques tierces (c'est-à-dire les composants open source) et recommander des corrections.
- Identifier les risques potentiels liés aux bibliothèques open source et les vulnérabilités associées dans les applications.

PROTECTION DE L'EXÉCUTION

- Protéger toute application vulnérable - aussi bien les applications tierces et les applications développées sur mesure - pendant l'exécution en production.
- Bloquer efficacement l'exploitation des vulnérabilités présentes dans le code.
- Protéger les applications contre le Top 10 des vulnérabilités de l'Open Web Application Security Project (OWASP), comme le cross-site scripting (XSS), l'injection SQL (SQLi), l'injection de commandes, l'injection XML et d'autres vulnérabilités et failles publiques courantes (CVE), sans avoir à appliquer de correctifs.
- Protéger les applications contre les exploits zero-day sans mises à jour disponibles.
- Fournir des points d'intégration avec les systèmes de surveillance les plus courants – Splunk, Securonix, AppDynamics, etc.
- Produire des données complètes (autrement dit horodatage, charge utile, URL, port, IP source/destination, type de violation/événement, ID de session, cookies, trace d'appels).

ÉTAT DE CYBERSÉCURITÉ

- Établir un score de risque pour une application et suivre son évolution dans le temps.
- Évaluer le score de risque par rapport au reste du portefeuille d'applications de l'organisation.
- Faire la distinction entre le score de risque des bibliothèques tierces/open source et celui du code personnalisé.
- Réévaluer le score de risque si la protection en production est activée.

RAPPORTS ET CONFORMITÉ

- Mesurer les taux de correction de la vulnérabilité
- Mesurer les temps de correction moyens
- Mesurer les attaques contre des vulnérabilités spécifiques
- Identifier les applications non conformes aux politiques – PCI (Payment Card Industry), OWASP, etc.
- Répondre directement aux directives de gestion des risques des publications spéciales 800-37 et 800-53 du National Institute of Standards and Technology (NIST) et à l'exigence 6 de la norme PCI DSS pour la conception et la maintenance de systèmes sécurisés.
- Consolider les résultats de plusieurs types de contrôle et politiques
- Prendre en charge des rapports personnalisés sur la plateforme

INTÉGRATION À GRANDE ÉCHELLE

- Vérifier la sécurité des applications sans envoyer le code source ou les binaires ou le bytecode de l'application aux serveurs cloud pour l'analyse
- Définir des politiques personnalisées pour chaque application
- Définir des périodes de grâce (la politique doit laisser le temps aux équipes de corriger les problèmes).
- Appliquer une politique à n'importe quel niveau de l'entreprise (une seule équipe ou un département, ou l'ensemble des structures).
- Définir les degrés de gravité des défauts, les catégories, les faiblesses communes (CWE) et les standards qui composent la politique.
- Inclure l'assistance de l'éditeur pendant les phases d'intégration de l'application, de chargement et de publication des résultats.

Contrast SECURITY

EXPÉRIENCE DES DÉVELOPPEURS

- Organiser l'intégration avec les systèmes de suivi des bugs, les outils de chat, les systèmes de gestion des tickets, ainsi que les outils de conteneurisation (par exemple, Docker, Kubernetes).
- Lancer des évaluations par le biais d'une ligne de commande (CLI)
- Scanner les dépôts de source publics ou privés
- Détecter les vulnérabilités sur l'ordinateur d'un développeur (par exemple : un environnement de développement intégré [IDE]).

EXIGENCES DE LA PLATEFORME

- Proposer une plateforme unique pour gérer toutes les solutions, y compris l'AST (pour le code personnalisé et les bibliothèques open source) et la protection de l'exécution.
- Utiliser les informations sur les attaques actives pour prioriser la correction des vulnérabilités.
- Utiliser les règles de protection de la production pour se défendre contre les CVE non corrigées.
- Utiliser les règles de protection de la production pour se défendre contre les vulnérabilités ouvertes non corrigées par les développeurs.
- Utiliser des règles de protection de la production pour se défendre contre les vulnérabilités zero-day.

Contrast Security est leader du marché de sécurité des applications grâce à une approche moderne qui intègre l'analyse du code et la prévention des attaques au cœur même des applications. Sa technologie d'instrumentation brevetée Deep Security bouscule totalement les approches traditionnelles par son observabilité intégrée et complète en fournissant une évaluation très précise et en assurant une protection continue du portefeuille d'applications tout entier. Ainsi, plus besoin de disposer d'infrastructures gourmandes en performances et d'experts spécialisés en sécurité. La plateforme de sécurité des applications Contrast accélère les cycles de développement, augmente l'efficacité et réduit les coûts, et permet un déploiement rapide tout en protégeant les applications contre les menaces connues et inconnues.

240 3rd Street
2nd Floor
Los Altos, CA 94022
Phone: 888.371.1333
Fax: 650.397.4133

contrastsecurity.com