Analyse des secrets : la checklist

Les 6 composantes essentielles d'une solution d'analyse des secrets « developer-first »

Grâce aux secrets codés en dur, les développeurs peuvent accéder ou authentifier facilement les services nécessaires au développement ou au déploiement d'applications. Mais attention, mal protégés, ces précieux sésames peuvent se retourner contre votre entreprise. Mots de passe, identifiants, clés API, jetons importants… les secrets codés en dur peuvent être exposés dans le code source, les journaux de build, les fichiers d'Infrastructure as Code (IaC), les référentiels, et bien plus encore. Or si ces identifiants tombent entre les mains de personnes mal intentionnées, ces dernières pourraient alors divulguer des données, modifier le code, dérober des informations sensibles, paralyser des services ou faire grimper la facture.

La prolifération des secrets codés en dur, notamment dans des entreprises de développement matriciel et au sein des applications cloud-native, n'a pas échappé aux acteurs malveillants qui en ont fait leur cible de prédilection.

D'où l'utilité des solutions de sécurité des secrets qui vous aident à prévenir ces risques et à adopter un programme complet de sécurité du code. Seulement voilà, tous ces outils ne se valent pas. Leurs capacités d'identification et de protection des secrets varient grandement.

Cette checklist vous présente les six grands critères qu'un fournisseur de solutions d'analyse des secrets doit impérativement remplir.

1. Analyse du code applicatif et des fichiers d'Infrastructure as Code (IaC)

Tout type de code dans votre entreprise peut renfermer des secrets. Votre solution doit donc être à même de passer au crible tous les principaux types de fichier : images de container, modèles IaC et code source. Faute d'une analyse complète, impossible de savoir où précisément se trouvent les secrets exposés dans votre entreprise.

Imaginons que votre solution recherche des secrets dans votre Infrastructure as Code, mais pas dans votre code applicatif. Certes, vos développeurs disposeront de la visibilité nécessaire pour remédier aux secrets compromis dans les modèles et fichiers IaC, mais le risque d'exploitation demeure toutefois si vous possédez du code applicatif ailleurs dans votre codebase.

Par contre, si votre solution passe au peigne fin aussi bien votre IaC que votre code applicatif, vous bénéficierez d'une visibilité à 360° et pourrez ainsi dénicher vos secrets dans les moindres recoins. Autre avantage : la réduction des alertes bénignes grâce à la contextualisation du fichier IaC autour du secret.

Critères recommandés pour le choix d'une solution d'analyse des secrets :

- Examen du code applicatif et de l'IaC
- Enrichissement de l'analyse des secrets grâce à la contextualisation du fichier IaC autour du secret

2. Intégrations intuitives pour les développeurs

Pour améliorer la détection, la remédiation et la prévention de l'exposition des identifiants, le plus simple est d'impliquer les développeurs dans le processus de sécurisation des secrets. Et pour ce faire, rien de tel qu'une solution d'analyse des secrets intégrée en natif aux outils de développement. Fluidité assurée. Ainsi, non seulement vous dotez vos développeurs des moyens de prévenir les problèmes de sécurité des secrets, mais vous réduisez aussi le context-switching tellement dommageable pour la productivité et le bien-être des développeurs.

En somme, au moment de choisir votre solution, veillez à inscrire au cahier des charges l'intégration directe du feedback dans les outils de développement et les workflows DevOps.

Critères recommandés pour le choix d'une solution d'analyse des secrets :

- Intégration en natif aux outils de développement existants, tels que les systèmes de contrôle de versions (VCS) et les environnements de développement intégré (IDE)
- Intégration aux workflows DevOps (p. ex. : pipelines CI/CD)
- Découverte des secrets exposés, mais aussi de leur contexte pour accélérer la priorisation des risques et la remédiation
- Blocage des envois de secrets vers un référentiel avant l'ouverture d'une pull request via un hook pre-commit ; découverte des secrets exposés dans le cadre d'une analyse de pull request

3. Adoption d'une approche multidimensionnelle de l'analyse des secrets

Parmi les identifiants les plus couramment reconnus comme exposés figurent les secrets recourant à des expressions régulières : jetons d'accès, clés API, clés de chiffrement, jetons OAuth, certificats, et bien d'autres encore. Grâce à l'analyse des expressions régulières, les solutions de sécurité des secrets déterminent si une chaîne donnée suit bien le schéma d'autres secrets de ce type (p. ex : clé d'accès AWS). Le problème c'est que les secrets peuvent adopter d'autres formes moins prévisibles qui compliquent toute identification systématique.

D'où la nécessité pour une solution complète de ne pas se limiter à l'analyse des expressions régulières et d'inclure également l'analyse des mots clés. Cela permet aux chaînes généralement associées à un secret, comme « password », d'être détectées. Autre forme plus avancée de sécurité des secrets, l'analyse des schémas d'entropie élevée consiste à évaluer les identifiants potentiellement exposés. L'objectif : déterminer s'ils sont suffisamment différents du langage réel pour se révéler être un secret (ex. : `EuN21!HHvaS%JPTQU&cx`.) Le bémol c'est que ce type d'analyse est aussi le plus complexe en raison de l'avalanche de faux positifs qu'il peut entraîner. Un point que nous examinerons plus en détail dans la partie suivante.

En clair, pour identifier l'intégralité de vos secrets exposés, vous n'avez d'autre choix que de miser sur une solution appliquant une triple analyse : expressions régulières, mots clés et entropie. Attention toutefois, nombre d'outils opèrent de manière disparate en privilégiant soit une seule catégorie de secrets soit une phase spécifique du cycle de développement. Ce procédé peut générer une cacophonie de faux positifs et permettre à certains secrets de passer entre les mailles du filet.

Par ailleurs, l'efficacité d'une solution de sécurité des secrets dépend à la fois de la portée de ses détecteurs et de la richesse de son analyse. Si votre outil s'appuie sur une vaste bibliothèque de signatures pour détecter et signaler un large éventail de secrets à l'aide d'expressions connues et prévisibles, alors il vous offrira l'ampleur d'analyse dont vous avez besoin.

Critères recommandés pour le choix d'une solution d'analyse des secrets :

- Recours à une analyse des expressions régulières, des mots clés et de l'entropie
- Utilisation de détecteurs de secrets spécifiques à un domaine
- Fonctionnalités s'appuyant sur une vaste bibliothèque de politiques basées sur les signatures
- Détection continue des identifiants exposés tout au long du cycle de développement, de la phase de build au runtime
- Examen de l'ensemble des fichiers de code source et des historiques de version pour mettre au jour les secrets enfouis dans le codebase

4. Reconnaissance affinée des schémas d'entropie élevée

Si les analyses des expressions régulières et des mots clés permettent de détecter une forte proportion des identifiants exposés, ces méthodes laisseront toutefois filer les secrets dépourvus de schémas cohérents ou identifiables. Avec ces techniques d'analyse basée sur les signatures, les chaînes aléatoires de mots de passe et de noms d'utilisateurs passeront sous le radar. Pour pallier ce problème et bénéficier d'une analyse des secrets multidimensionnelle, votre solution doit recourir également à l'analyse des secrets basée sur l'entropie qui s'appuie sur un modèle d'entropie optimisé.

Concrètement, ce type d'outil vise à déterminer dans quelle mesure une chaîne diffère d'un mot de la langue anglaise. Et lorsque c'est le cas, la chaîne est considérée comme un secret exposé. Côté pile, ces analyses mettent au jour des secrets que d'autres outils auraient laissé passer. Côté face, cette méthode basée sur l'entropie tend à générer des faux positifs. Imaginons que vous attribuez un nom particulièrement long à une variable, comme `redirectForOptimization`. Cette variable ne s'apparentant pas à un mot, elle risque d'être signalée, à tort, comme un secret par une solution d'analyse basée sur l'entropie.

Heureusement, ce taux de faux positifs est réductible de plusieurs manières. La première consiste à miser sur une solution capable d'optimiser ce modèle d'entropie à l'aide du contexte de la chaîne pour identifier plus précisément les types de secrets complexes. Votre outil devrait également combiner l'analyse par mots clés et celle basée sur l'entropie. L'adoption d'une approche multidimensionnelle présente l'avantage de réduire le taux de faux positifs tout en offrant une couverture plus complète. Ainsi, aucun secret exposé ne vous échappera plus.

Critères recommandés pour le choix d'une solution d'analyse des secrets :

- Mode d'analyse reposant à la fois sur les signatures et sur l'entropie
- Prise en compte du contexte de la chaîne par le modèle d'entropie pour mieux détecter les types de secrets complexes et réduire le taux de faux positifs

5. Partie intégrante d'une solution complète de sécurité du code

En fait, l'analyse des secrets n'est qu'une composante d'une solution plus large de sécurité du code. Vu le casse-tête que représentent les solutions disparates censées sécuriser chaque élément d'un codebase cloud-native, mieux vaut miser sur une solution unifiée de sécurité du code pour plus d'efficacité.

Analyse des secrets, sécurité de l'IaC, analyse de la composition logicielle (SCA)… la multiplication des cas d'usage de sécurité du code vous expose à une prolifération des outils qui crée d'importantes lacunes. En outre, elle complique la remédiation des secrets exposés, des vulnérabilités et des erreurs de configuration sur tout le cycle de développement. Par ailleurs, cet assemblage hétéroclite d'instruments génère des informations parasites qui déconcentrent les développeurs. Lorsque les équipes utilisent des outils de

sécurité disparates qui traitent la sécurité des applications et de l'infrastructure séparément, il devient difficile de prioriser les problèmes en fonction du niveau d'exposition.

Critères recommandés pour le choix d'une solution d'analyse des secrets et de sécurité du code :

- Intégration de l'analyse des secrets à une solution unifiée de sécurité du code qui consolide tous les outils et garantit une couverture intégrale
- Détection en continu des secrets exposés, des erreurs de configuration au sein des fichiers IaC et des vulnérabilités présentes dans le code open-source et les images de container
- Visibilité à 360° sur l'ensemble des dépendances et des problèmes de sécurité grâce à un graphique de la supply chain
- Surveillance et prévention des vulnérabilités et des erreurs de configuration tout au long du cycle de développement pour une couverture du code jusqu'au cloud

6. Connexion de l'environnement d'exécution avec une solution de gestion des droits sur l'infrastructure cloud

Pour une remédiation sans souci des identités trop permissives et de la prolifération des secrets, votre entreprise doit miser sur une solution d'analyse des secrets alignée sur un outil de gestion des droits sur l'infrastructure cloud (CIEM). Ce dernier vous permet de gérer les identités et les privilèges au sein des environnements cloud, et de comprendre les différents droits d'accès qui régissent vos environnements cloud et multicloud. Grâce à cette visibilité, vous pouvez identifier et réduire les risques liés aux droits inutilisés ou trop permissifs. Une solution CIEM complète inclut notamment le calcul des autorisations effectives qui vous éclaire sur qui a le droit de faire quoi dans votre entreprise. Mais ce n'est pas tout : elle jugule la multiplication des autorisations et fait appel à la puissance du machine learning (ML) avancé et des politiques de détections des anomalies basées sur les comportements tout en combinant les données relatives au runtime et à l'IaC.

Ainsi munie d'une solution unique alliant CIEM et analyse des secrets, votre entreprise bénéficiera d'une sécurité des identités cloud complète. Visibilité sur les utilisateurs autorisés à accéder aux ressources sensibles (key vault, secrets, etc.), compréhension de la manière dont sont utilisés les secrets, réduction des autorisations inutilisées et trop permissives susceptibles de nuire à la sécurité… les avantages sont multiples.

Critères recommandés pour le choix d'une solution conjuguant CIEM et analyse des secrets :

- Alignement du feedback de l'analyse des secrets sur les fonctionnalités CIEM de visibilité, de suivi et d'ajustement des autorisations
- Informations contextualisées sur les identités utilisant certains types de secrets
- Réduction des autorisations inutilisées et excessives

Lorsque vous évaluez ce type d'offres, gardez ces éléments à l'esprit afin de bénéficier de l'analyse des secrets – multidimensionnelle et full stack – qu'il vous faut pour prévenir les risques associés aux identifiants exposés.

paloalto® NETWORKS

Oval Tower, De Entrée 99 – 197
1101HE Amsterdam, Pays-Bas

Téléphone :
+31 20 888 1883

www.paloaltonetworks.fr