Politique d’entreprise pour la signature de code

Mis à jour le 22 mai 2025

1. Présentation

De nombreux groupes au sein des organisations publient des logiciels pour un usage interne et externe. Afin de se protéger contre toute altération des logiciels et de confirmer l’authenticité des différentes versions, les bonnes pratiques stipulent que les binaires des logiciels doivent être signés à l’aide de clés cryptographiques. Toutefois, la confiance dans ces signatures nécessite des contrôles et une surveillance appropriés des processus de signature de code.

Cette politique est destinée à aider toute organisation qui crée et distribue des logiciels, même si l’appellation « entreprise » peut évoquer une structure de plus grande taille.

2. Objectif

Cette politique décrit les exigences et les procédures de contrôle des processus de signature du code logiciel afin de garantir que seuls le code autorisé et vérifié est signé et distribué, renforçant par là même la sécurité et l’intégrité des logiciels.

Elle permettra à l’organisation concernée de savoir ce qu’elle publie et de s’assurer que le logiciel publié n’introduit aucun risque de sécurité inacceptable. Elle permettra également aux utilisateurs de vérifier l’intégrité des fichiers publiés et de s’assurer que ces derniers n’ont pas été altérés. De même, elle alertera les utilisateurs si un logiciel n’a pas été créé par l’éditeur officiellement associé au code.

Une politique rigoureuse de signature de code et de bonnes pratiques de documentation peuvent également faciliter le travail des auditeurs, tant internes qu’externes.

3. Champ d’application

Cette politique s’applique à l’ensemble des collaborateurs, des processus et des systèmes intervenant dans le processus de signature de code au sein de l’organisation, y compris les développeurs internes et externes, les signataires, les relecteurs et les auditeurs. Il s’applique au code développé par l’organisation pour un usage interne ou une vente externe, ainsi qu’aux composants logiciels créés ou utilisés dans le cadre du cycle de développement logiciel (SDLC) et du processus CI/CD (par exemple, les SBOM, les scripts, etc.).

Elle ne s’applique pas aux logiciels tiers que l’organisation utilise pour le développement ou d’autres activités (par exemple, des applications de productivité comme Microsoft Office ou des outils de développement comme JFrog).

4. Accès et privilèges

4.1 Authentification

Les utilisateurs qui accèdent à l'environnement de développement logiciel doivent utiliser des services ou des méthodes d'authentification approuvés. Il est interdit d'utiliser des comptes partagés, des identifiants stockés de manière non sécurisée, des comptes locaux ou d'autres moyens d'authentification qui enfreignent la politique de l'entreprise dans ce domaine.

Conformément aux exigences de base du CA/B Forum, l'authentification multifacteur (MFA) ou l'authentification de serveur à serveur doit être utilisée pour accéder à la signature et l'autoriser.

4.2 Définition des rôles – Contrôle d'accès basé sur les rôles (RBAC)

Les bonnes pratiques stipulent que l'organisation doit établir des rôles distincts, avec des privilèges définis, pour les activités de signature de code. Ces rôles doivent au minimum inclure le déposant (submitter), le signataire et le responsable de la sécurité, mais aussi s'étendre au relecteur, à l'auditeur et à l'administrateur du système.

4.3 Séparation des fonctions

Les utilisateurs peuvent avoir plusieurs rôles sur la base de pratiques commerciales raisonnables. Toutefois, aucun utilisateur ne peut avoir le contrôle exclusif d'un processus. Une telle pratique introduit en effet un point de défaillance unique (SPOF) en matière de sécurité.

Limitez l'accès au strict minimum des privilèges nécessaires afin de renforcer votre contrôle sur la sécurité, conformément au principe du moindre privilège.

4.4 Limiter l'accès aux ressources critiques

Tous les accès en général, et en particulier l'accès aux ressources privilégiées comme les serveurs de compilation et les modules HSM, doivent être octroyés sur la base des rôles définis et non d'utilisateurs spécifiques (ou de groupes sans lien avec ces ressources comme des administrateurs ou des membres du Comex).

Les ressource critiques doivent être soumises aux principes du moindre accès et du moindre privilège afin de renforcer le contrôle sur la sécurité.

4.5 Actions à haut risque

Certaines opérations (exportation de paires de clés, suppression de paires de clés et révocation de certificats) peuvent présenter des risques élevés pour la sécurité, voire entraîner des perturbations majeures dans les processus si elles ne sont pas effectuées correctement. Ces types d'actions peuvent justifier d'imposer plusieurs autorisations pour la signature, et ce en plus de privilèges de signature fortement restreints.

4.6 Risque de dommages causés par des logiciels compromis

Le risque de chaque produit ou projet logiciel doit être évalué au regard des dommages potentiels occasionnés si le logiciel venait à contenir un code malveillant ou des vulnérabilités susceptibles d'entraîner un accès non autorisé aux données, une perte de fonctionnalité du système ou une autre défaillance (par exemple, un logiciel avec accès au cœur du système, tel que l'accès root dans Linux).

Les logiciels présentant un risque suffisamment élevé peuvent justifier d'imposer plus d'une autorisation de signature, et ce en plus de privilèges de signature fortement restreints.

5. Gestion des certificats et des paires de clés

5.1 Environnements logiciels

Des paires de clés et des certificats de signature de code uniques doivent être utilisés pour les environnements de production et hors production. Les environnements de test qui sont accessibles à distance et qui traitent des données sensibles doivent être traités comme des environnements de production.

5.2 Stockage des certificats et des paires de clés

Les clés de signature du code de production doivent être générées et stockées en toute sécurité, à l'aide d'un module de sécurité matériel (HSM) conforme à la norme FIPS 140-2 niveau 2 ou au Common Criteria EAL 4+, et utilisées conformément aux politiques de l'entreprise en matière de gestion cryptographique et de certificats.

Les utilisateurs multiples ne sont pas autorisés à partager les clés de signature de code en production, sauf si le système de journalisation des activités permet d'enregistrer et d'identifier les actions d'utilisateurs uniques, y compris les utilisateurs des services.

5.3 Inventaire et cycle de vie des certificats

Maintenez un inventaire à jour de tous les certificats de signature de code. Surveillez les dates d'expiration et générez de nouveaux certificats de manière proactive.

De nouveaux certificats et de nouvelles clés peuvent être créés en toute sécurité et déployés automatiquement par le biais de la plupart des systèmes de gestion du cycle de vie des certificats (CLM).

5.4 Rotation des clés

La rotation des clés améliore la sécurité en réduisant l'impact en cas de compromission d'une clé. Cette rotation peut être effectuée manuellement, mais il est plus facile et plus rapide d'automatiser le processus.

5.5 Chiffrement des clés

Chiffrez les clés au niveau cryptographique le plus élevé que l'utilisateur final pourra déchiffrer. Encouragez ces utilisateurs à mettre à niveau leurs options de déchiffrement, de manière à pouvoir utiliser des algorithmes de chiffrement post-quantique (PQC) lorsque ceux-ci deviendront disponibles.

5.6 PKI publique vs PKI privée

Les logiciels utilisés ou accessibles en dehors de votre organisation doivent être signés à l'aide d'une clé publique qui peut être facilement vérifiée à l'aide de listes PKI standard.

Les logiciels qui ne seront utilisés et accessibles que sur le réseau interne de l'entreprise peuvent utiliser une PKI privée, dans laquelle les certificats sont rattachés à une racine de confiance, mais sont émis par l'entreprise via une autorité de certification intermédiaire (ICA). Il s'agit d'un cas d'usage courant pour les logiciels développés et utilisés uniquement en interne.

6. Processus de signature du code

6.1 Revues de sécurité

Le code source du logiciel doit faire l'objet d'une revue de sécurité, conformément à la politique de sécurité régissant le cycle de développement logiciel (SDLC).

Chaque version et ses composants associés doivent être analysés pour détecter tout indicateur de malveillance et des vulnérabilités connues. Ne signez pas un logiciel s'il contient un malware confirmé ou une vulnérabilité exploitable.

6.2 Processus automatisés

La meilleure pratique consiste à automatiser le processus de signature de code dans le cadre du SDLC et du pipeline CI/CD. Utilisez une autorité de certification (AC) agréée pour les certificats de signature de code. Dans l'idéal, les développeurs devraient utiliser une clé de signature unique pour signer tout code qu'ils soumettent. Les signatures individuelles doivent être vérifiées avant de signer la version dans son ensemble.

6.3 Horodatage

Les versions qui resteront valables après l'expiration du certificat de signature de code doivent inclure un horodatage provenant d'un service approuvé afin de prouver l'heure de la signature. Assurez-vous que vos horodatages et votre autorité d'horodatage (TSA) répondent aux nouvelles exigences de sécurité du CA/B Forum, lesquelles sont entrées en vigueur en 2025.

6.4 Signature des éléments logiciels publiés de la version et des nomenclatures logicielles (SBOM)

Les versions des éléments logiciels publiés , incluant la branche de code, la SBOM, les scripts de compilation et les documents VEX (Vulnerability Exploitability eXchange), doivent être signés et stockés afin de faciliter la détection de toute altération des évaluations et des archives.

Les composants logiciels doivent être stockés de manière sécurisée et peuvent exiger de suivre certaines méthodes pour le partage avec les clients ou les autorités de gouvernance. Les SBOM, en particulier, peuvent faire partie d'un « quote package » ou d'une soumission de produit.

6.5 Audits et journaux

Conserver un journal d'audit complet incluant les vérifications de code, les scans de sécurité et toute autre action sur la clé de signature du code. Les activités doivent être associées à des utilisateurs et à des comptes de service uniques. La signature et toutes les actions associées aux certificats, aux paires de clés, aux utilisateurs et aux modèles doivent être enregistrées dans les journaux.

7. Crypto-agilité – Préparation et réponse au changement

7.1 Préparation et réponse aux incidents et aux changements inattendus

Définissez et automatisez autant de procédures que possible pour documenter, révoquer et réémettre rapidement les clés compromises.

Définissez et automatisez les procédures de recertification et de redéploiement d'une version avec les composants logiciels qui ont été signés avec la clé compromise.

7.2 Préparation et action face aux changements attendus

Les réglementations sont en constante évolution. Cependant, elles sont souvent introduites progressivement ou donnent suffisamment de temps pour modifier les systèmes. Définissez et automatisez des procédures de documentation et de modification progressive, par exemple en effectuant des mises à jour lors de la réémission d'un certificat arrivant à expiration.

8. Revues et mises à jour

Cette politique sera revue chaque année et mise à jour si nécessaire afin de garantir son efficacité et son adéquation avec les standards du secteur et les changements organisationnels.

Téléchargez une version Word de ce fichier pour votre usage personnel en cliquant ici