Code-Signing-Richtlinie für Unternehmen

Aktualisiert am 22. Mai 2025

1. Überblick

In modernen Unternehmen veröffentlichen verschiedene Gruppen Software für den internen und externen Gebrauch. Zum Schutz vor Softwaremanipulation und zur Bestätigung der Authentizität von Softwareversionen sollten binäre Softwaredateien mit Hilfe von kryptografischen Schlüsseln signiert werden. Das Vertrauen in diese Signaturen setzt jedoch angemessene Kontroll- und Überwachungsmaßnahmen bei den Code-Signing-Prozessen voraus.

Die vorliegende Richtlinie richtet sich an jede Organisation, die Software erstellt und vertreibt, auch wenn die hier verwendete Bezeichnung „Unternehmen“ auf große Organisation hinweisen mag.

2. Zweck

Diese Richtlinie umreißt die Anforderungen und Verfahren für die Kontrolle von Prozessen beim Software-Code-Signing, mit denen sichergestellt werden soll, dass nur autorisierter, überprüfter Code signiert und verteilt wird. Das Ziel ist die Verbesserung der Sicherheit und Integrität der Software.

Diese Richtlinie stellt sicher, dass die Organisation weiß, was sie veröffentlicht, und dass die veröffentlichte Software keine inakzeptablen Sicherheitsrisiken birgt. Nutzer können die Integrität der veröffentlichten Dateien überprüfen, um nachzuweisen, dass sie nicht manipuliert wurden. Darüber hinaus werden Nutzer gewarnt, wenn die Software nicht von dem Herausgeber erstellt wurde, der als Codeersteller angegeben ist.

Eine strikte Code-Signing-Richtlinie mit sorgfältig dokumentierten Prozessen kann zudem internen und externen Prüfern die Arbeit erleichtern.

3. Geltungsbereich

Diese Richtlinie gilt für alle Mitarbeiter, Prozesse und Systeme, die am Code-Signing-Prozess innerhalb der Organisation beteiligt sind, einschließlich interner und externer Entwickler, Signierer, Prüfer und Auditoren. Sie gilt für Code, der von der Organisation für den internen Gebrauch oder den externen Verkauf entwickelt wird, sowie für Softwareartefakte, die im SDLC- und CI/CD-Prozess erstellt oder verwendet werden (z. B. SBOMs, Build Scripts).

Sie gilt nicht für Software von Drittanbietern, die das Unternehmen für die Entwicklung oder andere Aktivitäten verwendet (z. B. Produktivitätstools wie Microsoft Office oder Entwicklertools wie JFrog).

4. Zugriff und Zugriffsrechte

4.1 Authentifizierung

Nutzer, die auf die Softwareentwicklungsumgebung zugreifen, müssen genehmigte Authentifizierungsdienste und -methoden verwenden. Sie dürfen keine gemeinsam genutzten Konten, unsicher gespeicherte Anmeldeinformationen, lokalen Konten oder andere Authentifizierungsmethoden nutzen, die gegen die Authentifizierungsrichtlinie des Unternehmens verstoßen.

Für den Zugriff und die Autorisierung der Signierung muss gemäß den Mindestanforderungen des CA/Browser Forum eine Multifaktor-Authentifizierung oder eine Server-zu-Server-Authentifizierung verwendet werden.

4.2 Definieren von Rollen – Rollenbasierte Zugriffskontrolle (RBAC)

Im Rahmen von Best Practices sollten Organisationen separate Rollen mit definierten Zugriffsrechten für Code-Signing-Aktivitäten einrichten. Es sollte mindestens Rollen für Einreicher, Signierer und Sicherheitsbeauftragte geben, die dann um Rollen für Prüfer, Auditoren und Systemadministratoren erweitert werden.

4.3 Aufgabentrennung

Nutzer können im Rahmen wirtschaftlich vertretbarer Bedingungen mehrere Rollen haben. Allerdings darf kein einzelner Nutzer die alleinige Kontrolle über einen Prozess haben, da dies eine zentrale Schwachstelle verursachen würde.

Beschränken Sie den Zugriff auf ein Minimum an Zugriffsrechten, um eine striktere Sicherheitskontrolle gemäß dem Least-Privilege-Prinzip zu gewährleisten.

4.4 Beschränkung des Zugriffs auf wichtige Ressourcen

Jeglicher Zugriff, insbesondere aber der Zugriff auf Ressourcen, die erweiterte Zugriffsrechte erfordern, wie Build-Server oder Funktionen für das HSM-Management, sollte definierten Rollen zugewiesen werden und nicht einzelnen Nutzern (oder unspezifischen Gruppen wie „Administratoren“ oder „Führungsebene“).

Um eine bessere Kontrolle über die Sicherheit zu haben, sollte den wichtigsten Ressourcen nur das Minimum an Zugriff und Zugriffsrechten eingeräumt werden.

4.5 Vorgänge mit hohem Risiko

Bestimmte Vorgänge, wie z. B. der Export oder das Löschen von Schlüsselpaaren und der Widerruf von Zertifikaten, können ein hohes Sicherheitsrisiko darstellen bzw. erhebliche Störungen in den Abläufen verursachen, wenn sie nicht korrekt ausgeführt werden. Bei derartigen Vorgängen kann es sinnvoll sein, nicht nur die Signierberechtigungen stark zu beschränken, sondern auch mehrere Autorisierungsschritte für das Signieren einzurichten.

4.6 Schadensrisiko durch kompromittierte Software

Das mit einem Softwareprodukt oder -projekt verbundene Risiko muss auf der Grundlage des potenziellen Schadens bewertet werden, der entstehen würde, wenn die Software schädlichen Code oder ausnutzbare Schwachstellen enthielte, die zu unbefugtem Zugriff auf Daten, zum Verlust von Systemfunktionen oder zu anderen Fehlern führen würden (z. B. Software mit Low-Level-Systemzugriff wie Root-Zugriff in Linux).

Bei Software mit hohem Risiko kann es gerechtfertigt sein, nicht nur die Signierberechtigungen stark zu beschränken, sondern auch eine mehrstufige Autorisierung für das Signieren einzurichten.

5. Management von Zertifikaten und Schlüsselpaaren

5.1 Software-Umgebungen

Sowohl für Nicht-Produktions- als auch für Produktionsumgebungen müssen eindeutige Code-Signing-Schlüsselpaare und Zertifikate verwendet werden. Testumgebungen, die Remote zugänglich sind und sensible Daten verarbeiten, müssen wie Produktionsumgebungen behandelt werden.

5.2 Speicherung von Zertifikaten und Schlüsselpaaren

Signing-Keys für Produktionscode müssen auf sichere Weise generiert und gespeichert werden, und zwar über ein Hardware-Sicherheitsmodul (HSM), das FIPS 140-2 Level 2 oder dem Standard Common Criteria EAL 4+ entspricht und gemäß den Unternehmensrichtlinien für kryptografisches Management und Zertifikate verwendet wird.

Signing-Keys für Produktionscode dürfen nicht von mehreren Nutzern gemeinsam verwendet werden, es sei denn, die Aktivitätsprotokollierung kann die Aktivitäten einzelner Nutzer, einschließlich der Nutzer von Diensten, aufzeichnen und identifizieren.

5.3 Bestandsaufnahme und Lebenszyklus von Zertifikaten

Führen Sie ein Inventar aller Code-Signing-Zertifikate. Überwachen Sie Ablaufdaten und erstellen Sie proaktiv neue Zertifikate.

Neue Zertifikate und Schlüssel können über die meisten CLM-Systeme (Certificate Lifecycle Management) sicher erstellt und automatisiert in Betrieb genommen werden.

5.4 Schlüsselrotation

Das Rotieren von Schlüsseln verbessert die Sicherheit, indem die Folgen minimiert werden, die sich durch einen kompromittierten Schlüssel ergeben. Die Schlüsselrotation kann manuell erfolgen, gleichzeitig ist es einfacher und schneller, den Prozess zu automatisieren.

5.5 Key-Verschlüsselung

Verschlüsseln Sie die Schlüssel auf der höchstmöglichen Stufe, basierend auf den Entschlüsselungsmethoden, die ein Endnutzer verwenden kann. Fordern Sie die Gegenseite dazu auf, ihre Entschlüsselungsoptionen auf den aktuellen Stand zu bringen, damit PQC-fähige Verschlüsselungsalgorithmen genutzt werden können, sobald sie verfügbar sind.

5.6 Unterschied zwischen öffentlichen und privaten PKI

Software, die außerhalb Ihres Unternehmens verwendet wird oder auf die außerhalb Ihres Unternehmens zugegriffen wird, sollte mit einem öffentlichen Schlüssel signiert werden, der durch Standard-PKI-Listen leicht überprüft werden kann.

Software, auf die nur über ein Unternehmensnetzwerk zugegriffen wird, kann eine private PKI verwenden, bei der die Zertifikate an eine vertrauenswürdige Stammzertifizierungsstelle gekoppelt sind, aber vom Unternehmen über eine zwischengeschaltete Zertifizierungsstelle (ICA) ausgestellt werden. Dies ist eine Best Practice für Software, die intern entwickelt und nur intern verwendet wird.

6. Code-Signing-Prozess

6.1 Sicherheitsüberprüfungen

Der Software-Quellcode muss eine Sicherheitsüberprüfung gemäß den Sicherheitsrichtlinien des Softwareentwicklungszyklus (Software Development Life Cycle; SDLC) bestehen.

Jede Version und die zugehörigen Komponenten müssen auf Schadindikatoren und bekannte Schwachstellen gescannt werden. Signieren Sie keine Software, die nachweislich Malware oder eine ausnutzbare Sicherheitslücke enthält.

6.2 Automatisierte Prozesse

Am besten ist es, den Code-Signing-Prozess als Teil des SDLC und der CI/CD-Pipeline zu automatisieren. Verwenden Sie eine zugelassene Zertifizierungsstelle (CA) für Code-Signing-Zertifikate. Idealerweise sollten Entwickler einen eindeutigen Signierschlüssel verwenden, um jeden Code zu signieren, den sie einchecken. Die einzelnen Signaturen müssen überprüft werden, bevor ein Release signiert wird.

6.3 Zeitstempel

Releases, die nach Ablauf des Code-Signing-Zertifikats gültig sind, müssen einen Zeitstempel eines zugelassenen Zeitstempeldienstes enthalten, um den Zeitpunkt der Signierung nachzuweisen. Stellen Sie sicher, dass Ihre Zeitstempel und die TSA (Timestamp Authority) den neuen Sicherheitsanforderungen des CA/Browser Forum entsprechen, die seit April 2025 gelten.

6.4 Signieren von Release-Artefakten und Software-Stücklisten (SBOMs)

Release-Artefakte wie Code-Branches, SBOMs, Build-Skripte und VEX-Dokumente (Vulnerability Exploitability eXchange) müssen signiert und gespeichert werden, um Manipulationen an Stellungnahmen und Archiven zu erkennen.

Artefakte sollten sicher aufbewahrt werden und es müssen möglicherweise Methoden zum Teilen mit Kunden oder Leitungsgremien bereitgestellt werden. SBOMs können auch Teil eines Service- oder Produktangebots sein.

6.5 Protokollierung und Nachverfolgung

Pflegen Sie einen umfassenden Prüfpfad mit Nachweisen für Code-Reviews, Sicherheitsscans und Aktivitäten rund um Code-Signing-Keys. Aktivitäten müssen einzelnen Nutzern und Dienstkonten zugeordnet werden. Signiervorgänge und alle Aktionen im Zusammenhang mit Zertifikaten, Schlüsselpaaren, Nutzern und Vorlagen sollten in den Protokollen festgehalten werden.

7. Krypto-Agilität: Vorbereitung und Reaktion auf Veränderungen

7.1 Vorbereitung und Reaktion auf Zwischenfälle und unerwartete Veränderungen

Definieren und automatisieren Sie so viele Abläufe wie möglich, um kompromittierte Schlüssel schnell dokumentieren, widerrufen und neu ausstellen zu können.

Definieren und automatisieren Sie Abläufe zum Neusignieren und erneuten Bereitstellen von Code und Software-Artefakten, die mit einem kompromittierten Schlüssel signiert wurden.

7.2 Vorbereitung und Reaktion auf vorhersehbare Veränderungen

Vorschriften ändern sich ständig. Allerdings werden sie oft schrittweise eingeführt oder geben Zeit für Systemänderungen. Definieren und automatisieren Sie Verfahren zur Dokumentation und schrittweisen Durchführung von Änderungen, z. B. durch Aktualisierungen bei der Neuausstellung eines auslaufenden Zertifikats.

8. Überprüfung und Aktualisierung

Diese Richtlinie wird jährlich überprüft und bei Bedarf aktualisiert, um ihre Wirksamkeit und Anpassung an Branchenstandards und organisatorische Veränderungen zu gewährleisten.