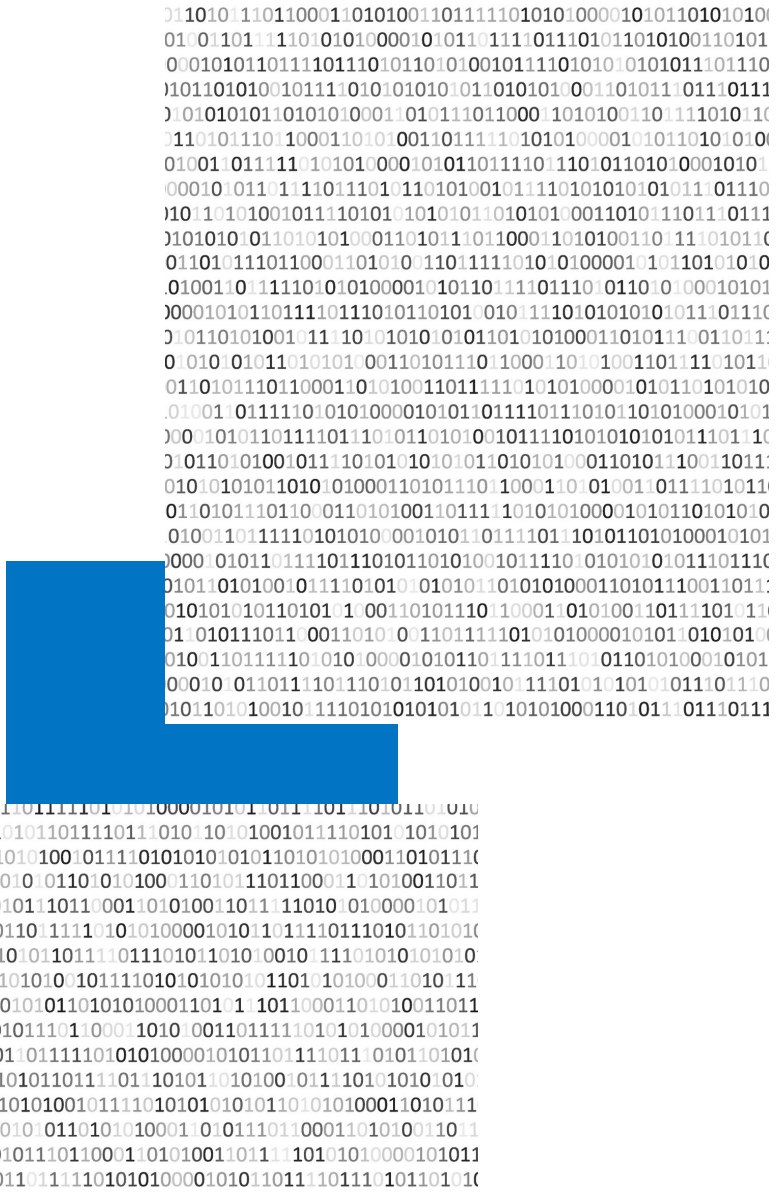
RICHTLINIEN-LEITFADEN

LEITFADEN FÜR EINE WIRKUNGSVOLLE RICHTLINIE ZUR SIGNIERUNG VON CODE UND SOFTWARE



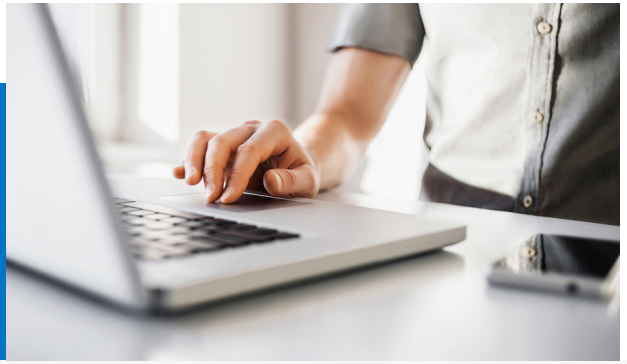
digicert®

SOFTWARE- UND CODE-SIGNING: DIE BEDEUTUNG EINER WIRKUNGSVOLLEN, DOKUMENTIERTEN SIGNIERRICHTLINIE

Code-Signing ist ein wichtiger Schritt bei der Entwicklung und zum Schutz von Software – aber viele Entwicklungsteams haben Mühe mit der Umsetzung einer konsistenten Signierpraxis. Vor allem, weil es die falsche Wahrnehmung gibt, dass die Signierung mit den Markteinführungszielen einer flexiblen Entwicklungspraxis nicht zu vereinbaren sei. Außerdem legt unsere Untersuchung nahe, dass in vielen Softwareentwicklungsteams keine umfassende Richtlinie für das Signierverfahren dokumentiert ist, sondern sie vielmehr verallgemeinerte Richtlinien und implizite gängige Praktiken nutzen.

Da der Bedarf an ausreiften Signierpraktiken zunimmt, müssen Unternehmen und Entwicklungsteams robuste Richtlinien für die Signierung sowie Prozesse zum Schutz ihrer Software erstellen und umsetzen. Mit der Erstellung und Dokumentation einer formellen Richtlinie für die Signierung von Code versetzen Sie Ihre Entwickler und Ingenieure in die Lage, die zum Schutz der von Ihnen entwickelten Software notwendigen Best Practices zu befolgen.

Dieser Leitfaden soll Sie dabei unterstützen, eine Strategie zu formulieren, die Software, Prozesse und Richtlinien vereint, um in Ihrer Signierpraxis durchgängigen Schutz zu erreichen. Als Grundlage für diesen Leitfaden haben wir im Rahmen von Umfragen und persönlichen Gesprächen Dutzende DevOps-Experten, Teamleiter und Experten für Softwaresicherheit von Unternehmen unterschiedlichster Größe aus verschiedenen Branchen und Ländern befragt. Mit ihrer Hilfe haben wir einen nützlichen Grundstandard für die Signierung bestimmt, der von praktisch jedem Softwareentwicklungsteam in jedem beliebigen Unternehmen angewandt werden kann.



Zweck und Umfang dieses Leitfadens

Im ersten Teil werden wesentliche Konzepte vorgestellt, mit denen Sie das Fundament einer Signierrichtlinie legen können, die sich an den Anforderungen Ihres Teams orientiert. Der Schwerpunkt liegt dabei darauf, ein Verständnis zu schaffen, wie Sie mit Code-Signing das Beste für die von Ihrem Team erstellte Software umsetzen können.

Im zweiten Teil wird die schriftliche Ausarbeitung einer Richtlinie skizziert. Dabei wird auf die Praktiken und Verfahren eingegangen, die die von uns befragten führenden Experten für Softwaresicherheit und DevOps als wesentlich einstufen.

Am Ende dieses Dokuments erhalten Sie eine Liste, anhand derer Sie Ihre eigene Richtlinie erstellen können.

TEIL I

WIE DENKEN SIE ÜBER GUTE SOFTWARESICHERHEIT?

Entwicklungsteams brauchen eine klare Vorstellung von ihrem Ziel

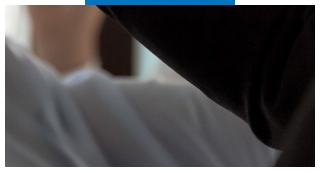
Welches Ziel hat Ihr Softwareentwicklungsteam?

Die Antwort auf diese Frage ist möglicherweise nicht so offenkundig, wie es zunächst scheint. Allzu oft steht das Ziel des Unternehmens oder anderer Teams oder Unternehmensbereiche an der Stelle des Ziels des Softwareentwicklungsteams, und auch wenn die Ziele einzelner Teams dem Unternehmensziel dienen sollen, können die beiden nicht gegeneinander ausgetauscht werden. Genauso wie die Finanzabteilung mit den Prinzipien und Richtlinien der Personalabteilung nicht optimal arbeiten kann und der Personalabteilung ein Vertriebssystem nicht weiterhilft, kann das Softwareentwicklungsteam seine Leistungsfähigkeit nicht voll entfalten, wenn es mit den Prinzipien und Richtlinien anderer Teams oder den übergeordneten Grundsätzen des Gesamtunternehmens arbeiten muss.

Ein Softwareentwicklungsteam arbeitet am besten basierend und ausgerichtet auf eine wirkungsvolle, klare Zielsetzung, in der die Grundsätze des Teams definiert sind, damit alle das Richtige für die Software tun – insbesondere in puncto Sicherheit. Wird das Ziel anhand der Parameter und Funktionsweise des Softwareentwicklungsteams definiert, bewirkt dies eine Verlagerung von einer reaktiven Arbeitsweise zur Erfüllung kurzfristiger Geschäfts- oder Verkaufsziele zu einer proaktiven Arbeitsweise auf langfristige Ziele hin.

Denken Sie während der Lektüre dieses Leitfadens an ihren eigenen Prozess zur Softwareentwicklung. Wenn Sie noch keine Zielsetzung für Ihr Softwareentwicklungsteam formuliert haben, holen Sie dies nun nach. Und denken Sie dabei daran, dass eine wirkungsvolle Zielsetzung nicht festlegt, wie das Team arbeitet oder welche Vorteile das Unternehmen oder seine Kunden erwarten können. Vielmehr definiert eine wirkungsvolle Zielsetzung, was das Beste für die Software ist, die Sie entwickeln, und warum das wichtig ist.

Eine gute Zielsetzung sollte einfach, einprägsam und ohne Jargon sein. Eine hilfreiche Faustregel ist: Vermeiden Sie verschachtelte Sätze und Auflistungen.

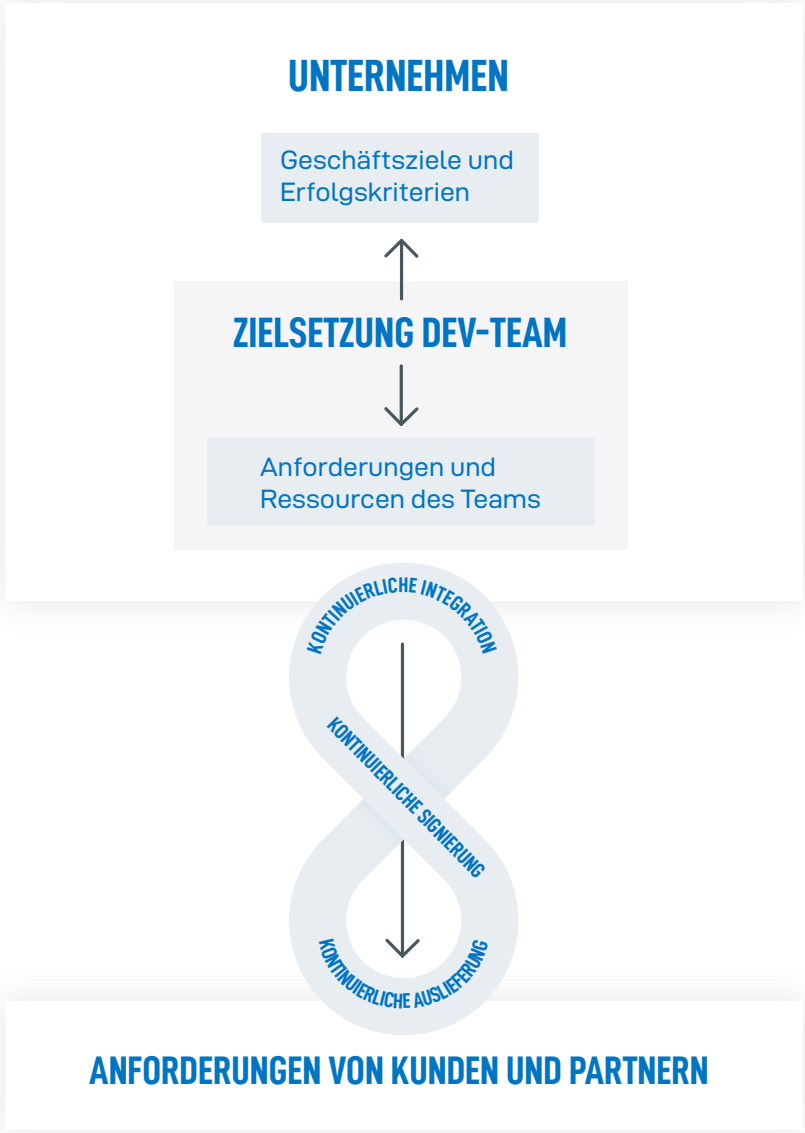


Sicherheit ergibt sich aus einer klaren Zielsetzung

Sicherheit in der Softwareentwicklung und besonders die Signierung sind traditionell langsam und mühevoll. Die zusätzlich erforderlichen Schritte zum ordnungsgemäßen Schutz von Code und Software sind unzählige Male mit der Erfüllung kurzfristiger Geschäftsziele kollidiert. Entwickler und Ingenieure wollen das umsetzen, was für ihre Software das Beste ist, aber die Signierung, sofern sie nicht von einer Plattform oder Softwarelieferkette eingefordert wird, entspricht eher der Definition eines proaktiven, langfristigen Ziels. Oft ist die Softwareentwicklung, insbesondere die flexible Entwicklung, anhand von Kriterien organisiert, die dazu führen können, dass langfristige Ziele zugunsten kurzfristiger, unmittelbarer Notwendigkeiten und Marktanforderungen geopfert werden.

In diesen Fällen kommt es auf eine klare Zielsetzung für die Softwareentwicklung an. Eine wirkungsvolle Zielsetzung berücksichtigt alle Schritte und Ressourcen, die zur Erfüllung der Aufgabe der Softwareentwicklung notwendig sind. Sie wirkt nach innen zur Erfüllung der Anforderungen des Entwicklungsteams. Sie wirkt auch entlang der Unternehmensstruktur zur Erfüllung der Geschäftsziele und Erfolgskriterien. Und nicht zuletzt wirkt sie nach außen, um den Anforderungen von Partnern und Kunden, die die Software nutzen, gerecht zu werden.

Mit einer klaren Zielsetzung für die Softwareentwicklung werden die Signierung und alle anderen Formen der Sicherheit zu einem festen Bestandteil eines proaktiven, langfristigen Entwicklungsprozesses, der Entwickler und Ingenieure in die Lage versetzt, kurzfristige Geschäftsziele zu erreichen, indem sie fristgerecht integren signierten Code liefern. Jede Richtlinie, die Sie verfassen, sollte auf dem Fundament Ihrer Zielsetzung für die Softwareentwicklung aufsetzen.



KULTUR UND ROLLEN VERSTEHEN

Ihre Richtlinie für die Softwaresignierung sollte nicht bloß ein reines Regelwerk sein. Die besten Leitlinien erklären, wie Softwaresignierung Entwicklungsteams dabei unterstützt, das Richtige für die von ihnen entwickelte Software zu tun. In unserem Kontext trägt eine wirkungsvolle Richtlinie für die Softwaresignierung dazu bei, dass alle Teamangehörigen verstehen, wie Signierung korrekt einzusetzen ist, um Ziele zu erreichen und die Aufgabe des Teams zu erfüllen.

Dazu ist es wichtig, die Beziehung zwischen den Rollen im Team und dem Einfluss dieser Rollen auf die Vermittlung und Umsetzung der Signierrichtlinie zu verstehen. Wenn die Richtlinie klar ist und diejenigen, die sie nutzen, ihre Vorteile für sich selbst und das gesamte Gelingen des Softwareentwicklungsprozesses verstehen, wird unserer Erfahrung nach die Codesicherheit zum Teil der Kultur eines Teams anstatt nur ein Schritt in einer Checkliste oder ein Spannungspunkt zwischen Entwicklern oder Ingenieuren und ihren Vorgesetzten zu sein.

Beginnen Sie mit den folgenden drei Grundsätzen:

1

Regeln sind für Administratoren, Prozesse für Entwickler und Ingenieure.

Es ist wichtig, in einer Richtlinie schriftlich zu definieren, wann und wie die Signierung vorzunehmen ist, aber diese Regeln sollten nur das Fundament bilden. Administratoren und Führungskräfte müssen dafür sorgen, dass Entwicklern und Ingenieuren Richtlinienprozesse zur Verfügung stehen, die berücksichtigen, wie sie die Signierung in ihrer täglichen Arbeit verwenden.

2

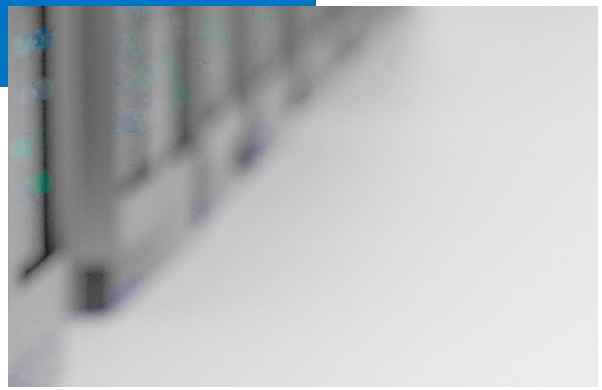
Signierung muss nicht nur gefordert, sondern ermöglicht werden.

In der Regel wollen die meisten Menschen das Richtige tun und so wollen Entwickler und Ingenieure das tun, was für den von ihnen geschriebenen Code das Richtige ist. Ihre Richtlinie für die Softwaresignierung sollte nicht als Wächter zur Durchsetzung von Sicherheitsregeln und Compliance-Vorgaben fungieren. Stattdessen sollte sie es Ihren Entwicklern und Ingenieuren erleichtern, hinter ihrer harten Arbeit zu stehen und sie zu schützen. Signierrichtlinien müssen eine gute Sicherheitspraxis ermöglichen.

3

Kommunikation ist grundlegend wichtig.

Eine Richtlinie, die verwirrend, unklar, unrealistisch oder strafbewehrt ist, behindert das Erreichen des Ziels. Vergewissern Sie sich, dass Ihre Richtlinie für die Softwaresignierung klar, logisch, leicht verständlich und einfach umzusetzen ist und dass alle, die sie verwenden, regelmäßig Auffrischungskurse erhalten, wie die Signierung ordnungsgemäß durchzuführen ist. Und denken Sie daran: Kommunikation ist keine Einbahnstraße. Eine wirksame Richtlinie muss regelmäßig überprüft und aktualisiert werden, wobei das Feedback von allen, die die Richtlinie in der Praxis umsetzen, einfließen sollte.



ERKENNUNG MÖGLICHER SCHWACHPUNKTE

Menschen sind leicht mit Vereinfachungen bei der Hand. Wenn etwas schiefgeht, neigen wir dazu, einen einzigen Schuldigen zu suchen – einen unbemerkten Fehler, das Versagen einer einzelnen Person, einen Moment der Unaufmerksamkeit oder einfach eine einzelne Anomalie. Die Wahrheit ist jedoch, dass Fehler oft eine Verkettung – oder sogar ein Netz – von Faktoren sind, die schließlich in einem Kipppunkt kulminieren.

Während die Nachrichten von Schlagzeilen dominiert werden, in denen von Sicherheitsverstößen in der Softwarelieferkette wegen eines einzelnen unachtsamen Mitarbeiters oder eines einzelnen heimtückischen Hackers die Rede ist, sind Schwachstellen in der Softwareentwicklung in Wahrheit komplex und nur allzu häufig aufzufinden. Diese Komplexität hat zuletzt noch zugenommen, da immer mehr Software entwickelt und an immer mehr Orten bereitgestellt wird. Ohne eine Lösung für die Komplexität vervielfältigt das System die Schwachstellen, bis der Punkt erreicht wird, an dem die Einzelfehler so zahlreich sind, dass es unausweichlich nur eine Frage der Zeit ist, bis einer gefunden und ausgenutzt wird.

Wenn wir weiterhin an der Vorstellung festhalten, dass Nachlässigkeit, Bequemlichkeit oder gelegentlich menschliches Versagen die Ursachen von Sicherheitsverstößen sind, werden Menge und Umfang von Angriffen auf Software garantiert zunehmen. Es stimmt zwar, dass Fehler manchmal einfach passieren, doch die meisten Schwachstellen bei der Softwaresicherheit lassen sich auf systemische Fehler in Richtlinien und Prozessen zurückführen. Üblicherweise sind sie das Resultat von schlechten Informationen, unrealistischen Erwartungen, Versehen und schwerfälligen oder fehlenden Tools.

Sicherheitslücken sind typischerweise nicht einfach nur das Ergebnis von menschlichem Versagen. Vielmehr sind sie die Folge eines Systemversagens bei der Ausstattung von Softwareentwicklungsteams mit den Tools und Leitlinien, die sie brauchen, um für die Software, die sie erstellen, das Richtige tun zu können.

DAS A UND O IN PUNCTO SICHERHEIT: EINE GUTE SICHERHEITSKULTUR

Eine wirkungsvolle Richtlinie für die Softwaresignierung ist die schriftliche Definition einer guten Sicherheitskultur. Eine Kultur, die das Softwareentwicklungsteam dazu befähigt, das zu tun, was für die Software richtig ist, konzentriert sich auf ein klares Softwareentwicklungsziel, auf Zusammenarbeit basierende Prozesse und Unterstützung, statt Regeln und deren Durchsetzung „von oben". Außerdem betrachtet sie Sicherheit als systemische Aufgabe und ermöglicht mit hervorragenden Tools eine gute Sicherheitspraxis.

SCHWACHE KULTUR

- kurzfristige, reaktive Ausrichtung
- hierarchische Struktur
- Prinzip der Durchsetzung
- unrealistische Verantwortung Einzelner
- manuelle Tools

STARKE KULTUR

- zielgerichteter, proaktiver Ansatz
- auf Gegenseitigkeit beruhend
- Prinzip der Befähigung
- systemische Unterstützung
- automatisierte Tools

„Sicherheit ist für jede einzelne Person in einem Unternehmen wichtig. Bauen Sie auf die Neugier und schaffen Sie den tief sitzenden Wunsch, immer weiter zu lernen, um eine Kultur der Sicherheit zu etablieren."

Jason Sabin, Chief Technology Officer, DigiCert

TEIL II

VERFASSEN EINER RICHTLINIE FÜR DIE SOFTWARESIGNIERUNG

Sammeln von Expertise

Bis hierhin hatten Sie Gelegenheit, über die Definition der Parameter für Ihre Richtlinie zur Softwaresignierung und eine gute Sicherheitskultur nachzudenken. Sie haben jetzt eine Vorstellung davon und haben eine klare Zielsetzung für Ihr Team schriftlich festgehalten. Nun ist es an der Zeit, die Details auszuarbeiten und eine Richtlinie zu verfassen, die Sie Ihrem Team vorlegen können. Im nächsten Schritt geht es darum, zu bestimmen, wie sich Ihr Team mit seinem ganz eigenen Ziel, seinen Prozessen, Anforderungen und Tools in das weitere Unternehmen einfügt.

Bei unserer Arbeit mit Softwareentwicklern auf der ganzen Welt aus Hunderten Branchen haben wir festgestellt, dass die Richtlinie für die Softwaresignierung oft auf das Wissen und die Erfahrung anderer Teams und Abteilungen im Unternehmen zurückgreift. Die folgenden Informationsquellen wurden uns mit am häufigsten im Zusammenhang mit der Verfassung einer Richtlinie für die Softwaresignierung genannt:

- CISO
- SecOps/InfoSec
- Risikomanagement
- Produktmanagement
- Compliance
- Kunden und Partner
- Qualitätssicherung und Standortzuverlässigkeit

Wenn Sie Ressourcen zu Rate ziehen, ist es nützlich, die folgenden Punkte zu berücksichtigen:

1 **Einbeziehung**
Was können sie dazu beitragen, dass Sie bei der Signierung einen proaktiven und reifen Sicherheitsstatus erlangen? Welche ihrer Richtlinien und Prozesse können Ihnen dabei helfen, Ihre eigene Aufgabe zu erfüllen?

2 **Gegenseitigkeit**
Wie passen Ihre Richtlinien und Prozesse mit denen Ihres Gegenübers zusammen? Wie können Sie zusammenarbeiten und zugleich sicherstellen, dass Ihre Richtlinie und Ihr Prozess für die Signierung bei der Erfüllung Ihrer Aufgabe nicht auf der Strecke bleiben?

3 **Umsetzung**
Wie können Sie Ihre Ressourcen in die Lage versetzen, zeitnah relevante Beiträge beizusteuern? Welche Beiträge sind unbedingt erforderlich und welche sind nicht praktikabel oder dienen ausschließlich der Information?

Der Wert forensischer Analysen

Der beste Sicherheitsansatz ist proaktiv. Aber Fehler bieten eine wertvolle Gelegenheit, zu lernen. Führen Sie regelmäßig forensische Untersuchungen aktueller Sicherheitsverstöße oder Pannen durch. Wenn bei Ihnen selbst keine Verstöße oder Fehler vorgekommen sind, untersuchen Sie die anderer. Fehler und Schwachstellen können Ihnen beim Verfassen einer Signierrichtlinie helfen, die die Bedrohungslage mildert und besseren Schutz Ihres Codes bietet.

GRUNDLAGEN DER NUTZUNG VON SOFTWARESIGNIERUNG

Die Konfiguration einzelner Signiersysteme und Teamstrukturen verschiedener Unternehmen können sich unterscheiden, dennoch gaben in unserer Untersuchung Entwicklungsteams unabhängig von der Teamgröße, der Art der Software oder der Branche eine Reihe gemeinsamer Komponenten an, die eine wirkungsvolle Richtlinie für die Softwaresignierung ausmachen. Diese Komponenten dienen als das Fundament für Best Practices für die Signierung.

Ausstellung von Schlüsseln

Diese Funktion gehört zwar zur Schlüsselverwaltung, doch zeigt unsere Untersuchung, dass sie das wichtigste Anliegen von DevOps-Experten ist. Richten Sie bei der Erstellung von Verwaltungsprotokollen Steuerungen und Verfahren dafür ein, wer wann Schlüssel ausstellen darf. Die Mitarbeiter im Team müssen wissen, wann es entsprechend ihrer Rolle angebracht ist, Schlüssel auszustellen. Legen Sie auch fest, welche Schlüsselarten ausgestellt werden sowie die ihnen zugeordneten Attribute. Dadurch vermeiden Sie die Ausstellung neuer Schlüssel mit schwachen Algorithmen, die Angreifern als Angriffsvektor dienen können.

Schlüsselverwaltung

Richten Sie für Benutzerrollen Steuerungen und Verfahren ein. Wer verwaltet die Schlüssel? Wie ist die Schlüsselverwaltung entsprechend der Rolle im Team, im Unternehmen oder entsprechend der Produktionsphase aufgeteilt? Diese Verfahren sollten eine Standortverfolgung für alle Schlüssel im gesamten Unternehmen einschließen.

Schlüsselspeicherung

Schlüssel müssen geschützt werden. Richten Sie Steuerungen und Verfahren für die Schlüsselspeicherung während der Nutzung und bei Nichtgebrauch ein. Diese sollten HSM, Token, USB und Zugriff auf Schlüssel mit Multifaktor-Authentifizierung umfassen. Teammitglieder müssen unterwiesen werden, wie sie Schlüssel ordnungsgemäß speichern und ihren Verlust vermeiden.

Signierberechtigungen

Die Mitarbeiter im Team müssen wissen, wann es angebracht ist, zu signieren, und wer dazu berechtigt ist. Sie müssen auch wissen, unter welchen Umständen keine Signierberechtigung gegeben wird und wie sie die Signierung beantragen können, wenn ihre eigene Rolle sie nicht dazu berechtigt.

Schlüsselnutzung

Wie Schlüssel verwendet werden – bzw. nicht verwendet werden – ist ein wesentlicher Aspekt ordnungsgemäßer Signierverfahren. Schlüssel müssen zur richtigen Zeit von den richtigen Personen genutzt werden.

Verhinderung von Schlüsselweitergabe

Die Weitergabe von Schlüsseln ist gängige Praxis. Sie ist auch eine der gefährlichsten Praktiken. Teamangehörige dürfen niemals Schlüssel weitergeben, auch nicht innerhalb von Repositorys oder unternehmensinternen Servern, Systemen und Netzwerken. Schlüssel müssen von der zuständigen Person entsprechend ihrer Rolle an eine Einzelperson ausgegeben, kontrolliert und gespeichert werden.

Kontinuierliches Signieren

Die Signierung von Code und Software sollte nicht als nachträgliche Maßnahme oder gar lästiger Compliance-Schritt aufgefasst werden. Jede CI/CD-Pipeline sollte auch CS (kontinuierliches Signieren) umfassen, damit der Schutz von Code ordnungsgemäß und konsistent praktiziert wird.

OFT VERNACHLÄSSIGT: UNTERNEHMENSINTERNES SIGNIEREN

Im Bereich der Softwaresicherheit gibt es eine unumstößliche Wahrheit: Die Befolgung von Best Practices und ihre Durchsetzung in Plattformen gehen meistens Hand in Hand. Code-Signing wird konsequenter betrieben, wenn die Software für die Bereitstellung in App-Shops, Betriebssystemen und regulierten Browsern bestimmt ist. Wenn eine Signatur notwendig ist, damit eine Software ausgeführt wird, dann wird der Code auch signiert. Doch diese Anforderungen sind nicht willkürlich oder aus der Luft gegriffen. Signaturen schützen Software wirklich vor Manipulation, Angriffen und anderen bösen Absichten. Wenn Plattformen die Signierung durchsetzen, verfolgen sie kein hohes Ideal, sondern verlangen eine praktische Maßnahme gegen Bedrohungen.

Warum also wird die Softwaresignierung schnell zu einer selten genutzten Option, wenn sie nicht für die Bereitstellung erforderlich ist? Wenn die Signierung als Best Practice bekanntermaßen eine im echten Leben bewährte Sicherheitsmaßnahme ist, sollte Software dann nicht immer signiert werden, auch wenn sie nicht für eine Plattform bestimmt ist, bei der Signierung durchgesetzt wird?

Das trügerische Gefühl der Sicherheit bei unternehmensinterner Veröffentlichung

In den meisten Fällen wird die Signierung unternehmensinterner Software ignoriert, weil Unternehmen den bekannten Entwicklerteams und den ihre Systeme umgebenden Schutzmaßnahmen vertrauen. In ihrer Wahrnehmung lauern Bedrohungen nur jenseits der Firewalls, wenn der Code bereits hinaus in die Welt geschickt wurde. Es erscheint durchaus plausibel, dass Software, die nur zwischen Teams innerhalb eines Unternehmens ausgetauscht wird oder zur internen Nutzung auf unternehmenseigenen Servern bereitgestellt wird, sicher ist.

In Wahrheit jedoch ist auch unternehmensinterne Software anfällig für Angriffe. Wenn sich jemand Zugang zum System verschafft, ist nicht signierter Code ein leichtes Ziel, das für Angriffe auf die Infrastruktur des gesamten Unternehmens ausgenutzt werden kann. Als Best Practice ist die unternehmensinterne Signierung kein Ideal, sondern eine praktische Maßnahme, die Ihre Software und Ihr Unternehmen genauso schützt wie die anderen Maßnahmen zum Schutz vor böswilligen Dritten.

Lösung durch Automatisierung

Dass die Durchsetzung die Signierung vorantreibt, ist nur eine Wahrheit der Softwaresicherheit. Eine andere ist, dass langsame, arbeitsintensive Sicherheitsmaßnahmen im Gegensatz zur DevOps-Philosophie stehen. Angesichts der umfangreichen Sicherheitsvorkehrungen, die durchschnittliche Unternehmen zum Schutz ihrer internen Systeme getroffen haben, ist es logisch, dass Entwickler und Ingenieure hinderliche Schritte auslassen, wenn das damit verbundene Risiko gering erscheint. Langsames und aufwendiges manuelles Code-Signing ist zwar keine Gefährdung im eigentlichen Sinn wie Hackerangriffe oder Manipulationen, öffnet diesen aber oft eine Hintertür.

Die Lösung ist Automatisierung in Kombination mit Verantwortbarkeit. Systeme, die kontinuierliches Signieren nutzen, erleichtern die Prozesse zum Schutz Ihrer Software unabhängig davon, wo sie entwickelt oder implementiert wird. Das erleichtert es Entwicklern, jedes Mal korrekt zu signieren.

Moderne, verwaltete Tools für die Softwaresignierung machen sicheres Signieren einfacher. „Mit diesen Tools können Sie die Signierung ohne Mehraufwand in Ihre Pipeline integrieren und so ein neues Sicherheitsniveau erreichen. Es ist ein praktisches Beispiel für das Motto ‚doppelt hält besser'."

Wade Choules, VP of Engineering bei DigiCert

SCHULUNG, WEITERBILDUNG UND FEEDBACK

Nach der Erstellung Ihrer Richtlinie für die Softwaresignierung brauchen Sie ein Schulungs- und Auffrischungssystem für Ihre Entwickler und Ingenieure. Die folgende Liste soll Ihnen als einfache Vorlage für den Schulungsplan zur sicheren Softwaresignierung für Ihre Mitarbeiter dienen.

Schulung

Diese Komponenten sollten Teil der Erstschulung sein:

- Ausrichtung und Zielsetzung des Softwareentwicklungsteams
- Stakeholder bei der Softwaresicherheit
- Einführung in Ihre Richtlinien
- Warum die Signierung genauso wichtig ist wie jeder andere Schritt des Entwicklungsprozesses
- Signierrollen im Unternehmen
- zentrale Grundsätze des Signierprozesses
- Signieren – wer, wann und wie
- ordnungsgemäße Schlüsselnutzung und -attribute
- Wann ein Zeitstempel notwendig ist
- Schlüsselsicherheit und -verwaltung
- Meldung von Problemen

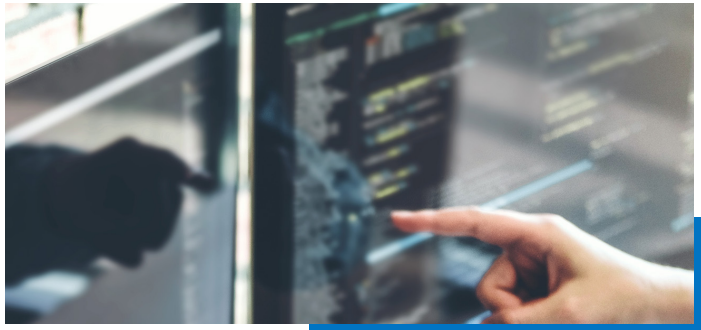
Binden Sie zusätzlich zu diesen Komponenten auch einen Test über das Verständnis der Inhalte in Ihr Schulungssystem ein. Außerdem empfiehlt es sich, neuen Teamangehörigen Raum für Feedback und Fragen zu geben, die von der Schulung nicht abgedeckt wurden. Nutzen Sie Feedback und Fragen auch zur Weiterentwicklung des Schulungsmoduls.

Weiterbildung

Laufende Weiterbildung trägt zur Vertiefung des Wissens und einem umfassenderen Verständnis bei. Wir empfehlen daher, regelmäßig Auffrischungskurse zu Richtlinien und Prozessen anzubieten. Vor allem ist es auch hilfreich, dem Wert der Schritte im Prozess einen Kontext zu geben. Erwägen Sie die folgenden Weiterbildungsangebote:

Regelmäßige Wiederholungen zu Signierrichtlinie und -prozess mit Raum für Feedback und Fragen von den Teammitgliedern. Dabei kann es sich um ein ausgewiesenes Schulungsmodul handeln, aber auch um eine moderierte Besprechung mit Fragerunde oder einen anderen Rahmen für Austausch und Diskussion.

Gelegentliche Updates und Erinnerungen an die reale Bedeutung der Softwaresignierung. Diskussionsforen zur Softwaresicherheit. Einen dedizierten Rahmen (zum Beispiel ein E-Mail-Konto oder einen privaten Slack-Kanal) für Anregungen oder Fragen von Teamangehörigen zum Thema Softwaresicherheit.



DIE GEMEINSCHAFT IST WICHTIG

Es liegt im Wesen des Themas Sicherheit, dass der Eindruck entstehen kann, ein offener Informationsaustausch mache uns anfälliger für Angriffe. Doch angesichts immer längerer Lieferketten und zunehmender Komplexität gewinnen Zusammenarbeit und Informationsaustausch für die Softwaresicherheit weltweit an Bedeutung.

Um der zunehmenden Zahl und Massivität von Angriffen etwas entgegenzusetzen, ist es wichtig, dass sich Softwareentwickler und Sicherheitsexperten über Best Practices und wirksame Richtlinien zur Softwaresignierung austauschen. Einzelne Sicherheitskonfigurationen und geistiges Eigentum sollten immer geheim bleiben, aber jegliche unbedenkliche Informationen, die zum Schutz Ihrer Software beitragen, können auch die Sicherheit anderer entlang der Lieferkette erhöhen und umgekehrt.

Wenn Sie Ihre eigene Richtlinie und Anleitung zur Softwaresignierung erstellt und in die Praxis umgesetzt haben, könnten Sie sich mit anderen Experten in Ihrem Netzwerk darüber austauschen, wie Sie mit Ihren Richtlinien und Prozessen eine robustere Sicherheit erreicht haben. Lernen Sie aus der Erfahrung anderer und regen Sie andere in Ihrer Gemeinschaft dazu an, ihren eigenen Sicherheitsstatus zu verbessern. Kein einziges Unternehmen kann es sich leisten, in puncto Sicherheit nicht proaktiv zu werden. Jedes Team in jedem Unternehmen muss seinen Code schützen, damit die gesamte Lieferkette geschützt wird.

VERFASSEN EINER RICHTLINIE FÜR DIE SOFTWARESIGNIERUNG

1 **Formulieren Sie die Zielsetzung für das Softwareentwicklungsteam.**

Definieren Sie Ihre Zielvorstellung. Ihre Zielsetzung sollte festlegen, was das Beste für die Software ist und warum das wichtig ist.

2 **Berücksichtigen Sie, welche Bedeutung Rollen für die Definition und Umsetzung Ihrer Richtlinie haben.**

Signierrichtlinien sollten im Hinblick auf Gegenseitigkeit und Zusammenarbeit erstellt werden. Sie müssen klar und einfach zu verstehen und umzusetzen sein.

3 **Legen Sie die Rolle von Signaturen in einer systemischen Umgebung guter Sicherheitspraxis dar.**

Sorgen Sie dafür, dass kontinuierliches Signieren in der gesamten CI/CD-Pipeline nicht nur durchgesetzt, sondern vor allem ermöglicht wird. Das System sollte die Best Practices für die Signierung zu einem einfachen, integrierten Bestandteil des Softwareentwicklungsprozesses machen.

4 **Sammeln Sie Informationen aus internen und externen Quellen.**

Nutzen Sie das Know-how und die Anforderungen der Betroffenen im Unternehmen und Ihrer Kunden für die Gestaltung einer Signierpraxis, die ausgereifte Sicherheitsfunktionen in hervorragender, termingerecht bereitgestellter Software ermöglicht. Beziehen Sie die forensische Analyse von Fehlern als Anleitung zur Vermeidung weiterer Sicherheitspannen ein.

5 **Dokumentieren Sie die Verfügbarkeit und Nutzung von Ressourcen und Tools.**

Überprüfen Sie Ihre Signierprozesse und die Tools, die Entwickler und Ingenieure zum Signieren nutzen. Ziehen Sie, wo möglich, Automatisierung in Betracht, um die Signierung zu beschleunigen und Angriffe auf die Lieferketten abzuwehren.

6 **Definieren Sie die Nutzung und Praxis des Signierens und die Signierkomponenten (extern und intern).**

Betrachten Sie den täglichen Entwicklungsprozess und wie sich das Signieren darin einfügt sowie die Ausstellung, Verwaltung und Nutzung von Schlüsseln und die beteiligten Rollen. Sorgen Sie dafür, dass alle Mitglieder Ihres Teams wissen, dass sie Schlüssel niemals weitergeben dürfen.

7 **Holen Sie Feedback von Administratoren, leitenden Mitarbeitern, Entwicklern und Ingenieuren ein.**

Legen Sie Ihre Richtlinie vor der Veröffentlichung und Umsetzung den Personen vor, die sie als Leitlinie nutzen werden. Arbeiten Sie Feedback ein, sodass die Richtlinien mit den täglichen Anforderungen der Beteiligten in der Softwareentwicklung in Einklang stehen.

8 **Erstellen Sie ein Schulungs- und Weiterbildungsprogramm mit regelmäßigen Auffrischungen.**

Zur Einführung sollte eine Schulung zur Richtlinie und Nutzung von Softwaresignierung gehören. Teamangehörige sollten ihr Wissen über Signierverfahren regelmäßig auffrischen, damit eine robuste Sicherheit bewahrt wird.

9 **Überprüfen Sie regelmäßig Ihre Zielsetzung und Richtlinie, nehmen Sie bei Bedarf Aktualisierungen und Änderungen vor.**

Unternehmensziele ändern sich, Projekte wechseln, Teams wachsen, schrumpfen, entstehen neu … Regelmäßige Überprüfungen Ihrer Richtlinie, in die auch das Feedback von Entwicklern und Ingenieuren einfließen sollte, stellen sicher, dass Richtlinien und Verfahren Veränderungen berücksichtigen und auf dem neuesten Stand, klar und nützlich bleiben.

10 **Nutzen Sie Gelegenheiten, durch Informationsaustausch und Zusammenarbeit Ihre eigene Sicherheit und die anderer in der Lieferkette zu stärken.**

Wenn sich die Signierfähigkeiten eines Unternehmens weiterentwickeln, kann die gesamte Welt der Software sicherer werden. Der gedankliche Austausch über die Umsetzung von Best Practices trägt zum Schutz von Unternehmen, Lieferketten und Nutzern bei.

DigiCert® Secure Software Manager stellt für Ihre ganze CI/CD-Pipeline von Anfang bis Ende kontinuierliches Signieren bereit, das automatisiert werden kann. Er bietet Nachverfolgung, Überprüfung und eine robuste, aber einfache zentralisierte Schlüsselverwaltung. Und dank der flexiblen Architektur und APIs, lässt sich DigiCert Secure Software Manager ohne Weiteres mit jeder beliebigen Entwicklungsumgebung kombinieren – auch mit Ihrer.

Wenn Sie mehr über die Automatisierung der Signierung Ihres Codes und Ihrer Software erfahren möchten, vereinbaren Sie mit uns eine technische Beratung.
PKI_Info@digicert.com

DIGICERT UND DAS BETRIEBSSYSTEM DES VERTRAUENS

DigiCert steht für mehr Sicherheit in unserer vernetzten Welt. Dieses Ziel zieht sich als roter Faden durch unsere gesamte Unternehmensgeschichte. Wir entwickelten das Betriebssystem des Vertrauens, um die Verwaltung von TLS/SSL, Identitäten, Servern, Netzwerken, Schlüsseln, Code, Signaturen, Dokumenten und IoT-Geräten einfacher und zuverlässiger zu machen. Wir haben das Vertrauen von 88 % der Fortune-500-Unternehmen, 97 der 100 größten internationalen Finanzinstitute und 81 % aller weltweit tätigen Online-Händler, dazu die branchenweit meisten Höchstbewertungen für Service und Support. Sie nutzen PKI Tag für Tag. Sollte Ihr Betriebssystem es nicht ermöglichen, einen noch größeren Nutzen daraus zu ziehen?