Engineer

Verschlüsselung

Daten ohne den richtigen Schlüssel unlesbar machen

Was es ist

Verschlüsselung, in einigen NIST-Veröffentlichungen auch enciphering genannt, verwandelt lesbare Daten, den Klartext, mit einem veröffentlichten Algorithmus und einem geheimen Wert namens Schlüssel in Geheimtext. Die Entschlüsselung liefert mit dem richtigen Schlüssel den Klartext Byte für Byte zurück. Der Algorithmus ist bewusst öffentlich: AES, das von den meisten Systemen gewählte Verfahren, gewann 2000 einen offenen NIST-Wettbewerb und ist als frei verfügbarer Standard veröffentlicht. Nur der Schlüssel ist geheim.

Verschlüsselungsverfahren gibt es in zwei Familien. Ein symmetrisches Verfahren verwendet in beiden Richtungen denselben Schlüssel. AES, spezifiziert in FIPS 197, verschlüsselt Daten in Blöcken von 128 Bit mit einem Schlüssel von 128, 192 oder 256 Bit. ChaCha20 aus RFC 8439 ist eine Stromchiffre, die auf Hardware ohne AES-Befehle etwa dreimal so schnell wie AES ist. Symmetrische Verfahren sind schnell genug für eine ganze Festplatte. Beide Seiten brauchen aber denselben Schlüssel, und seine sichere Übertragung ist der schwierige Teil.

Public-Key-Kryptografie beantwortet das mit einem Schlüsselpaar: einem veröffentlichten und einem privaten Schlüssel. Zwei Parteien können jeweils ihren privaten mit dem öffentlichen Schlüssel der anderen kombinieren und dasselbe gemeinsame Geheimnis erhalten, ohne es je zu versenden. Public-Key-Operationen sind langsam. Echte Systeme verwenden deshalb beides: Ein Public-Key-Austausch vereinbart einen neuen symmetrischen Schlüssel, und die eigentlichen Daten werden mit dem symmetrischen Verfahren verschlüsselt. So arbeitet TLS, das Protokoll hinter jeder `https://` -Adresse.

Die wichtigste moderne Idee ist Authentifizierung. Ein bloßes Verschlüsselungsverfahren verbirgt den Inhalt, lässt aber Änderungen zu: Im Zählermodus kehrt das Umlegen eines Bits im Geheimtext auch das

entsprechende Bit im Klartext um. Der Empfänger entschlüsselt die veränderte Nachricht, als wäre sie echt. Authentifizierte Verschlüsselung mit assoziierten Daten, AEAD, ergänzt einen Tag, den der Empfänger prüft, bevor er einem Byte vertraut. Die üblichen Verfahren sind AES-GCM und ChaCha20-Poly1305. TLS 1.3, 2018 als RFC 8446 veröffentlicht, behielt ausschließlich AEAD-Verfahren bei und machte jeden Public-Key-Austausch vorwärtsgeheim. Ein später gestohlener Serverschlüssel öffnet deshalb keine aufgezeichnete Sitzung.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist Verschlüsselung?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Konten, Passwörter und Berechtigungen werden so geordnet, dass alle genau den beruflich erforderlichen Zugriff haben und ihn am Tag ihres Ausscheidens verlieren.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. 1.000 RBAC-Regeln für sichere Linux-Server — 1,000 Zugriffsregeln zum Schließen einer Sicherheitslücke

1.000 RBAC-Regeln für sichere Linux-Server Kontakt aufnehmen →

Warum es wichtig ist

Daten bleiben selten auf Hardware unter Kontrolle ihres Eigentümers. Sie durchqueren fremde Netzwerke, liegen in fremden Rechenzentren und verlassen auf Laptops das Gebäude. Verschlüsselung lässt sie diese Orte sicher passieren: Eine gestohlene Festplatte, ein abgehörtes Kabel oder ein geleakter Bucket liefert Geheimtext. Geheimtext ohne Schlüssel ist Rauschen.

Das verschiebt das Problem, statt es zu beseitigen: Nach der Verschlüsselung hängt die Datensicherheit an der Sicherheit des Schlüssels. NISTs Empfehlung zur Schlüsselverwaltung, SP 800-57, vergleicht Schlüssel mit einer Tresorkombination. Ein Gegner, der sie kennt, überwindet den stärksten Tresor. Schlechte Schlüsselverwaltung kann einen starken Algorithmus leicht kompromittieren. Der größte Teil der tatsächlichen Arbeit gilt deshalb den Schlüsseln: getrennt von den Daten aufbewahren, den Schutzumfang jedes Schlüssels begrenzen, sie nach Zeitplan ersetzen und die Wiederherstellungskopie genauso gut schützen wie das Original.

Aufsichtsregeln ziehen dieselbe Grenze. Die DSGVO nennt Pseudonymisierung und Verschlüsselung personenbezogener Daten unter den Sicherheitsmaßnahmen in Artikel 32. Artikel 34 erlaubt einem Verantwortlichen,

auf die Benachrichtigung jeder betroffenen Person über eine Datenschutzverletzung zu verzichten, wenn die offengelegten Daten für Unbefugte unverständlich waren. Verschlüsselung ist das dort genannte Beispiel.

Verschlüsselung setzt an zwei Stellen an. Bei der Übertragung schützt meist TLS die Daten auf der Leitung. Bei der Speicherung schützt Verschlüsselung die gespeicherten Daten. Der Nutzungszeitpunkt liegt außerhalb beider Bereiche: Ein Server, der einen Datensatz zur Bearbeitung entschlüsselt, hält gleichzeitig Klartext und Schlüssel. Wer diesen Server kontrolliert, liest daher, was er liest.

Wie es in der Praxis aussieht

Bei der Übertragung besteht die tägliche Arbeit darin, Zertifikate gültig zu halten. Als hundert Docker-Anwendungen SSL/TLS-Zertifikate brauchten, erzeugte ein ACME-Ablauf jedes Zertifikat, erneuerte es vor Ablauf und stellte es auf Ubuntu-Hosts mit Apache und Nginx bereit. Ziel war, den Menschen aus dem Ablauf zu nehmen, denn er ist der Teil, der vergisst. Ein abgelaufenes Zertifikat legt eine Anwendung genauso sicher lahm wie ein kaputtes Verschlüsselungsverfahren.

Bei der Speicherung lautet die nützliche Frage, wer den Schlüssel hält. Die Implementierung verschlüsselter Sicherungen außerhalb des Hosts des Unternehmens wurde dafür entworfen, Wiederherstellung über einen Anbietersnapshot hinaus zu ermöglichen. Sie verwendet restic, das seine gespeicherten Daten mit AES-256 im Zählermodus verschlüsselt und mit Poly1305-AES authentifiziert. Die Skripte unterstützen den Versand von Snapshots per SFTP an Speicher bei einem anderen Anbieter. In diesem Entwurf erhält der Anbieter Geheimtext, und das Unternehmen hält die Passphrase in einer geschützten Datei. Das beschreibt die implementierten Kontrollen, nicht ihre betriebliche Aktivierung: Der Eintrag vom 2026-10-27 hält fest, dass die Timer ohne konfiguriertes Repository deaktiviert waren, und verzeichnet keine abgeschlossene Live-Wiederherstellung. Verschlüsselung ist eine Eigenschaft des vorgeschlagenen Speicherwegs. Um die Wiederherstellbarkeit der Sicherung zu belegen, muss der separate Wiederherstellungstest ausgeführt werden.

Selbst der Weg der Passphrase war wichtig. Sie wird aus einer Datei gelesen, weil die Dienstverwaltung Escape-Sequenzen in Umgebungswerten verarbeitet. Eine Passphrase mit einem umgekehrten Schrägstrich hätte sich so unbemerkt von der beim Anlegen des Repositorys verwendeten unterschieden. Der implementierte monatliche Ablauf für Wiederherstellungstests ist dafür

entworfen, eine bekannte Datei aus dem Repository wiederherzustellen und mit dem erwarteten Ergebnis zu vergleichen. Weil der Test die Passphrase benötigt, würde ein erfolgreicher Lauf auch bestätigen, dass der Schlüssel die Daten weiterhin öffnet.

Als die Plattform einen zweiten Host bekommen sollte, entschied dieselbe Überlegung, wie jedes für den Betrieb verwendete Geheimnis pro Host getrennt wurde. Eine Zeichenkette für ein Sicherungsrepository bezeichnet einen Speicherort. Getrennte restic-Repositories können unterschiedliche Hauptschlüssel haben, auch wenn sie dieselbe Passphrase verwenden. Wer diese Passphrase erhält und die verschlüsselten Daten sowie die Schlüsseldateien beider Repositories lesen kann, kann beide entsperren. Eine gemeinsame Passphrase verknüpft deshalb ihre Gefährdung. Jedes Geheimnis wurde zu einer nach Host indizierten Map, mit der Begründung neben der Konfiguration. SP 800-57 formuliert das allgemeine Prinzip: Den Schutzumfang eines Schlüssels zu begrenzen begrenzt den Schaden bei einem Leck.

Wo es schiefgeht

Der Schlüssel existiert nur an einem Ort. Ein starkes Verfahren behandelt einen Eigentümer mit vergessenem Passwort genauso wie einen Angreifer. Die restic-Dokumentation sagt es klar: Mit dem Passwort gehen die Daten unwiederbringlich verloren. Jeder verschlüsselte Speicher braucht eine zweite Schlüsselkopie an einem Ort außerhalb der Reichweite dessen, was die erste Kopie zerstört.

Der Schlüssel liegt neben den Daten. Eine verschlüsselte Datenbank mit Schlüssel in der Konfiguration desselben Servers oder eine Sicherung mit Passphrase im selben Bucket hält nur einen Dieb auf, der die Hälfte der Dateien mitnimmt. Festplattenverschlüsselung ist die weniger offensichtliche Variante: Sie schützt ein aus einem ausgeschalteten Rechner entnommenes Laufwerk. Ein laufender Server hat sein Laufwerk dagegen schon entsperrt, und ein Angreifer auf diesem Server liest Klartext.

Vertraulichkeit wird mit Integrität verwechselt. Ein bloßer Verschlüsselungsmodus verbirgt Inhalt und erlaubt Angreifern weiterhin Änderungen. Nutzen Sie ein AEAD-Verfahren und verwerfen Sie jede Nachricht mit ungültigem Tag, bevor Sie eines ihrer Bytes verarbeiten.

Eine Nonce wiederholt sich. AES-GCM und ChaCha20-Poly1305 verwenden eine Nonce, einen Wert, der für jede Nachricht unter einem Schlüssel eindeutig bleiben muss. RFC 8439 nennt dies die wichtigste Sicherheitsüberlegung: Zwei Nachrichten mit derselben Nonce teilen einen Schlüsselstrom und einen einmaligen Authentifizierungsschlüssel. Das legt das XOR der beiden Klartexte offen. Ein Zähler erzeugt Nonces sicher, solange pro Schlüssel nur ein Zähler zuständig ist. Zwei Prozesse mit gemeinsamem Schlüssel, die beide bei null zu zählen beginnen, erzeugen genau die vom RFC beanstandete Wiederholung.

Ein Passwort ersetzt einen Schlüssel. Menschen wählen Passwörter; Verschlüsselungsverfahren brauchen zufällige Schlüssel fester Länge. Eine Passphrase durchläuft zuerst eine Schlüsselableitungsfunktion, wie bei restic mit scrypt. Das Verfahren ist nach RFC 7914 auf hohen Speicheraufwand ausgelegt, damit massenhaftes Raten auf Spezialhardware teuer bleibt. Gespeicherte Anmeldepasswörter sind ein anderer Fall: Sie sollten als Hashwerte gespeichert werden, die mit einer solchen Funktion erzeugt wurden. Denn ein Server, der sie entschlüsseln kann, kann sie auch offenlegen.

Metadaten bleiben sichtbar. Verschlüsselung verbirgt Inhalte. Größe, Zeitpunkte und die Adressen beider Seiten bleiben sichtbar. TLS 1.3 erlaubt dem Absender, Datensätze aufzufüllen, um ihre Größe zu verbergen. Das Auffüllen ist optional, weshalb die Länge oft mehr verrät als vom Eigentümer erwartet.

Weiterführende Lektüre

- NIST-Glossar: Verschlüsselung — alle Definitionen aus NIST-Veröffentlichungen, jeweils mit ihrem Quelldokument.
- FIPS 197: Advanced Encryption Standard (AES) — die AES-Spezifikation samt Blockgröße und drei Schlüssellängen.
- NIST SP 800-57 Teil 1 Rev. 5: Empfehlung zur Schlüsselverwaltung — wie Schlüssel erzeugt, gespeichert, verwendet und außer Betrieb genommen werden und was bei einem Leck folgt.
- RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3 — das Protokoll hinter `https://`, ausschließlich AEAD-Datensätze und vorwärtsgeheimer Schlüsselaustausch.
- RFC 8439: ChaCha20 and Poly1305 for IETF Protocols — Stromchiffre, Authentifikator und ihre AEAD-Kombination samt Nonce-Regeln.
- restic: Referenzen — wie ein echtes Sicherungswerkzeug Schlüssel aus einem Passwort ableitet und seine Daten anschließend verschlüsselt und authentifiziert.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Bereitstellungsweg, die Betriebsrisiken und die Bereitschaftsverantwortlichen. Wir können eine Änderung mit Prüfungen, einem Wiederherstellungsweg und klaren Zuständigkeiten abgrenzen.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist Verschlüsselung?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/encryption/