Was ist schlüsselbasierte Authentifizierung, und warum verschaffen gestohlene Schlüssel dauerhaften Zugriff?

Kurzantwort: Schlüsselbasierte Authentifizierung weist die Identität mittels eines kryptografischen privaten Schlüssels statt eines Passworts nach — verwendet von SSH, vielen VPNs und Dateiübertragungs-Clients. Ein gestohlener privater Schlüssel gewährt Zugriff ohne jegliche Passwort- oder MFA-Abfrage, und da Schlüssel nur selten rotiert werden, kann dieser Zugriff über sehr lange Zeit bestehen bleiben.

Funktionsweise

Sie besitzen einen privaten Schlüssel; der Server besitzt den dazugehörigen öffentlichen Schlüssel. Beim Verbindungsaufbau stellt der Server eine Challenge, die nur der private Schlüssel beantworten kann. Der Schlüssel selbst überquert das Netzwerk niemals.

Dies ist tatsächlich stärker als Passwörter — es widersteht Erraten, Phishing und Wiederverwendung. Seine Schwäche ist anders gelagert: **Alles hängt davon ab, dass die private Schlüsseldatei geheim bleibt.**

Warum ein gestohlener Schlüssel so dauerhaft ist

- **Keine Passwortabfrage.** Der Schlüssel *ist* die Anmeldeinformation.
- **Häufig keine MFA.** Schlüsselbasierte Authentifizierung steht oft für sich allein, insbesondere bei automatisierten Systemen.
- **Selten rotiert.** Passwörter laufen nach einem festen Zeitplan ab; Schlüssel bestehen häufig über Jahre.
- **Vorhersehbare Speicherorte.** Standardverzeichnisse und Konfigurationspfade von VPN-Clients.

Ein gestohlener Schlüssel ist eine stille, dauerhafte Hintertür, die keine fehlgeschlagenen Anmeldungen erzeugt und vollständig legitim erscheint.

Schutz

VPN- und schlüsselbasiertes Authentifizierungsmaterial gehören ausdrücklich zu den geschützten Identitätskategorien. Der Zugriff wird auf Kernel-Ebene bewertet, sodass ein Programm, das Schlüsseldateien abgreift, gestoppt wird — abgewiesen oder mit gefälschten Daten versorgt, wenn es sich um ein vertrauenswürdiges Programm handelt, das seine Berechtigungen überschreitet, zurückgewiesen oder ausgesetzt, wenn es unbekannt oder kompromittiert ist — bevor der Schlüssel den Rechner verlässt.

Revision #1
Created 2026-07-24 04:30:05 UTC by Dennis Underwood
Updated 2026-07-24 04:30:05 UTC by Dennis Underwood