Warum ist das Abfangen von Ransomware-Verschlüsselungsschlüsseln keine gültige Verteidigung mehr?

Kurze Antwort: Das Abfangen von Schlüsseln war ab 2021 technisch nicht mehr zuverlässig. Moderne Ransomware umgeht die Standard-Verschlüsselungsbibliotheken des Betriebssystems, sodass es keine Schlüssel gibt, die ein Verteidigungstool abfangen könnte — und selbst wenn Schlüssel erlangt werden, lassen sie sich oft nicht anwenden.

Wie moderne Ransomware das Abfangen von Schlüsseln umgeht

- **Benutzerdefinierte Bibliotheken:** Der Verschlüsselungscode wird direkt in die Malware kompiliert, anstatt Betriebssystembibliotheken aufzurufen, sodass Schlüssel vollständig innerhalb des Prozesses des Angreifers erzeugt und verwendet werden — unsichtbar für Tools, die Standardaufrufe überwachen.
- **Einzigartige Implementierungen:** Manche Varianten verwenden nicht standardisierte, pro Ausführung unterschiedliche Verschlüsselung. Selbst ein abgefangener Schlüssel lässt sich nicht anwenden, da der Algorithmus selbst maßgeschneidert ist.
- **Streaming-Chiffren:** Diese basieren auf einem grundlegend anderen Konzept als AES und sind gegenüber dem Abfangen von AES-Schlüsseln schlicht immun.

Cyber Crucible hat die Technologie zum Abfangen von Schlüsseln ursprünglich entwickelt und patentiert und anschließend die Kryptoanalyse, die ihre Grenzen aufzeigt, 2021 auf der BSides Pittsburgh vorgestellt. Wir sind weitergezogen, weil die Beweislage dies nahelegte.

Auch ein Compliance-Problem

Das Abfangen von Schlüsseln erfordert die zentrale Speicherung von Verschlüsselungsschlüsseln — dies schafft einen Single Point of Failure und ein besonders lohnendes Angriffsziel. Eine Regierung könnte zudem Zugriff durch eine geheime Vorladung oder ein Schreiben zur nationalen Sicherheit erzwingen, ohne dass Sie davon Kenntnis erhalten. Das ist ein erhebliches Risiko für die Datenhoheit — zusätzlich zu einer Verteidigung, die ohnehin nicht mehr funktioniert.

Revision #1
Created 2026-07-24 04:32:32 UTC by Dennis Underwood
Updated 2026-07-24 04:32:32 UTC by Dennis Underwood