

Wie werden Identitätsdaten geschützt?

Kurze Antwort: Cyber Crucible identifiziert die Orte, an denen Identitätsdaten gespeichert sind, und schützt diese Orte auf Kernel-Ebene. Wenn ein Programm auf einen solchen Ort zugreift, wird dieser Zugriff verfolgt und analysiert — **ohne dass die Passwörter, Tokens oder Verschlüsselungsschlüssel selbst jemals abgerufen oder gesammelt werden**. Auf Basis dieser Analyse wird das Programm zugelassen, mit gefälschten Daten versorgt, abgelehnt oder ausgesetzt.

Schutz des Zugriffs, nicht des Geheimnisses

Die meisten Ansätze zum Schutz von Identitäten setzen beim Geheimnis selbst an: verschlüsseln, in einem Tresor ablegen oder beobachten, ob es in einem Datenleck auftaucht. Cyber Crucible setzt beim **Akt des Zugreifens** an.

Automatisierte Angriffe sind in einem bestimmten Punkt vorhersehbar — sie müssen bekannte Orte aufsuchen, um Zugangsdaten zu finden. Browser-Cookie-Speicher, VPN-Zugangsdatenpfade, der Windows-Anmeldeinformationsspeicher, Wallet-Dateien und Active-Directory-Daten befinden sich alle dort, wo Angreifer bereits wissen, wo sie suchen müssen. Diese Vorhersehbarkeit ist die defensive Chance.

Cyber Crucible identifiziert diese Identitätsdatenspeicher und schützt sie anschließend. Jedes Programm, das einen davon berührt, wird bewertet — zusammen mit seinen übergeordneten und untergeordneten Prozessen sowie den von ihm geladenen Bibliotheken.

Die Geheimnisse werden niemals gelesen

Dies ist eine bewusste Designentscheidung, kein Nebeneffekt: **Der Zugriff wird verfolgt und analysiert, ohne dass die Zugangsdaten selbst abgerufen werden**. Cyber Crucible liest, verarbeitet, sammelt oder überträgt keine Passwörter, Sitzungstokens, Cookies oder Verschlüsselungsschlüssel — an keinen Server, weder von Cyber Crucible noch vom Kunden gehostet.

Es stellt sich heraus, dass die Verhaltensanalyse des Identitätszugriffs die Identitätsdaten selbst nicht benötigt. Entscheidend ist, *welches Programm wo, in welchem Kontext und nach welchen vorherigen Aktionen zugreift* — nichts davon erfordert das Öffnen des Geheimnisses.

Die Folge ist, dass der Schutzmechanismus selbst nicht zur Sicherheitsverletzung werden kann und es keinen zentralen Speicher Ihrer Zugangsdaten gibt, der vorgeladen, geleakt oder verloren werden könnte.

Was die Verhaltens-Engine bewertet

Die Entscheidung lautet nicht „Steht dieses Programm auf einer Liste?“, sondern „Was tut dieser Code gerade tatsächlich?“ Zu den Signalen gehören:

- **Speicherzustand und -verhalten** — über den gesamten Lebenszyklus des Programms verfolgt, einschließlich Änderungen im Arbeitsspeicher, die niemals auf die Festplatte gelangen. Dies ist die grundlegende Sensor-Ebene der Plattform, die mit dem Datenschutz und FortressAI gemeinsam genutzt wird, statt spezifisch für Identität zu sein.
- **Prozessabstammung** — was dies gestartet hat und was es wiederum gestartet hat.
- **Zugriffsmuster** — handelt es sich um eine normale Anwendung, die ihre eigenen Zugangsdaten liest, oder um ein gleichzeitiges Durchsuchen mehrerer Identitätsorte?
- **Bibliotheks- und Injektionsaktivität** — wurde in diesen vertrauenswürdigen Prozess injiziert oder wurde er im Speicher verändert?

Die Reaktion ist abgestuft, nicht binär

Nicht jedes Programm, das auf Identitätsdaten zugreift, ist ein Angreifer. Daher hängt die Reaktion davon ab, um welches Programm es sich handelt und wie es sich verhält:

Situation	Reaktion
Eine bekannte Anwendung, nicht kompromittiert, greift auf das zu, was sie legitim benötigt	Zugelassen
Eine bekannte Anwendung, nicht kompromittiert, greift jedoch über ihren angemessenen Zugriff hinaus	Erhält gefälschte Daten oder wird der Zugriff verweigert — gesteuert durch Verhaltensregeln zum Datenschutz
Eine unbekannte Anwendung	Abgelehnt oder ausgesetzt , gemäß der Verhaltens-Engine und Ihren Einstellungen
Eine kompromittierte Anwendung — Prozess oder Bibliotheken ausgenutzt	Abgelehnt oder ausgesetzt , gemäß der Verhaltens-Engine und Ihren Einstellungen

Der mittlere Fall ist wichtiger, als es zunächst scheint. Viele legitime Softwareprogramme greifen weiter in Identitätsspeicher hinein, als sie eigentlich benötigen. Sie auszusetzen würde den Arbeitsalltag des Nutzers ohne jeglichen Sicherheitsgewinn stören — ihnen jedoch echte

Zugangsdaten zu übergeben, wäre eine unnötige Gefährdung. Die Rückgabe gefälschter Daten befriedigt das Programm, während das echte Geheimnis an seinem Ort verbleibt.

Die Unterscheidung, die die Engine trifft, liegt zwischen einem vertrauenswürdigen Programm, das *übergreift*, und einem Programm, das unbekannt ist oder *übernommen* wurde. Nur Letzteres wird als Angriff behandelt.

Warum dies auf dem Endpunkt geschieht

Ein automatisierter Infostealer kann innerhalb von Sekunden eindringen, Daten sammeln und seine eigenen speicherresidenten Werkzeuge löschen. Telemetriedaten an einen Cloud-Dienst zu senden und auf ein Urteil zu warten, kann dagegen nicht bestehen. Der vollständige Detect-Decide-Respond-Zyklus läuft daher lokal im Kernel — dies ist auch der Grund, warum er völlig ohne Konnektivität funktioniert.

“ Siehe auch: „*Sammeln Sie Identitätsdaten?*“, „*Welche Identitätsdaten werden geschützt?*“ und „*Was passiert, wenn ein Programm versucht, auf Identitätsdaten zuzugreifen, auf die es keinen Zugriff haben sollte?*“

Revision #1

Created 2026-07-24 04:28:14 UTC by Dennis Underwood

Updated 2026-07-24 04:28:14 UTC by Dennis Underwood