¿Cómo se protegen los datos de identidad?

Respuesta breve: Cyber Crucible identifica las ubicaciones donde se almacenan los datos de identidad y protege esas ubicaciones a nivel de kernel. Cuando un programa intenta acceder a una de ellas, su acceso se rastrea y se analiza, **sin que las contraseñas, tokens o claves de cifrado en sí mismos sean accedidos ni recopilados nunca**. Con base en ese análisis, el programa es permitido, recibe datos falsos, es denegado o suspendido.

Proteger el acceso, no el secreto

La mayoría de las soluciones de protección de identidad actúan sobre el secreto: lo cifran, lo almacenan en una bóveda o vigilan si aparece en una filtración de datos. Cyber Crucible actúa sobre **el acto de intentar acceder a él**.

Los ataques automatizados son predecibles en un aspecto concreto: deben dirigirse a ubicaciones conocidas para encontrar credenciales. Los almacenes de cookies del navegador, las rutas de credenciales de VPN, el almacén de credenciales de Windows, los archivos de wallets y los datos de Active Directory se encuentran todos en lugares donde los atacantes ya saben buscar. Esa previsibilidad constituye la oportunidad defensiva.

Cyber Crucible identifica esos almacenes de datos de identidad y luego los protege. Cualquier programa que toque uno de ellos es evaluado, junto con sus procesos padre e hijo y las bibliotecas que ha cargado.

Los secretos nunca se leen

Esto es una restricción de diseño deliberada, no un efecto secundario: **el acceso se rastrea y se analiza sin que se acceda a la credencial en sí**. Cyber Crucible no lee, procesa, recopila ni transmite contraseñas, tokens de sesión, cookies ni claves de cifrado, a ningún servidor, ya sea alojado por Cyber Crucible o por el cliente.

Resulta que el análisis conductual del acceso a la identidad no requiere los datos de identidad en sí. Lo que importa es *qué programa está accediendo a dónde, en qué contexto, y qué ha hecho previamente*, nada de lo cual requiere abrir el secreto.

La consecuencia es que la protección en sí misma no puede convertirse en la brecha, y no existe un almacén central de sus credenciales que pueda ser objeto de una citación judicial, filtrarse o perderse.

Qué evalúa el motor conductual

La decisión no es "¿está este programa en una lista?", sino "¿qué está haciendo realmente este código en este momento?". Las señales incluyen:

- **Estado y comportamiento de la memoria**: se rastrean a lo largo de todo el ciclo de vida del programa, incluidos los cambios en memoria que nunca llegan al disco. Esta es la capa de sensores fundamental de la plataforma, compartida con la protección de datos y FortressAI, en lugar de ser específica de la identidad.
- **Linaje del proceso**: qué lo inició y qué inició él a su vez.
- **Patrón de acceso**: ¿se trata de una aplicación normal que lee su propia credencial, o de un barrido simultáneo a través de múltiples ubicaciones de identidad?
- **Actividad de bibliotecas e inyección**: ¿se ha inyectado o alterado en memoria este proceso confiable?

La respuesta es gradual, no binaria

No todos los programas que intentan acceder a datos de identidad son atacantes, por lo que la respuesta depende de qué es el programa y cómo se está comportando:

Situación	Respuesta
Una aplicación conocida, no explotada, que accede a lo que legítimamente necesita	Permitido
Una aplicación conocida, no explotada, pero que **excede** su acceso apropiado	**Recibe datos falsos, o se le deniega el acceso** — regido por reglas conductuales de privacidad
Una aplicación **desconocida**	**Rechazada o suspendida**, según el motor conductual y su configuración
Una aplicación **comprometida** — proceso o bibliotecas explotados	**Rechazada o suspendida**, según el motor conductual y su configuración

El caso intermedio importa más de lo que parece a primera vista. Mucho software legítimo accede más allá de lo que necesita dentro de los almacenes de identidad. Suspenderlo arruinaría el día del usuario sin ninguna ganancia de seguridad, pero entregarle credenciales reales representa una exposición innecesaria. Devolver datos falsos satisface al programa mientras el secreto real permanece intacto.

La distinción que traza el motor es entre un programa confiable que *se extralimita* y un programa que es desconocido o que ha sido *tomado por control externo*. Solo este último se trata como un ataque.

Por qué esto ocurre en el endpoint

Un infostealer automatizado puede infiltrarse, recopilar datos y eliminar sus propias herramientas en memoria en cuestión de segundos. Enviar telemetría a un servicio en la nube y esperar un veredicto no puede competir con esa velocidad. Por ello, el ciclo completo de detectar-decidir-responder se ejecuta localmente en el kernel, lo cual también explica por qué funciona sin ninguna conectividad.

> Vea también: "*¿Recopilan algún dato de identidad?*", "*¿Qué datos de identidad se protegen?*" y "*¿Qué ocurre cuando un programa intenta acceder a datos de identidad que no debería?*"

Revision #1
Created 2026-07-24 01:43:14 UTC by Dennis Underwood
Updated 2026-07-24 01:43:14 UTC by Dennis Underwood