Como os dados de identidade são protegidos?

Resposta curta: A Cyber Crucible identifica os locais onde os dados de identidade são armazenados e protege esses locais no nível do kernel. Quando um programa tenta acessar um deles, o acesso é rastreado e analisado — **sem que as senhas, tokens ou chaves de criptografia em si sejam acessados ou coletados**. Com base nessa análise, o programa é permitido, recebe dados falsos, é negado ou suspenso.

Protegendo o acesso, não o segredo

A maior parte da proteção de identidade atua sobre o segredo: criptografá-lo, guardá-lo em um cofre ou monitorar seu aparecimento em um vazamento de dados. A Cyber Crucible atua sobre **o ato de tentar acessá-lo**.

Ataques automatizados são previsíveis de uma forma específica — eles precisam ir a locais conhecidos para encontrar credenciais. Os armazenamentos de cookies do navegador, os caminhos de credenciais de VPN, o armazenamento de credenciais do Windows, os arquivos de carteiras (wallets) e os dados do Active Directory estão todos localizados onde os atacantes já sabem procurar. Essa previsibilidade é a oportunidade de defesa.

A Cyber Crucible identifica esses armazenamentos de dados de identidade e, em seguida, os protege. Qualquer programa que toque em um deles é avaliado, junto com seus processos pai e filho e as bibliotecas que carregou.

Os segredos nunca são lidos

Essa é uma restrição de design deliberada, não um efeito colateral: **o acesso é rastreado e analisado sem que a própria credencial seja acessada.** A Cyber Crucible não lê, processa, coleta ou transmite senhas, tokens de sessão, cookies ou chaves de criptografia — para nenhum servidor, seja hospedado pela Cyber Crucible ou pelo cliente.

A análise comportamental do acesso à identidade acaba não exigindo os dados de identidade em si. O que importa é *qual programa está acessando onde, em que contexto, tendo feito o quê anteriormente* — nada disso exige abrir o segredo.

A consequência é que a proteção não pode, ela mesma, tornar-se a violação, e não existe um armazenamento central de suas credenciais para ser intimado judicialmente, vazado ou perdido.

O que o mecanismo comportamental avalia

A decisão não é "este programa está em uma lista?" — é "o que este código está realmente fazendo agora?" Os sinais incluem:

- **Estado e comportamento da memória** — rastreados ao longo de todo o ciclo de vida do programa, incluindo alterações em memória que nunca chegam ao disco. Essa é a camada de sensor fundamental da plataforma, compartilhada com a proteção de dados e o FortressAI, e não é específica da identidade.
- **Linhagem de processos** — o que iniciou isso e o que isso iniciou, por sua vez.
- **Padrão de acesso** — trata-se de uma aplicação normal lendo sua própria credencial, ou de uma varredura simultânea em vários locais de identidade?
- **Atividade de bibliotecas e injeção** — este processo confiável foi injetado ou alterado em memória?

A resposta é graduada, não binária

Nem todo programa que tenta acessar dados de identidade é um atacante, portanto a resposta depende do que o programa é e de como está se comportando:

Situação	Resposta
Uma aplicação conhecida, não explorada, acessando o que legitimamente precisa	Permitido
Uma aplicação conhecida, não explorada, mas que **excede** o acesso apropriado	**Recebe dados falsos, ou o acesso é negado** — regido por regras comportamentais de privacidade
Uma aplicação **desconhecida**	**Rejeitada ou suspensa**, conforme o mecanismo comportamental e suas configurações
Uma aplicação **comprometida** — processo ou bibliotecas explorados	**Rejeitada ou suspensa**, conforme o mecanismo comportamental e suas configurações

O caso intermediário importa mais do que parece à primeira vista. Muitos softwares legítimos acessam os armazenamentos de identidade mais profundamente do que precisam. Suspendê-los prejudicaria o dia do usuário sem qualquer ganho de segurança — mas entregar-lhes credenciais reais é uma exposição desnecessária. Retornar dados falsos satisfaz o programa enquanto o segredo real permanece intacto.

A distinção que o mecanismo está estabelecendo é entre um programa confiável que *excede seu escopo* e um programa que é desconhecido ou que foi *tomado por um atacante*. Somente o último é tratado como um ataque.

Por que isso acontece no endpoint

Um infostealer automatizado pode se infiltrar, coletar e apagar suas próprias ferramentas em memória em questão de segundos. Enviar telemetria para um serviço em nuvem e esperar por um veredito não consegue superar isso. O ciclo completo de detect-decide-respond, portanto, é executado localmente no kernel, o que também é o motivo pelo qual funciona sem qualquer conectividade.

> Veja também: "*Vocês coletam algum dado de identidade?*", "*Quais dados de identidade são protegidos?*" e "*O que acontece quando um programa tenta acessar dados de identidade que não deveria?*"

Revision #1
Created 2026-07-24 05:41:18 UTC by Dennis Underwood
Updated 2026-07-24 05:41:18 UTC by Dennis Underwood