How is identity data protected?

Short answer: Cyber Crucible identifies the locations where identity data is stored and protects those locations at the kernel level. When a program reaches for one, its access is traced and analyzed — **without the passwords, tokens, or encryption keys themselves ever being accessed or collected**. Based on that analysis the program is allowed, given fake data, denied, or suspended.

Protecting access, not the secret

Most identity protection works on the secret: encrypt it, vault it, or watch for it appearing in a breach dump. Cyber Crucible works on the **act of reaching for it**.

Automated attacks are predictable in one specific way — they must go to known locations to find credentials. Browser cookie stores, VPN credential paths, the Windows credential store, wallet files, and Active Directory data all live where attackers already know to look. That predictability is the defensive opportunity.

Cyber Crucible identifies those identity data stores and then protects them. Any program touching one is evaluated, along with its parent and child processes and the libraries it has loaded.

The secrets are never read

This is a deliberate design constraint, not a side effect: **the access is traced and analyzed without the credential itself being accessed.** Cyber Crucible does not read, process, collect, or transmit passwords, session tokens, cookies, or encryption keys — to any server, whether Cyber Crucible-hosted or customer-hosted.

Behavioral analysis of identity access turns out not to require the identity data. What matters is *which program is reaching where, in what context, having done what beforehand* — none of which requires opening the secret.

The consequence is that the protection cannot itself become the breach, and there is no central store of your credentials to subpoena, leak, or lose.

What the behavioral engine evaluates

The decision isn't "is this program on a list?" — it's "what is this code actually doing right now?" Signals include:

- **Memory state and behavior** — tracked across the program's whole lifecycle, including in-memory changes that never touch disk. This is the platform's foundational sensor layer, shared with data protection and FortressAI rather than specific to identity.
- **Process lineage** — what launched this, and what it launched in turn.
- **Access pattern** — is this a normal application reading its own credential, or a sweep across many identity locations at once?
- **Library and injection activity** — has this trusted process been injected into or altered in memory?

The response is graduated, not binary

Not every program reaching for identity data is an attacker, so the response depends on what the program is and how it is behaving:

Situation	Response
A known application, not exploited, accessing what it legitimately needs	Allowed
A known application, not exploited, but **exceeding** its appropriate access	**Given fake data, or access denied** — governed by privacy behavioral rules
An **unknown** application	**Rejected or suspended**, per the behavioral engine and your settings
A **compromised** application — process or libraries exploited	**Rejected or suspended**, per the behavioral engine and your settings

The middle case matters more than it first appears. Plenty of legitimate software reaches further into identity stores than it needs to. Suspending it would break the user's day for no security gain — but handing it real credentials is an unnecessary exposure. Returning fake data satisfies the program while the real secret stays put.

The distinction the engine is drawing is between a trusted program *overreaching* and a program that is unknown or has been *taken over*. Only the latter is treated as an attack.

Why this happens on the endpoint

An automated infostealer can infiltrate, collect, and delete its own in-memory tooling within seconds. Sending telemetry to a cloud service and waiting for a verdict cannot beat that. The full detect-decide-respond loop therefore runs locally in the kernel, which is also why it works with no connectivity at all.

> See also: "*Do you collect any of identity data?*", "*What identity data is protected?*", and "*What happens when a program tries to access identity data it shouldn't?*"

Revision #6
Created 2026-05-18 20:28:48 UTC by Dennis Underwood
Updated 2026-07-21 19:32:35 UTC by Dennis Underwood