How does Cyber Crucible secure and control endpoints?

Short answer: In two ways. The product itself is autonomous endpoint prevention operating at the kernel layer, and Cyber Crucible controls its own corporate and engineering endpoints under a formal program. Both rest on building the most privileged code in-house.

Product endpoint protection

The product provides autonomous prevention against ransomware, in-memory (fileless) attacks, and process injection, enforced at the kernel layer. It is deliberately decoupled from operating-system libraries, so when attackers compromise or hijack those libraries in memory the product continues to run, enforce, and report. Because detection and response execute locally, protection continues during a management-backend outage or in a deliberately air-gapped deployment.

Supply-chain integrity and independent validation

The discovery algorithms, kernel heuristics, sensors, and drivers are engineered and maintained in-house; no unverified third-party libraries or hidden telemetry hooks are embedded in the software. Because the highest-privilege component is proprietary, an entire class of third-party supply-chain exposure is removed rather than merely managed. All Windows kernel drivers are validated and signed under Microsoft's Windows Hardware Compatibility Program (WHCP).

Corporate endpoint controls

Managed endpoints follow approved secure configuration baselines, run endpoint protection and scanning, and apply the same controls to removable and portable media as to fixed media, with unauthorized media use restricted. Software installation is governed by policy, data at rest on endpoints is encrypted, and endpoints are kept current under the patch-management program.

Framework alignment and the honest boundary

Endpoint controls align to the CIS Critical Security Controls (v8) and NIST SP 800-53 (SI-3, CM-2, MP-7). No certification is claimed. Detailed configuration standards and evidence are provided to reviewers under NDA.

Revision #1
Created 2026-07-24 23:08:44 UTC by Dennis Underwood
Updated 2026-07-24 23:08:44 UTC by Dennis Underwood