

Architecture & Technology

How Cyber Crucible works under the hood - Genetic AI, kernel-level operation, Rogue Process Prevention, operating-system independence, and why decisions are made on the endpoint.

- [Why do you respond by suspending programs?](#)
- [Does Cyber Crucible stop all process injection?](#)
- [What are these strange, random looking files I see?](#)
- [How does Cyber Crucible use memory analytics?](#)
- [What is special about Cyber Crucible canary files?](#)
- [How is CC's Memory Behavioral Analysis different from memory scanning?](#)
- [Where do canary files exist?](#)
- [What Happens When CC Finds Memory Attacks](#)
- [What is Genetic AI, and how is it different from an LLM?](#)
- [How does Cyber Crucible make a decision in under 200 milliseconds?](#)
- [What is Rogue Process Prevention \(RPP\)?](#)
- [What role does memory analytics play across the platform?](#)
- [Why does Cyber Crucible operate at the kernel level?](#)
- [What does it mean that Cyber Crucible is independent of Windows libraries?](#)
- [Does Cyber Crucible work offline or in air-gapped environments?](#)
- [Why does Cyber Crucible respond by suspending a process instead of shutting down the system?](#)

Why do you respond by suspending programs?

Criminals have learned that operating in memory only, using techniques such as process injection, provide a variety of advantages:

1. Assuming control of a user's identity for that program.
2. Evasion from identity management checks.
3. Evasion from application-based firewall rules and whitelists.
4. Evasion from data loss prevention tools on servers.
5. Deletion of almost all in-memory evidence when the process is killed.
6. Deletion of almost all in-memory evidence once the computer is rebooted.

By suspending the programs under control of the attacker, evidence is frozen in time.

The attacker is also frozen.

Evidence is available for forensic tools as part of an investigation process.

If the organization is not equipped to perform forensics at the time, a simple reboot by the attacker will bring the system back to a healthy state. Force killing the suspending program, and restarting it, usually also has the desired effect.

Does Cyber Crucible stop all process injection?

No!

There are a variety of activities which occur on systems which are process injection, but are not necessarily malicious.

Cyber Crucible makes all of these process injection activities visible for threat hunting and root cause analysis, but do not stop the activity unless the data extortion activities of data theft or ransomware encryption are begun.

Below is an example of process injection which is performed by PC Matic endpoint security software, called PCPitStop. It is likely a behavior they are conducting to inspect running processes.

5373968.png?width=680

Due to Cyber Crucible's kernel-level behavioral analytics, you have complete visibility into all activities, regardless of the permission level of the software (to include highly privileged Microsoft or antivirus vendor software).

If an automated response is conducted, this page, and the Process Creations page, can be searched to provide fast investigation results.

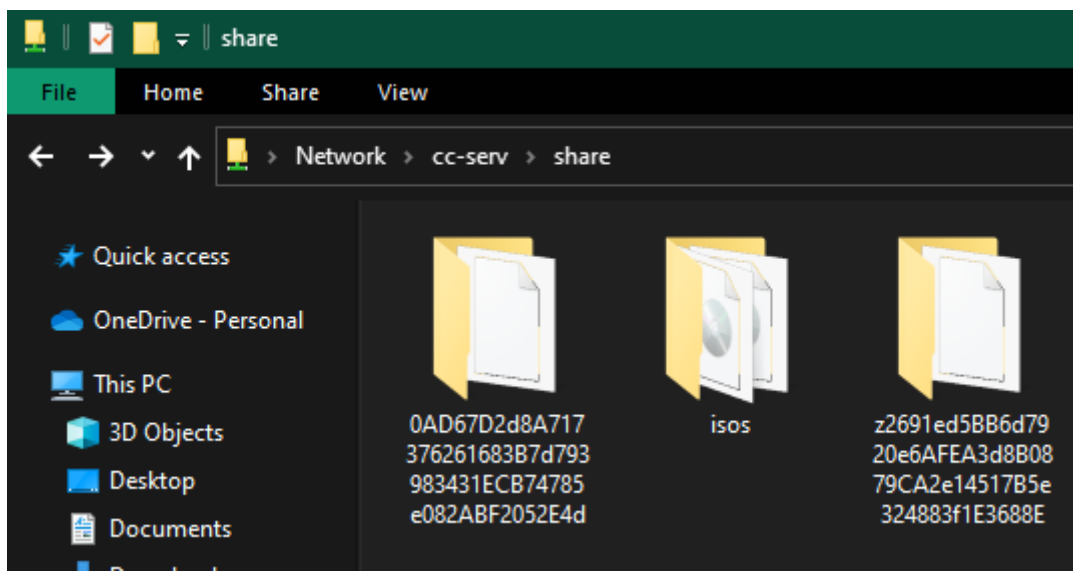
Additionally, the REST API may be leveraged by threat hunting teams to query for indicators of possible malicious in-memory tradecraft.

What are these strange, random looking files I see?

Our Data Extortion Prevention software receives behavioral indicators from several engines before making a split-second decision whether an attack is at hand. One of the engines, the file monitoring engine, uses special files in several locations.

Those files you sometimes see, sometimes not, are there by Cyber Crucible. Depending on how an application may list files, you may see them. For example, a program that use File Explorer to list files to save or open, will now show you them.

We call those files canaries, like, “canary in the coal mine” ([wiki](#)).



How does Cyber Crucible use memory analytics?

Cyber Crucible kernel analytics track the state of the memory of a program during its entire lifecycle, from program start to program end. It also tracks the state of in-memory interactions with other programs.

The behavioral monitoring tracks for unsafe states in the memory of the program. This could mean an exploit, a software bug, or known in-memory (aka, “file-less”) attacks occurring.

This monitoring is performed live on the programs in a highly efficient manner, to ensure system and customer operations are not affected by Cyber Crucible analysis.

At point of possible attacker behavior for identity or data theft or extortion, this memory behavioral tracking and analysis is included in other telemetry gathered, to determine whether a program should be suspended.

What is special about Cyber Crucible canary files?

Canary files have special data hidden in them, in the event they are ever stolen by an attacker. Cyber Crucible can trace the file back to the original owner.

The file names are mostly-random (called psuedo-random) in a way that attackers cannot identify them and avoid them. The names are created with a special Cyber Crucible math formula, that allows our software to recognize a file as a canary, even if the names are different from machine to machine or file to file.

The ability for Cyber Crucible software to recognize these files also means that there only needs to be one canary file per server, even if thousands of workstations are using the server.

How is CC's Memory Behavioral Analysis different from memory scanning?

What Is Memory Scanning?

Memory scanning is the practice of taking a block of memory, normally copying it to a different location in the system for analysis, and scanning it for analysis.

Photograph vs Live Action

This is not live scanning of the program as it behaves, but is a snapshot in time of the program. Think of it like a photograph, versus a live streaming. The state of the program may be dramatically different from one moment to the next, so it is most useful as a forensic exercise in a controlled environment. It is less useful in a non-controlled environment, like a customer environment. The busier the machine, or the more a program is doing the more difficult it is to have an accurate picture of the current state of a program.

Cyber Crucible started with heavy use of memory scanning, trying to take lots of “pictures” of the memory states of programs. As we became faster, we hit multiple bottlenecks on how fast memory snapshots could be taken, and across multiple programs.

CPU or GPU resource consumption concerns

This memory scanning can be extremely resource intensive, and can take some time for large programs. When Cyber Crucible was using memory scanning, instead of intelligence behavioral memory monitoring, an entire CPU core was dedicated to the scanning.

Intel has an innovative technology called [Intel TDT AMS](#), which is available on specific Intel based architectures with an embedded Intel GPU.

It scans the memory of a program when it is first started, using the GPU to help not bog down the CPU. Also, the GPU is more performant for the type of scanning that is performed (more on that later).

Scanning Methodology

Memory scanning is performed to look for specific codes or “strings”. These are often the same pieces of data that antivirus/EDR tools look for in potentially malicious files.

Memory scanning specifically is more limited than some of the complex searching that can be done against a file.

Therefore, the scanning by default is looking for already known items, like older malware, but is limited in its ability to do so.

Cyber Crucible used to scan memory segments for cryptographic material, but it proved too limited for modern environments, and we invented a different method to observe behaviors, using kernel-based logic.

Behavior Based vs Scanning Methodologies

Cyber Crucible migrated from a scanning-based methodology to a behavior-based one for a couple reasons.

Scanning Cons

The first is that a live picture needed to be used across all programs. Snapshots were really difficult to complete even limited analysis, before another one was needed. Too much program activity was being missed, almost like having a video camera operating at too low a frame rate. This was compounded with the need to monitor all programs simultaneously. Servers or busy workstations, arguably the computers that needed the most monitoring and were the most important, ended up with the lowest percentage of surveillance being conducted. Regardless of the company or product, the technology limitations were obvious and unworkable.

The second was that, beyond resource consumption, scanning requires, by default, looking for **something known**. So, memory scanning has to be signature based. Also, the signature scanning must be a limited subset of the more fully featured signature scanning found in file-based antivirus

scanning. Attackers have already evaded file-based signature defenses for year; there is very little chance of attackers being discovered by a less capable signature-based system.

Behavioral Analysis Pros

Cyber Crucible's behavior-based memory monitoring does not require signatures, and leverages kernel memory access to track behaviors of all programs simultaneously. Let's break that apart.

First, behavioral monitoring means that new malware, or malware that has been edited by attackers to look new, can be easily discovered in real time. Signature based scanning, whether of memory or files, relies on having seen that specific piece of malware before. Attackers have been evading signature based defenses for years, despite a full suite of scanning techniques against files. The scanning techniques available against memory is a subset of that, and is even less effective.

Second, Cyber Crucible's novel memory behavioral monitoring is able to stay real time across all programs simultaneously, for the full execution of programs. Memory scanning suffers from severe resource bottlenecks that only get worse as more programs run. Attackers normally run dozens of malware samples at once, choking memory scanning analysis through the speed and number of programs to examine at once, regardless of whether it is zero day or old malware.

Where do canary files exist?

Canaries are found in various places in your file system.

Anytime a new source of files appears on your system, from any possible user, Cyber Crucible automatically checks to see if canaries have already been placed. That includes removable drives like USB sticks, network file sources, and cloud-based file storage.

What Happens When CC Finds Memory Attacks

First, it is very important to note that memory alterations happen for a couple different reasons:

1. In-memory attacks
2. Interaction between programs
3. Software bugs
4. Exploits

So, a process injection or memory altering technique used, even between programs, is not necessarily a malicious action.

If a program which has suffered a likely malicious memory event (so, #1 or #4 above) is involved in an automated response by Cyber Crucible, an automated collection activity occurs.

The portion of memory that was altered in a program, as well as the “before” is saved to a structured Cyber Crucible format called `.ccdiff`

That injected or altered portion of memory is saved and is available for download in the Cyber Crucible web portal.

Why Is This Not Automatically Submitted to Virustotal (etc) for analysis?

In some circumstances, the malicious code that is used in an attack has variables which are unique to an attack, specifically to a victim.

Submitting this malicious code to a public analysis engine such as Virustotal would represent an undesired victim-identifying information disclosure to the security community.

Why Would Submitting This Binary Data to Virustotal (etc) be helpful?

Signature based antivirus and EDR tools look for strings or bytes of code seen in last attacks. Even if this malware has never been seen before, it may after a bit of time be known to the security community. These scanning tools usually do not need the code to be in a properly formatted binary to flag an association with a known piece of malware.

What is Genetic AI, and how is it different from an LLM?

Short answer: Genetic AI is Cyber Crucible's purpose-built security AI. It models how programs behave — in memory, across processes, and against files — and detects malicious deviations in real time. It is not a large language model; LLMs analyze text patterns, which is a different problem entirely.

What makes it different

- **Purpose-built for security**, not adapted from a general text model.
- **Behavioral, not signature-based** — no signatures, no threat intelligence feeds, no waiting for someone else to discover the attack first.
- **Fully autonomous** — traditional tools rely on human-machine teaming; Genetic AI decides and acts on its own.
- **Pre-execution** — it blocks malicious behavior before it runs, rather than describing it afterward.

Vertical AI

Genetic AI is an example of vertical AI: artificial intelligence customized for the specific challenges of one sector rather than general-purpose reasoning. In security, that specialization is what allows a sub-200-millisecond decision on intent.

The results are measurable — Cyber Crucible has stopped multiple zero-day attacks on customer networks up to 90 days before any other vendor reported or responded to them.

How does Cyber Crucible make a decision in under 200 milliseconds?

Short answer: It runs a three-step loop — Detect, Decide, Respond — entirely on the endpoint at the kernel level, using a patented edge computing analytics engine. Because nothing has to travel to a cloud service and back, and no human is in the loop, the whole cycle completes in under 200 milliseconds.

The loop

1. **Detect** — novel kernel-level sensors identify malicious behavioral patterns in memory, process activity, and file access.
2. **Decide** — the AI-enhanced behavioral model assesses the intent of the program's data access instantly.
3. **Respond** — if intent is malicious, the program is intercepted and suspended.

Why local processing is non-negotiable

Sending telemetry to a cloud server, waiting for analysis, and receiving a response back adds milliseconds the attack will happily use. For context on the scale involved: this decision happens faster than a nerve impulse travels from your eye to your brain. Any architecture with a network round trip in the critical path has already lost that race.

Raw telemetry is still copied to servers afterward for evidence and enrichment — but the protective decision never waits for it.

What is Rogue Process Prevention (RPP)?

Short answer: Rogue Process Prevention is Cyber Crucible's patented approach of interceding against a malicious process rather than merely observing it. Whether it's ransomware starting to encrypt or an infostealer reaching for a session token, the process is stopped before data leaves the device.

Watching versus intervening

Most of the endpoint market is built to watch. RPP is built to intervene. The distinction is the difference between a security camera that records a burglary for later review and a guard who stops the thief at the vault.

What it protects against

RPP addresses the three core threats together, because modern attacks are usually multi-pronged:

- **Identity theft** — access to keys, tokens, and passwords, often with no employee interaction at all.
- **Information theft** — bulk exfiltration of data at machine speed.
- **Ransomware** — behavioral analysis plus hacker deception technology that prevents encryption of even a single file.

A defense narrowly focused on ransomware alone would miss the first two stages, which is where the real damage usually begins.

What role does memory analytics play across the platform?

Short answer: Memory analytics is the foundation everything else is built on. It is the sensor and analytic source feeding all behavioral modeling — data protection, identity protection, and FortressAI alike. Every decision those capabilities make rests on knowing what a program's memory is actually doing.

One foundation, three capabilities

Cyber Crucible tracks the state of a program's memory across its entire lifecycle, from start to end, including its in-memory interactions with other programs. That single stream of analysis is what makes the rest possible:

- **Data protection** — is this process behaving like something about to encrypt or exfiltrate?
- **Identity protection** — is this process the legitimate owner of the credential it's reaching for, or has it been taken over?
- **FortressAI** — is this AI tool still itself, and is its data access within policy?

These aren't three separate engines with three separate sensor sets. They are three policies expressed on top of the same memory behavioral model.

Why memory is the right foundation

Modern attacks increasingly leave no file to inspect. Code is injected into trusted processes and built in memory, so the only place the activity is visible is the memory state of a running program. A defense that can't see there is blind to the entire class.

Memory analysis also answers the question that matters most before granting access to anything sensitive: **is this program still what it claims to be?** A trusted application whose process or libraries have been tampered with is no longer trustworthy, and only memory-level inspection reveals that.

Not memory scanning

This is live behavioral tracking, not periodic scanning. Scanning copies a block of memory and analyzes the snapshot — a photograph rather than live footage. On a busy machine the program's state may be entirely different a moment later, which makes snapshots useful for controlled forensics and much weaker for real-time defense.

“ See also the existing *Memory Analytics* articles for the technical detail on how this differs from memory scanning, and what happens when a memory attack is found.

Why does Cyber Crucible operate at the kernel level?

Short answer: The kernel is the deepest, most privileged layer of the operating system. Operating there lets Cyber Crucible see and control all system activity, identify a malicious process's intent, and stop it in milliseconds — and it makes the tool far harder for an attacker to tamper with or disable.

What kernel access provides

- **Complete visibility** into memory behavior, process behavior, and file access, including activity that never touches disk.
- **The ability to act**, not just observe — permissions can be revoked and processes suspended at the moment of malicious intent.
- **Resilience**, because attackers will inevitably try to disable the security tool itself.

On the criticism of kernel development

Some incumbents and analysts discourage kernel-level development. Kernel engineering is genuinely difficult and expensive, which is precisely why much of the market avoided it — and why endpoints were left dependent on user-space telemetry that root-level attacks can neutralize.

Cyber Crucible's quality and tamper resistance are validated through Microsoft's Windows Hardware Compatibility Program (WHCP).

What does it mean that Cyber Crucible is independent of Windows libraries?

Short answer: Cyber Crucible does not rely on the Windows system libraries it is protecting. It was deliberately decoupled from them, so that when attackers compromise, crash, or hijack those libraries, Cyber Crucible keeps running, keeps enforcing, and keeps reporting.

Why this was built

In the second half of 2025, attackers shifted to infecting core Windows libraries in memory. Because nearly every security vendor depends on those same libraries to function, that single move rendered a large portion of the market blind.

Cyber Crucible spent roughly nine months inventing novel sensors and a further five-month engineering effort to remove Windows library dependencies entirely — including eliminating volatile MiniFilter dependencies.

What it looks like architecturally

The architecture runs seven fully independent subsystems inside the kernel, including its own isolated encryption, networking, metadata analysis, and event tracking. A further phase of this work replaces the standard Windows networking library (`ws32_2.dll`) with a custom kernel-level network stack.

Proven in the field

With two customers, attackers blocked by the kernel-only decision engine then tried to prevent the customer from being notified by attacking the OS networking layer. Because Cyber Crucible's network communications are independent of the operating system, protection held and the intelligence still reached the customer.

Does Cyber Crucible work offline or in air-gapped environments?

Short answer: Yes. All analysis and enforcement happen locally on the endpoint, so Cyber Crucible provides full protection with no internet connection, no cloud service, and no signal of any kind. It is suitable for air-gapped networks, OT and ICS environments, and disconnected field operations.

Why it works without connectivity

Because the Detect-Decide-Respond loop runs entirely in the kernel on the device, connectivity is never required to make a protective decision. Telemetry syncs opportunistically for evidence and reporting when a connection is available, but protection does not depend on it.

Proven at sea

In a maritime deployment, Cyber Crucible operated independently across multi-day voyages with no IT staff aboard, adapted to low-bandwidth and bursty satellite links with weather interruptions, and retained rich forensic logs through long disconnected stretches for post-voyage analysis. Protected systems ranged from passenger-facing recreational terminals to mission-critical maritime controls.

This is also why the same agent suits manufacturing floors, logistics fleets, and other edge environments where cloud connectivity is unreliable or deliberately absent.

Why does Cyber Crucible respond by suspending a process instead of shutting down the system?

Short answer: Suspending only the malicious process — sometimes called a Selective Freeze — neutralizes the attack while everything else keeps running. There's no full-system lockdown, no reboot, and no interruption to legitimate work.

The problem with lockdown-style responses

When many detection tools identify an anomaly, the containment options are blunt: isolate the machine or take it offline. In a hospital, a plant, or a trading operation, that response can be nearly as damaging as the attack.

What selective suspension achieves

The attack is suspended in memory. Legitimate transactions never stop. In the financial services deployment where nearly 10,000 malicious processes were intercepted, legitimate SQL transactions continued uninterrupted throughout — operations were never taken down.

This is what makes prevention compatible with business continuity rather than a trade-off against it.