

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

Created by Dennis Underwood, last modified on Feb 08, 2023

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.

Revision #1

Created 2026-05-18 20:28:43 UTC by Dennis Underwood

Updated 2026-05-18 20:28:43 UTC by Dennis Underwood