Explained: Smart Memory Dumping

vmray.com/cyber-security-blog/smart-memory-dumping/

In a recent major update of our flagship platform, VMRay Analyzer 3.0, we made dramatic improvements in the system's memory dumping capabilities. In an automated approach we call *smart memory dumping*, VMRay Analyzer now triggers more frequent and more relevant memory dumps to capture a comprehensive view of malware characteristics and behavior. In turn, this increases the speed and accuracy of malware analysis and detection.

For incident responders, smart memory dumping makes their jobs easier by automating many tasks that previously required manual intervention. In addition, more frequent and relevant dumps enhance digital forensics by capturing a more complete record of artifacts created by threats and attacks.

What Gets Packed, Must be Unpacked

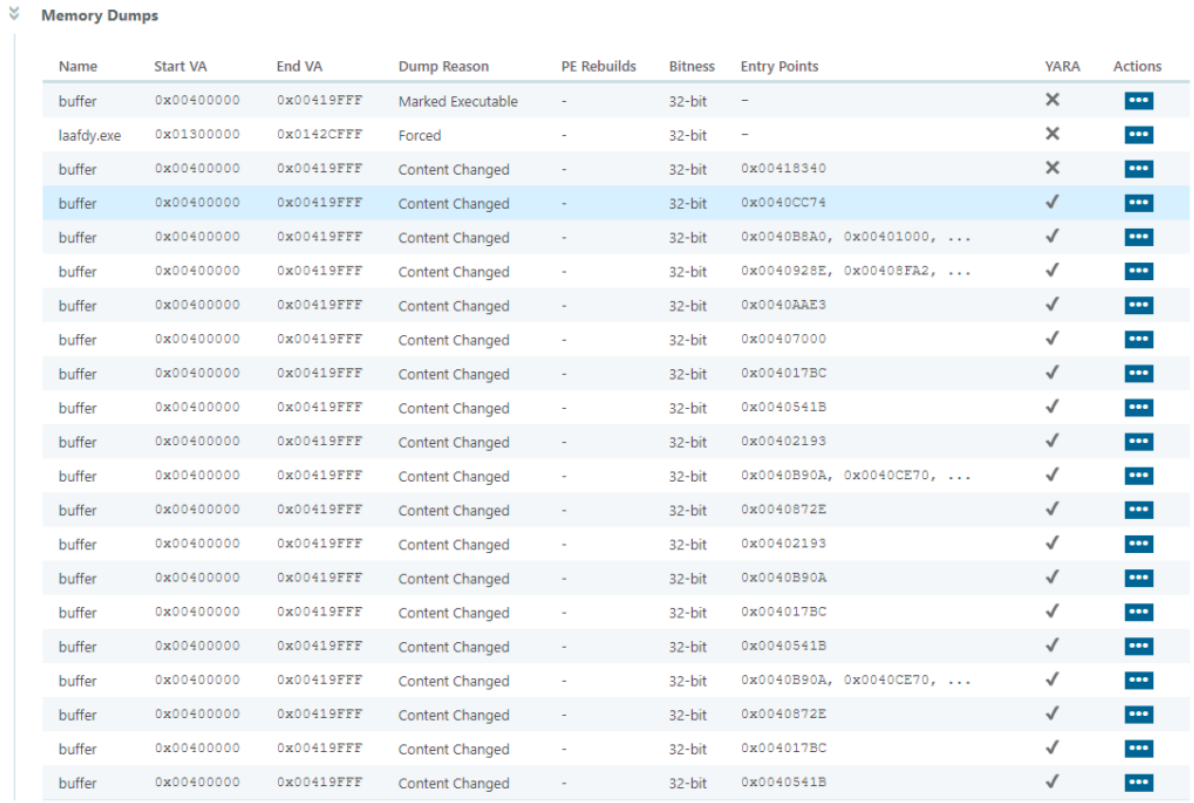
Malware authors routinely combine multiple techniques to evade detection and analysis by layered defenses: anti-virus, static analysis, and dynamic analysis tools. A popular approach is to compress, encrypt and/or obfuscate malicious files, making it harder and more time-consuming to identify and classify the malware—and address the threat it poses.

To do their dirty work, however, malware files need to be restored to their original state in memory so they can execute as intended. And that creates an opportunity to detect potentially malicious behavior in the safe environment of the sandbox.

As successive layers of protection are removed—the packer layer, the compression layer, the obfuscation layer—the characteristics and behavior of malware are revealed. For instance, memory has to be allocated to store decrypted or deobfuscated data and code. In response to such changes, the VMRay Platform triggers a succession of memory dumps, creating snapshots of telltale information about a potential threat or attack. (See Figure 1.)

Some examples of changes that trigger memory dumps:

- Private memory regions/buffers are marked as executable
- Code has executed in the memory region
- The memory region was written into



Memory Dumps

Name	Start VA	End VA	Dump Reason	PE Rebuilds	Bitness	Entry Points	YARA	Actions
buffer	0x00400000	0x00419FFF	Marked Executable	-	32-bit	-	✕	•••
laafdy.exe	0x01300000	0x0142CFFF	Forced	-	32-bit	-	✕	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x00418340	✕	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x0040CC74	✓	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x0040B8A0, 0x00401000, ...	✓	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x0040928E, 0x00408FA2, ...	✓	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x0040AAE3	✓	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x00407000	✓	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x004017BC	✓	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x0040541B	✓	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x00402193	✓	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x0040B90A, 0x0040CE70, ...	✓	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x0040872E	✓	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x00402193	✓	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x0040B90A	✓	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x004017BC	✓	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x0040541B	✓	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x0040B90A, 0x0040CE70, ...	✓	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x0040872E	✓	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x004017BC	✓	•••
buffer	0x00400000	0x00419FFF	Content Changed	-	32-bit	0x0040541B	✓	•••

Figure 1: VMRay triggers a series of dumps for a memory region at important phases of memory usage

In the screen capture above, taken from a VMRay Analyzer report on Remcos, a series of dumps is created in response to code being executed in the memory region that starts at 0x400000.

(Remcos is a remote access tool widely used for malicious purposes.) Here's what this high-level view shows:

- Because VMRay Analyzer automatically applies YARA rules to memory dumps, all the resulting YARA matches are displayed.
- The dump listed on Line 1 appears to be benign as no YARA matches were found.

- Likewise, there are no YARA matches for the dump listed on Line 3. However, the content of memory has changed, and that triggers a subsequent dump that we'll examine shortly in Figure 2.
- Line 4 shows that, after monitoring and dumping the region, YARA matching detected potentially malicious indicators in that particular dump. Later in the post, we'll show the memory dump−after it has been loaded into IDA for easier analysis−and we'll explore it in a bit more detail. (See Figure 3.)

Why Timing and Frequency Matter

With memory dumping, frequency and timing matter. Dump too often and you'll create more information than you need. Dump infrequently or at the wrong times and you'll miss information that could enhance analysis and detection, such as identifying malicious URLs or registry keys the malware is designed to access.

Smart memory dumping addresses these challenges. In a process analogous to burst mode on a camera, triggers built into VMRay dynamically initiates a memory dump any time something changes (see Figure 2).

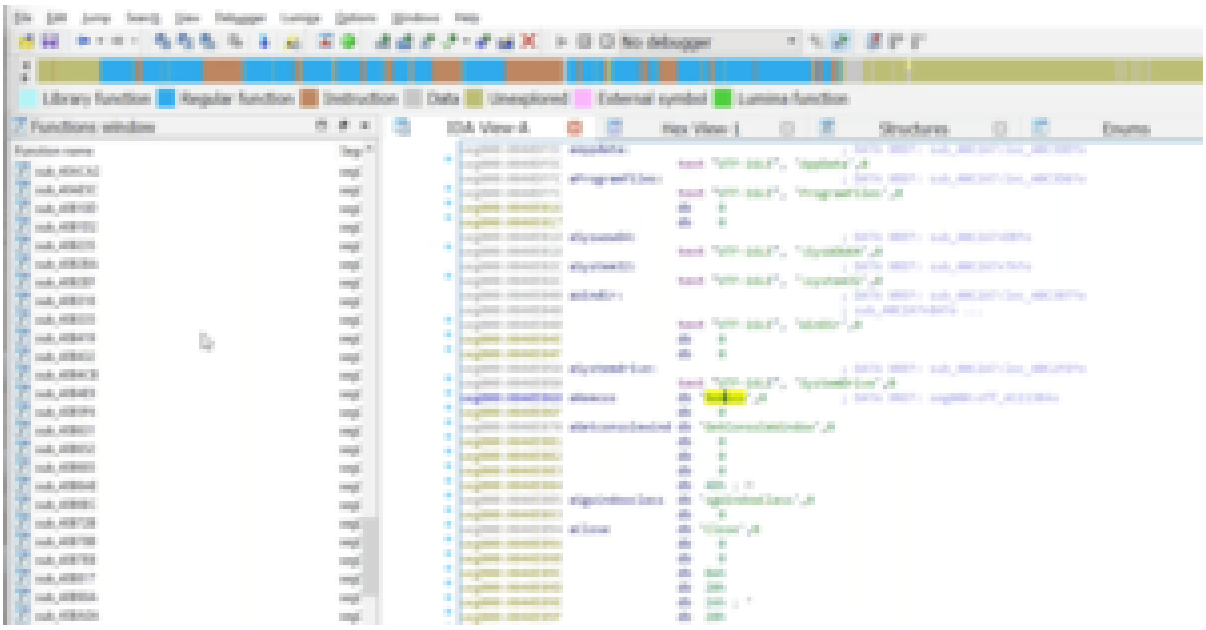


Figure 2: VMRay Analyzer memory dump loaded in IDA

This screenshot of a VMRay Analyzer memory dump loaded in IDA shows details from one of the first memory dumps we saw initiated in Figure 1 when the code starting at Entry Point 0x00418340 executed.

- The first several lines, highlighted in green, show areas of memory that are empty, indicating they're free of threats.

- However, the next line, in red, shows a set of instructions, starting with the pusha instruction in the previously mentioned memory location 0x418340. VMRay captures the whole sequence of instructions that follow, also shown in red.

Though not illustrated here, the importance of timely memory dumps is easily understood in the context of malware that “cleans up” by re-encrypting parts of itself that have been temporarily decrypted so they can execute. The decrypted data may only be visible for an infinitesimal period of time, but VMRay takes a snapshot that captures it before it disappears again.

Better Malware Classification & Streamlined Manual Analysis

Improved memory dumping creates a more complete and detailed picture of malware activity, for example by revealing suspect code and strings associated with known signatures and malware families.

- Continuing with the example we’ve been discussing, Figure 3 now shows that the benign-looking region in Figure 2 has changed, revealing strings and other indicators of the malware.
- You would typically see this on malware that unpacks and decrypts parts of itself in memory. Our memory dumping feature was able to capture this transition.
- Highlighted in yellow, VMRay identifies the file as a likely Remcos sample. Remcos is a remote access tool (RAT) widely used for malicious purposes

In turn, that bolsters the automated pattern-matching power of YARA rules, resulting in faster, more accurate malware classification and better detection rates.

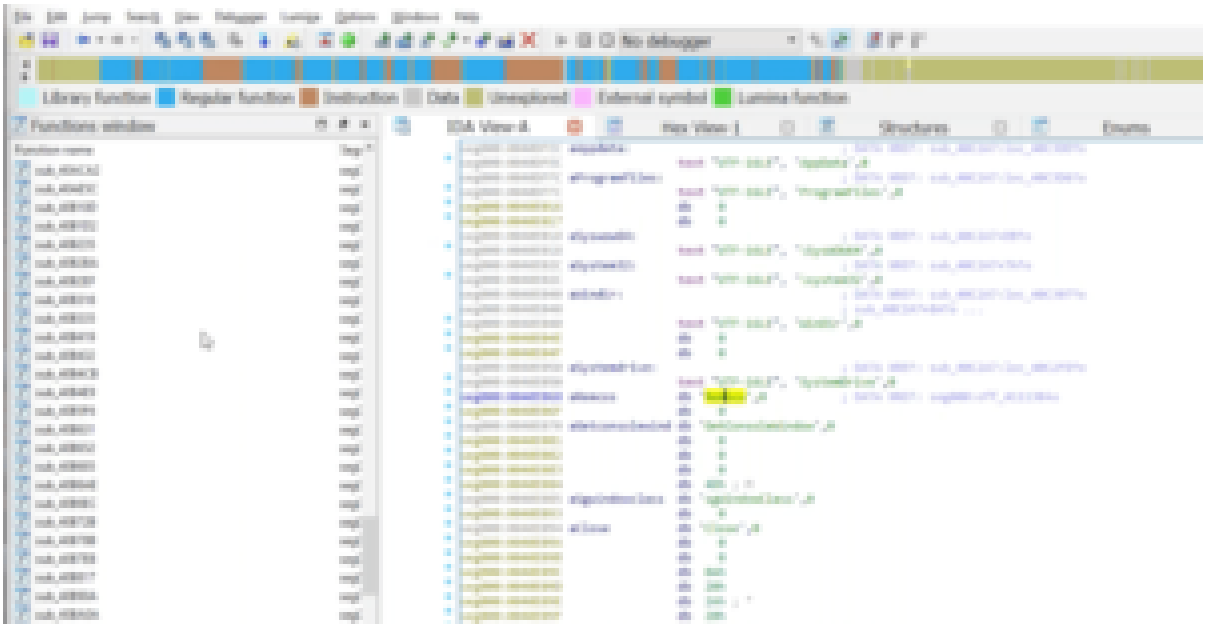


Figure 3: VMRay memory dumps, as shown here in the IDA interface, reveal concealed malware behavior to analysts

With VMRay, security teams can utilize their own custom YARA rules along with VMRay-provided rules. And they can more easily write new rules, for example, to automate the extraction of new C2 servers and malware samples.

Version 3.0's enhanced memory dumping also streamlines manual analysis. Decompressed or decrypted code that has been captured via a memory dump can be directly examined in a disassembler, such as IDA or GHIDRA. Translating machine language into higher-level languages makes human analysis easier and eliminates the tedious process of debugging packed files.

Get-hands on with VMRay Analyzer's enhanced memory dumping capabilities. Start your trial today!