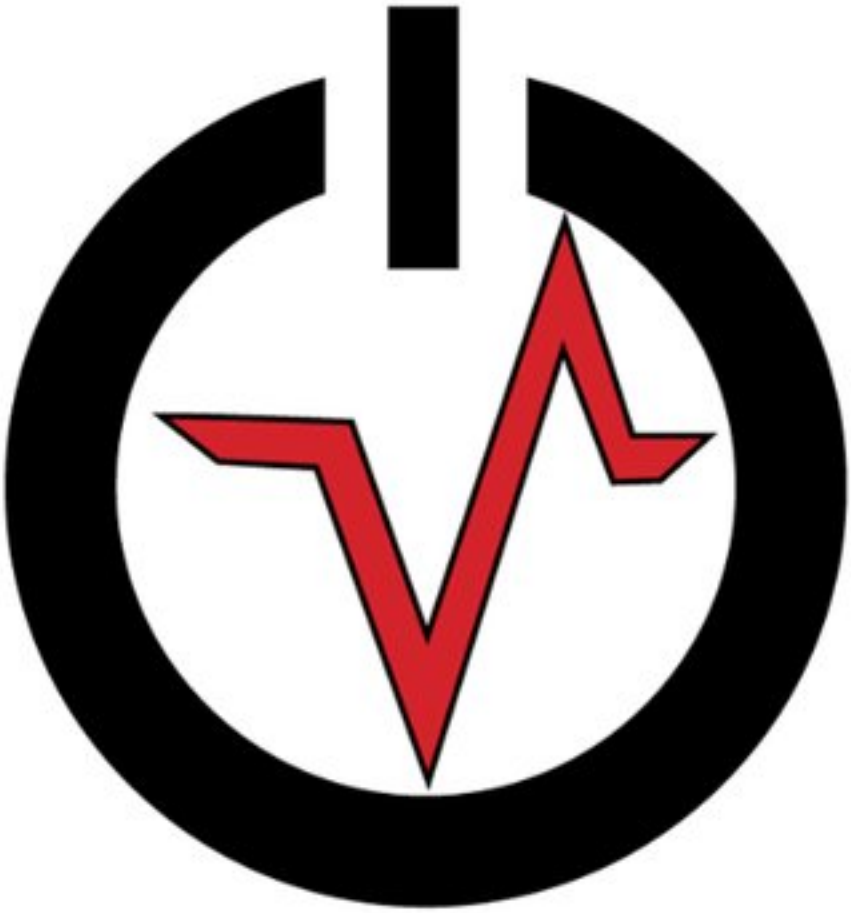
BP:Blue Primer

BP: Volatility



BP: Volatility

Part of the Blue Primer series, learn how to perform memory forensics with Volatility!

[Task 1] Intro

Volatility is a free memory forensics tool developed and maintained by Volatility labs. Regarded as the gold standard for memory forensics in incident response, Volatility is wildly expandable via a plugins system and is an invaluable tool for any Blue Teamer.

The virtual machine for this room can also be downloaded from https://darkstar7471.com, the credentials for the machine are as follows: voluser:volatility

#1
Install Volatility onto your workstation of choice or use the provided virtual machine. On Debian-based systems such as Kali this can be done via `apt-get install volatility`

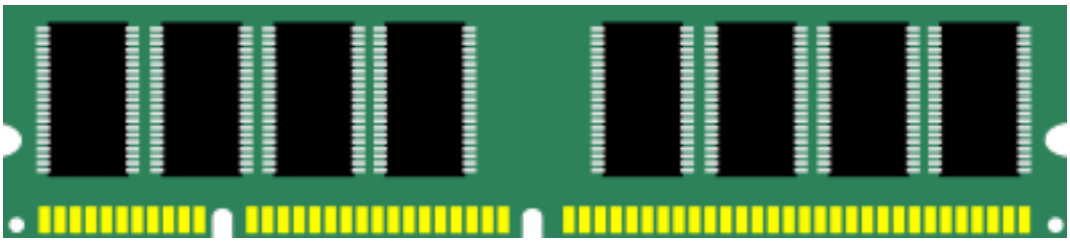
No answer needed

[Task 2] Obtaining Memory Samples

Obtaining a memory capture from machines can be done in numerous ways, however, the easiest method will often vary depending on what you're working with. For example, live machines (turned on) can have their memory captured with one of the following tools:

- **FTK Image - Link**
- **Redline - Link *Requires registration but Redline has a very nice GUI***
- **DumpIt.exe**
- **win32dd.exe / win64dd.exe - *Has fantastic psexec support, great for IT departments if your EDR solution doesn't support this***

These tools will typically output a .raw file which contains an image of the system memory. The .raw format is one of the most common memory file types you will see in the wild.



Offline machines, however, can have their memory pulled relatively easily as long as their drives aren't encrypted. For Windows systems, this can be done via pulling the following file:

%SystemDrive%/hiberfil.sys

hiberfil.sys, better known as the Windows hibernation file contains a compressed memory image from the previous boot. Microsoft Windows systems use this in order to provide faster boot-up times, however, we can use this file in our case for some memory forensics!

Things get even more exciting when we start to talk about virtual machines and memory captures. Here's a quick sampling of the memory capture process/file containing a memory image for different virtual machine hypervisors:

- **VMware - .vmem file**
- **Hyper-V - .bin file**
- **Parallels - .mem file**
- **VirtualBox - .sav file *_This is only a partial memory file. You'll need to dump memory like a normal bare-metal_**
- ***_system for this hypervisor_***

These files can often be found simply in the data store of the corresponding hypervisor and often can be simply copied without shutting the associated virtual machine off. This allows for virtually zero disturbance to the virtual machine, preserving it's forensic integrity.

#1
What memory format is the most common?

.raw

#2
The Window's system we're looking to perform memory forensics on was turned off by mistake. What file contains a compressed memory image?

hiberfil.sys

#3
How about if we wanted to perform memory forensics on a VMware-based virtual machine?

.vmem

[Task 3] Examining Our Patient

Now that we've collected our memory image let's dig into it! For those using their own workstation for this activity, I've provided a download link to our memory sample attached to this task. If you're using the workstation I've provided as a VM for this activity you'll find the memory image in the 'voluser' home directory.

My Memory by Dmitry Posudin

#1
First, let's figure out what profile we need to use. Profiles determine how Volatility treats our memory image since every version of Windows is a little bit different. Let's see our options now with the command `volatility -f MEMORY_FILE.raw imageinfo`

No answer needed

#2
Running the imageinfo command in Volatility will provide us with a number of profiles we can test with, however, only one will be correct. We can test these profiles using the pslist command, validating our profile selection by the sheer number of returned results. Do this now with the command `volatility -f MEMORY_FILE.raw --profile=PROFILE pslist`. What profile is correct for this memory image?

Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)

WinXPSP2x86

#3
Take a look through the processes within our image. What is the process ID for the smss.exe process? If results are scrolling off-screen, try piping your output into less

368

#4

In addition to viewing active processes, we can also view active network connections at the time of image creation! Let's do this now with the command `volatility -f MEMORY_FILE.raw --profile=PROFILE netscan`. Unfortunately, something not great is going to happen here due to the sheer age of the target operating system as the command netscan doesn't support it.

No answer needed

#5

It's fairly common for malware to attempt to hide itself and the process associated with it. That being said, we can view intentionally hidden processes via the command `psxview`. What process has only one 'False' listed?

csrss.exe

#6

In addition to viewing hidden processes via psxview, we can also check this with a greater focus via the command 'ldrmodules'. Three columns will appear here in the middle, InLoad, InInit, InMem. If any of these are false, that module has likely been injected which is a really bad thing. On a normal system the grep statement above should return no output. Which process has all three columns listed as 'False' (other than System)?

csrss.exe

#7

Processes aren't the only area we're concerned with when we're examining a machine. Using the 'apihooks' command we can view unexpected patches in the standard system DLLs. If we see an instance where Hooking module: <unknown> that's really bad. This command will take a while to run, however, it will show you all of the extraneous code introduced by the malware.

No answer needed

#8

Injected code can be a huge issue and is highly indicative of very very bad things. We can check for this with the command `malfind`. Using the full command `volatility -f MEMORY_FILE.raw --profile=PROFILE malfind -D <Destination Directory>` we can not only find this code, but also dump it to our specified directory. Let's do this now! We'll use this dump later for more analysis. How many files does this generate?

12

#9

Last but certainly not least we can view all of the DLLs loaded into memory. DLLs are shared system libraries utilized in system processes. These are commonly subjected to hijacking and other side-loading attacks, making them a key target for forensics. Let's list all of the DLLs in memory now with the command `dlllist`

No answer needed

#10

Now that we've seen all of the DLLs running in memory, let's go a step further and pull them out! Do this now with the command `volatility -f MEMORY_FILE.raw --profile=PROFILE --pid=PID dlldump -D <Destination Directory>` where the PID is the process ID of the infected process we identified earlier (questions five and six). How many DLLs does this end up pulling?

PID=584

volatility -f cridex.vmem --profile=WinXPSP2x86 --pid=584 dlldump -D /home/taj702/Downloads

12

[Task 4] Post Actions

Now that we've performed some basic forensics, let's go a step further and see what the community at large has to say about the items we've discovered. Check out the following two sites and upload the injected code we yanked out of our previous section. You can pull this code either via SCP with the box above, your local volatility workstation, or via a download link attached to this task.

VirusTotal - Link
Hybrid Analysis - Link

#1
Upload the extracted files to VirusTotal for examination.

No answer needed

#2
Upload the extracted files to Hybrid Analysis for examination - Note, this will also upload to VirusTotal but for the sake of demonstration we have done this separately.

No answer needed

#3
What malware has our sample been infected with? You can find this in the results of VirusTotal and Hybrid Anaylsis.

cridex

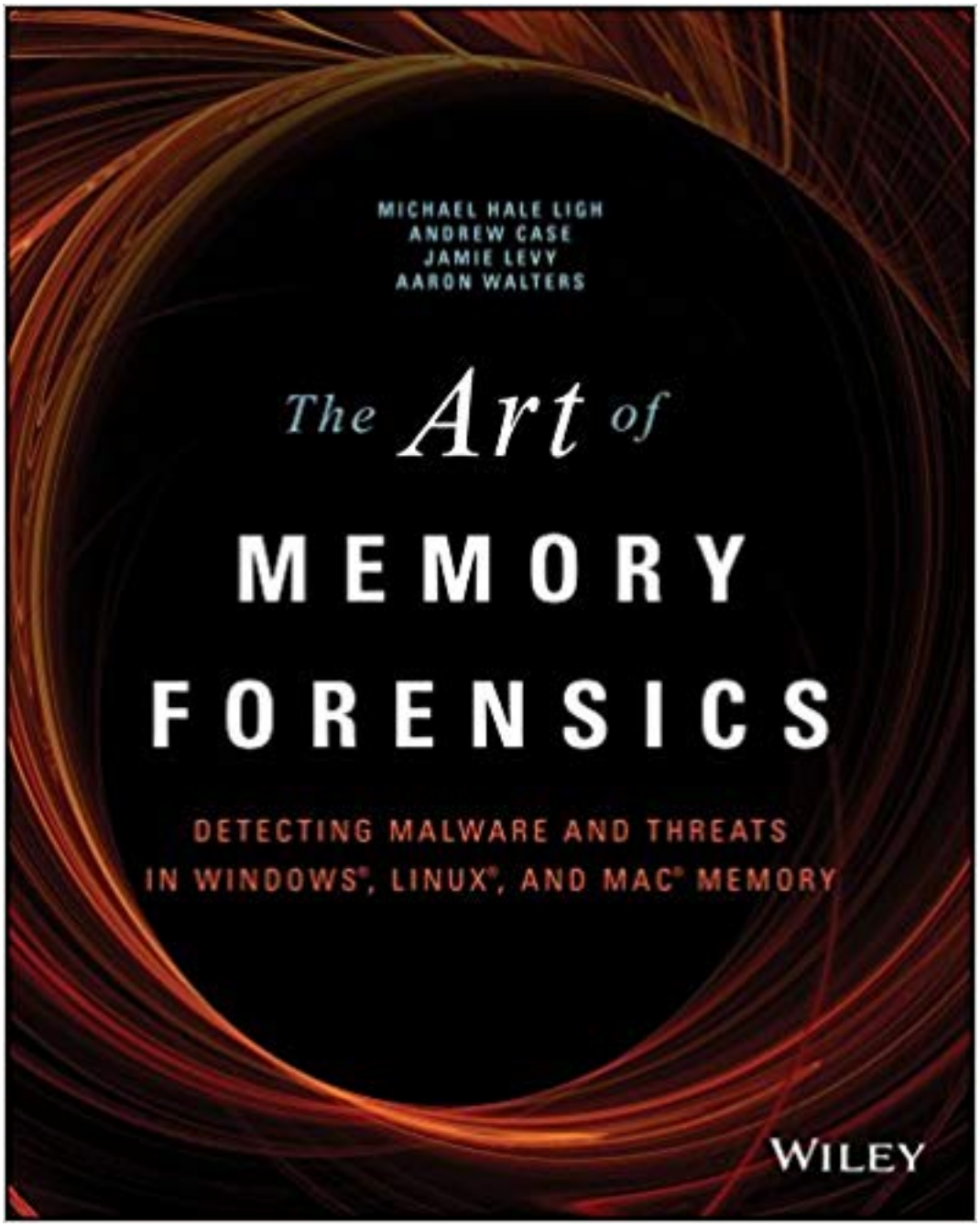
[Task 5] Extra Credit

Interested in going further? Here's a slew of awesome resources (both paid and free in no particular order) to check out and learn more!
AlienVault Open Threat Exchange (OTX) - An open-source threat tracking system. Create pulses based on your

malware analysis work and check out the work of others. Link

SANS 408 - Windows Forensic Analysis Link

"Memory Forensics with Vol(a|u)tility" - A great talk on learning the basics of Volatility and the GUI plugin VolUtility made by @chupath1ngee
"The Art of Memory Forensics" - Link



MemLabs - A collection of CTF-style memory forensic labs Link

#1
Check out the resources provided above!

No answer needed