Finding Metasploit's Meterpreter Traces With Memory Forensics

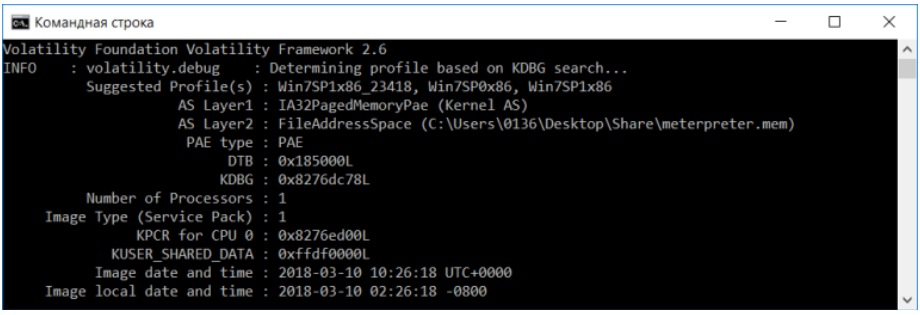
scar 5 mins ago

by Oleg Skulkin & Igor Mikhaylov

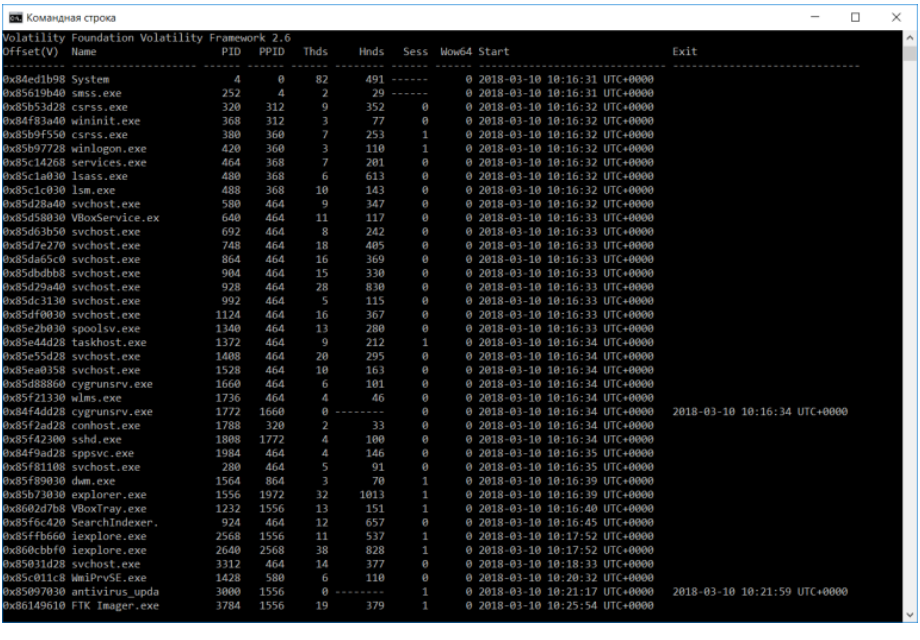
Metasploit Framework is not only very popular among pentesters, but is also quite often used by real adversaries. So why is memory forensics important here? Because Meterpreter, for example – an advanced, dynamically extensible Metasploit payload – resides entirely in the memory and writes nothing to the victim's drive. In this article we will show you how to use the Volatility Framework to find Metasploit traces with memory forensics.

As we are analyzing a memory image, first of all we should gather information about the operating system to choose the right Volatility profile. If you ask us, the best practice here is to document the OS version

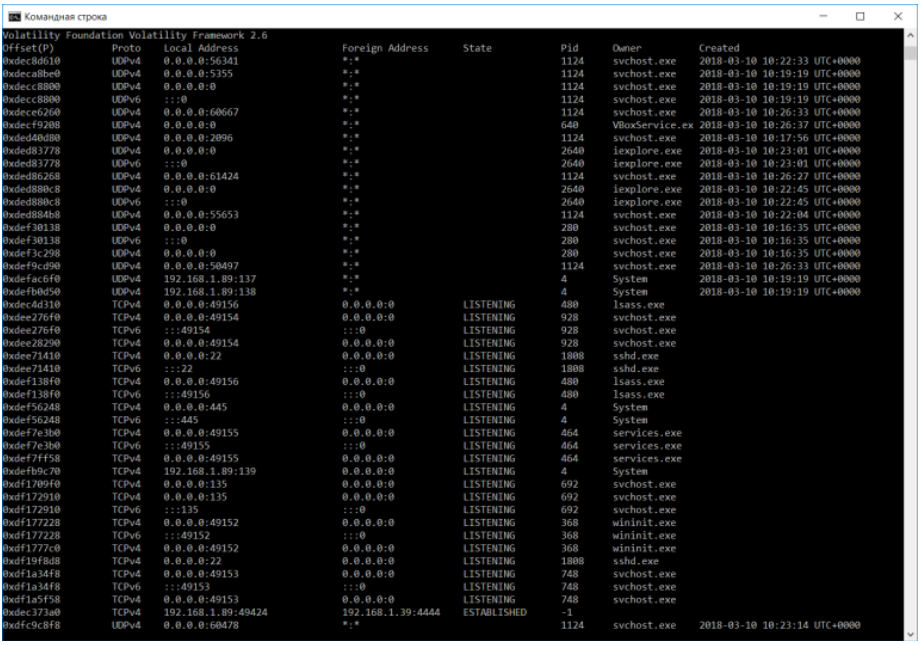
during memory imaging process, as Volatility does not always detect it correctly. Anyway, if you get the memory image from the third party and the OS version is unknown, use the **imageinfo** plugin:



So this time Volatility guessed the OS version right – it really was Windows 7 x86 with SP1. Ok, let's look at the process list using the **pslist** plugin:

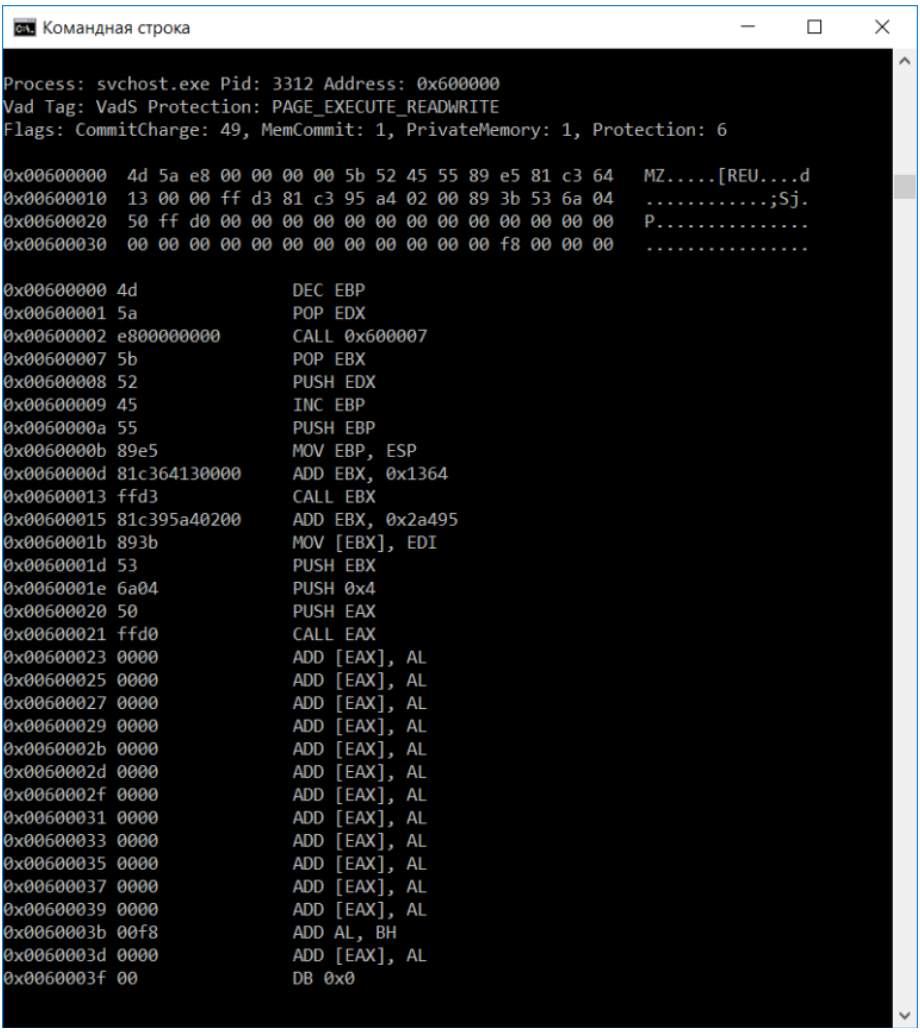


Do you see anything potentially malicious? What about the process with PID 3000? Hmm, probably the user initiated an antivirus updating process? But the strange thing is that this process exited 42 seconds after starting. Let's go further and look at network connections using the **netscan** plugin:



Ouch, an unknown process has established a connection to 192.168.1.39:4444. If you don't know, 4444 is the default Metasploit port to connect back to. As Meterpreter injects itself into the compromised process, let's try to find it using

the **malfind** plugin:



It seems like Meterpreter migrated to svchost.exe with PID 3312. Let's dump it to a file and check if it's detected by antiviruses:

Wait, wha-a-a-a-at?! Only joking, it's not that bad:

Anyway, it's not detected by lots of popular antiviruses like McAfee, Malwarebytes, DrWeb, etc. Shame on them!

If you like using YARA rules for malware detection, you can write your own rule or find some rules online, and use the **yarascan** plugin:

In this example we used a simple rule we have written:

So it seems that everything started from running that process with PID 3000. If we go back to pslist output, we see that the only web browser running is Internet Explorer (iexplore.exe, PIDs 2568 and 2640). Let's check browsing history using the **iehistory** plugin:

Bingo! The victim downloaded **antivirus_update.exe** from the server with the IP-address we have already seen! But what made them to do it? Let's dump Internet Explorer's processes memory with the **memdump** plugin and search for the "**antivirus**" string:

Ok, as you can see, the attacker used social engineering and shortened link to trick the victim. So when the victim downloaded the

file and ran it, the attacker got the meterpreter session and migrated it to svchost.exe (PID 3312).

But did the victim really run it? Let's find the evidence of execution! First of all, let's use **shimcache** plugin, as it's used to track compatibility issues with executed programs and may contain evidence we are looking for:

Yes, we got it! Let's go further, and use Registry forensics running the **userassist** plugin:

Wow! Two times! Our victim isn't very smart! What else can be used for getting the evidence of execution? For example, prefetch files. Yes, you can find these pieces of evidence in memory too, Volatility even have a plugin for it – **prefetchparser**.

Unfortunately, prefetching was disabled on our victim's system, so we haven't got any evidence.

Ok, we have gathered quite a lot, but there is one more thing to check – persistence! There is a very nice plugin to detect most common persistence techniques used by adversaries – **autoruns**:

As you can see, our victim doesn't have to run the "Antivirus Update" anymore, it will be started automatically with each reboot. That's it.

Happy forensicating!

About The Authors

Oleg Skulkin, GCFA, MCFE, ACE, is a DFIR enthusional (enthusiast + professional), Windows Forensics Cookbook and Practical Mobile Forensics co-author.

Igor Mikhaylov, MCFE, EnCE, ACE, OSFCE, is a digital forensic examiner with more than 20 years of experience and Mobile Forensics Cookbook author.

Categories: Forensics 101

Tags: Digital Forensics, malware, malware forensics, metasploit

Leave a Comment