

Forensic Focus – Articles

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

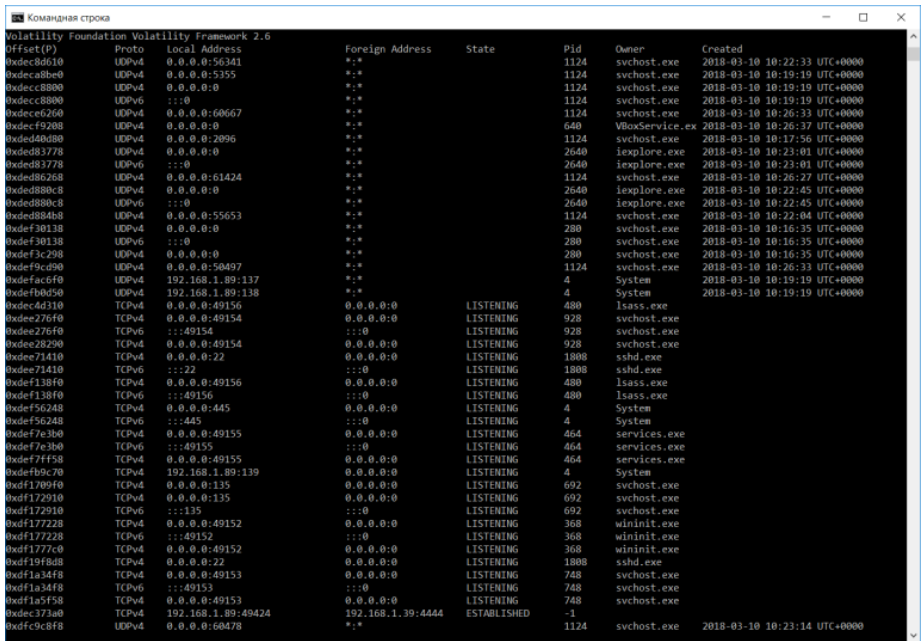
during memory imaging process, as Volatility does not always detect it correctly. Anyway, if you get the memory image from the third party and the OS version is unknown, use the **imageinfo** plugin:

```
Командная строка
Volatility Foundation Volatility Framework 2.6
INFO : volatility.debug : Determining profile based on KDBG search...
      Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86
      AS Layer1 : IA32PagedMemoryPae (Kernel AS)
      AS Layer2 : FileAddressSpace (C:\Users\0136\Desktop\Share\meterpreter.mem)
      PAE type : PAE
      DTB : 0x185000L
      KDBG : 0x8276dc78L
      Number of Processors : 1
      Image Type (Service Pack) : 1
      KPCR for CPU 0 : 0x8276ed00L
      KUSER_SHARED_DATA : 0xffdf0000L
      Image date and time : 2018-03-10 10:26:18 UTC+0000
      Image local date and time : 2018-03-10 02:26:18 -0800
```

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:

```
Командная строка
Volatility Foundation Volatility Framework 2.6
Offset(V)  Name                PID  PPID  Thds  Hnds  Sess  Wow64  Start                Exit
-----
0x84ed1b98 System              4      0    491  -----  0  2018-03-10 10:16:31 UTC+0000
0x85619b40 smss.exe       252    4      2  -----  0  2018-03-10 10:16:31 UTC+0000
0x85b53d28 csrss.exe     320   312    9   352    0  2018-03-10 10:16:32 UTC+0000
0x84f83a40 wininit.exe    368   312    3    77    0  2018-03-10 10:16:32 UTC+0000
0x85b9f550 csrss.exe     380   360    7   253    1  2018-03-10 10:16:32 UTC+0000
0x85b97728 winlogon.exe 420   360    3   110    1  2018-03-10 10:16:32 UTC+0000
0x85c14268 services.exe 464   368    7   201    0  2018-03-10 10:16:32 UTC+0000
0x85c1a030 lsass.exe      480   368    6   613    0  2018-03-10 10:16:32 UTC+0000
0x85c1c030 lsass.exe      488   368   10   143    0  2018-03-10 10:16:32 UTC+0000
0x85d28a40 svchost.exe 580   464    9   347    0  2018-03-10 10:16:32 UTC+0000
0x85d58030 VBoxService.exe 640   464   11   117    0  2018-03-10 10:16:33 UTC+0000
0x85d63b50 svchost.exe 692   464    8   242    0  2018-03-10 10:16:33 UTC+0000
0x85d7e270 svchost.exe 748   464   18   405    0  2018-03-10 10:16:33 UTC+0000
0x85da65c0 svchost.exe 864   464   16   369    0  2018-03-10 10:16:33 UTC+0000
0x85dbdbb8 svchost.exe 904   464   15   330    0  2018-03-10 10:16:33 UTC+0000
0x85d29a40 svchost.exe 928   464   28   830    0  2018-03-10 10:16:33 UTC+0000
0x85dc3130 svchost.exe 992   464    5   115    0  2018-03-10 10:16:33 UTC+0000
0x85df0030 svchost.exe 1124  464   10   367    0  2018-03-10 10:16:33 UTC+0000
0x85e2b030 spoolsv.exe 1340  464   13   280    0  2018-03-10 10:16:33 UTC+0000
0x85e44d28 taskhost.exe 1372  464    9   212    1  2018-03-10 10:16:34 UTC+0000
0x85e55d28 svchost.exe 1408  464   20   295    0  2018-03-10 10:16:34 UTC+0000
0x85ea0358 svchost.exe 1528  464   10   163    0  2018-03-10 10:16:34 UTC+0000
0x85d88860 cygrunsrv.exe 1660  464    6   101    0  2018-03-10 10:16:34 UTC+0000
0x85f21330 wlm.exe      1736  464    4    46    0  2018-03-10 10:16:34 UTC+0000
0x84f4dd28 cygrunsrv.exe 1772 1660  0  -----  0  2018-03-10 10:16:34 UTC+0000 2018-03-10 10:16:34 UTC+0000
0x85f2ad28 conhost.exe 1788  320    2    33    0  2018-03-10 10:16:34 UTC+0000
0x85f42300 sshd.exe     1808 1772    4   100    0  2018-03-10 10:16:34 UTC+0000
0x84f99d28 spoolsv.exe 1904  464    4   146    0  2018-03-10 10:16:35 UTC+0000
0x85f81188 svchost.exe 280   464    5    91    0  2018-03-10 10:16:35 UTC+0000
0x85f89030 dm.exe       1564  864    3    70    1  2018-03-10 10:16:39 UTC+0000
0x85b7f030 explorer.exe 1556 1972   32   1013   1  2018-03-10 10:16:39 UTC+0000
0x8602d7b8 VBoxTray.exe 1232 1556   13   151    1  2018-03-10 10:16:40 UTC+0000
0x85f6c420 SearchIndexer.exe 924  464   12   657    0  2018-03-10 10:16:45 UTC+0000
0x85f6b600 iexplore.exe 2568 1556   11   537    1  2018-03-10 10:17:52 UTC+0000
0x860cbbf0 iexplore.exe 2640 2568   38   828    1  2018-03-10 10:17:52 UTC+0000
0x85031d28 svchost.exe 3312  464   14   377    0  2018-03-10 10:18:33 UTC+0000
0x85c011c8 cmd.exe     1428  580    6   110    0  2018-03-10 10:20:32 UTC+0000
0x85097030 antivirus.upda 3000 1556    0  -----  1  2018-03-10 10:21:17 UTC+0000 2018-03-10 10:21:59 UTC+0000
0x86149610 FTK Imager.exe 3784 1556   19   379    1  2018-03-10 10:25:54 UTC+0000
```

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:



Offset(P)	Proto	Local Address	Foreign Address	State	Pid	Owner	Created
0xdec8d610	UDPv4	0.0.0.0:56341	*	*	1124	svchost.exe	2018-03-10 10:22:33 UTC+0000
0xdec8d610	UDPv4	0.0.0.0:5355	*	*	1124	svchost.exe	2018-03-10 10:19:19 UTC+0000
0xdec8d800	UDPv4	0.0.0.0:0	*	*	1124	svchost.exe	2018-03-10 10:19:19 UTC+0000
0xdec8d800	UDPv6	:::0	*	*	1124	svchost.exe	2018-03-10 10:19:19 UTC+0000
0xdec8e260	UDPv4	0.0.0.0:60667	*	*	1124	svchost.exe	2018-03-10 10:26:33 UTC+0000
0xdecf9208	UDPv4	0.0.0.0:0	*	*	640	VBoxService.exe	2018-03-10 10:26:37 UTC+0000
0xdec8d4080	UDPv4	0.0.0.0:2096	*	*	1124	svchost.exe	2018-03-10 10:17:56 UTC+0000
0xdec83778	UDPv4	0.0.0.0:0	*	*	2640	iexplore.exe	2018-03-10 10:23:01 UTC+0000
0xdec83778	UDPv6	:::0	*	*	2640	iexplore.exe	2018-03-10 10:23:01 UTC+0000
0xdec8d6268	UDPv4	0.0.0.0:61424	*	*	1124	svchost.exe	2018-03-10 10:26:27 UTC+0000
0xdec8d80c8	UDPv4	0.0.0.0:0	*	*	2640	iexplore.exe	2018-03-10 10:22:45 UTC+0000
0xdec8d80c8	UDPv6	:::0	*	*	2640	iexplore.exe	2018-03-10 10:22:45 UTC+0000
0xdec8d84b8	UDPv4	0.0.0.0:55653	*	*	1124	svchost.exe	2018-03-10 10:22:04 UTC+0000
0xdef30138	UDPv4	0.0.0.0:0	*	*	280	svchost.exe	2018-03-10 10:16:35 UTC+0000
0xdef30138	UDPv6	:::0	*	*	280	svchost.exe	2018-03-10 10:16:35 UTC+0000
0xdef3c298	UDPv4	0.0.0.0:0	*	*	280	svchost.exe	2018-03-10 10:16:35 UTC+0000
0xdef9cd90	UDPv4	0.0.0.0:50497	*	*	1124	svchost.exe	2018-03-10 10:26:33 UTC+0000
0xdefac6f0	UDPv4	192.168.1.89:137	*	*	4	System	2018-03-10 10:19:19 UTC+0000
0xdefb0d50	UDPv4	192.168.1.89:138	*	*	4	System	2018-03-10 10:19:19 UTC+0000
0xdef4310	TCPv4	0.0.0.0:49156	0.0.0.0:0	LISTENING	480	lsass.exe	
0xdec276f0	TCPv4	0.0.0.0:49154	0.0.0.0:0	LISTENING	928	svchost.exe	
0xdec276f0	TCPv6	:::49154	:::0	LISTENING	928	svchost.exe	
0xdec28290	TCPv4	0.0.0.0:49154	0.0.0.0:0	LISTENING	928	svchost.exe	
0xdec71410	TCPv4	0.0.0.0:22	0.0.0.0:0	LISTENING	1808	sshd.exe	
0xdec71410	TCPv6	:::22	:::0	LISTENING	1808	sshd.exe	
0xdef138f0	TCPv4	0.0.0.0:49156	0.0.0.0:0	LISTENING	480	lsass.exe	
0xdef138f0	TCPv6	:::49156	:::0	LISTENING	480	lsass.exe	
0xdef56248	TCPv4	0.0.0.0:445	0.0.0.0:0	LISTENING	4	System	
0xdef56248	TCPv6	:::445	:::0	LISTENING	4	System	
0xdef7e3b0	TCPv4	0.0.0.0:49155	0.0.0.0:0	LISTENING	464	services.exe	
0xdef7e3b0	TCPv6	:::49155	:::0	LISTENING	464	services.exe	
0xdef77f58	TCPv4	0.0.0.0:49155	0.0.0.0:0	LISTENING	464	services.exe	
0xdefb9c70	TCPv4	192.168.1.89:139	0.0.0.0:0	LISTENING	4	System	
0xdef1709f0	TCPv4	0.0.0.0:135	0.0.0.0:0	LISTENING	692	svchost.exe	
0xdef172910	TCPv4	0.0.0.0:135	0.0.0.0:0	LISTENING	692	svchost.exe	
0xdef172910	TCPv6	:::135	:::0	LISTENING	692	svchost.exe	
0xdef17228	TCPv4	0.0.0.0:49152	0.0.0.0:0	LISTENING	368	wininit.exe	
0xdef17228	TCPv6	:::49152	:::0	LISTENING	368	wininit.exe	
0xdef1777c0	TCPv4	0.0.0.0:49152	0.0.0.0:0	LISTENING	368	wininit.exe	
0xdef19f8d8	TCPv4	0.0.0.0:22	0.0.0.0:0	LISTENING	1808	sshd.exe	
0xdef1a34f8	TCPv4	0.0.0.0:49153	0.0.0.0:0	LISTENING	748	svchost.exe	
0xdef1a34f8	TCPv6	:::49153	:::0	LISTENING	748	svchost.exe	
0xdef1a5f58	TCPv4	0.0.0.0:49153	0.0.0.0:0	LISTENING	748	svchost.exe	
0xdec373a0	TCPv4	192.168.1.89:49424	192.168.1.39:4444	ESTABLISHED	-1		
0xdef9c8f8	UDPv4	0.0.0.0:60478	*	*	1124	svchost.exe	2018-03-10 10:23:14 UTC+0000

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:

```
Командная строка
Process: svchost.exe Pid: 3312 Address: 0x600000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 49, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x00600000 4d 5a e8 00 00 00 00 5b 52 45 55 89 e5 81 c3 64 MZ.....[REU....d
0x00600010 13 00 00 ff d3 81 c3 95 a4 02 00 89 3b 53 6a 04 .....;Sj.
0x00600020 50 ff d0 00 00 00 00 00 00 00 00 00 00 00 00 00 P.....
0x00600030 00 00 00 00 00 00 00 00 00 00 00 00 f8 00 00 00 .....

0x00600000 4d          DEC EBP
0x00600001 5a          POP EDX
0x00600002 e800000000 CALL 0x600007
0x00600007 5b          POP EBX
0x00600008 52          PUSH EDX
0x00600009 45          INC EBP
0x0060000a 55          PUSH EBP
0x0060000b 89e5        MOV EBP, ESP
0x0060000d 81c364130000 ADD EBX, 0x1364
0x00600013 ffd3        CALL EBX
0x00600015 81c395a40200 ADD EBX, 0x2a495
0x0060001b 893b        MOV [EBX], EDI
0x0060001d 53          PUSH EBX
0x0060001e 6a04        PUSH 0x4
0x00600020 50          PUSH EAX
0x00600021 ffd0        CALL EAX
0x00600023 0000        ADD [EAX], AL
0x00600025 0000        ADD [EAX], AL
0x00600027 0000        ADD [EAX], AL
0x00600029 0000        ADD [EAX], AL
0x0060002b 0000        ADD [EAX], AL
0x0060002d 0000        ADD [EAX], AL
0x0060002f 0000        ADD [EAX], AL
0x00600031 0000        ADD [EAX], AL
0x00600033 0000        ADD [EAX], AL
0x00600035 0000        ADD [EAX], AL
0x00600037 0000        ADD [EAX], AL
0x00600039 0000        ADD [EAX], AL
0x0060003b 00f8        ADD AL, BH
0x0060003d 0000        ADD [EAX], AL
0x0060003f 00          DB 0x0
```

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 enthushional (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](#)

Forensic Focus – Articles