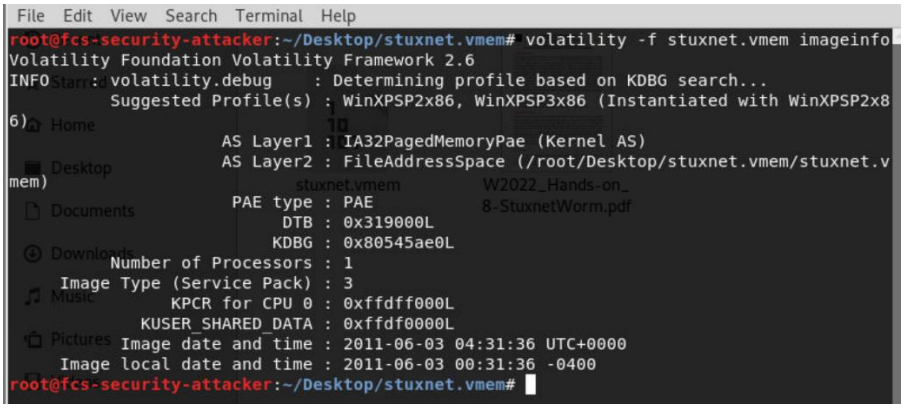
ASSIGNMENT 7

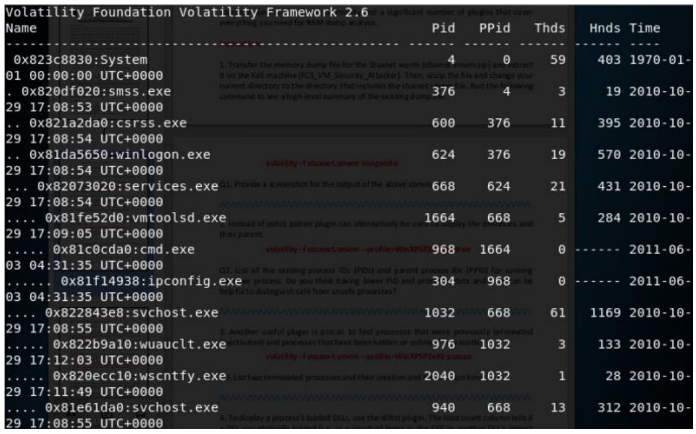
Network Security

Soheil Shirvani 3720505

1) $volatility\ -f\ stuxnet.vmem\ imageinfo$, Provide a screenshot for the output of the above command.



2) $volatility\ -f\ stuxnet.vmem\ --profile = WinXPSP2x86\ pstree$, List all the existing process IDs (PIDs) and parent process IDs (PPID) for running lsass.exe process. Do you think having lower PID and process's data and time can be helpful to distinguish safe from unsafe processes?



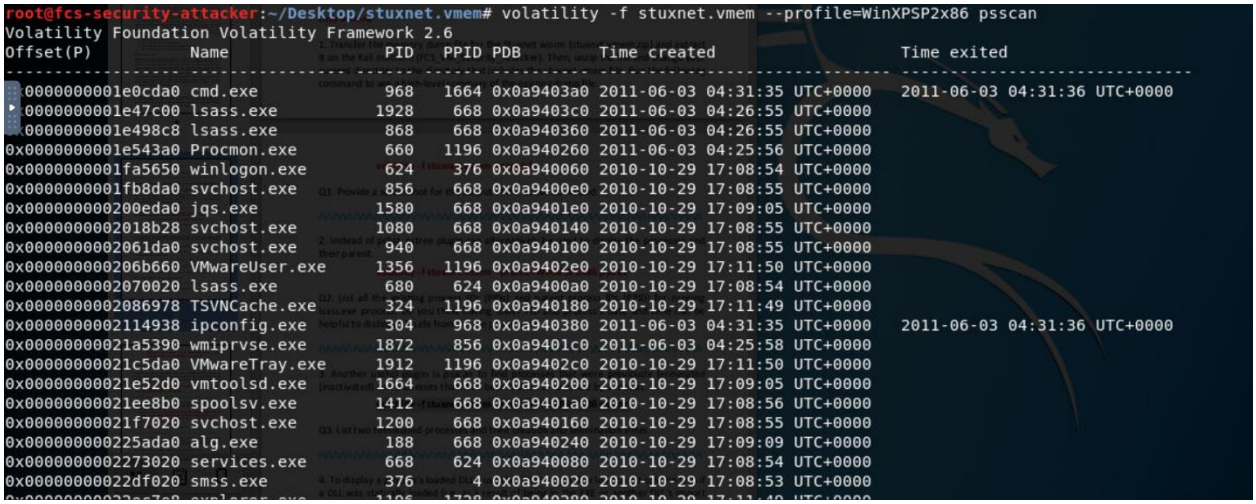
lsass.exe process:

Offset(P)	Name	PID	PPid
0x0000000001e47c00	lsass.exe	1928	668
0x0000000001e498c8	lsass.exe	868	668
0x0000000002070020	lsass.exe	680	624

There are 3 lsass.exe in the process list but there should be one lsass instance in the process list. For the first two rows the creation time is the same and for the

last one is around one year earlier which is suspicious. Also, the PID for all these three processes are different which shows this file is a malicious file.

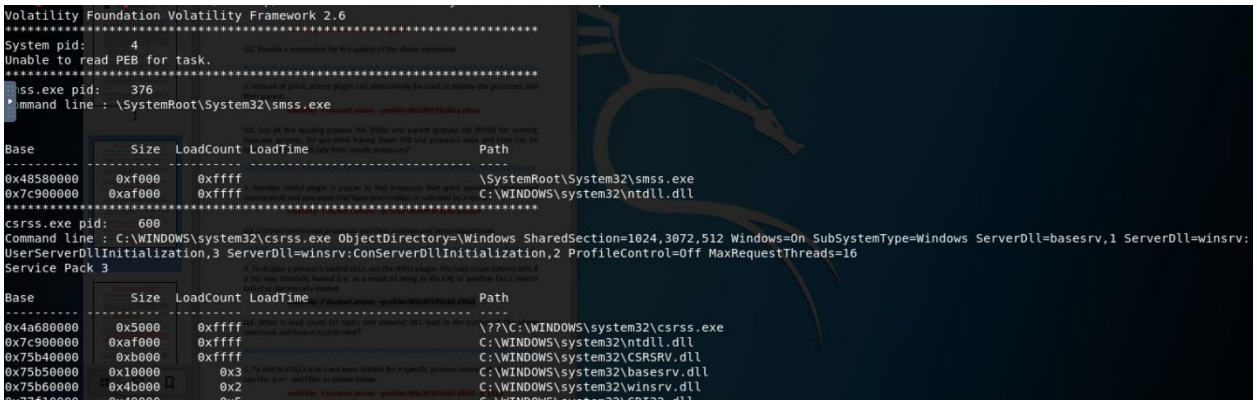
3) $volatility\ -f\ stuxnet.vmem\ --profile=WinXPSP2x86\ psscan$, *List two terminated processes and their creation and termination times*

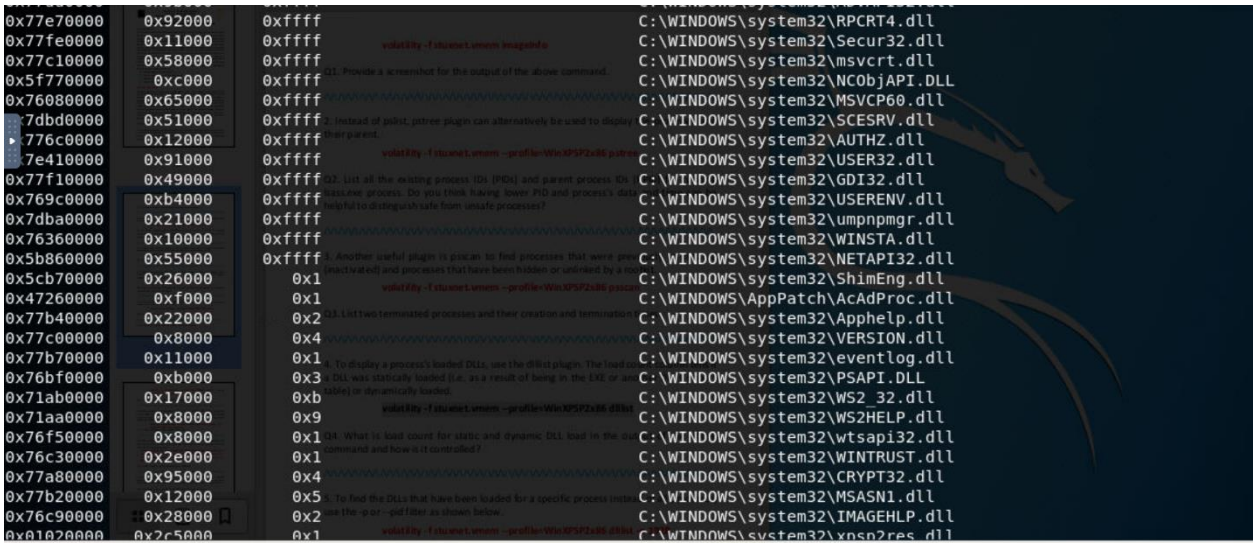


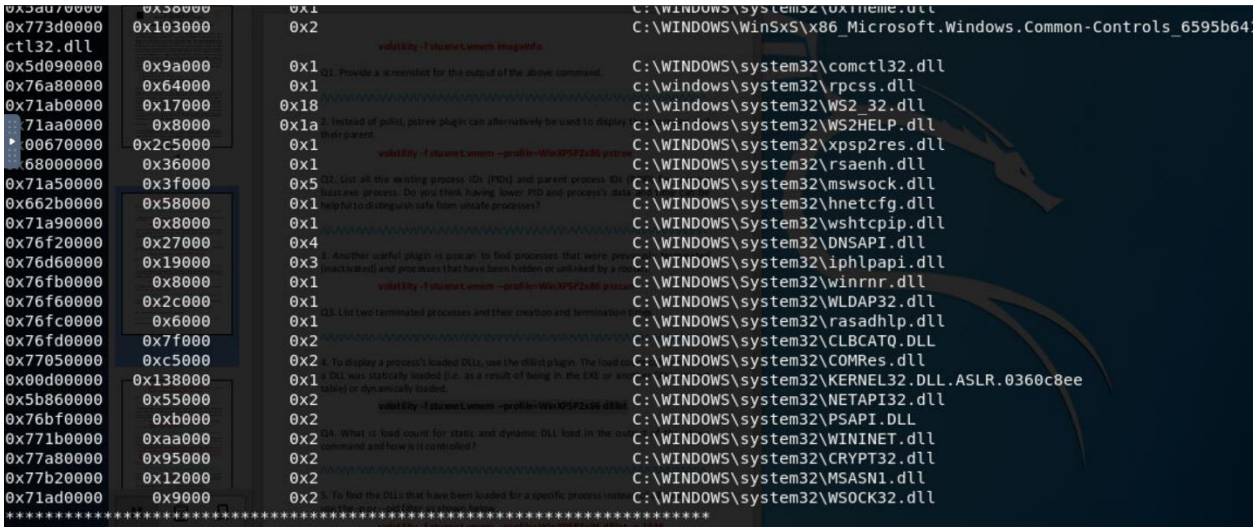
```
root@fcs-security-attacker:~/Desktop/stuxnet.vmem# volatility -f stuxnet.vmem --profile=WinXPSP2x86 psscan
Volatility Foundation Volatility Framework 2.6
Offset(P)          Name                PID   PPID PDB        Time created                   Time exited
------------------ ---------------- ------ ------ ---------- ------------------------------ ------------------------------
0x0000000001e0cda0 cmd.exe             968   1664 0x0a9403a0 2011-06-03 04:31:35 UTC+0000   2011-06-03 04:31:36 UTC+0000
0x0000000001e47c00 lsass.exe          1928    668 0x0a9403c0 2011-06-03 04:26:55 UTC+0000
0x0000000001e498c8 lsass.exe           868    668 0x0a940360 2011-06-03 04:26:55 UTC+0000
0x0000000001e543a0 Procmon.exe         660   1196 0x0a940260 2011-06-03 04:25:56 UTC+0000
0x0000000001fa5650 winlogon.exe        624    376 0x0a940060 2010-10-29 17:08:54 UTC+0000
0x0000000001fb8da0 svchost.exe         856    668 0x0a9400e0 2010-10-29 17:08:55 UTC+0000
0x000000000200eda0 jqs.exe            1580    668 0x0a9401e0 2010-10-29 17:09:05 UTC+0000
0x0000000002018b28 svchost.exe        1080    668 0x0a940140 2010-10-29 17:08:55 UTC+0000
0x0000000002061da0 svchost.exe         940    668 0x0a940100 2010-10-29 17:08:55 UTC+0000
0x000000000206b660 VMwareUser.exe     1356   1196 0x0a9402e0 2010-10-29 17:11:50 UTC+0000
0x0000000002070020 lsass.exe           680    624 0x0a9400a0 2010-10-29 17:08:54 UTC+0000
0x0000000002086978 TSVNCache.exe       324   1196 0x0a940180 2010-10-29 17:11:49 UTC+0000
0x0000000002114938 ipconfig.exe        304    968 0x0a940380 2011-06-03 04:31:35 UTC+0000   2011-06-03 04:31:36 UTC+0000
0x00000000021a5390 wmiprvse.exe       1872    856 0x0a9401c0 2011-06-03 04:25:58 UTC+0000
0x00000000021c5da0 VMwareTray.exe     1912   1196 0x0a9402c0 2010-10-29 17:11:50 UTC+0000
0x00000000021e52d0 vmtoolsd.exe       1664    668 0x0a940200 2010-10-29 17:09:05 UTC+0000
0x00000000021ee8b0 spoolsv.exe        1412    668 0x0a9401a0 2010-10-29 17:08:56 UTC+0000
0x00000000021f7020 svchost.exe        1200    668 0x0a940160 2010-10-29 17:08:55 UTC+0000
0x000000000225ada0 alg.exe             188    668 0x0a940240 2010-10-29 17:09:09 UTC+0000
0x0000000002273020 services.exe        668    624 0x0a940080 2010-10-29 17:08:54 UTC+0000
0x00000000022df020 smss.exe            376      4 0x0a940020 2010-10-29 17:08:53 UTC+0000
```

Offset(P)	Name	PID	PPID	PBD	Time\C	Time\E
0x0000000001e0cda0	cmd.exe	968	1664	0x0a9403a0	**2011-06-03 04:31:35 UTC+0000**	**2011-06-03 04:31:36 UTC+0000**
0x0000000002114938	ipconfig.exe	304	968	0x0a940380	**2011-06-03 04:31:35 UTC+0000**	**2011-06-03 04:31:36 UTC+0000**

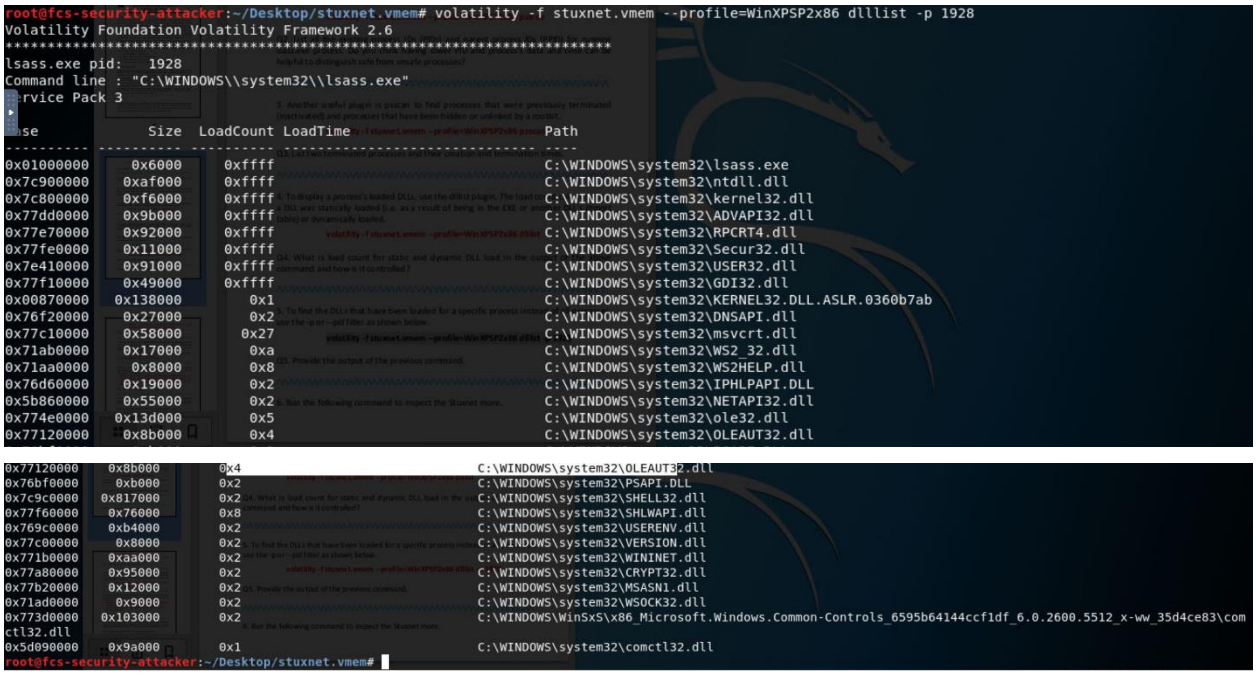
4) $volatility\ -f\ stuxnet.vmem\ --profile = WinXPSP2x86\ dlllist$, What is load count for static and dynamic DLL load in the output of the above command and how is it controlled?







5) $volatility\ -f\ stuxnet.vmem\ --profile=WinXPSP2x86\ dlllist\ -p\ 1928$, *Provide the output of the previous command.*

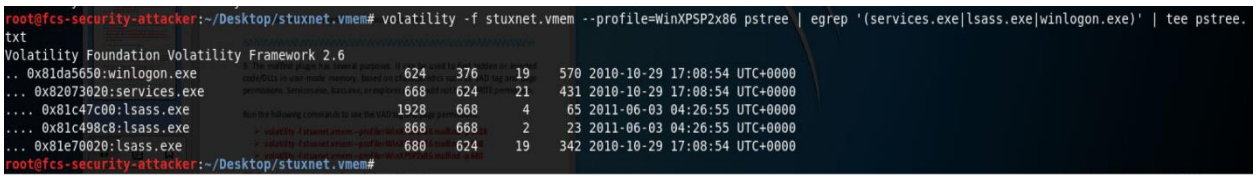


```
root@fcs-security-attacker:~/Desktop/stuxnet.vmem# volatility -f stuxnet.vmem --profile=WinXPSP2x86 dlllist -p 1928
Volatility Foundation Volatility Framework 2.6
************************************************************************
lsass.exe pid:   1928
Command line : "C:\WINDOWS\\system32\\lsass.exe"
ervice Pack 3

ase             Size  LoadCount LoadTime                       Path
---------- ---------- ---------- ------------------------------ ----
0x01000000    0x6000     0xffff                                C:\WINDOWS\system32\lsass.exe
0x7c900000   0xaf000     0xffff                                C:\WINDOWS\system32\ntdll.dll
0x7c800000   0xf6000     0xffff                                C:\WINDOWS\system32\kernel32.dll
0x77dd0000   0x9b000     0xffff                                C:\WINDOWS\system32\ADVAPI32.dll
0x77e70000   0x92000     0xffff                                C:\WINDOWS\system32\RPCRT4.dll
0x77fe0000   0x11000     0xffff                                C:\WINDOWS\system32\Secur32.dll
0x7e410000   0x91000     0xffff                                C:\WINDOWS\system32\USER32.dll
0x77f10000   0x49000     0xffff                                C:\WINDOWS\system32\GDI32.dll
0x00870000  0x138000        0x1                                C:\WINDOWS\system32\KERNEL32.DLL.ASLR.0360b7ab
0x76f20000   0x27000        0x2                                C:\WINDOWS\system32\DNSAPI.dll
0x77c10000   0x58000       0x27                                C:\WINDOWS\system32\msvcrt.dll
0x71ab0000   0x17000        0xa                                C:\WINDOWS\system32\WS2_32.dll
0x71aa0000    0x8000        0x8                                C:\WINDOWS\system32\WS2HELP.dll
0x76d60000   0x19000        0x2                                C:\WINDOWS\system32\IPHLPAPI.DLL
0x5b860000   0x55000        0x2                                C:\WINDOWS\system32\NETAPI32.dll
0x774e0000  0x13d000        0x5                                C:\WINDOWS\system32\ole32.dll
0x77120000   0x8b000        0x4                                C:\WINDOWS\system32\OLEAUT32.dll
```



```
0x77120000   0x8b000        0x4                                C:\WINDOWS\system32\OLEAUT32.dll
0x76bf0000    0xb000        0x2                                C:\WINDOWS\system32\PSAPI.DLL
0x7c9c0000  0x817000        0x2                                C:\WINDOWS\system32\SHELL32.dll
0x77f60000   0x76000        0x8                                C:\WINDOWS\system32\SHLWAPI.dll
0x769c0000   0xb4000        0x2                                C:\WINDOWS\system32\USERENV.dll
0x77c00000    0x8000        0x2                                C:\WINDOWS\system32\VERSION.dll
0x771b0000   0xaa000        0x2                                C:\WINDOWS\system32\WININET.dll
0x77a80000   0x95000        0x2                                C:\WINDOWS\system32\CRYPT32.dll
0x77b20000   0x12000        0x2                                C:\WINDOWS\system32\MSASN1.dll
0x71ad0000    0x9000        0x2                                C:\WINDOWS\system32\WSOCK32.dll
0x773d0000  0x103000        0x2                                C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83\com
ctl32.dll
0x5d090000   0x9a000        0x1                                C:\WINDOWS\system32\comctl32.dll
root@fcs-security-attacker:~/Desktop/stuxnet.vmem#
```

6) $volatility\ -f\ stuxnet.vmem\ -profile=WinXPSP2x86\ pstree\ |\ egrep\ '(services.exe|lsass.exe|winlogon.exe)'\ |\ tee\ pstree.txt$, *Provide the output of the previous command*



```
root@fcs-security-attacker:~/Desktop/stuxnet.vmem# volatility -f stuxnet.vmem --profile=WinXPSP2x86 pstree | egrep '(services.exe|lsass.exe|winlogon.exe)' | tee pstree.
txt
Volatility Foundation Volatility Framework 2.6
.. 0x81da5650:winlogon.exe                624    376     19    570 2010-10-29 17:08:54 UTC+0000
... 0x82073020:services.exe               668    624     21    431 2010-10-29 17:08:54 UTC+0000
.... 0x81c47c00:lsass.exe                1928    668      4     65 2011-06-03 04:26:55 UTC+0000
.... 0x81c498c8:lsass.exe                 868    668      2     23 2011-06-03 04:26:55 UTC+0000
... 0x81e70020:lsass.exe                  680    624     19    342 2010-10-29 17:08:54 UTC+0000
root@fcs-security-attacker:~/Desktop/stuxnet.vmem#
```

7) $volatility\ -f\ stuxnet.vmem\ --profile=WinXPSP2x86\ sockets\ |\ egrep\ '(Off|---|680|1928|868)'\ |\ tee\ sockets.txt$, *Provide the output of the previous command.*

```
root@fcs-security-attacker:~/Desktop/stuxnet.vmem# volatility -f stuxnet.vmem --profile=WinXPSP2x86 sockets | egrep '(Off|---|680|1928|868)' | tee sockets.txt
Volatility Foundation Volatility Framework 2.6
Offset(V)      PID   Port  Proto Protocol        Address         Create Time
---------- -------- ------ ------ --------------- --------------- -----------
0x81dc2008      680    500     17 UDP             0.0.0.0         2010-10-29 17:09:05 UTC+0000
0x81da4d18      680      0    255 Reserved        0.0.0.0         2010-10-29 17:09:05 UTC+0000
0x82060008      680   4500     17 UDP             0.0.0.0         2010-10-29 17:09:05 UTC+0000
```

8) $volatility\ -f\ stuxnet.vmem\ --profile=WinXPSP2x86\ dlllist\ -p\ 1928\ 2>/dev/null\ |\ wc\ -l$

```
root@fcs-security-attacker:~/Desktop/stuxnet.vmem# volatility -f stuxnet.vmem --profile=WinXPSP2x86 dlllist -p 1928 2>/dev/null | wc -l
35
```

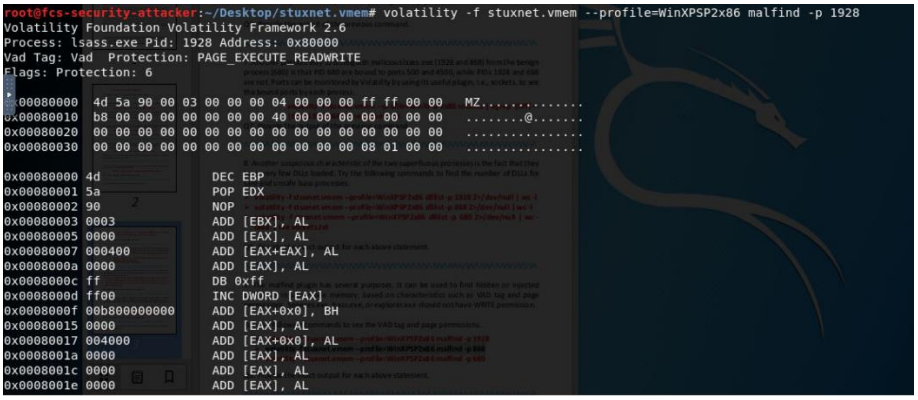
volatility −f stuxnet.vmem −−profile = WinXPSP2x86 dlllist −p 868 2 >/dev/null | wc –

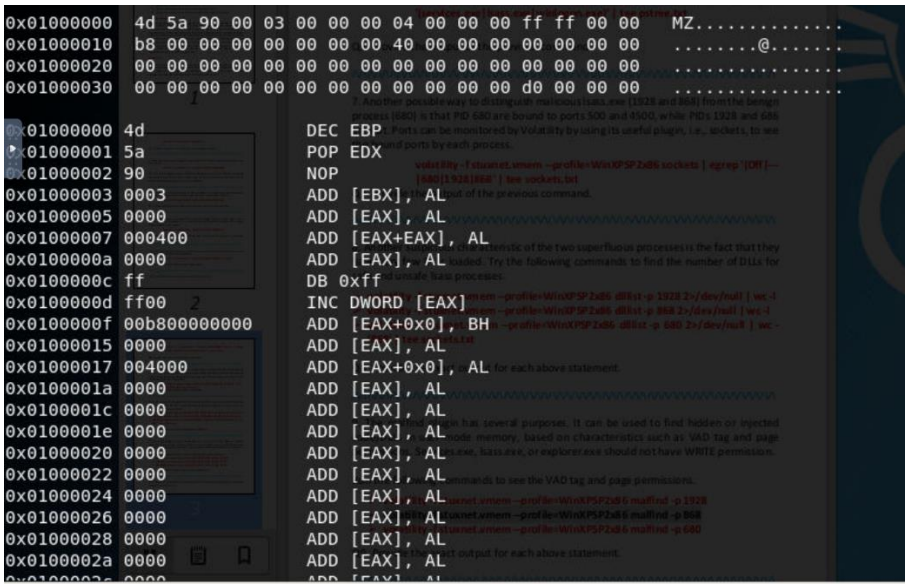
```
root@fcs-security-attacker:~/Desktop/stuxnet.vmem# volatility -f stuxnet.vmem --profile=WinXPSP2x86 dlllist -p 868 2>/dev/null | wc -l
15
```

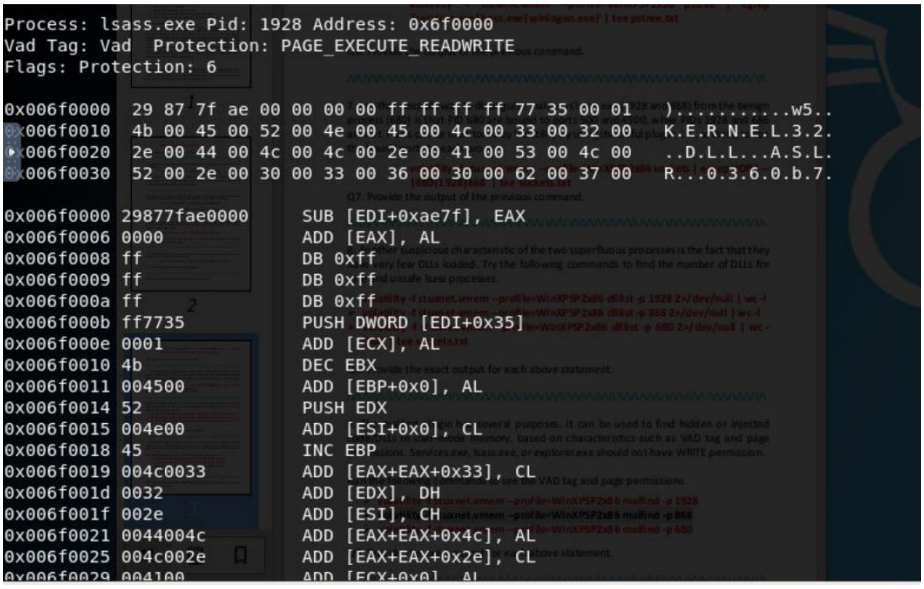
volatility −f stuxnet.vmem −−profile = WinXPSP2x86 dlllist −p 680 2 >/dev/null | wc −l

```
root@fcs-security-attacker:~/Desktop/stuxnet.vmem# volatility -f stuxnet.vmem --profile=WinXPSP2x86 dlllist -p 680 2>/dev/null | wc -l
64
```

9) $volatility\ -f\ stuxnet.vmem\ --profile = WinXPSP2x86\ malfind\ -p\ 1928$





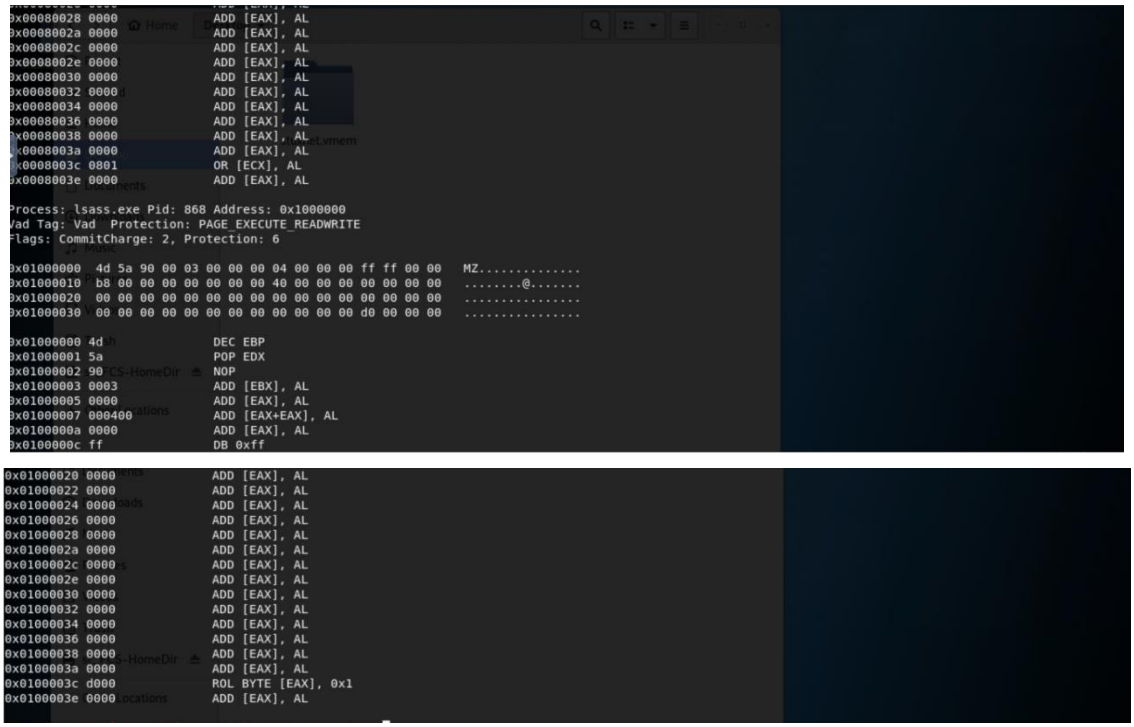


```
Process: lsass.exe Pid: 1928 Address: 0x680000
Vad Tag: Vad  Protection: PAGE_EXECUTE_READWRITE
Flags: Protection: 6

0x00680000  90 06 68 00 c6 07 68 00 24 00 68 00 a5 04 00 00   ..h...h.$.h.....
0x00680010  f2 04 68 00 48 06 00 00 c9 04 68 00 29 00 00 00   ..h.H.....h.)...
0x00680020  00 00 6f 00 e8 13 00 00 00 5a 77 4d 61 70 56 69   ..o......ZwMapVi
0x00680030  65 77 4f 66 53 65 63 74 69 6f 6e 00 5a 51 81 c1   ewOfSection.ZQ..

0x00680000 90              NOP
0x00680001 06              PUSH ES
0x00680002 6800c60768      PUSH DWORD 0x6807c600
0x00680007 002400          ADD [EAX+EAX], AH
0x0068000a 6800a50400      PUSH DWORD 0x4a500
0x0068000f 00f2            ADD DL, DH
0x00680011 0468            ADD AL, 0x68
0x00680013 004806          ADD [EAX+0x6], CL
0x00680016 0000            ADD [EAX], AL
0x00680018 c9              LEAVE
0x00680019 0468            ADD AL, 0x68
0x0068001b 0029            ADD [ECX], CH
0x0068001d 0000            ADD [EAX], AL
0x0068001f 0000            ADD [EAX], AL
0x00680021 006f00          ADD [EDI+0x0], CH
0x00680024 e813000000      CALL 0x68003c
0x00680029 5a              POP EDX
0x0068002a 774d            JA 0x680079
```

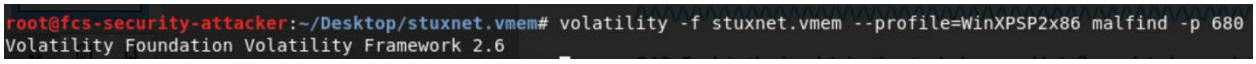
```
Process: lsass.exe Pid: 868 Address: 0x1000000
Vad Tag: Vad  Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 2, Protection: 6

0x01000000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x01000010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x01000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x01000030  00 00 00 00 00 00 00 00 00 00 00 00 d0 00 00 00   ................

0x01000000 4d              DEC EBP
0x01000001 5a              POP EDX
0x01000002 90              NOP
0x01000003 0003            ADD [EBX], AL
0x01000005 0000            ADD [EAX], AL
0x01000007 000400          ADD [EAX+EAX], AL
0x0100000a 0000            ADD [EAX], AL
0x0100000c ff              DB 0xff
0x0100000d ff00            INC DWORD [EAX]
0x0100000f 00b800000000    ADD [EAX+0x0], BH
0x01000015 0000            ADD [EAX], AL
0x01000017 004000          ADD [EAX+0x0], AL
0x0100001a 0000            ADD [EAX], AL
0x0100001c 0000            ADD [EAX], AL
0x0100001e 0000            ADD [EAX], AL
0x01000020 0000            ADD [EAX], AL
0x01000022 0000            ADD [EAX], AL
```

volatility − f stuxnet. vmem − −profile = WinXPSP2x86 malfind − p 868

```
root@fcs-security-attacker:~/Desktop/stuxnet.vmem# volatility -f stuxnet.vmem --profile=WinXPSP2x86 malfind -p 868
Volatility Foundation Volatility Framework 2.6
Process: lsass.exe Pid: 868 Address: 0x80000
Vad Tag: Vad  Protection: PAGE_EXECUTE_READWRITE
Flags: Protection: 6

0x00080000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x00080010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x00080020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00080030  00 00 00 00 00 00 00 00 00 00 00 00 08 01 00 00   ................

0x00080000 4d              DEC EBP
0x00080001 5a              POP EDX
0x00080002 90              NOP
0x00080003 0003            ADD [EBX], AL
0x00080005 0000            ADD [EAX], AL
0x00080007 000400          ADD [EAX+EAX], AL
0x0008000a 0000            ADD [EAX], AL
0x0008000c ff              DB 0xff
0x0008000d ff00            INC DWORD [EAX]
0x0008000f 00b800000000    ADD [EAX+0x0], BH
0x00080015 0000            ADD [EAX], AL
0x00080017 004000          ADD [EAX+0x0], AL
0x0008001a 0000            ADD [EAX], AL
0x0008001c 0000            ADD [EAX], AL
0x0008001e 0000            ADD [EAX], AL
0x00080020 0000            ADD [EAX], AL
0x00080022 0000            ADD [EAX], AL
0x00080024 0000            ADD [EAX], AL
0x00080026 0000            ADD [EAX], AL
```



volatility − f stuxnet. vmem − −profile = WinXPSP2x86 malfind − p 680



10) Explain the hook injection technique and briefly explain how volatility can be applied to detect malicious hooks.

Hook injection describes a way to load malware that takes advantage of Windows hooks, which are used to intercept messages destined for applications. Malware authors can use hook injection to accomplish two things: To be sure that malicious code will run whenever a particular message is intercepted, to be sure that a particular DLL will be loaded in a victim process's memory space. users generate events that are sent to the OS, which then sends messages created by those events to threads registered to receive them. The right side of the figure shows one way that an attacker can insert a malicious DLL to intercept messages.

The plugin dlllist in the Volatility Framework can also be used to list all DLLs for a given process in memory and find DLLs injected with the CreateRemoteThread and LoadLibrary technique. This technique does not hide the DLL and therefore will not be detected by the plugin malfind. The first command well use is the malfind command. This command is used to find injected code inside the processes memory. It does this by looking for sections of allocated memory (by looking at the VAD tree data structure) and checking if they have hints of executable code that are not mapped to any file on the disk