What Malware Authors Don't Want you to Know - Evasive Hollow Process Injection

Monnappa K A

Who AM I

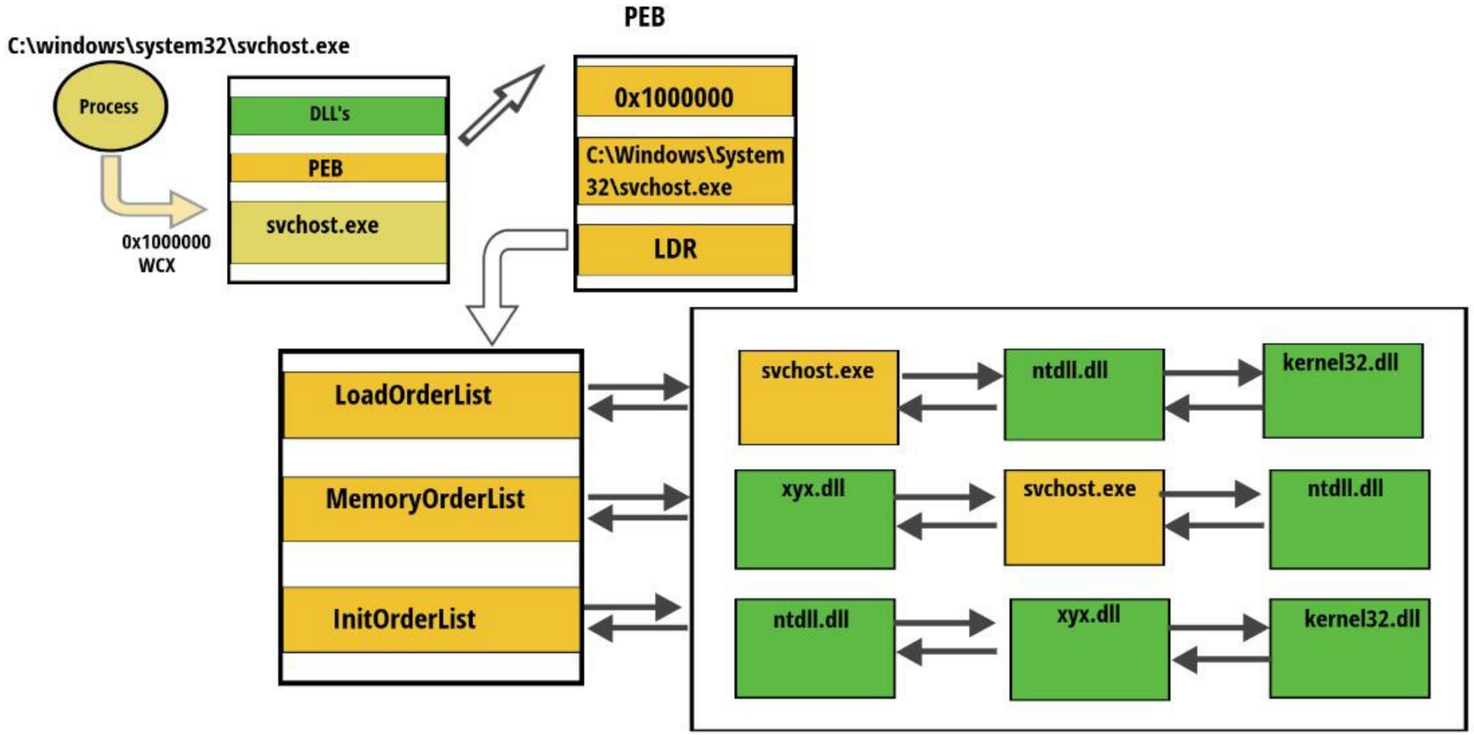
Monnappa K A

- *Info Security Investigator - Cisco CSIRT*
- *Co-founder Cysinfo Security Community*
- *Author of Limon Sandbox*
- *Winner of Volatility Plugin Contest 2016*
- *Conferences - Black Hat, FIRST, 4SICS*
- *Articles - eForensics, Hakin9, Hack Insight*



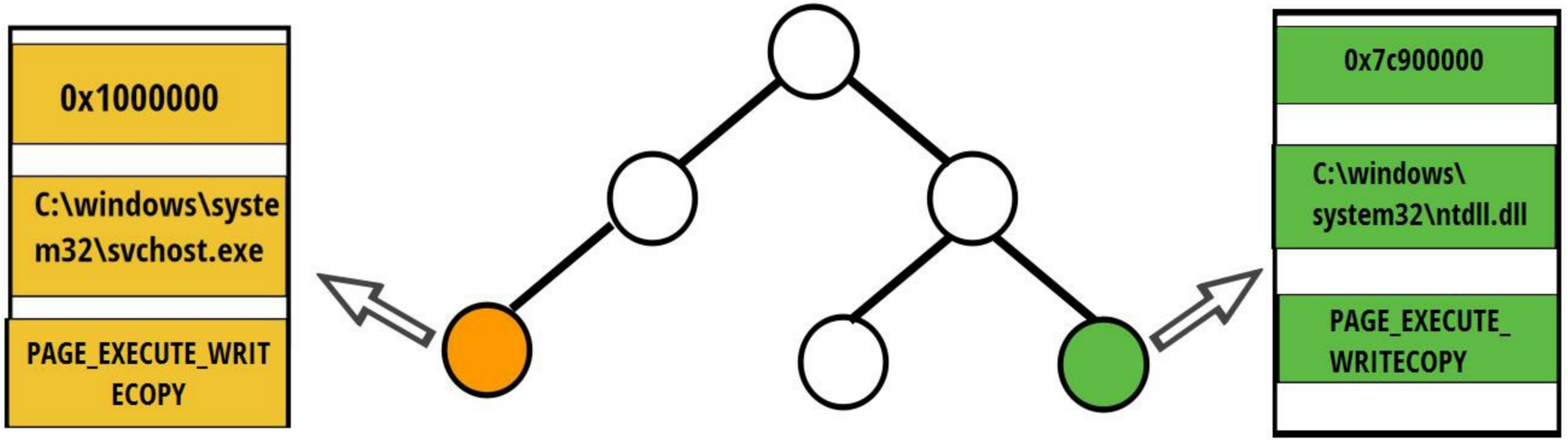
Process Memory Internals

Process Memory

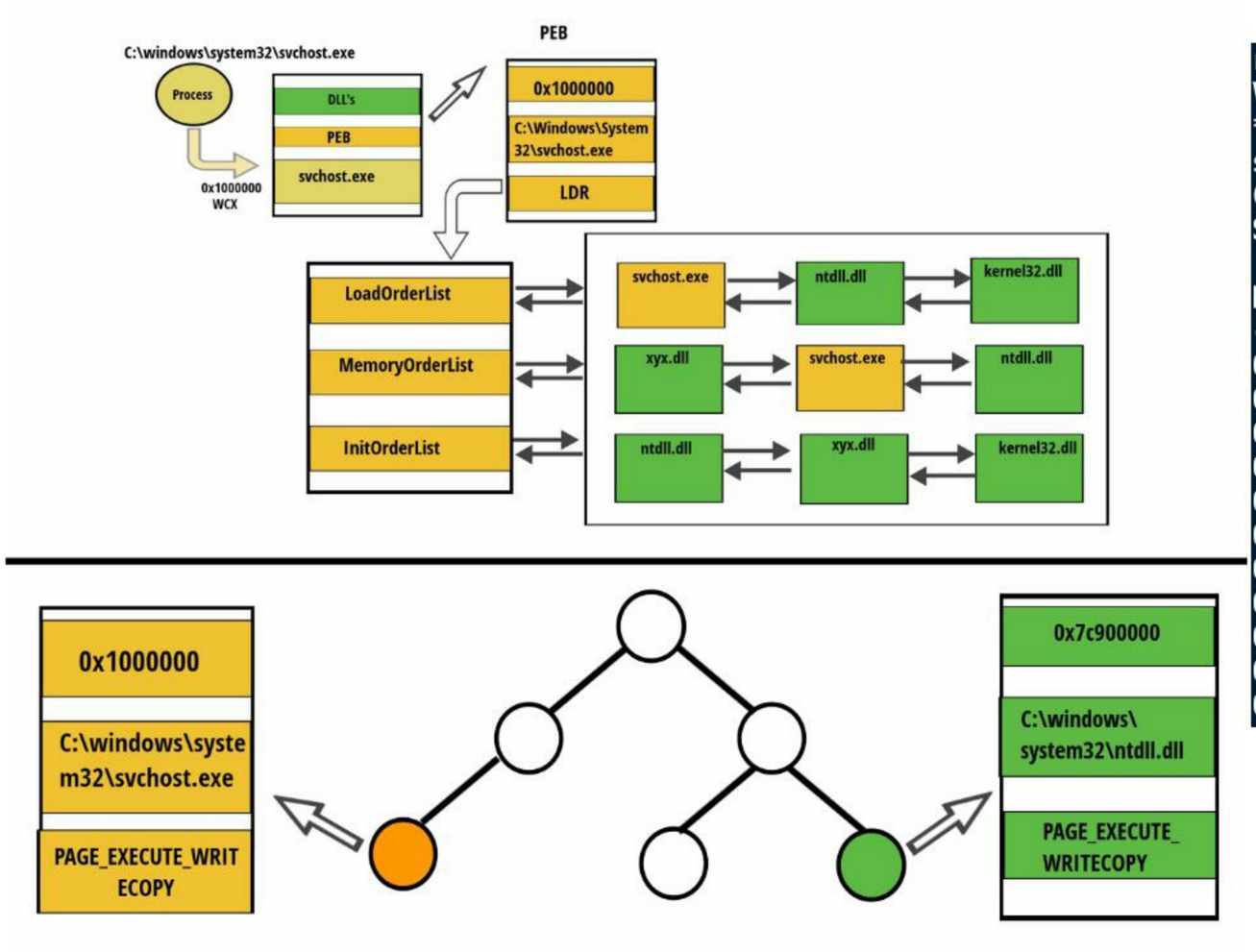


User Memory

Kernel Memory

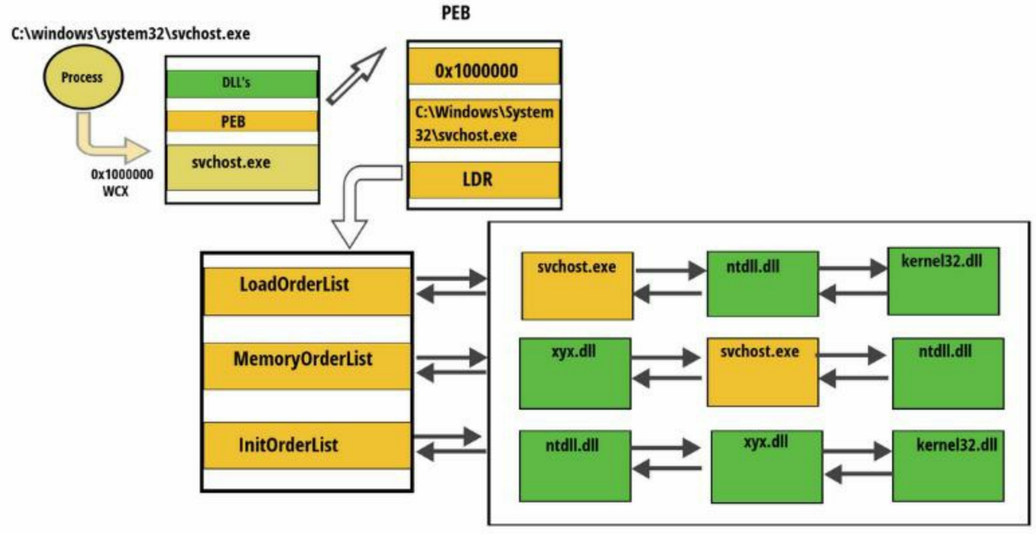


Dlllist Volatility plugin displays loaded modules by walking the InLoadOrderModuleList

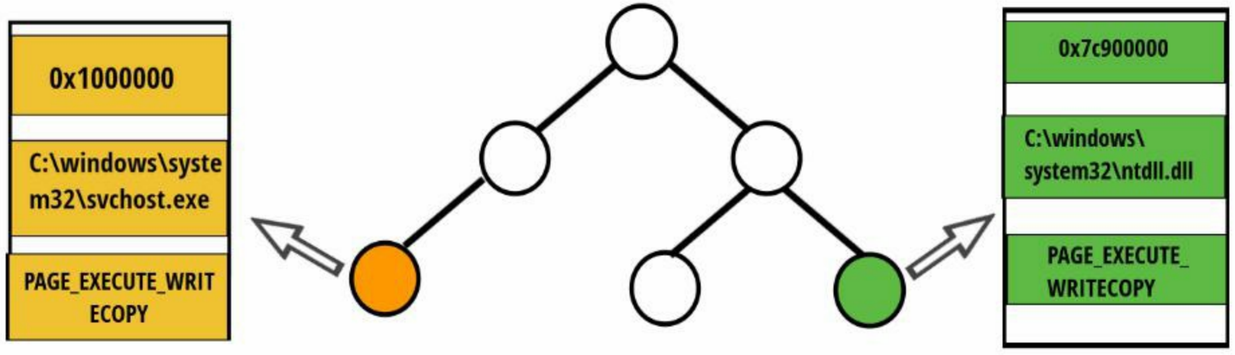



```
root@kratos:~/Volatility# python vol.py -f mem_image.vmem dlllist -p 884
Volatility Foundation Volatility Framework 2.5
************************************************************************
svchost.exe pid:    884
Command line : C:\WINDOWS\system32\svchost -k DcomLaunch
Service Pack 3

Base             Size  LoadCount Path
---------- ---------- ---------- ----
0x01000000     0x6000     0xffff C:\WINDOWS\system32\svchost.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
0x5cb70000    0x26000        0x1 C:\WINDOWS\system32\ShimEng.dll
0x6f880000   0x1ca000        0x1 C:\WINDOWS\AppPatch\AcGenral.DLL
0x7e410000    0x91000       0x39 C:\WINDOWS\system32\USER32.dll
0x77f10000    0x49000       0x28 C:\WINDOWS\system32\GDI32.dll
0x76b40000    0x2d000        0x2 C:\WINDOWS\system32\WINMM.dll
```

ldrmodules volatility plugin compares the PEB list with data in the VAD




```
root@kratos:~/Volatility# python vol.py -f mem_image.vmem ldrmodules -p 884
Volatility Foundation Volatility Framework 2.5
Pid      Process              Base       InLoad InInit InMem MappedPath
-------- -------------------- ---------- ------ ------ ----- ----------
     884 svchost.exe          0x01000000 True   False  True  \WINDOWS\system32\svchost.exe
     884 svchost.exe          0x7c900000 True   True   True  \WINDOWS\system32\ntdll.dll
     884 svchost.exe          0x74f70000 True   True   True  \WINDOWS\system32\icaapi.dll
     884 svchost.exe          0x76f60000 True   True   True  \WINDOWS\system32\wldap32.dll
     884 svchost.exe          0x77c00000 True   True   True  \WINDOWS\system32\version.dll
     884 svchost.exe          0x5ad70000 True   True   True  \WINDOWS\system32\uxtheme.dll
     884 svchost.exe          0x76c90000 True   True   True  \WINDOWS\system32\imagehlp.dll
     884 svchost.exe          0x76bc0000 True   True   True  \WINDOWS\system32\regapi.dll
     884 svchost.exe          0x77dd0000 True   True   True  \WINDOWS\system32\advapi32.dll
     884 svchost.exe          0x77a80000 True   True   True  \WINDOWS\system32\crypt32.dll
     884 svchost.exe          0x77be0000 True   True   True  \WINDOWS\system32\msacm32.dll
     884 svchost.exe          0x68000000 True   True   True  \WINDOWS\system32\rsaenh.dll
     884 svchost.exe          0x76e10000 True   True   True  \WINDOWS\system32\adsldpc.dll
```





Introduction to Hollow Process Injection

Code Injection

Injects code into the legitimate process

Forces legitimate process to execute malicious code

Performs malicious actions within the context of legitimate process

Hollow Process Injection

Code Injection technique which replaces the executable section of the runnning process with malicious executable

Malware Creates a legitimate process in the suspended state

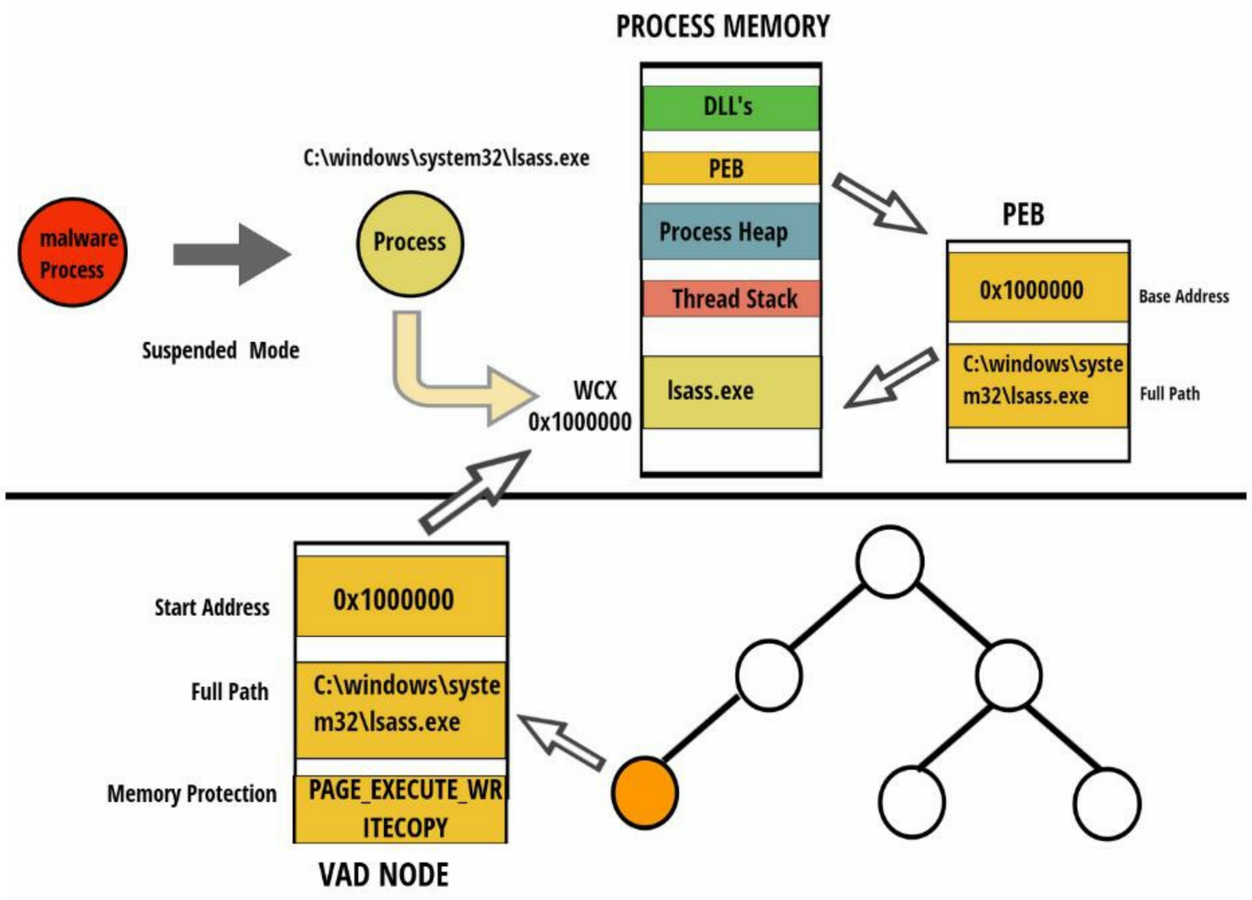
Process executable section is freed, reallocated and copied with malicious executable

Suspended thread's start address is pointed to the malicious exectuable's address of entry point and thread is resumed

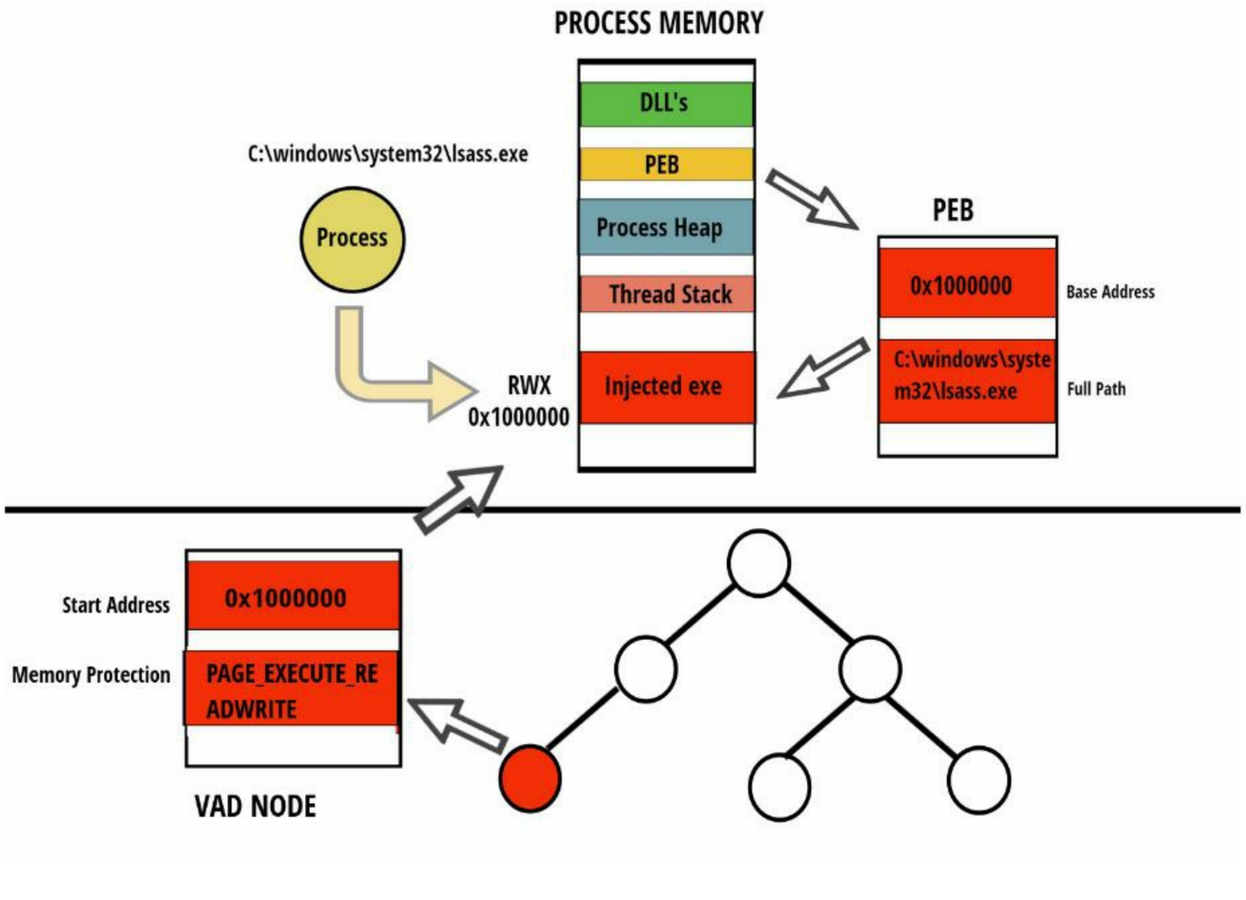
Used to disguise malware process as legitimate process



Hollow Process Injection - Stuxnet



BEFORE HOLLOWING



AFTER HOLLOWING



Detecting Hollow Process Injection Using Memory Forensics - Stuxnet

lsass.exe process (pid 868 and pid 1928) was not started by winlogon.exe or wininit.exe but these processes were started by services.exe (pid 668)

```
root@kratos:~/Volatility# python vol.py -f stuxnet.vmem pslist | grep -i lsass
Volatility Foundation Volatility Framework 2.5
0x81e70020 lsass.exe             680    624     19      342      0      0 2010-10-29
 17:08:54 UTC+0000
0x81c498c8 lsass.exe             868    668      2       23      0      0 2011-06-03
 04:26:55 UTC+0000
0x81c47c00 lsass.exe            1928    668      4       65      0      0 2011-06-03
 04:26:55 UTC+0000
root@kratos:~/Volatility# python vol.py -f stuxnet.vmem pslist -p 668
Volatility Foundation Volatility Framework 2.5
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start
                    Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ----------
------------------ ------------------------------
0x82073020 services.exe <-       668    624     21      431      0      0 2010-10-29
 17:08:54 UTC+0000
```

```
root@kratos:~/Volatility# python vol.py -f stuxnet.vmem pslist -p 624
Volatility Foundation Volatility Framework 2.5
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start
                    Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ----------
------------------ ------------------------------
0x81da5650 winlogon.exe <-       624    376     19      570      0      0 2010-10-29
 17:08:54 UTC+0000
```

Comparing Dlllist and Ldrmodules plugin shows the discrepancy

```
root@kratos:~/Volatility# python vol.py -f stuxnet.vmem dlllist -p 868
Volatility Foundation Volatility Framework 2.5
************************************************************************
lsass.exe pid:    868
Command line : "C:\WINDOWS\\system32\\lsass.exe"
Service Pack 3

Base             Size  LoadCount 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
```

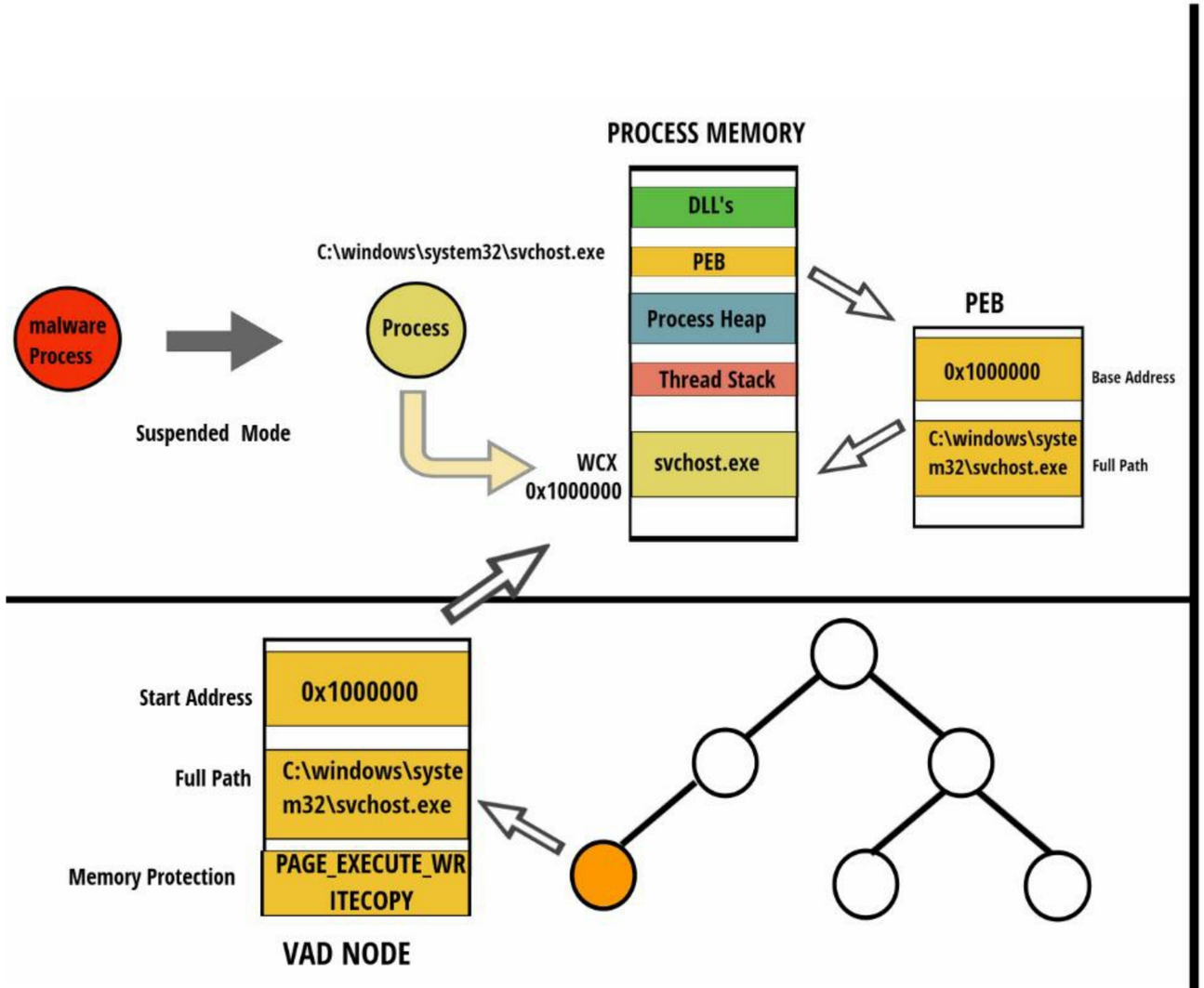
```
root@kratos:~/Volatility# python vol.py -f stuxnet.vmem ldrmodules -p 868
Volatility Foundation Volatility Framework 2.5
Pid      Process              Base       InLoad InInit InMem MappedPath
-------- -------------------- ---------- ------ ------ ----- ----------
     868 lsass.exe            0x00080000 False  False  False
     868 lsass.exe            0x7c900000 True   True   True  \WINDOWS\system32\ntdll.d
ll
     868 lsass.exe            0x77e70000 True   True   True  \WINDOWS\system32\rpcrt4.
dll
     868 lsass.exe            0x7c800000 True   True   True  \WINDOWS\system32\kernel3
2.dll
     868 lsass.exe            0x77fe0000 True   True   True  \WINDOWS\system32\secur32
.dll
     868 lsass.exe            0x7e410000 True   True   True  \WINDOWS\system32\user32.
dll
     868 lsass.exe            0x01000000 True   False  True
     868 lsass.exe            0x77f10000 True   True   True  \WINDOWS\system32\gdi32.d
ll
     868 lsass.exe            0x77dd0000 True   True   True  \WINDOWS\system32\advapi3
2.dll
```

Malfind Volatility plugin shows suspicious memory protection (PAGE_EXECUTE_READWRITE) at address 0x1000000

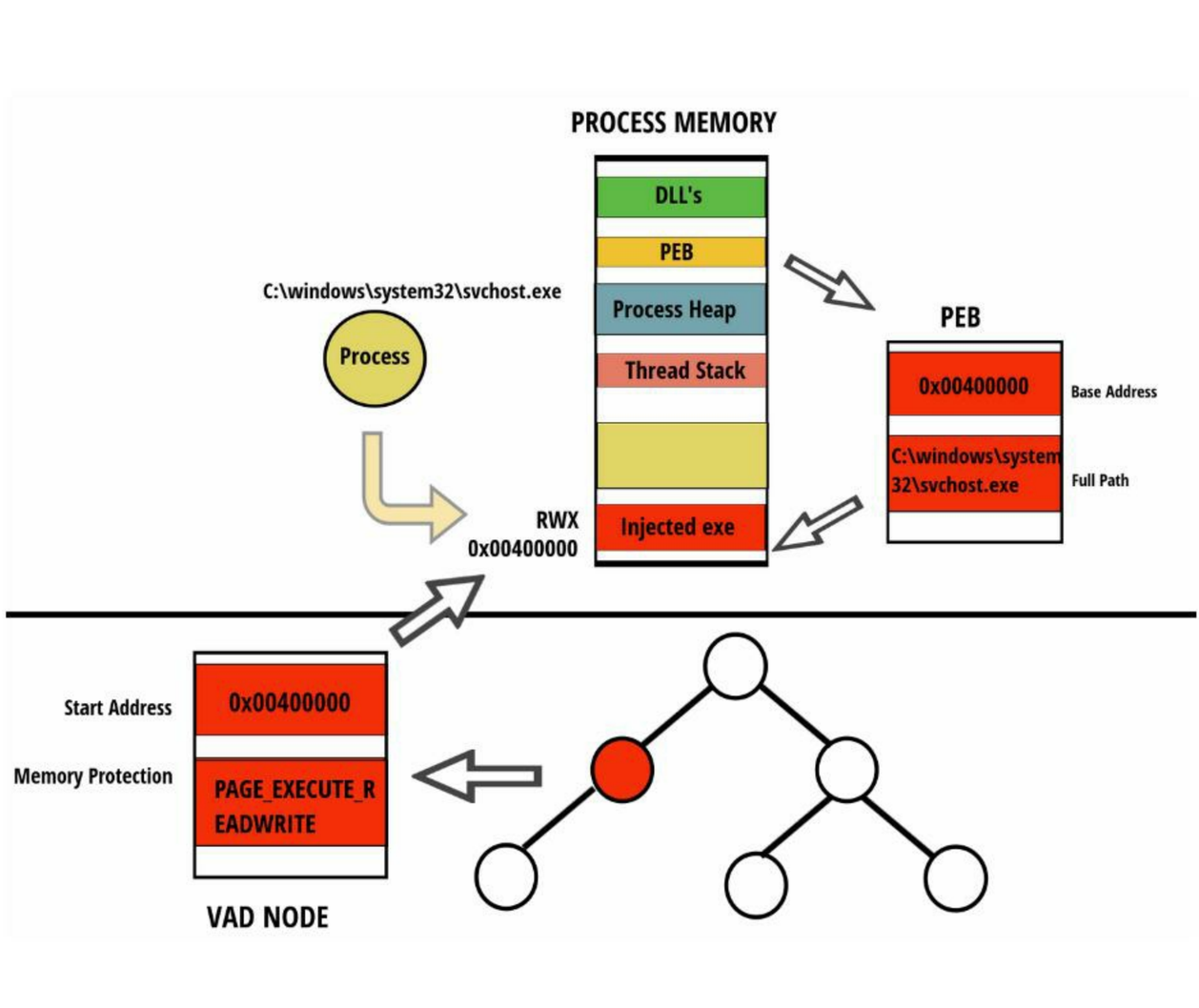
```
root@kratos:~/Volatility# python vol.py -f stuxnet.vmem malfind -p 868
Volatility Foundation Volatility Framework 2.5
Process: lsass.exe Pid: 868 Address: 0x1000000
Vad Tag: Vad  Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 2, Protection: 6

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
```



Hollow Process Injection - Skeeyah



BEFORE HOLLOWING



AFTER HOLLOWING



Detecting Hollow Process Injection using Memory Forensics - Skeeyah

svchost.exe process (pid 1824) was not started by services.exe

```
root@kratos:~/Volatility# python vol.py -f infected.vmem pslist | grep -i svchost
Volatility Foundation Volatility Framework 2.5
0x815cfaa0 svchost.exe             876    696     20      202      0      0 2016-05-10
 06:47:25 UTC+0000
0x818c5a78 svchost.exe             960    696      9      227      0      0 2016-05-10
 06:47:25 UTC+0000
0x8181e558 svchost.exe            1044    696     68     1227      0      0 2016-05-10
 06:47:25 UTC+0000
0x818c7230 svchost.exe            1104    696      5       59      0      0 2016-05-10
 06:47:25 UTC+0000
0x81743da0 svchost.exe            1144    696     15      210      0      0 2016-05-10
 06:47:25 UTC+0000
0x817ba390 svchost.exe            1824   1768      1       26      0      0 2016-05-12
 14:43:43 UTC+0000
```

```
root@kratos:~/Volatility# python vol.py -f infected.vmem pslist -p 696
Volatility Foundation Volatility Framework 2.5
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start
                        Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ----------
-------------------- ------------------------------
0x8186c980 services.exe <--       696    652     16      264      0      0 2016-05-10
 06:47:24 UTC+0000
root@kratos:~/Volatility# python vol.py -f infected.vmem pslist -p 1768
Volatility Foundation Volatility Framework 2.5
ERROR   : volatility.debug    : Cannot find PID 1768. If its terminated or unlinked, u
se psscan and then supply --offset=OFFSET <--
root@kratos:~/Volatility#
```

Comparing Dlllist and Ldrmodules plugin shows the discrepancy

```
root@kratos:~/Volatility# python vol.py -f infected.vmem dlllist -p 1824
Volatility Foundation Volatility Framework 2.5
************************************************************************
svchost.exe pid:   1824
Command line : "C:\WINDOWS\system32\svchost.exe"
Service Pack 3

Base             Size  LoadCount Path
---------- ---------- ---------- ----
0x00400000     0x7000     0xffff C:\WINDOWS\system32\svchost.exe
0x7c900000    0xaf000     0xffff C:\WINDOWS\system32\ntdll.dll
0x7c800000    0xf6000     0xffff C:\WINDOWS\system32\kernel32.dll
0x7e410000    0x91000     0xffff C:\WINDOWS\system32\USER32.dll
0x77f10000    0x49000     0xffff C:\WINDOWS\system32\GDI32.dll
0x5cb70000    0x26000        0x1 C:\WINDOWS\system32\ShimEng.dll
0x6f880000   0x1ca000        0x1 C:\WINDOWS\AppPatch\AcGenral.DLL
0x77dd0000    0x9b000       0x18 C:\WINDOWS\system32\ADVAPI32.dll
0x77e70000    0x92000        0xa C:\WINDOWS\system32\RPCRT4.dll
0x77fe0000    0x11000        0x5 C:\WINDOWS\system32\Secur32.dll
0x76b40000    0x2d000        0x2 C:\WINDOWS\system32\WINMM.dll
0x774e0000   0x13d000        0x2 C:\WINDOWS\system32\ole32.dll
0x77c10000    0x58000        0x9 C:\WINDOWS\system32\msvcrt.dll
0x77120000    0x8b000        0x1 C:\WINDOWS\system32\OLEAUT32.dll
```

```
root@kratos:~/Volatility# python vol.py -f infected.vmem ldrmodules -p 1824
Volatility Foundation Volatility Framework 2.5
Pid      Process              Base       InLoad InInit InMem MappedPath
-------- -------------------- ---------- ------ ------ ----- ----------
    1824 svchost.exe          0x7c900000 True   True   True  \WINDOWS\system32\ntdll.dll
    1824 svchost.exe          0x7c800000 True   True   True  \WINDOWS\system32\kernel32.dll
    1824 svchost.exe          0x773d0000 True   True   True  \WINDOWS\WinSxS\x86_Microsoft.W
-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83\comctl32.dll
    1824 svchost.exe          0x77f60000 True   True   True  \WINDOWS\system32\shlwapi.dll
    1824 svchost.exe          0x769c0000 True   True   True  \WINDOWS\system32\userenv.dll
    1824 svchost.exe          0x77dd0000 True   True   True  \WINDOWS\system32\advapi32.dll
    1824 svchost.exe          0x77be0000 True   True   True  \WINDOWS\system32\msacm32.dll
    1824 svchost.exe          0x77c00000 True   True   True  \WINDOWS\system32\version.dll
    1824 svchost.exe          0x76b40000 True   True   True  \WINDOWS\system32\winmm.dll
    1824 svchost.exe          0x77e70000 True   True   True  \WINDOWS\system32\rpcrt4.dll
    1824 svchost.exe          0x6f880000 True   True   True  \WINDOWS\AppPatch\AcGenral.dll
    1824 svchost.exe          0x774e0000 True   True   True  \WINDOWS\system32\ole32.dll
    1824 svchost.exe          0x7e410000 True   True   True  \WINDOWS\system32\user32.dll
    1824 svchost.exe          0x77f10000 True   True   True  \WINDOWS\system32\gdi32.dll
    1824 svchost.exe          0x77120000 True   True   True  \WINDOWS\system32\oleaut32.dll
    1824 svchost.exe          0x5cb70000 True   True   True  \WINDOWS\system32\shimeng.dll
    1824 svchost.exe          0x76390000 True   True   True  \WINDOWS\system32\imm32.dll
    1824 svchost.exe          0x7c9c0000 True   True   True  \WINDOWS\system32\shell32.dll
    1824 svchost.exe          0x77c10000 True   True   True  \WINDOWS\system32\msvcrt.dll
    1824 svchost.exe          0x5ad70000 True   True   True  \WINDOWS\system32\uxtheme.dll
    1824 svchost.exe          0x5d090000 True   True   True  \WINDOWS\system32\comctl32.dll
    1824 svchost.exe          0x77fe0000 True   True   True  \WINDOWS\system32\secur32.dll
root@kratos:~/Volatility#
```

Malfind Volatility plugin shows suspicious memory protection (PAGE_EXECUTE_READWRITE) at address 0x400000

```
root@kratos:~/Volatility# python vol.py -f infected.vmem malfind -p 1824
Volatility Foundation Volatility Framework 2.5
Process: svchost.exe Pid: 1824 Address: 0x400000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 7, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x00400000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x00400010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x00400020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00400030  00 00 00 00 00 00 00 00 00 00 00 00 e0 00 00 00   ................

0x00400000 4d               DEC EBP
0x00400001 5a               POP EDX
```



Automating Process Hollow Detection using HollowFind Volatility Plugin

Detecting Stuxnet's process hollowing using hollowfind plugin

```
root@kratos:~/Volatility# python vol.py -f stuxnet.vmem hollowfind
Volatility Foundation Volatility Framework 2.5
Hollowed Process Information:
        Process: lsass.exe PID: 868
        Parent Process: services.exe PPID: 668
        Creation Time: 2011-06-03 04:26:55 UTC+0000
        Process Base Name(PEB): lsass.exe
        Command Line(PEB): "C:\WINDOWS\\system32\\lsass.exe"
        Hollow Type: Invalid EXE Memory Protection and Process Path Discrepancy

VAD and PEB Comparison:
        Base Address(VAD): 0x1000000
        Process Path(VAD):
        Vad Protection: PAGE_EXECUTE_READWRITE
        Vad Tag: Vad

        Base Address(PEB): 0x1000000
        Process Path(PEB): C:\WINDOWS\system32\lsass.exe
        Memory Protection: PAGE_EXECUTE_READWRITE
        Memory Tag: Vad
```

```
Similar Processes:
        lsass.exe(868) Parent:services.exe(668) Start:2011-06-03 04:26:55 UTC+0000
        lsass.exe(680) Parent:winlogon.exe(624) Start:2010-10-29 17:08:54 UTC+0000
        lsass.exe(1928) Parent:services.exe(668) Start:2011-06-03 04:26:55 UTC+0000

Suspicious Memory Regions:
        0x80000(PE Found)  Protection: PAGE_EXECUTE_READWRITE  Tag: Vad
        0x1000000(PE Found)  Protection: PAGE_EXECUTE_READWRITE  Tag: Vad
------------------------------------------------
```

Detecting Skeeyah's process hollowing using hollowfind plugin

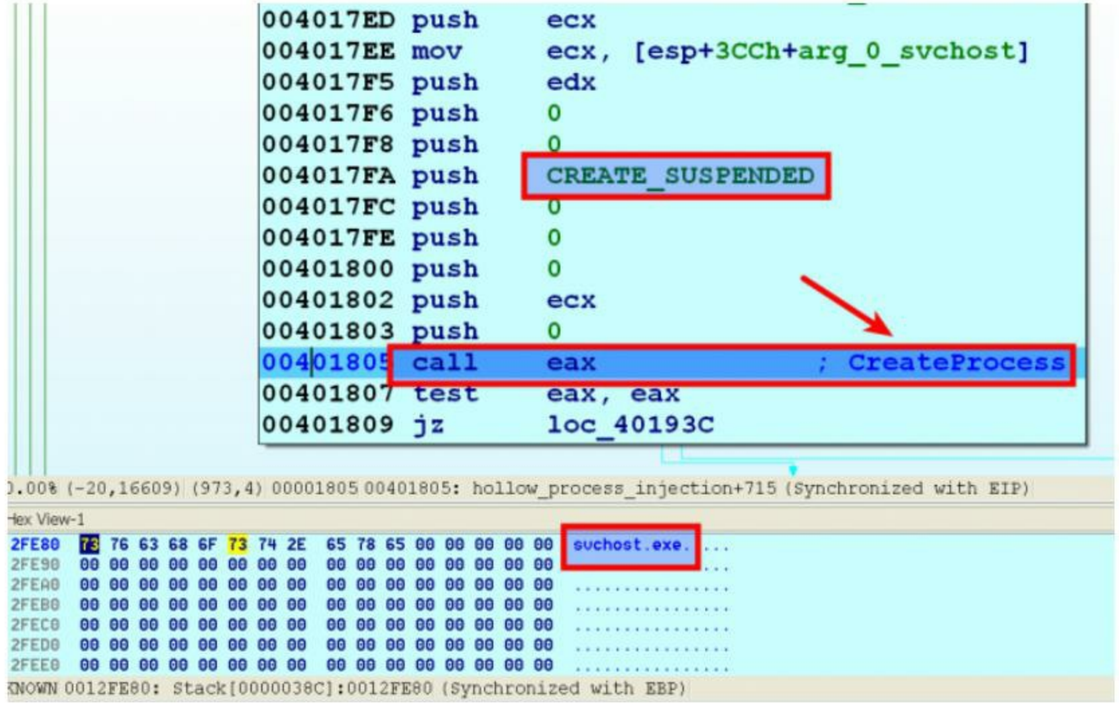
```
root@kratos:~/Volatility# python vol.py -f infected.vmem hollowfind
Volatility Foundation Volatility Framework 2.5
Hollowed Process Information:
        Process: svchost.exe PID: 1824 PPID: 1768
        Process Base Name(PEB): svchost.exe
        Hollow Type: No VAD Entry For Process Executable

VAD and PEB Comparison:
        Base Address(VAD): 0x0
        Process Path(VAD): NA
        Vad Protection: NA
        Vad Tag: NA

        Base Address(PEB): 0x400000
        Process Path(PEB): C:\WINDOWS\system32\svchost.exe
        Memory Protection: PAGE_EXECUTE_READWRITE
        Memory Tag: VadS

0x00400000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x00400010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x00400020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00400030  00 00 00 00 00 00 00 00 00 00 00 00 e0 00 00 00   ................
```

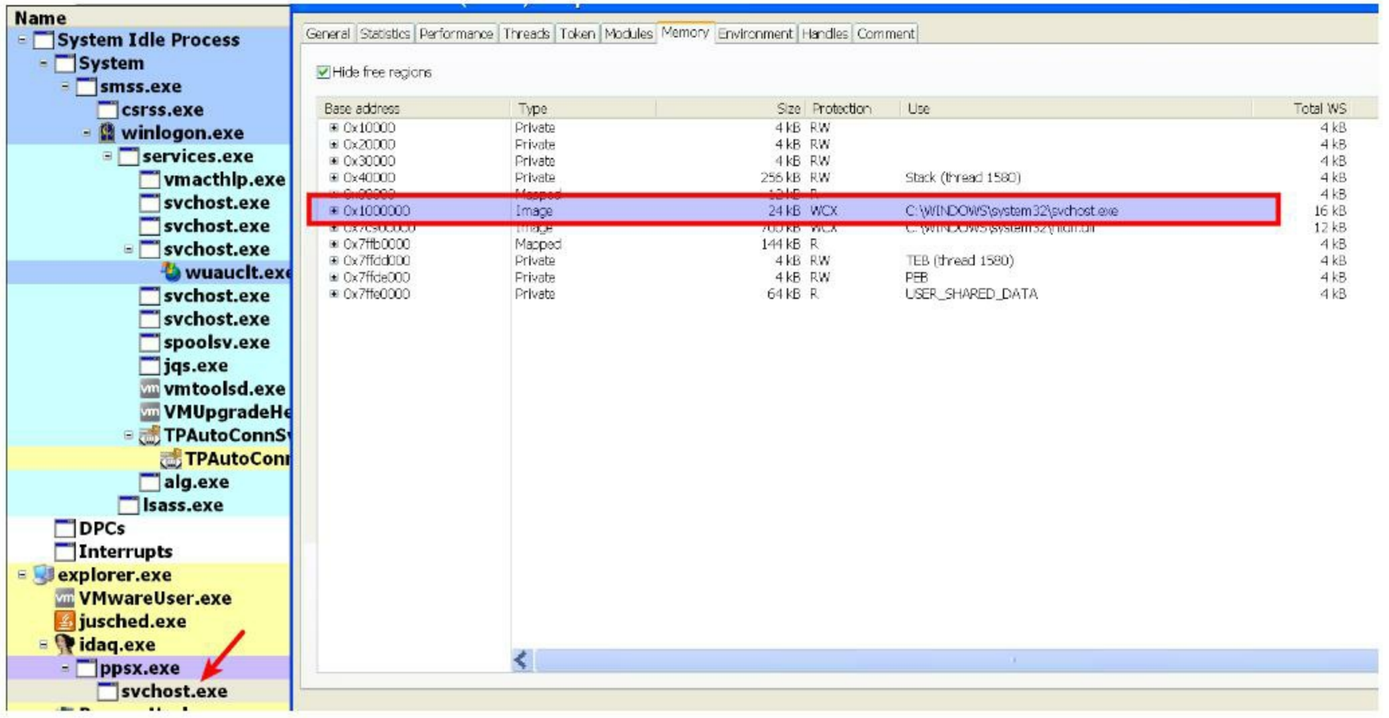
```
Similar Processes:
        svchost.exe(1824) Parent:NA(1768) Start:2016-05-12 14:43:43 UTC+0000
        svchost.exe(960) Parent:services.exe(696) Start:2016-05-10 06:47:25 UTC+0000
        svchost.exe(1104) Parent:services.exe(696) Start:2016-05-10 06:47:25 UTC+0000
        svchost.exe(1144) Parent:services.exe(696) Start:2016-05-10 06:47:25 UTC+0000
        svchost.exe(876) Parent:services.exe(696) Start:2016-05-10 06:47:25 UTC+0000
        svchost.exe(1044) Parent:services.exe(696) Start:2016-05-10 06:47:25 UTC+0000

Suspicious Memory Regions:
        0x400000(PE Found)  Protection: PAGE_EXECUTE_READWRITE  Tag: VadS
-----------------------------------------------
```



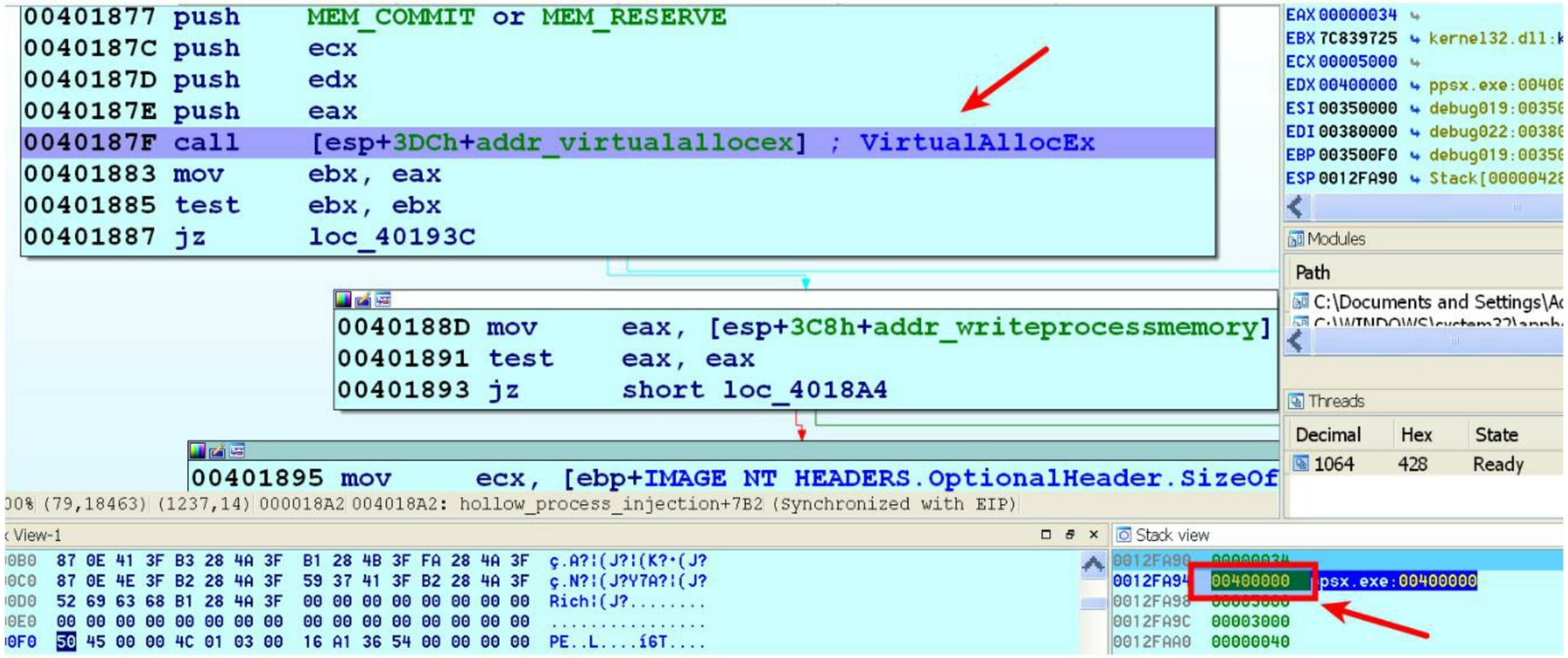
Evasive Hollow Process Injection - Taidoor

Malware creates svchost.exe process in suspended state, svchost.exe is loaded at 0x1000000

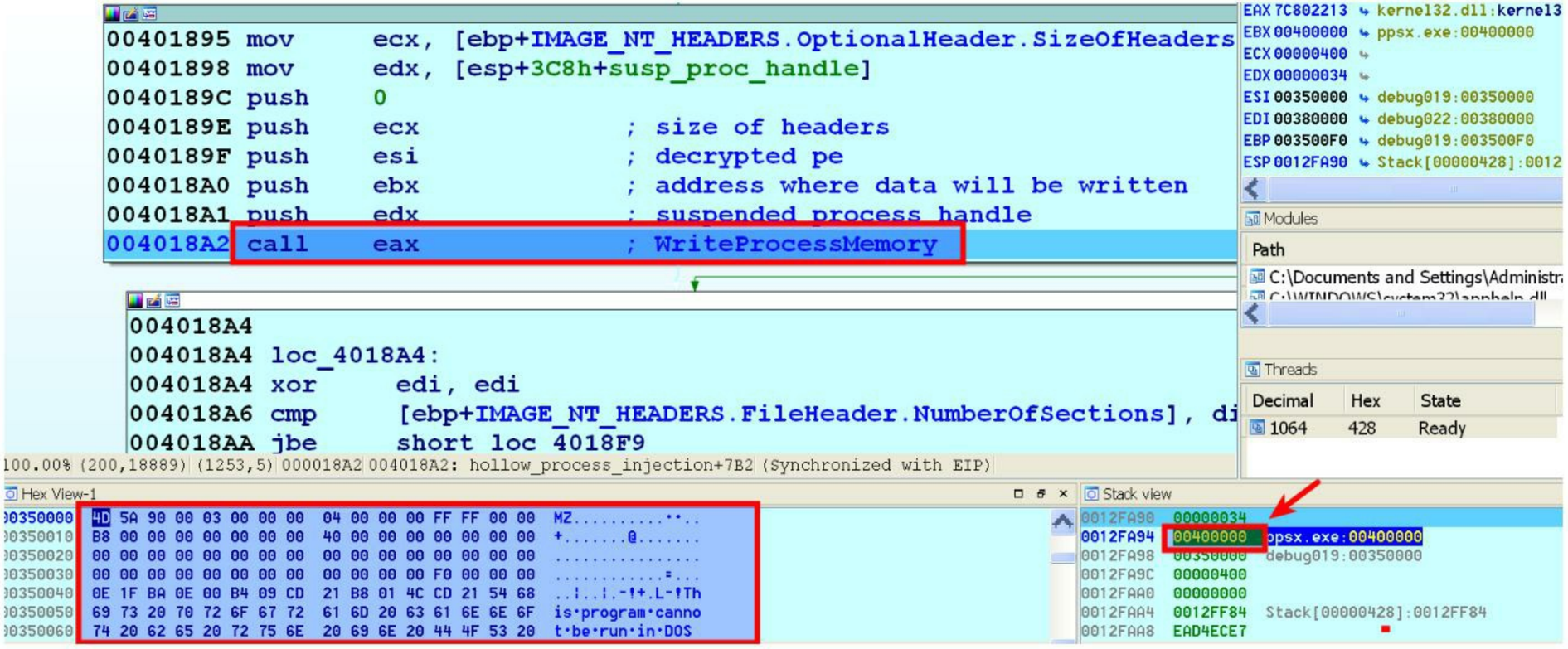




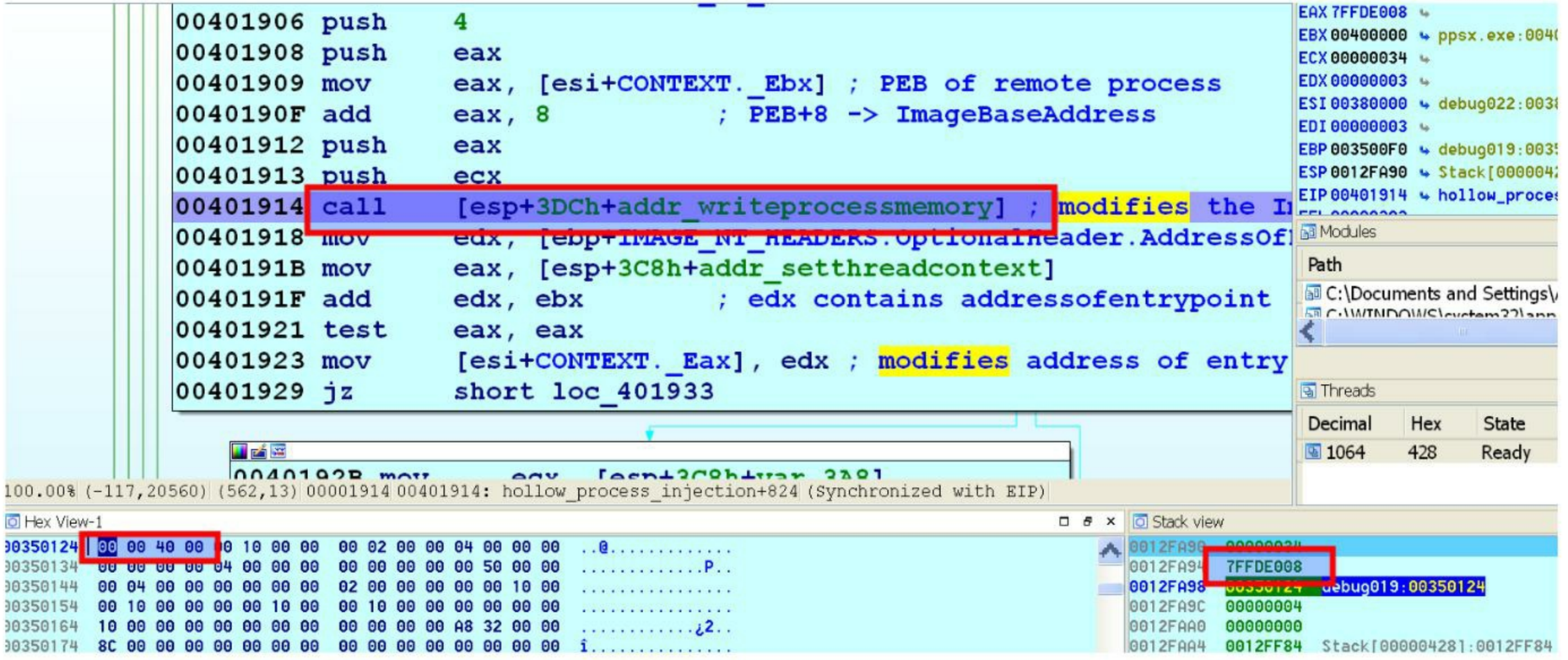
Malware allocates a memory in the remote process (svchost.exe) at address 0x00400000 but does not unmap (hollow out) the memory at the address 0x01000000



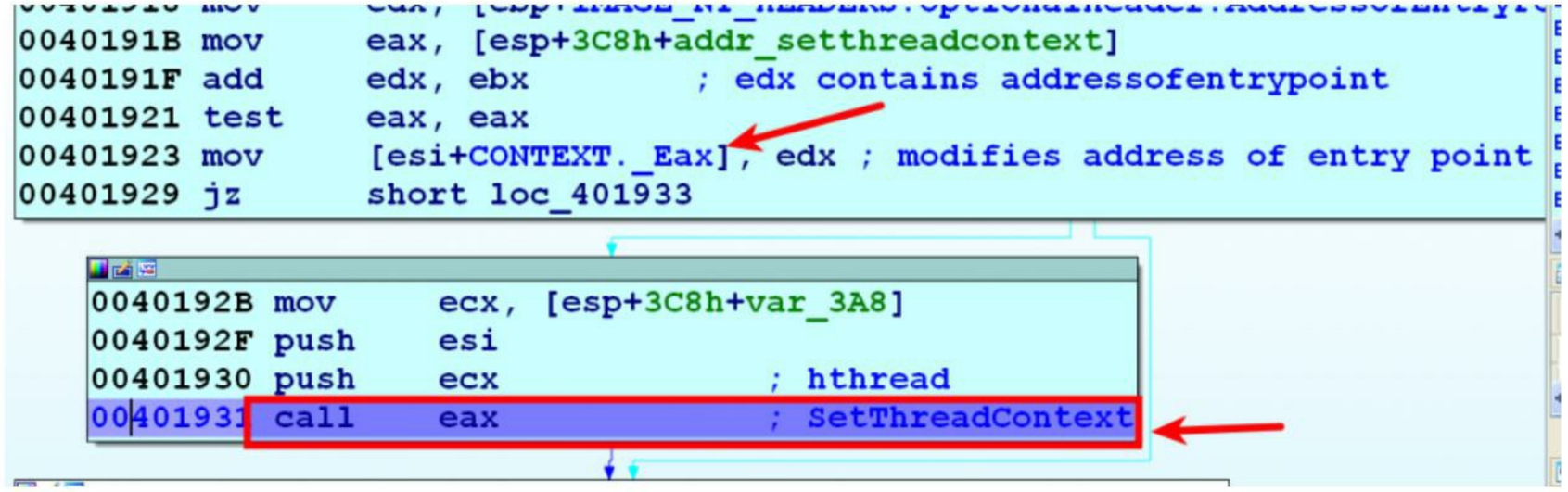
Malware writes the PE file to inject into the remote process (svchost.exe) at the allocated address 0x00400000

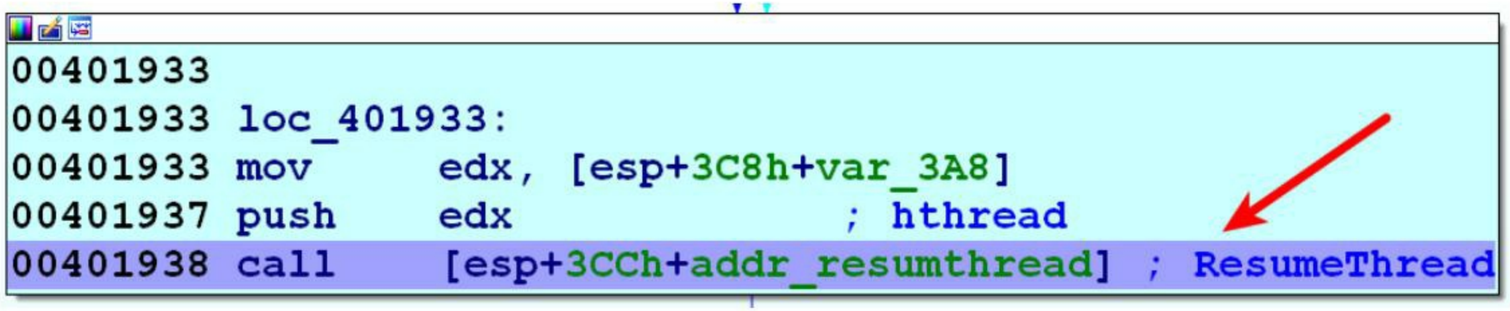


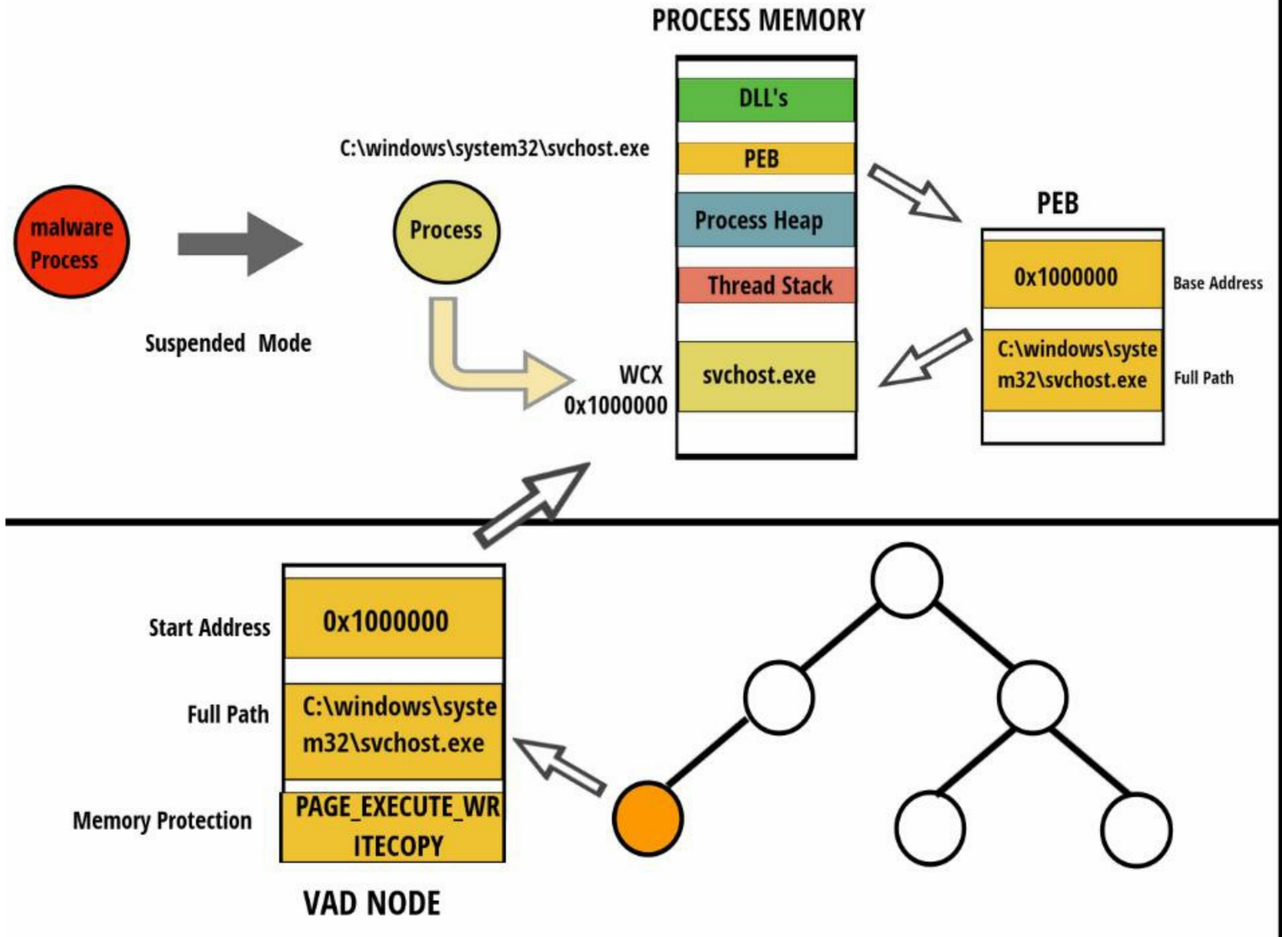
Malware then overwrites the PEB.ImageBaseAddress of svchost.exe with the new address (0x00400000), this changes the base address of svchost.exe from 0x01000000 to 0x00400000 (this address contains the injected executable)



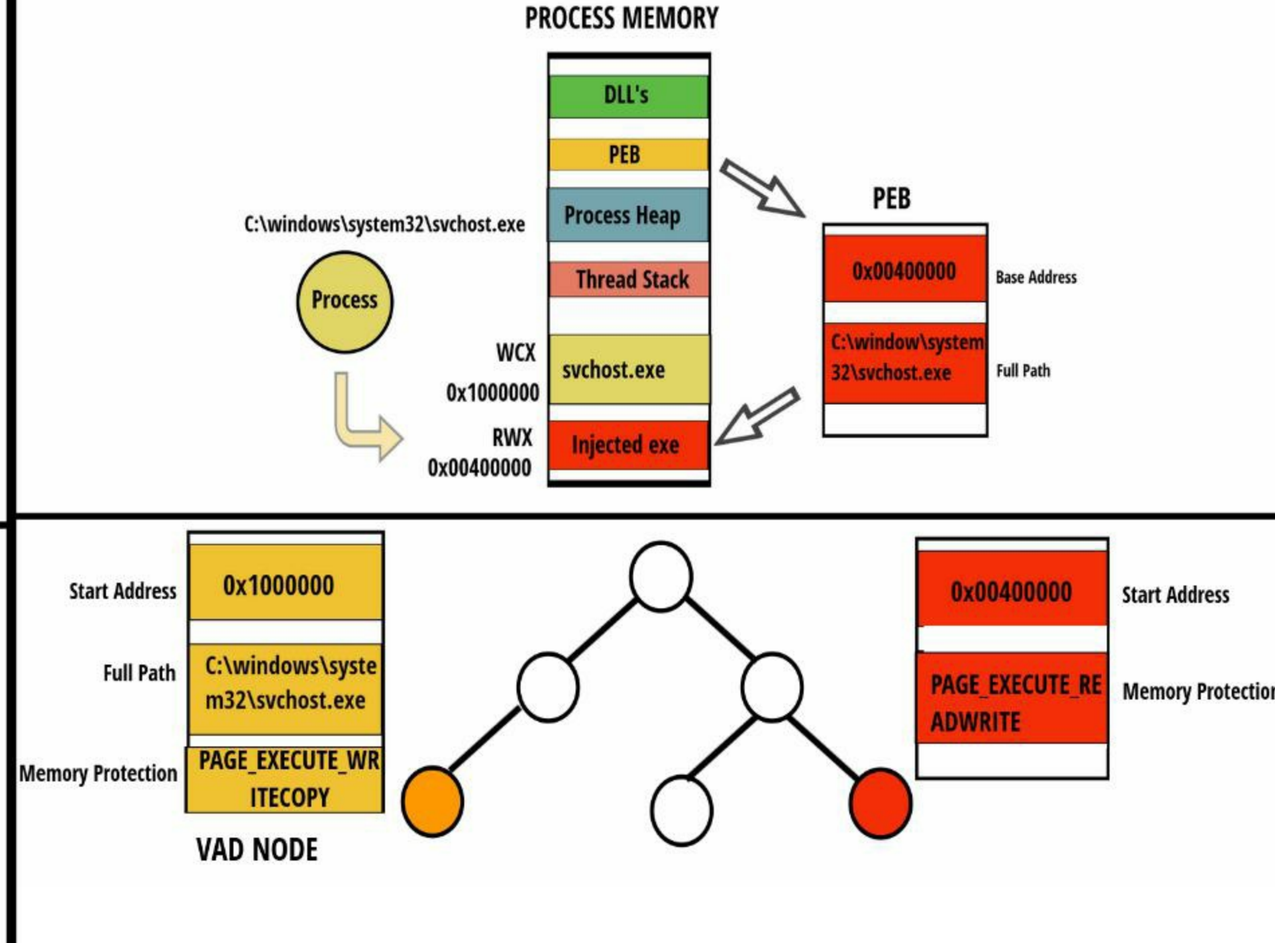
Malware then changes the start address of the suspended thread to the address of entry point of the injected executable by setting CONTEXT._Eax and using SetThreadContext api and then it resumes the thread







BEFORE HOLLOWING



AFTER HOLLOWING



Demo1 - Taidoor's Process Hollowing Confuses Analysis Tools

Hollowfind Plugin Detects Taidoor's Hollow Process Injection

```
Hollowed Process Information:
        Process: svchost.exe PID: 2020 PPID: 2012
        Process Base Name(PEB): svchost.exe
        Hollow Type: Process Base Address and Memory Protection Discrepancy

VAD and PEB Comparison:
        Base Address(VAD): 0x1000000
        Process Path(VAD): \WINDOWS\system32\svchost.exe
        Vad Protection: PAGE_EXECUTE_WRITECOPY
        Vad Tag: Vad

        Base Address(PEB): 0x400000
        Process Path(PEB): C:\WINDOWS\system32\svchost.exe
        Memory Protection: PAGE_EXECUTE_READWRITE
        Memory Tag: VadS

0x00400000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x00400010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x00400020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x00400030  00 00 00 00 00 00 00 00 00 00 00 00 f0 00 00 00   ................
```

```
Similar Processes:
        svchost.exe(2020) Parent:NA(2012) Start:2016-04-09 15:36:18 UTC+0000
        svchost.exe(960) Parent:services.exe(572) Start:2016-04-03 18:44:53 UTC+0000
        svchost.exe(1064) Parent:services.exe(572) Start:2016-04-03 18:44:55 UTC+0000
        svchost.exe(832) Parent:services.exe(572) Start:2016-04-03 18:44:53 UTC+0000
        svchost.exe(748) Parent:services.exe(572) Start:2016-04-03 18:44:53 UTC+0000
        svchost.exe(892) Parent:services.exe(572) Start:2016-04-03 18:44:53 UTC+0000

Suspicious Memory Regions:
        0x400000(PE Found)  Protection: PAGE_EXECUTE_READWRITE  Tag: VadS
------------------------------------------------
```

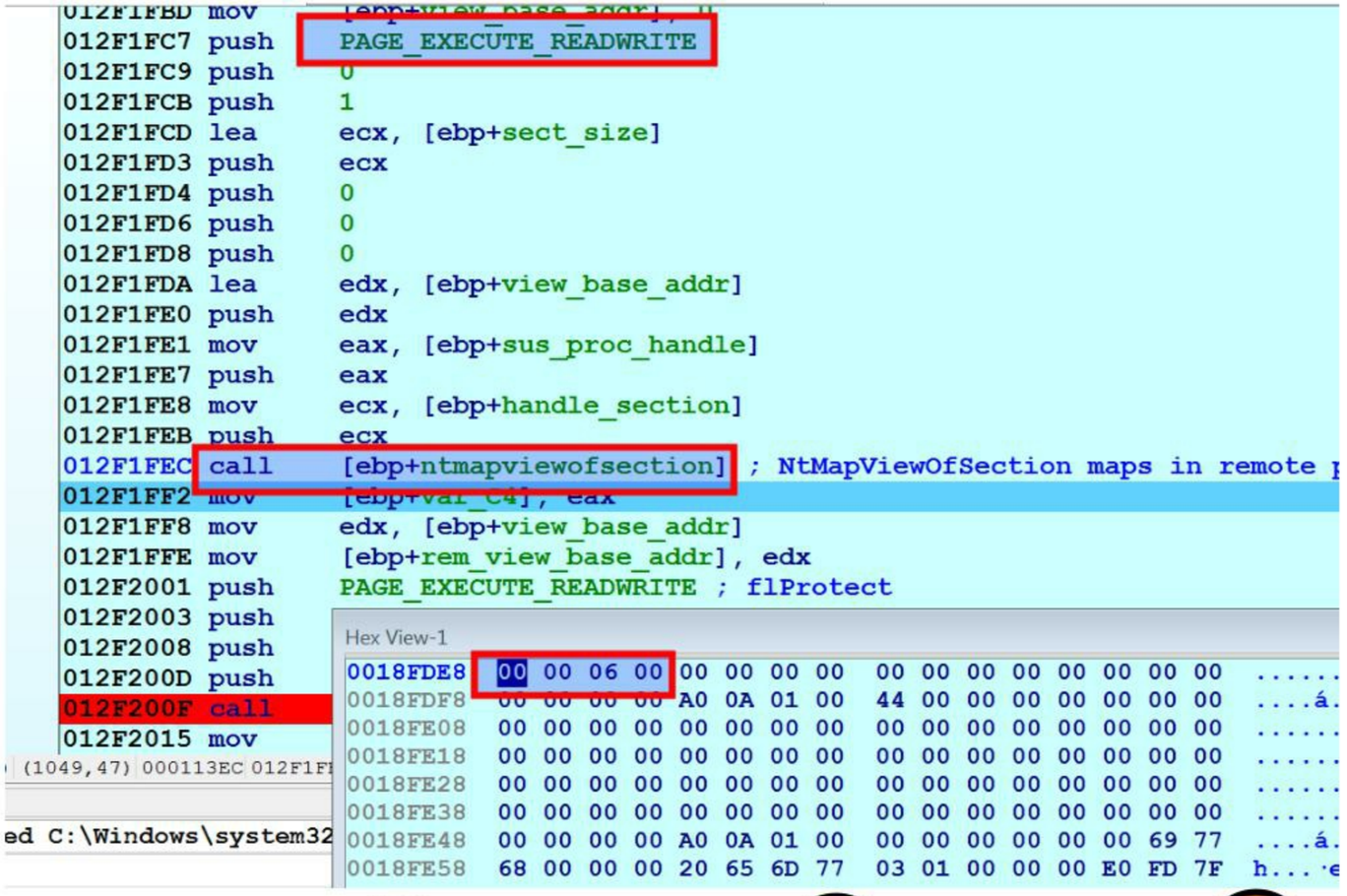


Evasive Hollow Process Injection - Kuluoz

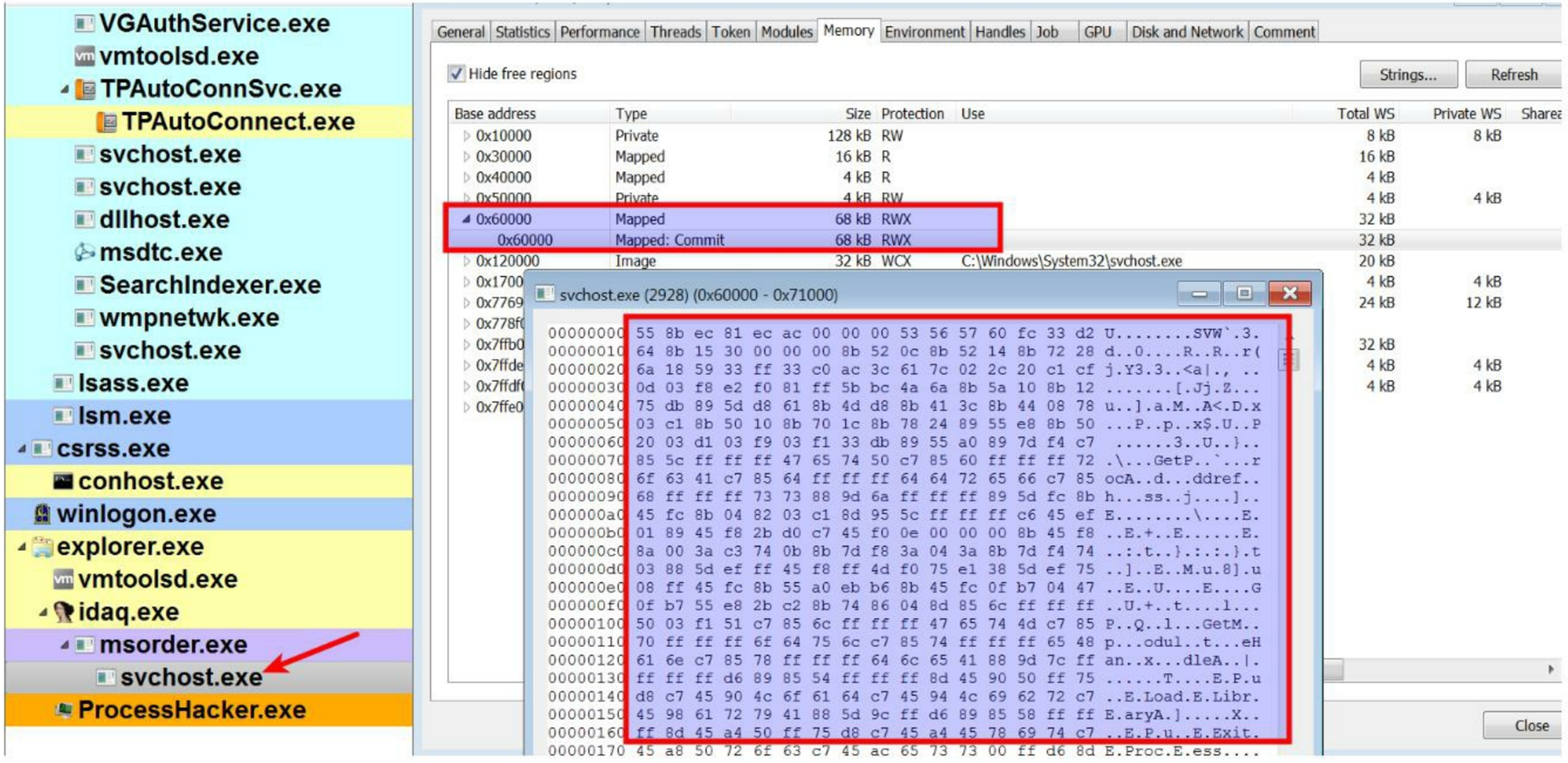
Kuluoz creates svchost.exe process in the suspended mode which loads svchost.exe at address 0x120000

```
012F1E89 push    edx                ; lpProcessInformation
012F1E8A lea     eax, [ebp+StartupInfo]
012F1E8D push    eax                ; lpStartupInfo
012F1E8E push    0                  ; lpCurrentDirectory
012F1E90 push    0                  ; lpEnvironment
012F1E92 push    CREATE_SUSPENDED   ; dwCreationFlags
012F1E94 push    0                  ; bInheritHandles
012F1E96 push    0                  ; lpThreadAttributes
012F1E98 push    0                  ; lpProcessAttributes
012F1E9A push    offset CommandLine ; "svchost.exe"
012F1E9F push    0                  ; lpApplicationName
012F1EA1 call    CreateProcessA
012F1EA7 mov     ecx, [ebp+ProcessInformation.hProcess]
012F1EAD mov     [ebp+sus_proc_handle], ecx
012F1EB3 mov     edx, [ebp+ProcessInformation.hThread]
012F1EB9 mov     [ebp+var_9C], edx
012F1EBF push    0
```

Instead of using VirtualAllocEX and WriteProcessMemory api call, kuluoz creates a section in its own address space, copies the malicious code into the created section and then maps a view of this section with read, write, execute(RWX) protections in the remote process (at address 0x60000 using NtMapViewOfSection api call



Code injected into the remote process (svchost.exe) at address 0x60000 with RWX memory protection



Malware copies the executable content of svchost.exe and It then unmaps the section in the svchost.exe where its executable is loaded (i.e 0x120000)

```
012F21A0
012F21A0 loc_12F21A0:
012F21A0 nop
012F21A1 mov     edx, [ebp+baseaddr_svchost]
012F21A7 push    edx
012F21A8 mov     eax, [ebp+sus_proc_handle]
012F21AE push    eax
012F21AF call    [ebp+Ntunmapviewofsection] ; NtUnmapViewOfSection
012F21B2 mov     [ebp+var_C4], eax
012F21B8 mov     ecx, [ebp+size_of_image]
012F21BB mov     [ebp+sect_size], ecx
```

Stack view

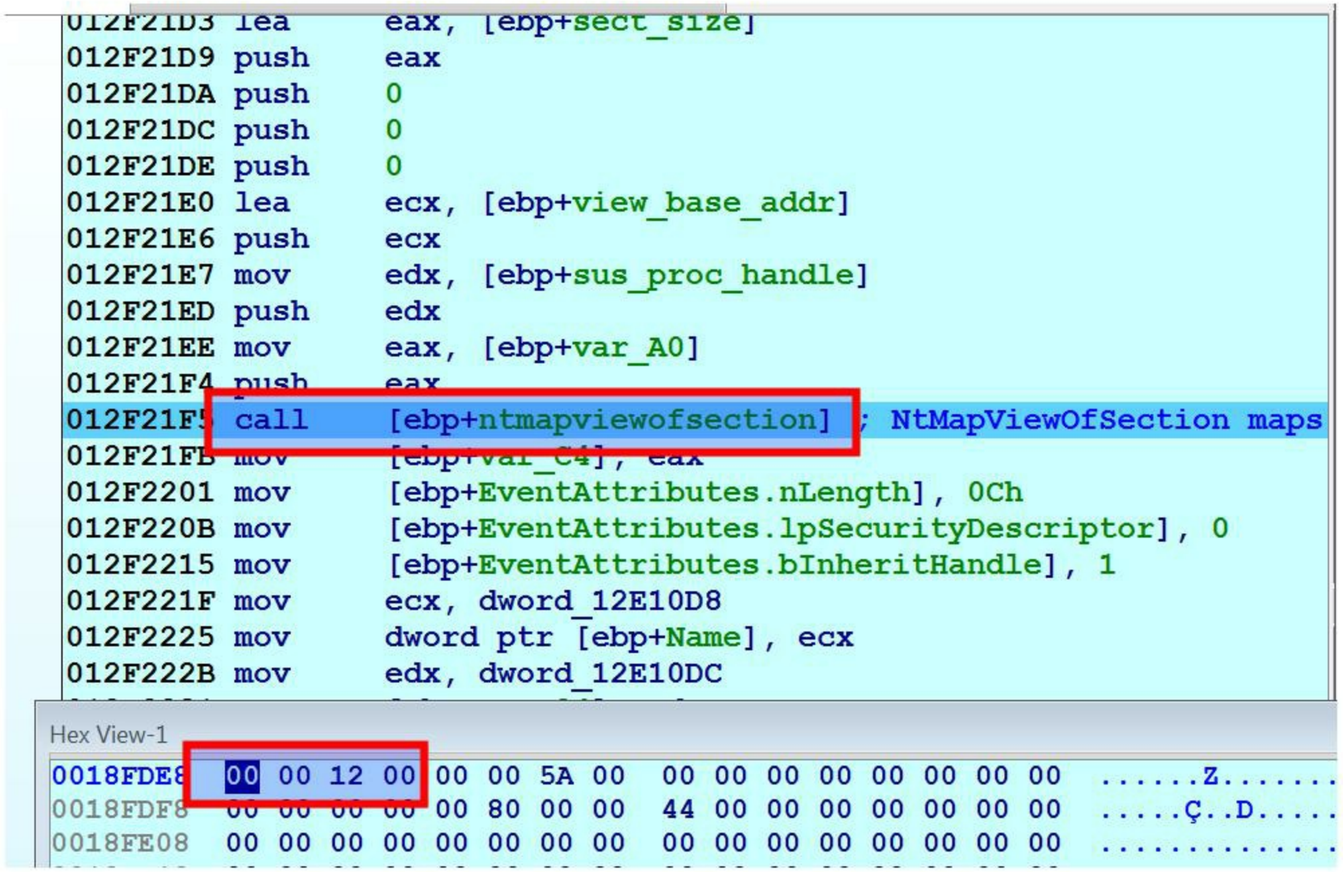
```
0018FC70  00000060
0018FC74  00120000
0018FC78  7FFDE008  TIB[
0018FC7C  00000005
0018FC80  00008000
0018FC84  00000000
0018FC88  00000000
0018FC8C  00000060
0018FC90  0000005C
0018FC94  00000B70
0018FC98  00000F94
```

General | Statistics | Performance | Threads | Token | Modules | Memory | Environment | Handles | Job | GPU | Disk and Network | Comment

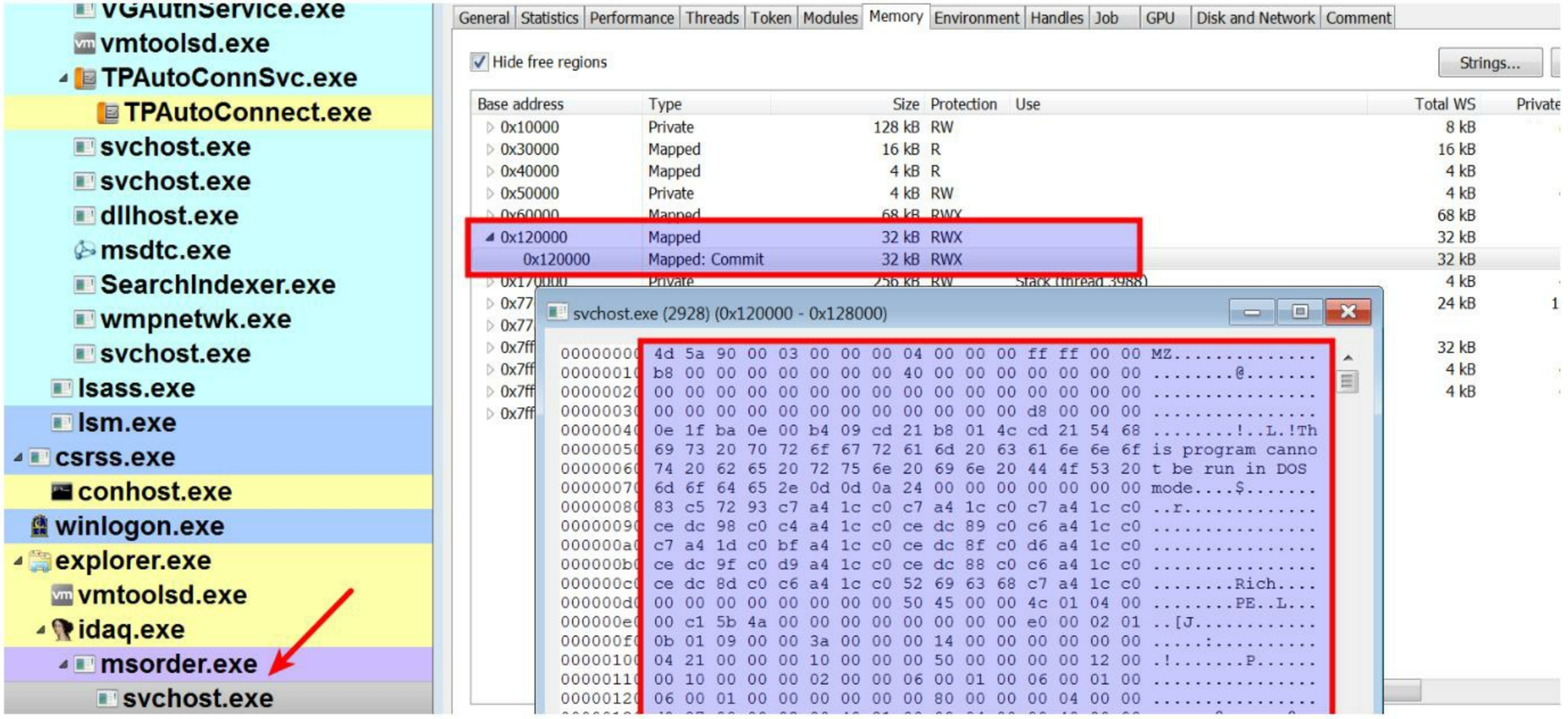
☑ Hide free regions Strings... Refresh

Base address	Type	Size	Protection	Use	Total WS	Private WS	Sharea
0x10000	Private	128 kB	RW		8 kB	8 kB	
0x30000	Mapped	16 kB	R		16 kB		
0x40000	Mapped	4 kB	R		4 kB		
0x50000	Private	4 kB	RW		4 kB	4 kB	
0x60000	Mapped	68 kB	RWX		68 kB		
0x170000	Private	256 kB	RW	Stack (thread 3988)	4 kB	4 kB	
0x77690000	Image	1,288 kB	WCX	C:\Windows\System32\ntdll.dll	24 kB	12 kB	
0x778f0000	Image	4 kB	WCX	C:\Windows\System32\apisetschema.dll			
0x7ffb0000	Mapped	140 kB	R		32 kB		
0x7ffde000	Private	4 kB	RW	PEB	4 kB	4 kB	
0x7ffdf000	Private	4 kB	RW	TEB (thread 3988)	4 kB	4 kB	
0x7ffe0000	Private	64 kB	R	USER_SHARED_DATA			

Malware modifies the copied exe content, then maps the section (which contains patched svchost.exe) into the remote process (svchost.exe) at the same address 0x120000 with read, write, execute (RWX) protection. At this point first 7 bytes of address of entry point of svchost.exe is modified.



Below screen shot shows the patched executable injected into the remote process (svchost.exe) at the address 0x120000 and once the patched exectuable is injected the suspended thread is resumed

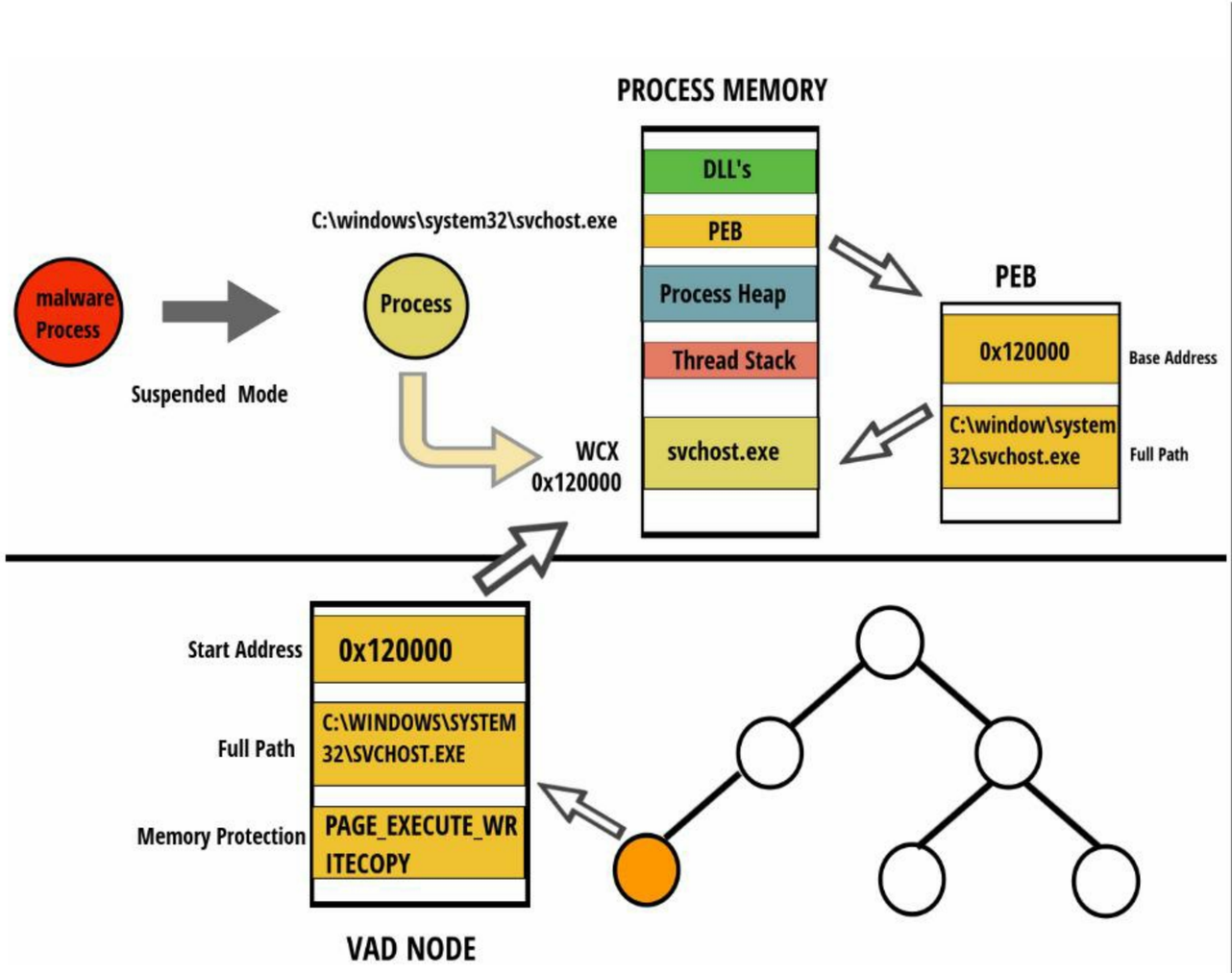


Comparing address of entry point of the legitimate svchost.exe (on the left) and the patched svchost.exe (on the right) process shows the difference in the 7 bytes at the address of entry point, whereas all other bytes are same. These 7 bytes turn out to be 3 instructions which will redirect the control flow to the malicious code that was injected before (at address 0x60000)

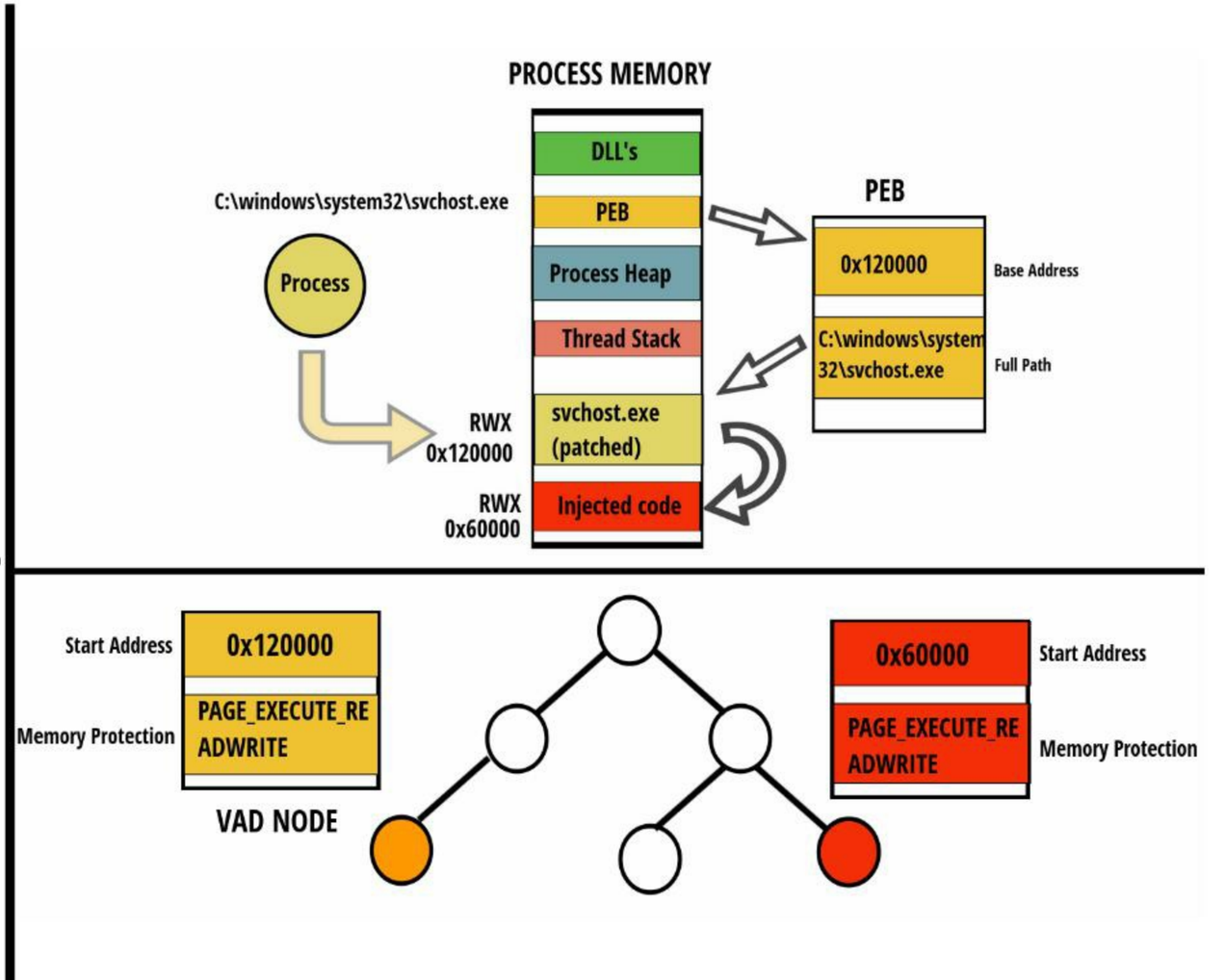
```
00001100 90 90 90 90 e8 d3 fc ff ff 6a 10 68 f8 21 12 00 .........j.h.!..
00001110 e8 75 fc ff ff 33 db 89 5d fc 64 a1 18 00 00 00 .u...3..].d.....
00001120 8b 70 04 89 5d e4 bf 80 50 12 00 53 56 57 ff 15 .p..]...P..SVW..
00001130 70 10 12 00 3b c3 0f 85 bc 0f 00 00 33 f6 46 a1 p...;.......3.F.
00001140 68 50 12 00 3b c6 0f 84 cb 0f 00 00 a1 68 50 12 hP..;.........hP.
00001150 00 85 c0 75 78 89 35 68 50 12 00 68 f0 21 12 00 ...ux.5hP..h.!..
00001160 68 e4 21 12 00 e8 71 ff ff ff 59 59 85 c0 0f 85 h.!...q...YY....
00001170 b0 0f 00 00 a1 68 50 12 00 3b c6 75 1b 68 e0 21 .....hP..;.u.h.!
00001180 12 00 68 d8 21 12 00 e8 f8 fb ff ff 59 59 c7 05 ..h.!.......YY..
00001190 68 50 12 00 02 00 00 00 39 5d e4 75 08 53 57 ff hP......9].u.SW.
000011a0 15 78 10 12 00 39 1d 58 55 12 00 0f 85 84 0f 00 .x...9.XU.......
000011b0 00 ff 35 bc 50 12 00 ff 35 c0 50 12 00 ff 35 b8 ..5.P...5.P...5.
000011c0 50 12 00 e8 64 02 00 00 e9 8a 0f 00 00 89 35 a4 P...d.........5.
000011d0 55 12 00 eb 9f 90 90 90 00 00 00 00 39 1d 12 00 U...........9...
000011e0 00 00 00 00 00 00 00 00 19 22 12 00 71 1d 12 00 .........."..q...
000011f0 00 00 00 00 90 90 90 90 fe ff ff ff 00 00 00 00 ................
00001200 d0 ff ff ff 00 00 00 00 fe ff ff ff 6e 31 12 00 ............n1..
00001210 82 31 12 00 90 90 90 90 90 b8 4d 5a 00 00 66 39 .1........MZ..f9
00001220 05 00 00 12 00 0f 85 8f 00 00 00 a1 3c 00 12 00 ............<...
00001230 8d 80 00 00 12 00 81 38 50 45 00 00 75 7c 0f b7 .......8PE..u|..
00001240 48 18 81 f9 0b 01 00 00 0f 85 70 0f 00 00 83 78 H.........p....x
00001250 74 0e 76 66 33 c9 39 88 e8 00 00 00 0f 95 c1 8b t.vf3.9.........
00001260 c1 6a 01 a3 48 50 12 00 e8 ce 01 00 00 50 ff 15 .j..HP.......P..
```

```
00002100 90 90 90 90 90 68 00 00 06 00 c3 68 f8 21 12 00 .....h.....h.!..
00002110 e8 75 fc ff ff 33 db 89 5d fc 64 a1 18 00 00 00 .u...3..].d.....
00002120 8b 70 04 89 5d e4 bf 80 50 12 00 53 56 57 ff 15 .p..]...P..SVW..
00002130 70 10 12 00 3b c3 0f 85 bc 0f 00 00 33 f6 46 a1 p...;.......3.F.
00002140 68 50 12 00 3b c6 0f 84 cb 0f 00 00 a1 68 50 12 hP..;.........hP.
00002150 00 85 c0 75 78 89 35 68 50 12 00 68 f0 21 12 00 ...ux.5hP..h.!..
00002160 68 e4 21 12 00 e8 71 ff ff ff 59 59 85 c0 0f 85 h.!...q...YY....
00002170 b0 0f 00 00 a1 68 50 12 00 3b c6 75 1b 68 e0 21 .....hP..;.u.h.!
00002180 12 00 68 d8 21 12 00 e8 f8 fb ff ff 59 59 c7 05 ..h.!.......YY..
00002190 68 50 12 00 02 00 00 00 39 5d e4 75 08 53 57 ff hP......9].u.SW.
000021a0 15 78 10 12 00 39 1d 58 55 12 00 0f 85 84 0f 00 .x...9.XU.......
000021b0 00 ff 35 bc 50 12 00 ff 35 c0 50 12 00 ff 35 b8 ..5.P...5.P...5.
000021c0 50 12 00 e8 64 02 00 00 e9 8a 0f 00 00 89 35 a4 P...d.........5.
000021d0 55 12 00 eb 9f 90 90 90 00 00 00 00 39 1d 12 00 U...........9...
000021e0 00 00 00 00 00 00 00 00 19 22 12 00 71 1d 12 00 .........."..q...
000021f0 00 00 00 00 90 90 90 90 fe ff ff ff 00 00 00 00 ................
00002200 d0 ff ff ff 00 00 00 00 fe ff ff ff 6e 31 12 00 ............n1..
00002210 82 31 12 00 90 90 90 90 90 b8 4d 5a 00 00 66 39 .1........MZ..f9
00002220 05 00 00 12 00 0f 85 8f 00 00 00 a1 3c 00 12 00 ............<...
00002230 8d 80 00 00 12 00 81 38 50 45 00 00 75 7c 0f b7 .......8PE..u|..
00002240 48 18 81 f9 0b 01 00 00 0f 85 70 0f 00 00 83 78 H.........p....x
00002250 74 0e 76 66 33 c9 39 88 e8 00 00 00 0f 95 c1 8b t.vf3.9.........
00002260 c1 6a 01 a3 48 50 12 00 e8 ce 01 00 00 50 ff 15 .j..HP.......P..
```

```
root@localhost:~# rasm2 -d 906800000600c3
nop
push 0x60000
ret
```



BEFORE HOLLOWING



AFTER HOLLOWING



Demo2 - Kuluoz's Process Hollowing Diverts Analysis

Hollowfind plugin detects Kuluoz's Hollow Process Injection

```
Hollowed Process Information:
        Process: svchost.exe PID: 3056 PPID: 3040
        Process Base Name(PEB): svchost.exe
        Hollow Type: Invalid EXE Memory Protection and Process Path Discrepancy

VAD and PEB Comparison:
        Base Address(VAD): 0xa00000
        Process Path(VAD):
        Vad Protection: PAGE_EXECUTE_READWRITE
        Vad Tag: Vad

        Base Address(PEB): 0xa00000
        Process Path(PEB): C:\Windows\system32\svchost.exe
        Memory Protection: PAGE_EXECUTE_READWRITE
        Memory Tag: Vad

Disassembly(Entry Point):
        0x00a02104 90              NOP
        0x00a02105 6800000600      PUSH DWORD 0x60000
        0x00a0210a c3              RET
        0x00a0210b 68f821a000      PUSH DWORD 0xa021f8
        0x00a02110 e875fcffff      CALL 0xa01d8a
```

```
Similar Processes:
        svchost.exe(3056) Parent:order.exe(3040) Start:2016-05-11 07:31:52 UTC+0000
        svchost.exe(1152) Parent:services.exe(496) Start:2016-05-11 06:35:30 UTC+0000
        svchost.exe(1068) Parent:services.exe(496) Start:2016-05-11 06:35:30 UTC+0000
        svchost.exe(1328) Parent:services.exe(496) Start:2016-05-11 06:35:30 UTC+0000
        svchost.exe(624) Parent:services.exe(496) Start:2016-05-11 06:35:29 UTC+0000
        svchost.exe(712) Parent:services.exe(496) Start:2016-05-11 06:35:29 UTC+0000
        svchost.exe(764) Parent:services.exe(496) Start:2016-05-11 06:35:29 UTC+0000
        svchost.exe(876) Parent:services.exe(496) Start:2016-05-11 06:35:30 UTC+0000
        svchost.exe(916) Parent:services.exe(496) Start:2016-05-11 06:35:30 UTC+0000

Suspicious Memory Regions:
        0x60000(No PE/Possibly Code)  Protection: PAGE_EXECUTE_READWRITE  Tag: Vad
        0x310000(No PE/Possibly Code)  Protection: PAGE_EXECUTE_READWRITE  Tag: VadS
        0xa00000(PE Found)  Protection: PAGE_EXECUTE_READWRITE  Tag: Vad
```



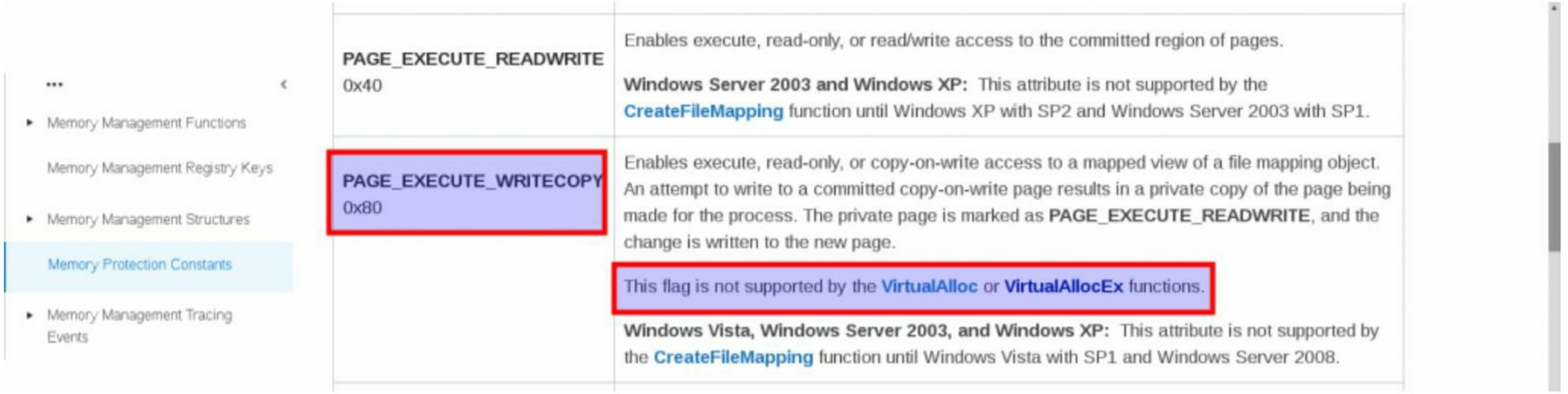
Evasive Hollow Process Injection - Modifying Kuluoz to be Stealthy

Steps to Make Kuluoz Stealthy

- *Instead of creating svchost.exe in the suspended mode, created explorer.exe in the suspended mode, the reason is because explorer.exe is normally started by userinit.exe and it terminates itself which means userinit.exe will not show up in the process listing (and will not show as parent for explorer.exe). So if malware starts explorer.exe, injects code and terminates itself, it can become hard to tell based on the parent child relationship*

- *Kuluoz injects code into the remote process using NtMapViewOfSection with read, write, execute(RWX) permission but if we can some how map that memory section containing malicious code with PAGE_EXECUTE_WRITECOPY protection we should be able to bypass the malfind plugin thereby hiding the memory region containing malicious code*

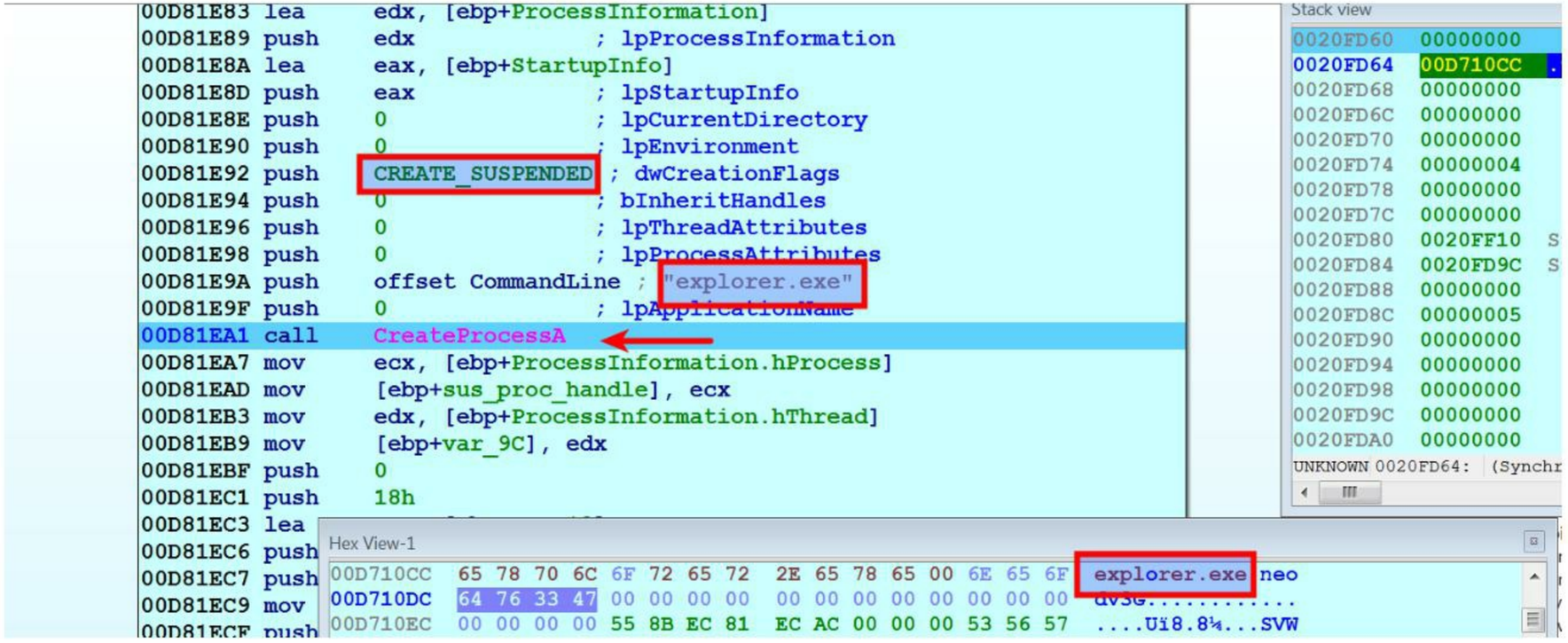
As per API documentation flag PAGE_EXEXCUTE_WRITECOPY is not supported by memory allocation API's like VirtualAllocEx but it turns out that PAGE_EXECUTE_WRITECOPY protection can be set by using the native api like NtMapViewOfSection, so Kuluoz code was modified to do that.



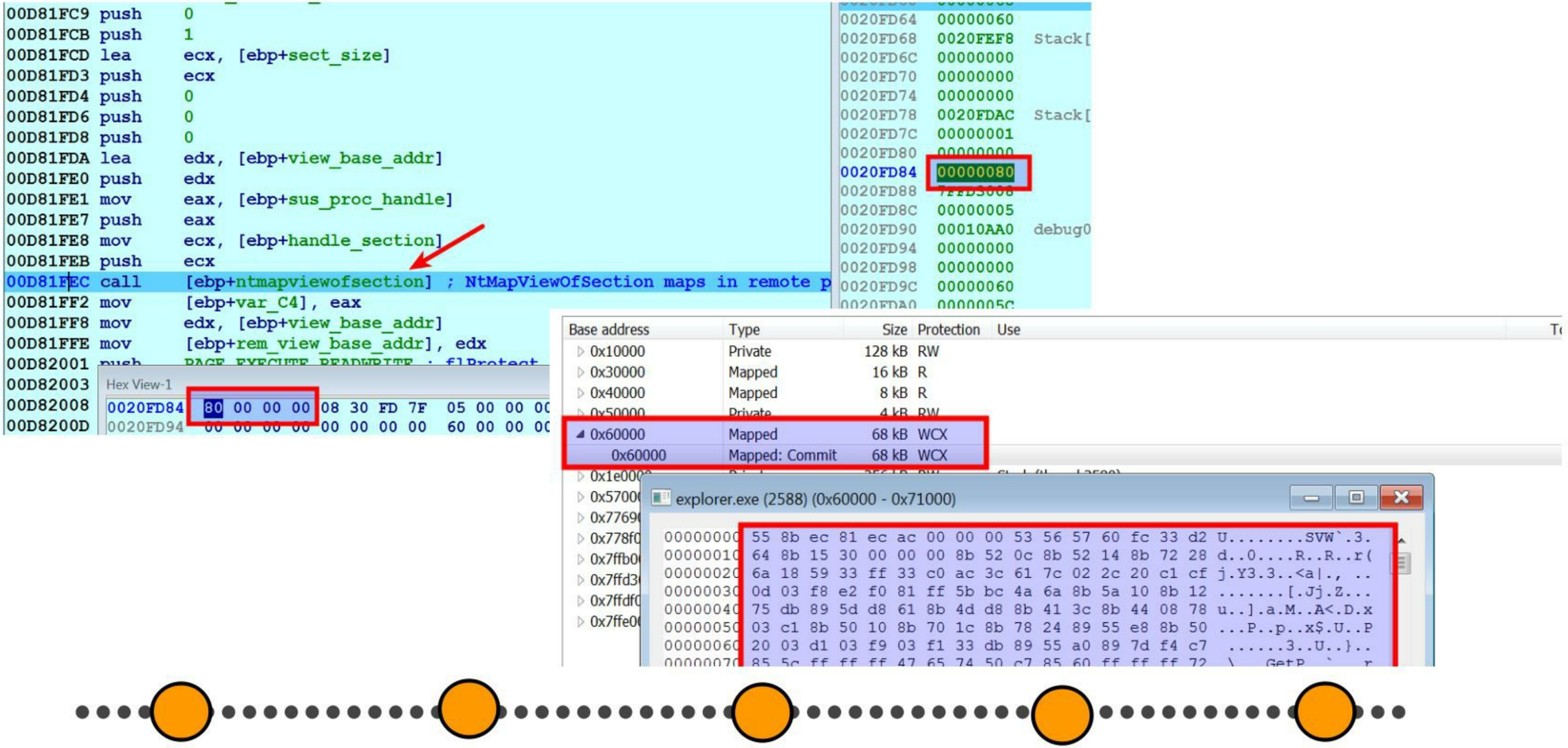
- Memory Management Functions
- Memory Management Registry Keys
- Memory Management Structures
- Memory Protection Constants
- Memory Management Tracing Events

PAGE_EXECUTE_READWRITE 0x40	Enables execute, read-only, or read/write access to the committed region of pages. **Windows Server 2003 and Windows XP:** This attribute is not supported by the **CreateFileMapping** function until Windows XP with SP2 and Windows Server 2003 with SP1.
PAGE_EXECUTE_WRITECOPY 0x80	Enables execute, read-only, or copy-on-write access to a mapped view of a file mapping object. An attempt to write to a committed copy-on-write page results in a private copy of the page being made for the process. The private page is marked as **PAGE_EXECUTE_READWRITE**, and the change is written to the new page. This flag is not supported by the **VirtualAlloc** or **VirtualAllocEx** functions. **Windows Vista, Windows Server 2003, and Windows XP:** This attribute is not supported by the **CreateFileMapping** function until Windows Vista with SP1 and Windows Server 2008.

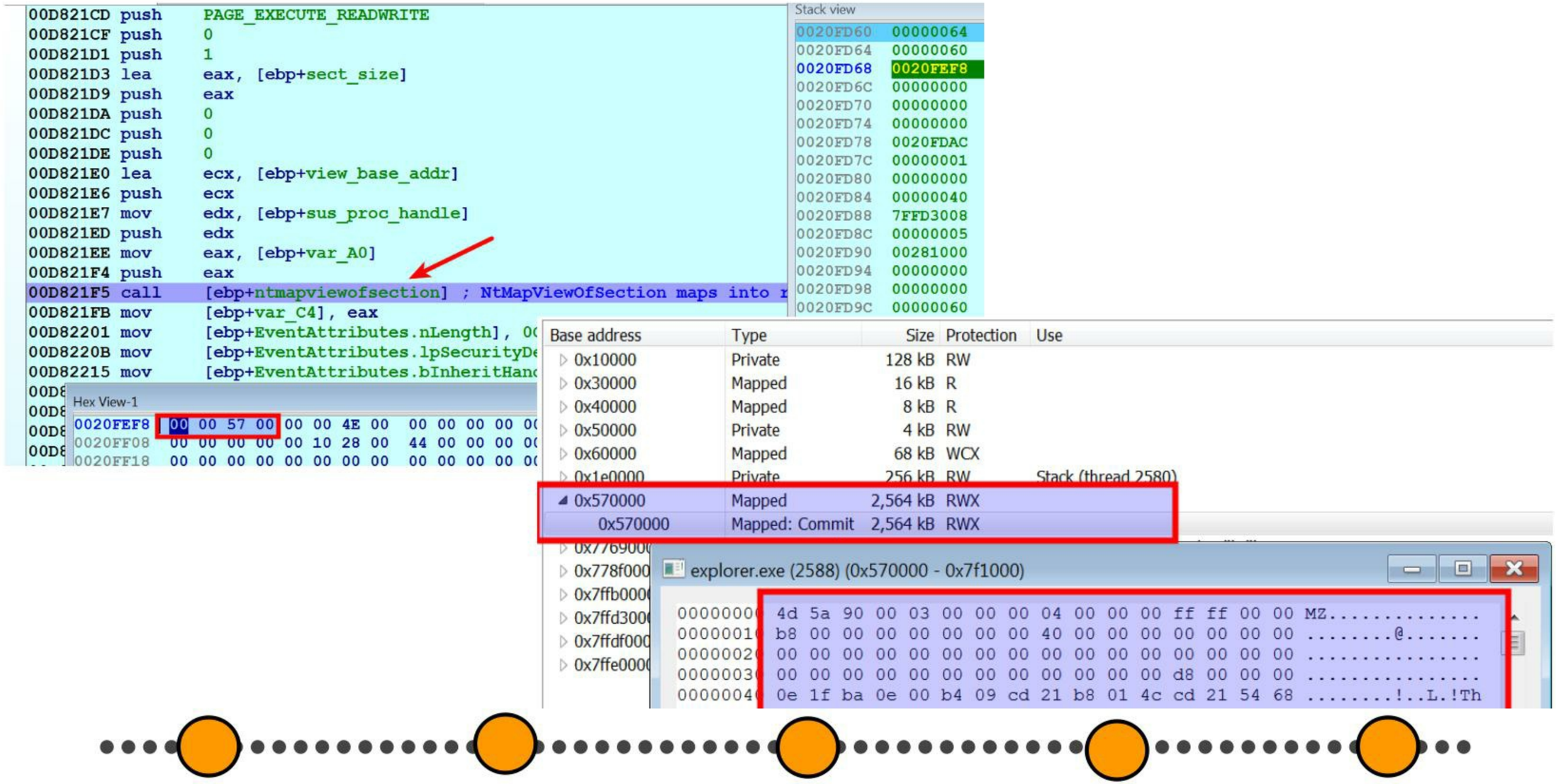
Kuluoz malwares sample was modified to create explorer.exe in the suspended mode instead of svchost.exe. The explorer.exe was loaded at base address 0x570000 with the PAGE_EXECUTE_WRITECOPY(WCX)

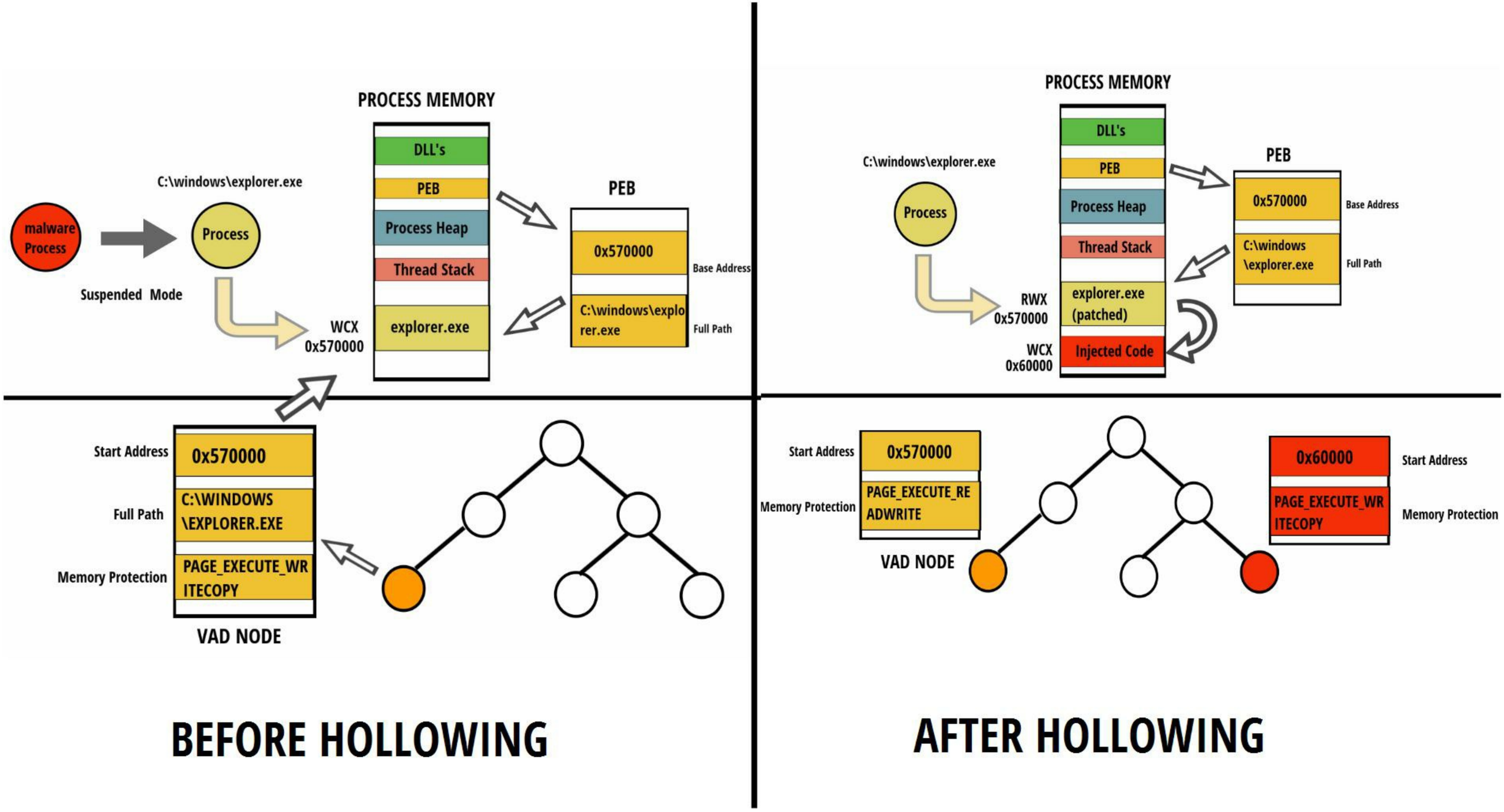


kuluoz was allowed to inject code into remote process (explorer.exe) at address 0x60000 but instead of allowing the malware to map the section with RWX protection, it was modified to map the section with writecopy(WCX) protection by changing the constant value to 0x80.



Malware was allowed to copy the content of explorer.exe, patch the copied content, hollow out explorer.exe and write the patched content back into remote process (explorer.exe) at 0x570000. At this point first 7 bytes of address of entry point of explorer.exe is modified




PROCESS MEMORY
DLL's
PEB
Process Heap
Thread Stack
explorer.exe
C:\windows\explorer.exe
malware Process
Process
Suspended Mode
WCX 0x570000
PEB
0x570000
Base Address
C:\windows\explorer.exe
Full Path
Start Address
0x570000
Full Path
C:\WINDOWS \EXPLORER.EXE
Memory Protection
PAGE_EXECUTE_WRITECOPY
VAD NODE
BEFORE HOLLOWING
PROCESS MEMORY
DLL's
PEB
Process Heap
Thread Stack
explorer.exe (patched)
Injected Code
C:\windows\explorer.exe
Process
RWX 0x570000
WCX 0x60000
PEB
0x570000
Base Address
C:\windows \explorer.exe
Full Path
Start Address
0x570000
Memory Protection
PAGE_EXECUTE_READWRITE
VAD NODE
0x60000
Start Address
PAGE_EXECUTE_WRITECOPY
Memory Protection
AFTER HOLLOWING



Demo3 - Kuluoz's modified Process Hollowing Hides from Forensic Tools

Hollowfind plugin detects Kuluoz's Modified Hollow Process Injection

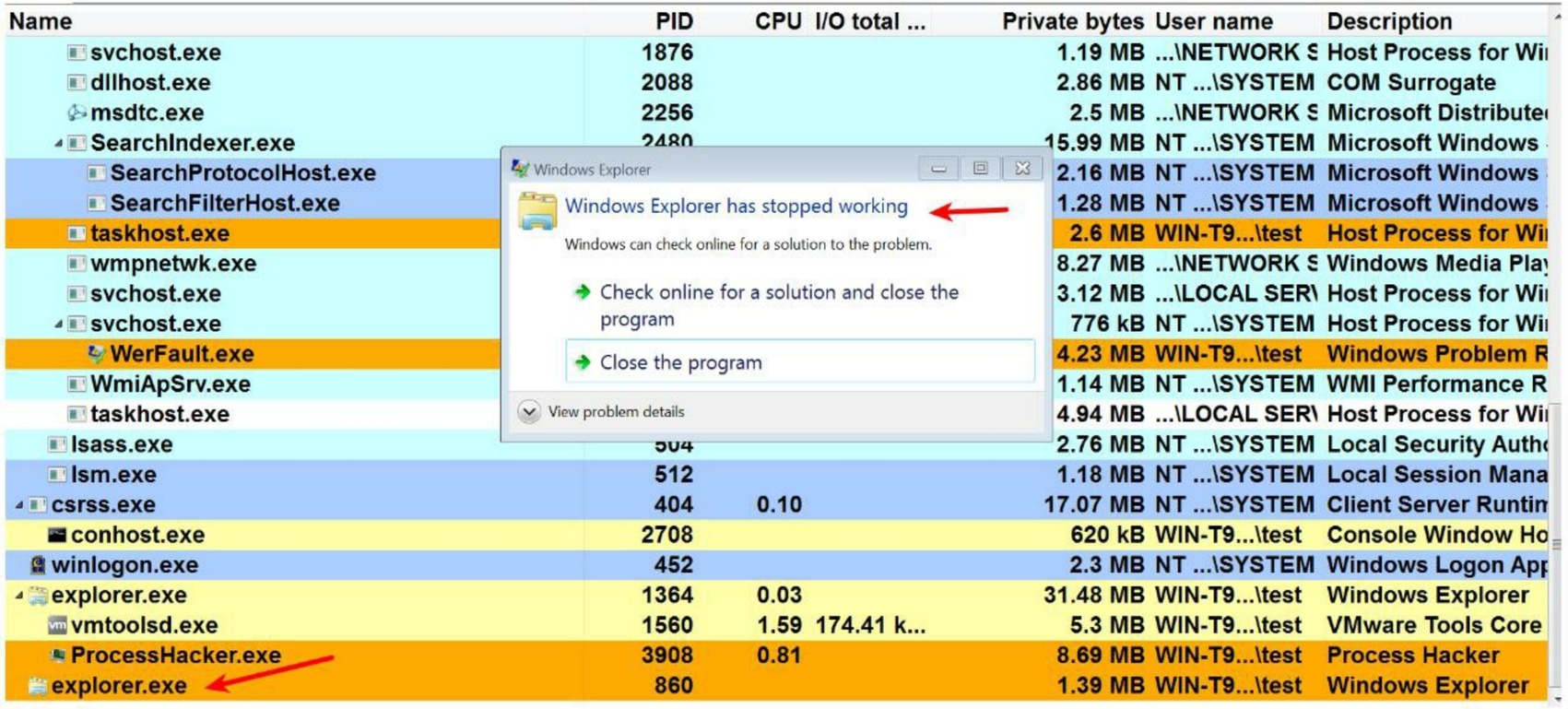
```
Hollowed Process Information:
        Process: explorer.exe PID: 2588 PPID: 160
        Process Base Name(PEB): explorer.exe
        Hollow Type: Invalid EXE Memory Protection and Process Path Discrepancy

VAD and PEB Comparison:
        Base Address(VAD): 0x570000
        Process Path(VAD):
        Vad Protection: PAGE_EXECUTE_READWRITE
        Vad Tag: Vad

        Base Address(PEB): 0x570000
        Process Path(PEB): C:\Windows\explorer.exe
        Memory Protection: PAGE_EXECUTE_READWRITE
        Memory Tag: Vad

Disassembly(Entry Point):
        0x005a0efa 90              NOP
        0x005a0efb 6800000600      PUSH DWORD 0x60000
        0x005a0f00 c3              RET
        0x005a0f01 6830105a00      PUSH DWORD 0x5a1030
        0x005a0f06 e8c11d0000      CALL 0x5a2ccc
```

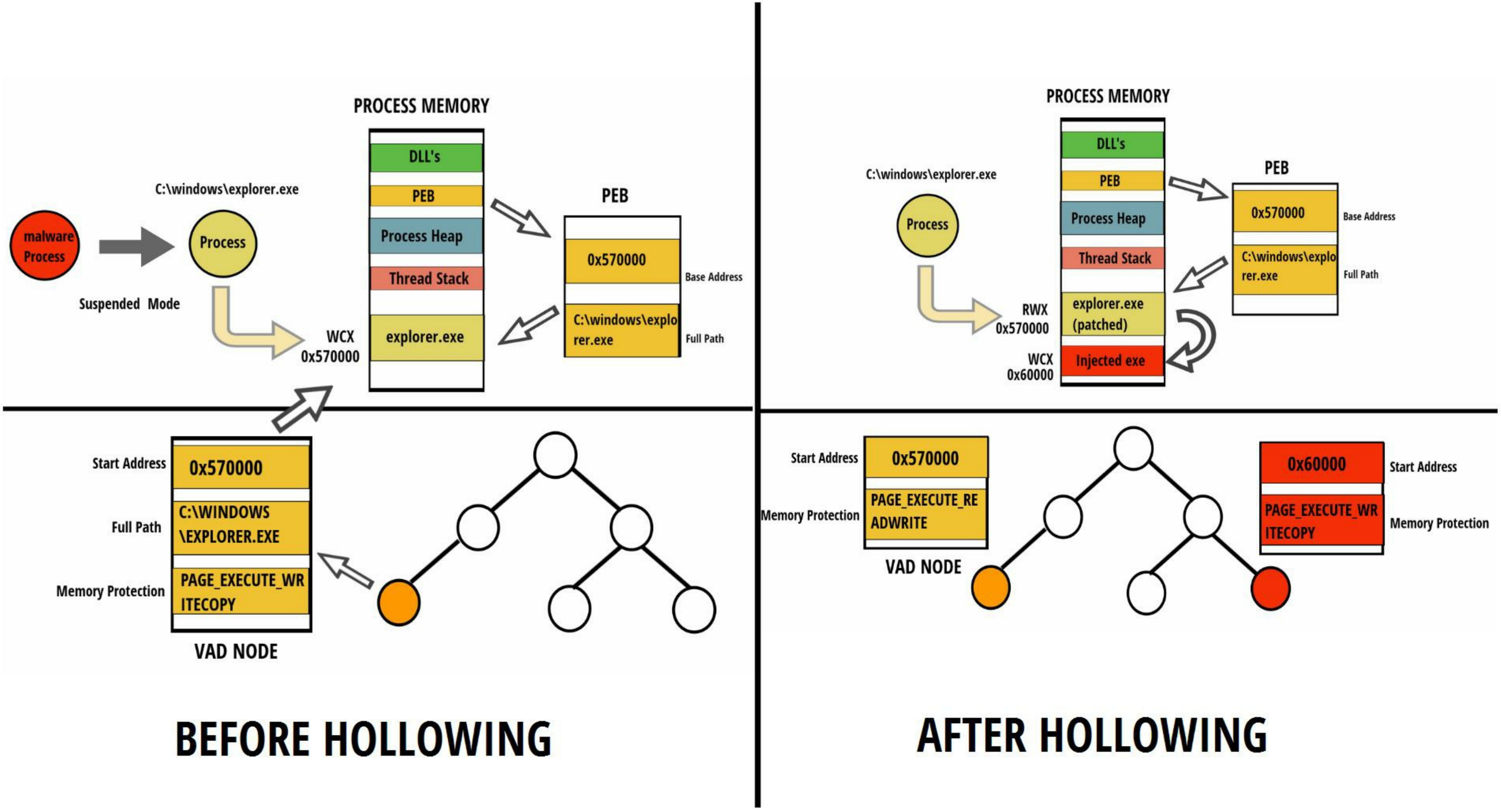
```
Similar Processes:
        explorer.exe(2588) Parent:msorder.exe(160) Start:2016-06-26 10:04:34 UTC+0000
        explorer.exe(1348) Parent:NA(1328) Start:2016-06-24 13:28:21 UTC+0000

Suspicious Memory Regions:
        0x60000(No PE/Possibly Code)  Protection: PAGE_EXECUTE_WRITECOPY  Tag: Vad
        0x370000(No PE/Possibly Code)  Protection: PAGE_EXECUTE_READWRITE  Tag: VadS
        0x570000(PE Found)  Protection: PAGE_EXECUTE_READWRITE  Tag: Vad
```



Evasive Hollow Process Injection - Kronos

Kronos perfoms similar hollow process redirection technique as kuluoz, this sample hollows out the explorer.exe process, patches the address of entry point and attempts to redirect execution flow inside an executable (instead of code) which was injected with PAGE_EXECUTE_WRITECOPY protections. While testing this executable the explorer.exe crashed as shown below, but still the memory image was taken for further analysis.




PROCESS MEMORY
DLL's
PEB
Process Heap
Thread Stack
explorer.exe
C:\windows\explorer.exe
malware Process
Process
Suspended Mode
WCX 0x570000
PEB
0x570000
Base Address
C:\windows\explorer.exe
Full Path
Start Address
0x570000
Full Path
C:\WINDOWS \EXPLORER.EXE
Memory Protection
PAGE_EXECUTE_WRITECOPY
VAD NODE
BEFORE HOLLOWING
PROCESS MEMORY
DLL's
PEB
Process Heap
Thread Stack
explorer.exe (patched)
Injected exe
C:\windows\explorer.exe
Process
RWX 0x570000
WCX 0x60000
PEB
0x570000
Base Address
C:\windows\explorer.exe
Full Path
Start Address
0x570000
Memory Protection
PAGE_EXECUTE_READWRITE
VAD NODE
0x60000
Start Address
PAGE_EXECUTE_WRITECOPY
Memory Protection
AFTER HOLLOWING



Demo4 - Kronos modified Process Hollowing Hides from Forensic Tools

Black Hat Sound Bytes

Key Take Aways:

- ***Attackers find new ways to bypass, confuse & divert analysis***
- ***Tools don't work always as expected***
- ***Understanding the working of these stealth techniques will help better detect & counter such attacks***

References

- *https://cysinfo.com/detecting-deceptive-hollowing-techniques/*
- *http://www.volatilityfoundation.org/#!releases/component_71401*
- *https://www.trustwave.com/Resources/SpiderLabs-Blog/Analyzing-Malware-Hollow-Processes/*
- *http://mnin.blogspot.in/2011/06/examining-stuxnets-footprint-in-memory.html*
- *http://journeyintoir.blogspot.in/2015/02/process-hollowing-meets-cuckoo-sandbox.html*

THANK YOU

@monnappa22

http://www.youtube.com/c/MonnappaKA

monnappa22@gmail.com

https://www.cysinfo.com