Lab 8 Solutions - The Case of Taidoor Espionage

Lab 8 - The Case of Taidoor Espionage

Your security device alerts on a malware callback connection from 192.168.1.60 to 200.2.126.61 on port 443. you suspect the host 192.168.1.60 to be infected. you collect the memory image from the host (taidoor.vmem). Analyze the memory image and answer the below questions

- Can you confirm if the host made the connection to the C2 server?
- What is the process id of the malicious process?
- Can you determine the full path of the malicious process and based on the path do you think its a legitimate operating system process?
- If this is a legitimate process, is there is anything that makes it different from the legitimate process?
- Dump the process onto disk, can you recognize any interesting strings?
- Based on your observation, what code injection technique malware is using?

Answers

01. Can you confirm if the host made the connection to the C2 server?

Running the ***netscan*** plugin shows a closed connection to the C2 server on port ***443***, this confirms that the host had established a network connection with the C2 server and associated process is ***svchost.exe*** (***pid 1412***)

02. What is the process id of the malicious process?

The process id of the malicious process is ***1412*** and it is associated with ***svchost.exe*** as shown in the screenshot.

```
 System
0xf327ac0        TCPv6    :::2869                 :::0                    LISTENING       4
 System
0xf38d2a8        TCPv4    0.0.0.0:554             0.0.0.0:0               LISTENING       3088
 wmpnetwk.exe
0xf2ae898        TCPv4    192.168.1.60:49161      200.2.126.61:443        CLOSED          1412
 svchost.exe
```

03. Can you determine the full path of the malicious process and based on the path do you think its a legitimate operating system process?

The full path of the process is **C:\Windows\system32\svchost.exe**. Based on the path it looks like a legitimate process but this process does not have any command line parameters, normally the ***svchost.exe*** processes running on a clean system has command line parameters.

```
root@kratos:~/Volatility# python vol.py -f taidoor.vmem --profile=Win7SP0x86 dlllist -p 1412
Volatility Foundation Volatility Framework 2.5
************************************************************************
svchost.exe pid:   1412
Command line : svchost.exe


Base             Size  LoadCount Path
---------- ---------- ---------- ----
0x00400000     0x5000     0xffff C:\Windows\system32\svchost.exe
0x76f60000   0x13c000     0xffff C:\Windows\SYSTEM32\ntdll.dll
0x75530000    0xd4000     0xffff C:\Windows\system32\kernel32.dll
0x75160000    0x4a000     0xffff C:\Windows\system32\KERNELBASE.dll
```

04. If this is a legitimate process, is there is anything that makes it different from the legitimate process?

Process listing shows a suspicious ***svchost.exe*** process (***pid 1412***) which was not started by ***services.exe*** but this process was started by some process with process id ***2504*** (which is terminated), whereas other legitimate ***svchost.exe*** processes were started by ***services.exe*** (pid ***448***)

```
root@kratos:~/Volatility# python vol.py -f taidoor.vmem --profile=Win7SP0x86 pslist | grep -i svchost
Volatility Foundation Volatility Framework 2.5
0x889f8308 svchost.exe            616    448     10      348      0      0 2017-03-04 09:44:57 UTC+0000
0x861d2968 svchost.exe            724    448      7      298      0      0 2017-03-04 09:44:58 UTC+0000
0x889e7d40 svchost.exe            812    448     20      442      0      0 2017-03-04 09:44:58 UTC+0000
0x86203030 svchost.exe            848    448     19      419      0      0 2017-03-04 09:44:58 UTC+0000
0x86213030 svchost.exe            880    448     43      976      0      0 2017-03-04 09:44:58 UTC+0000
0x8622d990 svchost.exe           1012    448     12      545      0      0 2017-03-04 09:44:59 UTC+0000
0x862ed908 svchost.exe           1120    448     16      371      0      0 2017-03-04 09:44:59 UTC+0000
0x864b8180 svchost.exe           1284    448     18      315      0      0 2017-03-04 09:45:00 UTC+0000
0x8597f370 svchost.exe           3040    448     11      300      0      0 2017-03-04 14:36:15 UTC+0000
0x85a53418 svchost.exe           3128    448     12      226      0      0 2017-03-04 14:36:16 UTC+0000
0x85ada458 svchost.exe           1412   2504      9      214      1      0 2017-03-04 14:38:48 UTC+0000
```

```
root@kratos:~/Volatility# python vol.py -f taidoor.vmem --profile=Win7SP0x86 pslist -p 2504
Volatility Foundation Volatility Framework 2.5
ERROR   : volatility.debug    : Cannot find PID 2504. If its terminated or unlinked, use psscan and then
supply --offset=OFFSET
root@kratos:~/Volatility#
```

```
root@kratos:~/Volatility# python vol.py -f taidoor.vmem --profile=Win7SP0x86 pslist -p 448
Volatility Foundation Volatility Framework 2.5
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start
  Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ----------------------------
- ------------------------------
0x88980d40 services.exe           448    396      9      221      0      0 2017-03-04 09:44:57 UTC+0000
```

Running the malfind plugin shows that the address ***0x400000*** where ***svchost.exe*** is loaded has suspicious memory protection "***PAGE_EXECUTE_READWRITE***" if an executable is normally loaded it should have a memory protection of "***PAGE_EXECUTE_WRITECOPY***".

```
root@kratos:~/Volatility# python vol.py -f taidoor.vmem --profile=Win7SP0x86 malfind -p 1412
Volatility Foundation Volatility Framework 2.5
Process: svchost.exe Pid: 1412 Address: 0x400000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 5, 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 f0 00 00 00   ................
```

05. Dump the process onto disk, can you recognize any interesting strings?

Dumping the process from memory to disk and extracting the strings show references to the additional C2 domains which can be used as network indicator, apart from that strings also contain references to some of the http patterns used by the malware and references to the file name.

```
iphlpapi.dll
211.232.98.9
128.91.197.123
200.2.126.61
/%s.php?id=%06d%s&ext=%s
http://%s:%d/%s.php?id=%06d%s&ext=%s
%temp%\
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
http://%s:%d/%s.php?id=%06d%s
%c%c%c%c%c%c
%systemroot%\system32\sprxx.dll
/%s.php?id=%06d%s
%%temp%%\%u
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Content-Type: application/x-www-form-urlencoded
POST
HTTP/1.1
%02X-%02X-%02X-%02X-%02X-%02X
```

```
root@kratos:~/Volatility# python vol.py -f taidoor.vmem --profile=Win7SP0x86 procdump -p 1412 -D dum
p/
Volatility Foundation Volatility Framework 2.5
Process(V) ImageBase  Name                 Result
---------- ---------- -------------------- ------
0x85ada458 0x00400000 svchost.exe          OK: executable.1412.exe
```

Submitting the dumped executable to VirusTotal confirms it to be malicious as shown in the screenshot.

Analysis | File detail | Additional information | Comments 0 | Votes | Behavioural information

Antivirus	Result	Update
Ad-Aware	Generic.Malware.Fdld!.05C5C271	20170304
AhnLab-V3	Trojan/Win32.Agent.C74807	20170304
ALYac	Generic.Malware.Fdld!.05C5C271	20170304
Antiy-AVL	Trojan[Downloader]/Win32.Rubinurd	20170304
Arcabit	Generic.Malware.Fdld!.05C5C271	20170304
Avast	Win32:Malware-gen	20170304

06. Based on your observation, what code injection technique malware is using?

Looking at the parent-child relationship and the anomaly in the memory protection of executable section of ***svchost.exe*** suggests that this svchost.exe was started by another process (not ***services.exe***) and then the executable section was replaced with malicious executable. This suggests the use of hollow process injection (also called as process hollowing)