Lab 13 Solutions - The Case of Spybot

Lab 13: The Case of Spybot

While you are reading an article on malware called '***spybot***' you find that spybot creates a mutex (mutant) "***krnel***" to mark its presence. Analyze the memory image (***spybot.vmem***) and answer the below questions:

- Is the host infected with spybot?
- What is the malicious process id?
- What is the name of the malicious process?
- Which other process is related to the malicious process and what is its pid?

Answers

01. Is the host infected with spybot?

Looking for the handles to the mutex shows the presence of mutex "***krnel***" on the system since this is mutex is associated with spybot the system is possibly infected.

```
File  Edit  View  Search  Terminal  Help
root@kratos:~/Volatility# python vol.py -f spybot.vmem handles -t Mutant | grep -i krnel
Volatility Foundation Volatility Framework 2.5
0x8173df48   1700       0x50   0x1f0001 Mutant          krnel
root@kratos:~/Volatility#
```

02. What is the malicious process id?

The process with process id ***1700*** has acquired the handle to the mutex as shown in the previous screenshot. This process probably created this mutex.

```
root@kratos:~/Volatility# python vol.py -f spybot.vmem pslist -p 1700
Volatility Foundation Volatility Framework 2.5
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------
0x81754020 wuaumqr.exe             1700   1676      4       37      0      0 2014-10-22 17:09:32 UTC+0000
```

03. What is the name of the malicious process?

The process id ***1700*** is associated with a process "***wuaumqr.exe***" as shown in the screenshot

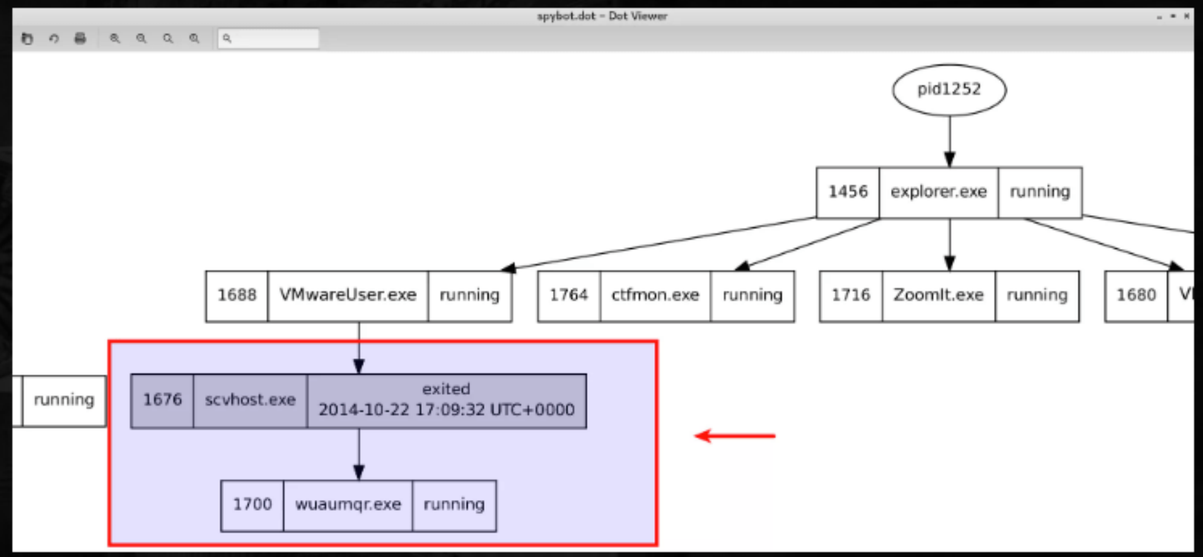
04. Which other process is related to the malicious process and what is its pid?

Running the ***pstree*** plugin shows that ***wuaumqr.exe*** process was created by another suspicious process "***scvhost.exe***" (***pid 1676***), the reason this process is suspicious is that its name is similar to the legitimate process "***svchost.exe***".

```
0x81387710:explorer.exe                1456   1252     15    436 2014-06-11 14:49:55 UTC+0000
. 0x8173b850:VMwareUser.exe            1688   1456      8    214 2014-06-11 14:49:56 UTC+0000
.. 0x8175a020:scvhost.exe              1676   1688      0 ------ 2014-10-22 17:09:32 UTC+0000
... 0x81754020:wuaumqr.exe             1700   1676      4     37 2014-10-22 17:09:32 UTC+0000
. 0x81612b28:GrooveMonitor.e           1708   1456      2    108 2014-06-11 14:49:56 UTC+0000
```

The parent-child relationship can also be determined using ***psscan*** plugin; this can be done by dumping in the dot format as shown below. The below screenshot shows the relationship between these processes.

```
root@kratos:~/Volatility# python vol.py -f spybot.vmem psscan --output=dot --output-file=spybot.dot
Volatility Foundation Volatility Framework 2.5
Outputting to: spybot.dot
root@kratos:~/Volatility#
```



Lab 13.1 Solutions - The Case of Spybot (contd.)

Lab 13.1: The Case of Spybot (contd.)

Use the memory image (***spybot.vmem***)

- Can you identify the C2 IP address?
- What port/protocol is the malware using for communication?

Answers

01. Can you identify the C2 IP address?

Running the connscan plugin shows the communication to the C2 IP made by the process with pid ***1700*** (wuaumqr.exe). The c2 IP is ***209.126.201.22***

```
root@kratos:~/Volatility# python vol.py -f spybot.vmem connscan
Volatility Foundation Volatility Framework 2.5
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------ ---
0x01949690 192.168.1.100:1033       209.126.201.22:6666      1700
root@kratos:~/Volatility#
```

02. What port/protocol is the malware using for communication?

The communication to the C2 IP was made on port **6666**, this port is associated with IRC protocol. IRC communication was used by malwares to take command from the attackers.

Lab 13.2 Solutions - The Case of Spybot (contd.)

Lab 13.2: The Case of Sypbot (contd.)

From the memory image (***spybot.vmem***)

- Dump the malicious process to the disk?
- Can you confirm if the dumped process is associated with spybot?
- Can you confirm if the malware is using IRC for its communication?

Answers

01. Dump the malicious process to the disk?

The malicious executable can be dumped from the memory to disk using the "***procdump***" plugin as shown below

```
root@kratos:~/Volatility# python vol.py -f spybot.vmem connscan
Volatility Foundation Volatility Framework 2.5
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------ ---
0x01949690 192.168.1.100:1033       209.126.201.22:6666       1700
root@kratos:~/Volatility#
```

02. Can you confirm if the dumped process is associated with spybot?

Submitting the dumped executable to VirusTotal confirms the sample to be associated with Spybot as shown in the screen shot below

Antivirus	Result	Update
Ad-Aware	Generic.Keylogger.2.98176F51	20160618
AegisLab	Backdoor.W32.Ircbot!c	20160618
AhnLab-V3	Win32/IRCBot.worm.Gen	20160617
ALYac	Generic.Keylogger.2.98176F51	20160618
Antiy-AVL	Worm[P2P]/Win32.SpyBot	20160618
Arcabit	Generic.Keylogger.2.98176F51	20160618
Avast	Win32:IRCBot-SQ [Trj]	20160618
AVG	Worm/Spybot	20160618
Avira (no cloud)	TR/Drop.Agent.CR	20160617
AVware	BehavesLike.Win32.Malware.ssc (mx-v)	20160618
Baidu	Win32.Trojan.WisdomEyes.151026.9950.9998	20160618
Baidu-International	Backdoor.Win32.IRCBot.gen	20160614
BitDefender	Generic.Keylogger.2.98176F51	20160618

03. Can you confirm if the malware is using IRC for its communication?

Extracting the strings from the dumped executable shows references to IRC commands as shown below. This indicates the use of IRC

```
PRIVMSG
KICK
PART
NICK
NICK %s
JOIN %s
JOIN %s %s
PONG %s
PING
Found: %i files and %i dirs
</PRE></HTML>
PRIVMSG %s :Found %i files and %i dirs
%s  (%i bytes)
<p><A href="%s%s">%s</A> (%i bytes)
PRIVMSG %s :%s (%i bytes)
<%s>
<li><A href="%s%s/">%s</A></li> <b><u>(Directory)</b></u>
PRIVMSG %s :[%s]
<li><A href="%s">Parent Directory</A></li>
Searsing for: %s
<HTML><PRE>
PRIVMSG %s :Searsing for: %s
PRIVMSG %s :%s
PRIVMSG %s :(%s)
```