ZLAB

The stealth process injection of the new Ursnif malware

Malware Analysts:

Antonio Pirozzi
Antonio Farina
Luigi Martire

CSE CyberSec Enterprise SPA
Via G.B. Martini 6 Italy 00100, Roma - Italia
Email: info@csecybsec.com
Website: www.csecybsec.com

11/01/18

Introduction

Whereas the malware LockPos, famous for its new incredibly advanced and sophisticated evasion technique, spread and affected many Points of Sale, another variant spread in the wild and adopts a similar but not identical advanced evasion trick. It is likely a new variant of “ursnif v3”, another evolution of an old banking trojan that was spreading since November 2017. Moreover, the command and control of this new malware, oretola[.]at has been sinkholed by authorities, so it is difficult to reconstruct the entire behavior and the real purpose of this malware.

However, it is very interesting to analyze its stealth evasion technique that allows it to be invisible to many modern antivirus software. In fact, its final stage is to hide itself as a thread of “explorer.exe” process and this make the analysis very difficult. To reach its goal, the malware uses a sort of “***double process hollowing***” technique based on Windows Native API, leveraging the “svchost.exe” system process as a way to make privilege escalation and to get to inject malicious code in “explorer.exe”.

Only after the concealment in “explorer.exe” it starts to make its malicious operations that consist of contacting a series of compromised sites the host encrypted additional payloads. The final step of its malicious behavior is to periodically communicate with its C2C, “oretola[.]at”, where it sends information about the victim host.

This malware probably spreads up through spam mails, the message contains an URL that points to a compromised site on which the sample is hosted. We discovered the malware sample just on one of these compromised sites, in particular it is an Italian blog dedicated to dolls “marinellafashiondolls[.]com/_private/php3.exe”.

```
DNS      70 Standard query 0xcb91 A dmclain.ca
DNS      86 Standard query response 0xcb91 A dmclain.ca A 10.10.10.4
DNS      72 Standard query 0x265e A sahara.to.it
DNS      88 Standard query response 0x265e A sahara.to.it A 10.10.10.4
DNS      78 Standard query 0x56a8 A longegamaurizio.it
DNS      94 Standard query response 0x56a8 A longegamaurizio.it A 10.10.10.4
DNS      72 Standard query 0xb0fd A agriweek.com
DNS      88 Standard query response 0xb0fd A agriweek.com A 10.10.10.4
DNS      81 Standard query 0x15e9 A secondglancedesign.ca
DNS      97 Standard query response 0x15e9 A secondglancedesign.ca A 10.10.10.4
DNS      70 Standard query 0xac93 A incomes.at
DNS      86 Standard query response 0xac93 A incomes.at A 10.10.10.4
DNS      81 Standard query 0xcf1f A resolver1.opendns.com
DNS      97 Standard query response 0xcf1f A resolver1.opendns.com A 10.10.10.4
DNS      76 Standard query 0x0002 A myip.opendns.com
DNS      92 Standard query response 0x0002 A myip.opendns.com A 10.10.10.4
DNS      76 Standard query 0x0003 AAAA myip.opendns.com
DNS      76 Standard query response 0x0003 AAAA myip.opendns.com
DNS      72 Standard query 0x4b4b A curlmyip.net
DNS      88 Standard query response 0x4b4b A curlmyip.net A 10.10.10.4
DNS      70 Standard query 0xcf27 A mogolik.at
DNS      86 Standard query response 0xcf27 A mogolik.at A 10.10.10.4
DNS      70 Standard query 0xa1d6 A oretola.at
DNS      86 Standard query response 0xa1d6 A oretola.at A 10.10.10.4
```

Figure 1 - List of some domains resolved by the malware

Technique

First of all, this malware uses almost exclusively the Native API of Windows with also its undocumented functions. The use of them causes a more difficult monitoring by antiviruses.

Once the php3.exe file is executed, it deletes itself from the original path and recopy itself in "%APPDATA%\Roaming\Microsoft\Brdgplua\ddraxpps.exe" path.

Once completed this operation, the malware starts its malicious behavior, summarizing in these phases:

1. Create a new "svchost.exe" process in suspended mode, using CreateProcessA.

Figure 2 - svchost.exe process creation

CreateProcessA | Kernel32.dll

Module: KERNELBASE.dll
Process: (ddraxpps.exe)

Process ID: 2376 Kill
Thread ID: 2584 Kill

#	Type	Name	Value
1	LPCTSTR	lpApplicationName	NULL
2	LPTSTR	lpCommandLine	0x03158d18 "C:\Windows\system32\svchost.exe"
3	LPSECURITY_AT...	lpProcessAttributes	NULL
4	LPSECURITY_AT...	lpThreadAttributes	NULL
5	BOOL	bInheritHandles	FALSE
6	DWORD	dwCreationFlags	CREATE_DEFAULT_ERROR_MODE | CREATE_SUSPENDED
7	LPVOID	lpEnvironment	NULL
8	LPCTSTR	lpCurrentDirectory	NULL
9	LPSTARTUPINFO	lpStartupInfo	0x0018fe70 = { cb = 68, lpReserved = NULL, lpDesktop = NULL ...}
10	LPPROCESS_IN...	lpProcessInformation	0x0018feb8 = { hProcess = 0x000000fc, hThread = 0x000000f8, dwProcess...
	BOOL	Return	TRUE

Figure 3 - Parameters of CreateProcessA

2. Create a new thread of “explorer.exe” process in suspended mode using OpenProcess with PROCESS_CREATE_THREAD and PROCESS_SUSPEND_RESUME flags enabled.

Parameters: OpenProcess (Kernel32.dll)

#	Type	Name	Pre-Call Value
1	DWORD	dwDesiredAccess	STANDARD_RIGHTS_ALL | PROCESS_CREATE_PROCESS | PROCESS_CREATE_THREAD |
2	BOOL	bInheritHandle	FALSE
3	DWORD	dwProcessId	2672

ddraxpps.exe	2140
svchost.exe	3448
explorer.exe	2672
procexp64.exe	3044

Figure 4 - Creation of a new thread of explorer.exe process (PID 2672) in suspended mode

3. Create a new section in memory in which it is loaded the code to map in “svchost.exe” process.

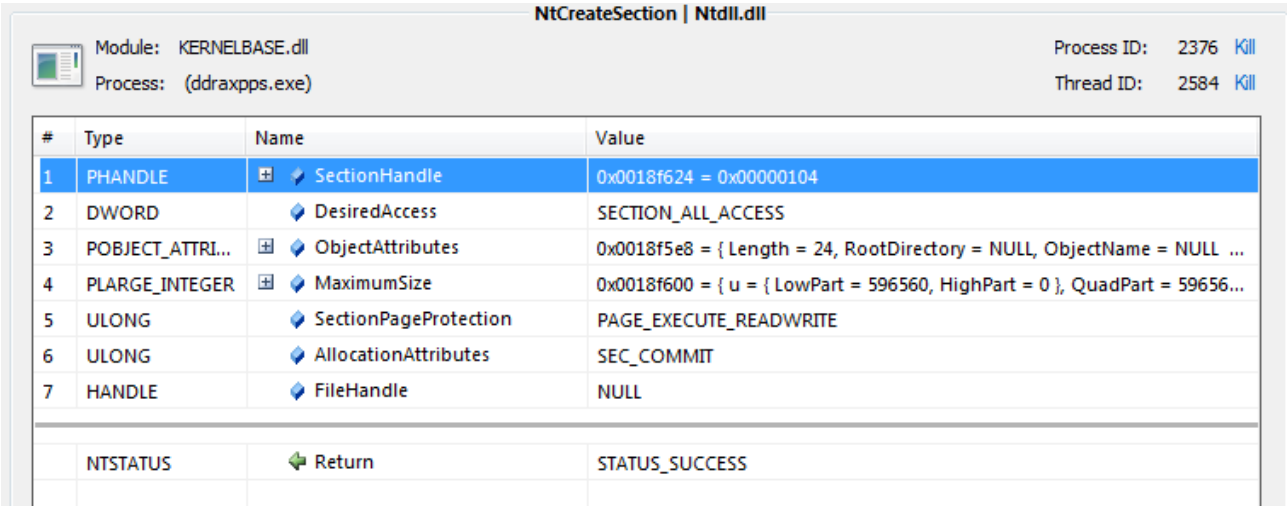

Figure 5 - Section creation

At this moment, the section is empty and it will be filled in the next step

4. Copy the payload into the previous section using “memcpy” function

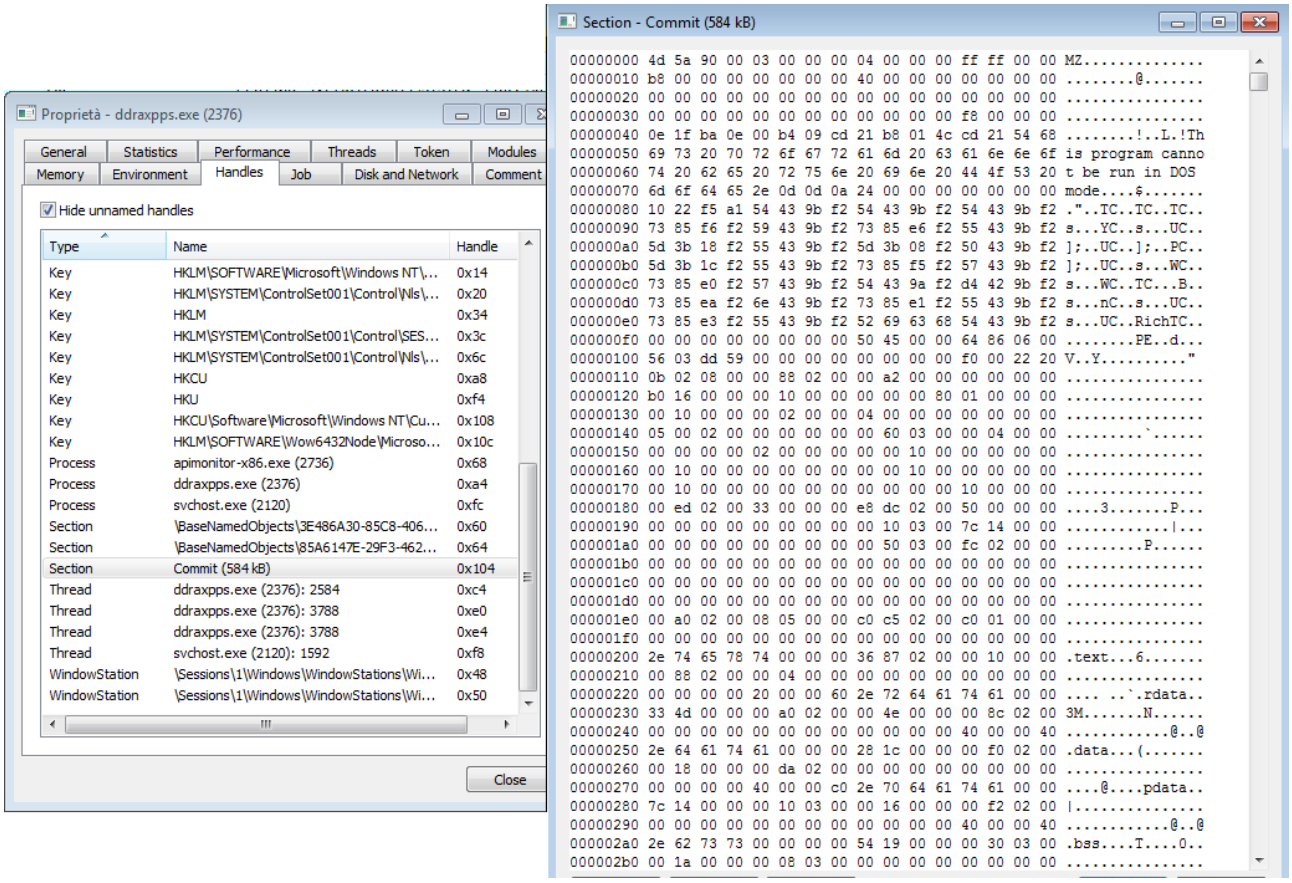

Figure 6 - Payload's copy in the section previously created through memcpy function

CSE CyberSec Enterprise SPA
Via G.B. Martini 6 Italy 00100, Roma - Italia
Email: info@csecybsec.com
Website: www.csecybsec.com

5. Map the filled section to “svchost.exe” process using the Windows Native API function NtMapViewOfSection.

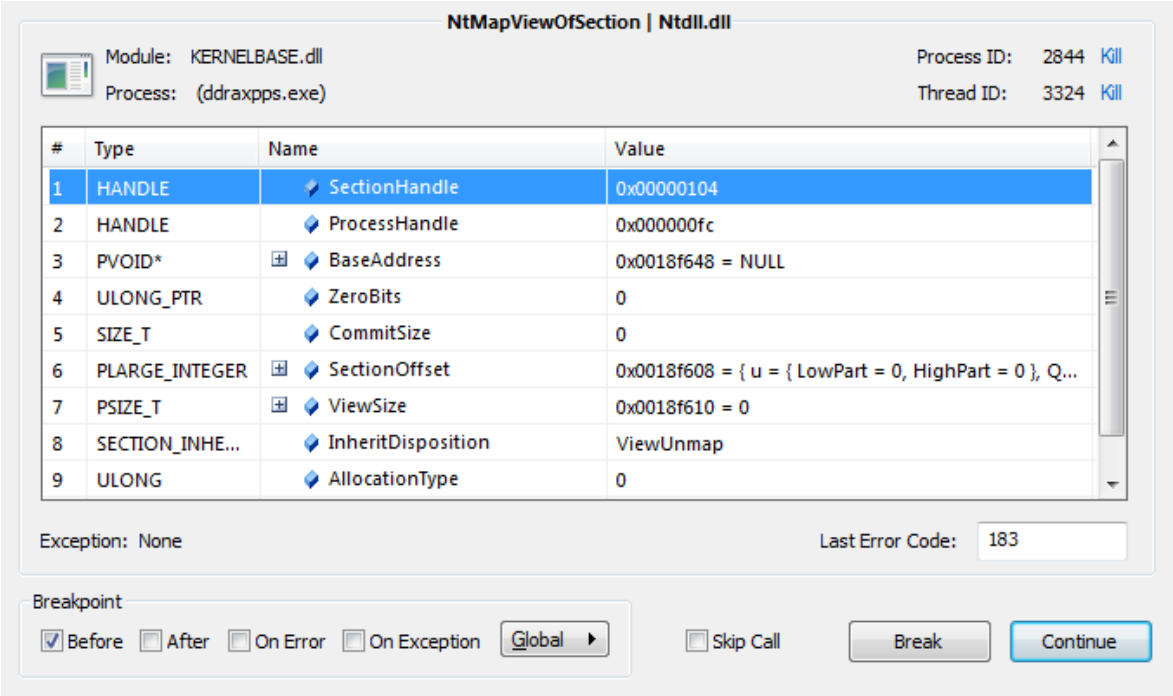

Figure 7 - Mapping of the previously filled section to svchost.exe process through NtMapViewOfSection

6. Resume “svchost.exe” thread in order to act in the section previously allocated.

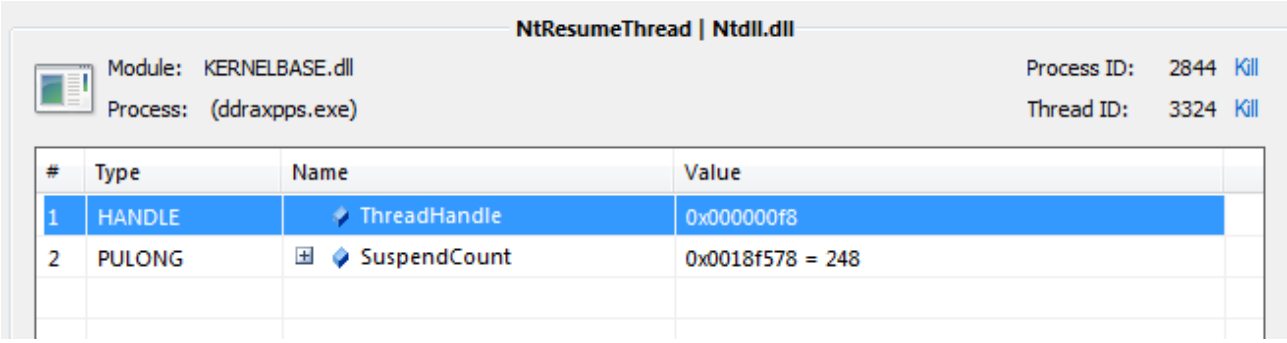

Figure 8 - svchost.exe resuming in order to execute the payload loaded into the section

7. Exit

After this step, we lose the control of the behavior, because “svchost.exe” is a system process and we are not able to monitor the activities performed by it. But we can see that

- Both malicious “svchost.exe” and its father “ddraxpps.exe” terminate

- “explorer.exe” process start to have a malicious behavior, in particular it generates internet traffic to compromised websites.

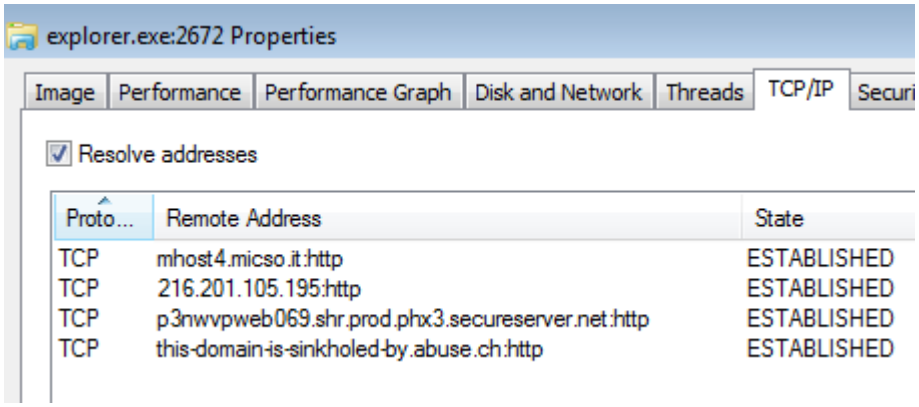

Figure 9 - Abnormal traffic performed by explorer.exe process

Thus, we can deduce with a good confidence that the effective payload is injected in “explorer.exe” thread and “svchost.exe” is only a proxy used to transfer the malicious code into the explorer process in order to make stealthier the malware execution. In fact, it is highly likely that “svchost.exe” performs the same actions viewed above to reach its goal. It seems that the first stage of process hollowing is used to perform a privilege escalation, starting from a user-space project to a system one; the second stage is to totally hide the payload to a user.

In conclusion, in this malware analysis the real challenging part was reversing this absolutely unusual and powerful hiding technique. In fact, it’s true that lots of sophisticated malware adopt process hollowing for conceal themselves, but not this two-step version. The malware adopts the principles of privilege escalation and process hollowing, and make the analysis very hard.

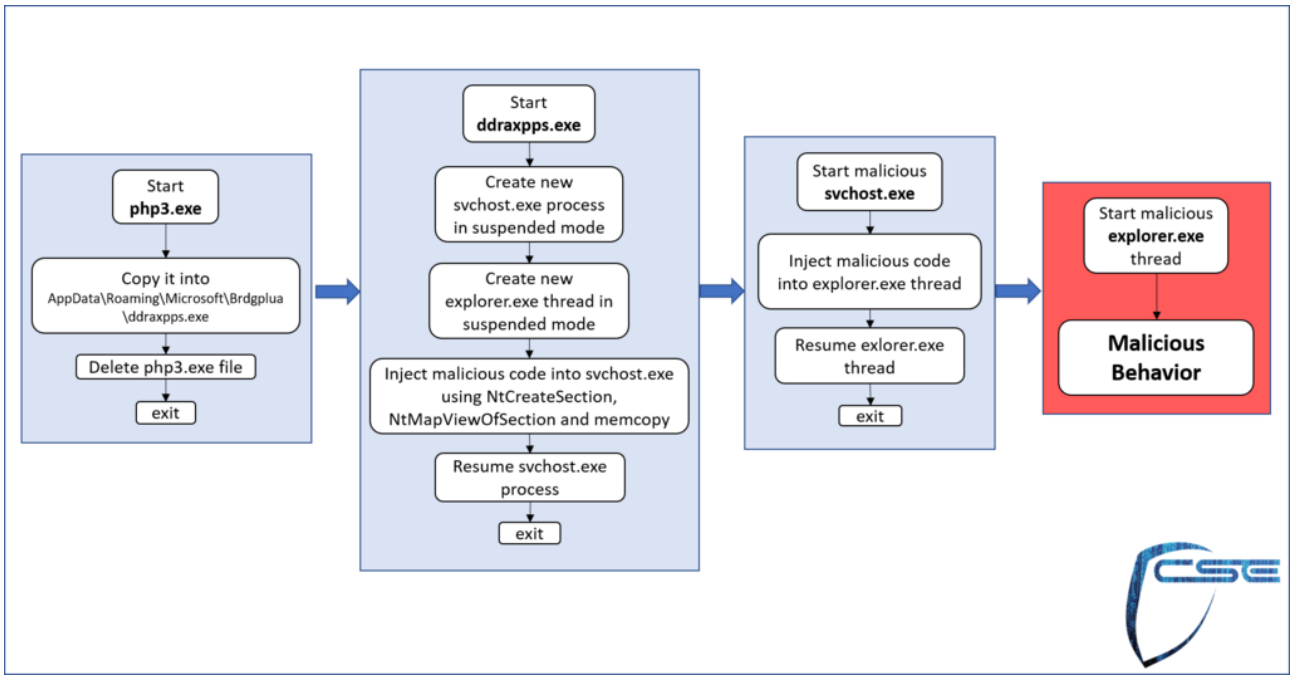

Figure 10 - Double Process Hollowing used by the malware.