# ZLAB

The stealth process injection of the new Ursnif malware



Malware Analysts:

Antonio Pirozzi Antonio Farina Luigi Martire



### 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.

ddraxpps.exe		6.596 K	11.400 K 2376
svchost.exe	Suspended	336 K	260 K 2120 Microsoft Corporation

Figure 2 - svchost.exe process creation



	CreateProcessA   Kernel32.dll						
	Module: KERNEL	BASE.dll	Process ID: 2376				
	Process: (ddrax)	pps.exe)	Thread ID: 2584				
#	Туре	Name	Value				
1	LPCTSTR	🗉 🧳 IpApplicationName	NULL				
2	LPTSTR	🗄 🧼 IpCommandLine	0x03158d18 "C:\Windows\system32\svchost.exe"				
3	LPSECURITY_AT	🗄 🧳 IpProcessAttributes	NULL				
4	LPSECURITY_AT	🗄 🧳 IpThreadAttributes	NULL				
5	BOOL	bInheritHandles	FALSE				
6	DWORD	dwCreationFlags	CREATE_DEFAULT_ERROR_MODE   CREATE_SUSPENDED				
7	LPVOID	IpEnvironment	NULL				
8	LPCTSTR	🗄 🧳 IpCurrentDirectory	NULL				
9	LPSTARTUPINFO	🗄 🧳 IpStartupInfo	0x0018fe70 = { cb = 68, IpReserved = NULL, IpDesktop = NULL}				
10	LPPROCESS_IN	IpProcessInformation	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.

Pa	ameters: OpenProce	ss (Kernel32.dll)	ddraxpps.exe	2140	
#	Туре	Name	Pre-Call Value	svchost.exe	3448
1	DWORD	dwDesiredAccess	STANDARD_RIGHTS_ALL   PROCESS_CREATE_PROCESS   PROCESS_CREATE_THREAD	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	2672
2	BOOL	bInheritHandle	FALSE	≫procexp64.exe	3044
3	DWORD	dwProcessId	2672		

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.



#	Туре	Name	Value
1	PHANDLE	🗉 🧳 SectionHandle	0x0018f624 = 0x00000104
2	DWORD	DesiredAccess	SECTION_ALL_ACCESS
3	POBJECT_ATTRI	🗉 🧳 ObjectAttributes	0x0018f5e8 = { Length = 24, RootDirectory = NULL, ObjectName = NULL .
4	PLARGE_INTEGER	🗄 🧼 MaximumSize	0x0018f600 = { u = { LowPart = 596560, HighPart = 0 }, QuadPart = 59656.
5	ULONG	SectionPageProtection	PAGE_EXECUTE_READWRITE
6	ULONG	AllocationAttributes	SEC_COMMIT
7	HANDLE	FileHandle	NULL

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

			💽 Section - Commit (584 kB)
Proprietà - ddra: General Stat	rpps.exe (2376) istics Performance Threads Toke	n Modules	00000000 4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00 MZ
Memory Enviro	nment Handles Job Disk and Netwo	ork Comment	00000060 74 20 62 65 20 72 75 6e 20 69 6e 20 44 4f 53 20 t be run in DOS
Hide unnamed	handles		00000000 10 22 f5 a1 54 43 9b f2 54 43 9b f2 54 43 9b f2TCTCTC
Turpe	Name	Handle 🔺	00000090 73 85 f6 f2 59 43 9b f2 73 85 e6 f2 55 43 9b f2 sYCsUC
Type	Name	nariue	00000000 5d 3b 1c 12 55 43 9b 12 5d 3b 0c 12 50 43 9b 12 j;UCsWC
Key	HKLM\SOFTWARE\Microsoft\Windows NT\	0x14	000000c0 73 85 e0 f2 57 43 9b f2 54 43 9a f2 d4 42 9b f2 sWCTCB
Key	HKLM\SYSTEM\ControlSet001\Control \Wls\	0x20	000000d0 73 85 ea f2 6e 43 9b f2 73 85 e1 f2 55 43 9b f2 snCsUC
Key	HKLM	0x34	000000e0 73 85 e3 f2 55 43 9b f2 52 69 63 68 54 43 9b f2 sUCRichTC
Key	HKLM\SYSTEM\ControlSet001\Control\SES	0x3c	000000f0 00 00 00 00 00 00 00 00 50 45 00 00 64 86 06 00PEd
Key	HKLM\SYSTEM\ControlSet001\Control\Vis\	0x6c	00000100 56 03 dd 59 00 00 00 00 00 00 00 00 f0 00 22 20 VY
Key	HKCU	0xa8	00000110 0b 02 08 00 00 88 02 00 00 a2 00 00 00 00 00 00
Key	HKU	0xf4	00000120 b0 16 00 00 00 10 00 00 00 00 00 80 01 00 00 00
Key	HKCU\Software\Microsoft\Windows NT\Cu	0x108	00000130 00 10 00 00 00 02 00 00 04 00 00 00 00 00 00 00
Key	HKLM\SOFTWARE\Wow6432Node\Microso	0x10c	
Process	apimonitor-x86.exe (2736)	0x68	
Process	ddraxpps.exe (2376)	0xa4	
Process	sychost.exe (2120)	0xfc	00000180 00 ed 02 00 33 00 00 00 e8 dc 02 00 50 00 00 00 00
Section	\BaseNamedObjects\3E486A30-85C8-406	0x60	
Section	\BaseNamedObjects\8546147E-29E3-462	0x64	000001a0 00 00 00 00 00 00 00 00 00 50 03 00 fc 02 00 00
Section	Commit (584 kB)	0x104	000001b0 00 00 00 00 00 00 00 00 00 00 00 00 0
Thread	ddravpos eve (2276): 2584	0xc4	000001c0 00 00 00 00 00 00 00 00 00 00 00 00 0
Thread	ddraxpps.exe (2376): 2304	0xe4	000001d0 00 00 00 00 00 00 00 00 00 00 00 00 0
Thread	ddraxpps.exe (2276): 3788	0xe0	000001e0 00 a0 02 00 08 05 00 00 c0 c5 02 00 c0 01 00 00
Thread	uuraxpps.exe (2376): 3786	0x64	000001f0 00 00 00 00 00 00 00 00 00 00 00 00 0
meau Weder Challen	SVCHOSt. EXE (2120): 1592	0x10	00000200 2e 74 65 78 74 00 00 00 36 87 02 00 00 10 00 00 .text6
windowStation	Sessions (1) windows (windows tations (wi	0x48	
windowstation	Sessions (1) windows (windows tations (wi	0x50 -	00000220 00 00 00 00 20 00 00 80 22 72 64 61 74 61 00 00
•		•	
			00000250 2e 64 61 74 61 00 00 00 28 1c 00 00 00 f0 02 00 .data(
			00000260 00 18 00 00 00 da 02 00 00 00 00 00 00 00 00 00
		Close	00000270 00 00 00 00 40 00 00 c0 2e 70 64 61 74 61 00 00@pdata
			00000280 7c 14 00 00 00 10 03 00 00 16 00 00 00 f2 02 00
			00000290 00 00 00 00 00 00 00 00 00 00 00 00 40 00 0
			000002a0 2e 62 73 73 00 00 00 00 54 19 00 00 00 30 03 00 .bssT0
			000002b0 00 1a 00 00 00 08 03 00 00 00 00 00 00 00 00 00

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



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

ŧ	Туре	Name	Value	-
1	HANDLE	SectionHandle	0x0000104	
2	HANDLE	ProcessHandle	0x000000fc	
3	PVOID*	🗄 🧼 BaseAddress	0x0018f648 = NULL	
4	ULONG_PTR	ZeroBits	0	Ξ
5	SIZE_T	CommitSize	0	
6	PLARGE_INTEGER	🗄 🧼 SectionOffset	0x0018f608 = { u = { LowPart = 0, HighPart = 0 }, Q	
7	PSIZE_T	🗄 🧼 ViewSize	0x0018f610 = 0	
8	SECTION_INHE	InheritDisposition	ViewUnmap	
9	ULONG	AllocationType	0	-

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.

		NtResumeThre	ad   Ntdil dil			
	Module: KERNELI Process: (ddraxp	BASE.dll pps.exe)		Process ID: Thread ID:	2844 3324	Kill Kill
#	Туре	Name	Value			
1	HANDLE	ThreadHandle	0x000000f8			
2	PULONG	🗉 🧼 SuspendCount	0x0018f578 = 248			

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.

📄 ex	plorer.e	xe:2672 Pr	operties				
Ima	ge Per	formance	Performance Graph	Disk and Network	Threads	TCP/IP	Securi
	Resolve addresses						
	Proto Remote Address					State	
TCP mnost4.micso.it.nttp E: TCP 216.201.105.195:http E:					ESTABLIS	SHED	
T	TCP p3nwvpweb069.shr.prod.phx3.secureserver.net:http					ESTABLIS	SHED
T	CP	this-domai	n-is-sinkholed-by.abus	e.ch:http	1	ESTABLIS	SHED

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.

