

Sebdraven Follow

OSINT, Python,Malware Analysis, Botnet Tracker, SIEM and IPS/IDS and Threats Expert / coorganizer #BotConf / co-creator of #FastIR Aug 2 · 4 min read

Goblin Panda against the Bears

During my last investigation (<u>here</u>), I've found two RTFs malware documents with the same techniques of exploitation of CVE-2017–11882:

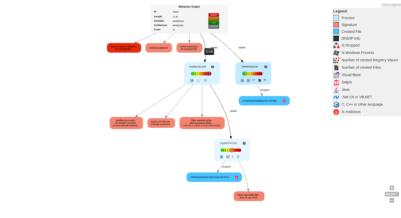
A file 8.t in %TMP% with Package Ole Object

The same loop of decryption

The same runPE after overwriting in memory EQNEDT32.exe

But the payload is really different. It's not a version of PlugX but a version of Sisfider studied by Ncc group. <u>https://www.nccgroup.trust/uk/about-us/newsroom-and-</u> events/blogs/2018/june/cve-2017-8570-rtf-and-the-sisfader-rat/

With the behaviour graph of Joe Sandbox, we can recognize the same interactions with operating system than my last article and the paper of NCC Group.



Behaviour of malwares

The difference with the version studied by NCC Group is the Package Ole Object. In the article of NCC Group, the researchers talk about a SCT File and many javascript manipulations for dropping the RAT on the disk and to start it. Here, the payload is encrypted in 8.t file

If we analyze EQNEDT32.exe overwritten to recognise the payload, we have the same technics anti emulation with the same value.

In a thread, the process posts in a queue the value 5ACE8D0Ah.

	; Attributes: bp-based frame
	, ACCIDACCS. Sp-Sabou Hamo
	; DWORDstdcall StartAddress(LPVOID lpThreadParameter)
	StartAddress proc near
	hWnd= dword ptr 8
	push ebp
	mov ebp, esp
	* *
	loc_401A23:
	mov eax, 1 test eax, eax
	test eax, eax jz short loc_401A51
🔲 🚄 🖗	
push	1 : bAlertable
push push	1 ; bAlertable 7D0h ; dwMilliseconds loc_401A51:
push push call	7D0h ; dwMilliseconds loc_401A51: ds:SleepEx pop ebp
push push call push	7D0h ; dwMilliseconds loc_401A51: ds:SleepEx pop ebp 0 ; lParam retn
push push call push push	7D0h ; dwMilliseconds loc_401A51: ds:SleepEx pop ebp 0 ; lParam retn 4 SACE8D0Ah ; wParam StartAddress endp
push push call push	7D0h ; dwMilliseconds loc_401A51: pop ebp 0 ; lParam retn 4 3ACE8D0Ah ; wParam StartAddress endp 500h ; Msg
push push call push push push mov	7D0h ; dwMilliseconds loc_401A51: ds:SleepEx pop ebp 0 ; lParam retn 5ACESDOAh ; wParam StartAddress endp 500h ; Msg
push push call push push mov push	7D0h ; dwMilliseconds ds:SleepEx loc_401A51: 0 ; lParam 5ACE8D0Ah ; wParam 500h ; Msg ecx, [ebp+hWnd] startAddress endp
push push call push push mov push call	7D0h ; dwMilliseconds ds:SleepEx ; 0 ; lParam ; retn 4 5ACE8D0Ah ; wParam ; 500h ; kMsg ecx, [ebp+hWnd] ecx ; hWnd ds:PostMessageA
push push call push push mov push	7D0h ; dwMilliseconds ds:SleepEx loc_401A51: 0 ; lParam 5ACE8D0Ah ; wParam 500h ; Msg ecx, [ebp+hWnd] startAddress endp

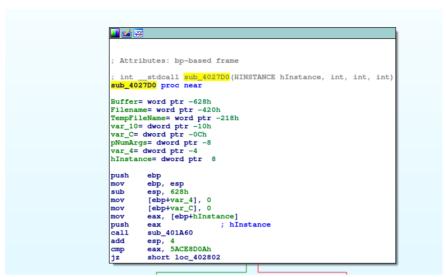
Anti emulation tricks

	•		
🗾 🚄 🖼			
loc_401	B7E:	;	nCmdShow
push	5		
mov	ecx, [ebp+hWnd]		
push	ecx	;	hWnd
call	ds:ShowWindow		
mov	edx, [ebp+hWnd]		
push	edx	;	hWnd
call	ds:UpdateWindow		
push	0	;	lpThreadId
push	0	7	dwCreationFlags
mov	eax, [ebp+hWnd]		
push	eax		lpParameter
-	offset <mark>StartAdd</mark>		<mark>ss</mark> ; lpStartAddress
push	0	7	dwStackSize
push	0	7	lpThreadAttributes
call	ds:CreateThread		
mov	eax, 1		

Anti emulation tricks

The verification is calling GetMessage() and the value is stored in EAX in the function sub_401A60.

The comparaison is made in the calling function sub_4027D0.

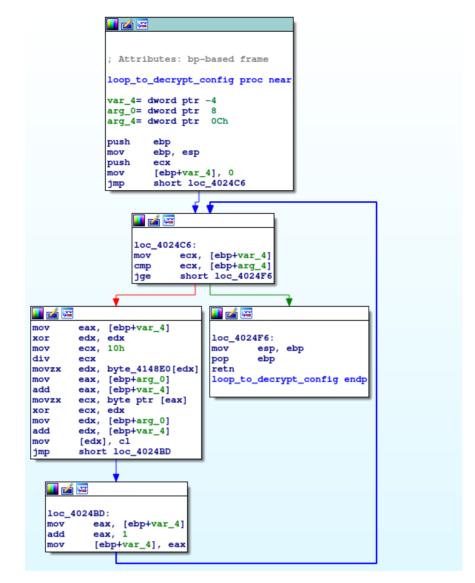


Anti emulation tricks verification

Juste after we found again the loop of decryption for the config.

📕 🗾 🖥	
1oc_40	2820:
push	160h
push	offset Data
call	<pre>loop_to_decrypt_config</pre>
add	esp, 8
push	104h ; nSize
lea	<pre>eax, [ebp+Filename]</pre>
push	eax ; lpFilename

call to loop of decryption



Loop of decrypting config

It's the same algorithm described: a simple XOR loop with rolling key.

The mechanism of persistent is the same with a service creation just after dropping differents files and a privilege escalation.

Ý Ý
loc_601448: nov eax, [bp+1pServiceName] push eax ; lpServiceName call check_service description ; 4 beer eax, eax just short loc_61430
V ico_401450: puch edd.; (abp+oNumberOfBytesTOWcite] puch edd.; (abp+isDutter) puck edd.; (abp+1sDutter) puck edd.; (abp+1sDutter) puck edd.; (abp+1sDutter) puck edd.; (abp+1sDutter) call dropfile edd.; (abp+1sDutter) call dropfile edd.; (abp+1sDutter) call dropfile edd.; (abp+1sDutter) call dropfile edd.; (abp+1sDutter) test edd.; (abp+1sDutter) test edd.; (abp+1sDutter) test edd.; (abp+1sDutter)
Implies loc_60177A: mor eas: eas: borrion: eas: borrion:

We found the same name of the dll files.

	•				•	
1 🚄 🖼			📕 🚄 🔛			
ea ecx, [ebp4 ush ecx all ds:lstrler ov [ebp4var_] ush offset aD] ea edx, [ebp4 ush edx all ds:lstroat ea eax, [ebp4 ush eax all ds:Deletef ov [ebp4pSer ov ecx, [ebp4 ov [ebp4cx45]	<pre>; jpString), sappngs ; "\dlaach\\s Buffer] ; jpStringi Buffer] iba; jpFileHame iba; jpFileHame viceMame], offset aAppngmt; yurfee], 0 gmpetsDil; "\\appmgnts.dll" Buffer] Buffer] Suffer] JpStringi</pre>		loc_402 mov push lea push call lea push call add	<pre>[ebp+lpServiceName] offset aIscsiexeDll eax, [ebp+Buffer] eax ; 1 ds:lstrcatA ecx, [ebp+Buffer]</pre>	pString1 ObjectName	"MSISCSI"
1 🗹 🖂		I 📕 🚄 🖂				
oc_402794: ov edx, [ebp- ush edx ov eax, [ebp- ush eax ea ecx, [ebp- ush ecx	; lpFileName lpServiceName] ; lpServiceName eate	loc_4027B1: mov eax push eax mov ecx push ecx push off call dro	, [ebp+] set aApp	NumberOfBytesToWrite ; nNumberOfByte Buffer] ; lpBuffer uhelperDl; = \\App comobject	sToWrite	

Persistence and loading agent

The malware overwrite the comobject

{9BA05972-F6A8–11CF-A442–00A0C90A8F39} to execute when this com object is called to make a persistence

lea push push	edx	p+pszPath] ; lpString Lid_class ; "{9BA05972	-F6A8-11CF-A442-00A0C90A8F3	39}"
call	add_comob			2
🛄 🚄 😼	-			
	-		—	
1oc_40	1053-	; lpdwDisposition		
push	0	, 194851090010101		
lea	ecx, [ebp+	hKey]		
push	ecx	; phkResult		
push	0	; lpSecurityAttrib	utes	
push push	0F003Fh 0	; samDesired ; dwOptions		
push	0	; lpClass		
push	ō	; Reserved		
mov	edx, [ebp+			
push	edx	; lpSubKey		
mov	eax, [ebp+			
push call	eax	; hKey		
mov	ds: <mark>RegCrea</mark> [ebp+var_4			
cmp	[ebp+var_4	1. 0		
jz	short loc_			
			•	
		🔲 🚄 🖼		
		loc_401082:	; lpdwDisposition	
		push 0		
		lea ecx, [ebp+var		
		push ecx push 0	<pre>; phkResult ; lpSecurityAttributes</pre>	
		push 0F003Fh	; ipsecurityAttributes ; samDesired	
		push 0	; dwOptions	
		push 0	; lpClass	
		push 0	; Reserved	
			cserver32 ; "InprocServer32"	
		mov edx, [ebp+hKe		
		push edx call ds:RegCreateR	; hKey	
		mov [ebp+var_4],		
		cmp [ebp+var_4],		
		jz short loc_401		
		-		

ComObject Adding

All evidences show is the same payload Sisfader RAT.

Threat Intel

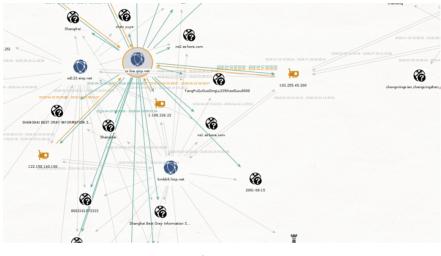
The toolset for exploiting the module of equation is the same using of the compromission for Vietnameses Officials used by Goblin Panda. (APT 1937CN) If we check the domain contacted by EQNEDT32.exe is kmbk8.hicp.net. This address is a real good pivot. It makes the link with Goblin Panda and SisFader RAT.

And the infrastructure is very interesting this domains resolved on three IPs:

122.158.140.100, 122.158.140.100 and 103.255.45.200

Theses addresses can permit to found others domains:

Sd123.eicp.net with new IP 180.131.58.9 and cv3sa.gicp.net with new IP 1.188.233.201

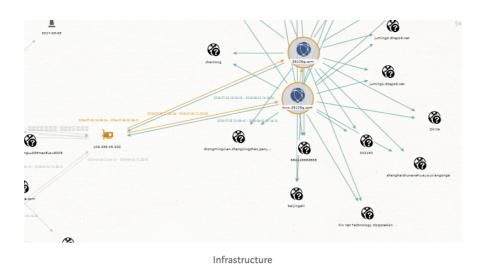


Infrastructure

The Ip Address 103.255.45.200 has two domains:

www.36106g.com

36106g.com



All infrastructure is based at Shanghai.

The victims are different than the Vietnameses campaign.

They targeted Telecom Firms pretending to be the Intelligence Service of Russia (FSB)

Buang	Condusion The hardware and software complex SOBM on TZUS / OPTS "SS000" of local telephone network LLC '' basicaly meets the technical requirements for SOBM and is recommended for commissioning in trial operation.
Алпаратно-программный комплекс ССРМ на ТЗУС/ОПТС «53000» сети местной телефонной связи СОО в сосновном соответствует техническим требованиям к СОРМ и рекомендуется к воду в опытную эксплуатацию.	Representative FSB of Russia
Продставитель ФСБРоссии 	A. 5. Kondratev 2018 Representative LTD*
«s2018 г. Представитель 000 «»	< <u></u> 2018
«2018 r.	



So Gobelin Panda targets like the report of CrowdStrike

https://go.crowdstrike.com/rs/281-OBQ-

<u>266/images/ReportGlobalThreatIntelligence.pdf</u> he telecom industries in Russia.

Conclusion

Goblin Panda used Sisfader RAT to target the Telecom Firms russian with the same exploitation techniques for Vietnameses Officials. They updated theirs technics than the report of NCC group.

IOCs:

Rtfs:

722e5d3dcc8945f69135dc381a15b5cad9723cd11f7ea20991a3ab867 d9428c7

71c94bb0944eb59cb79726b20177fb2cd84bf9b4d33b0efbe9aed58bb 2b43e9c

Domains IP:

- 1.188.233.201 cv3sa.gicp.net
- 1.188.236.22 cv3sa.gicp.net

1.188.236.22 kmbk8.hicp.net

1.188.236.22 sd123.eicp.net

103.255.45.200 36106g.com

103.255.45.200 cv3sa.gicp.net

103.255.45.200 kmbk8.hicp.net

103.255.45.200 sd123.eicp.net

103.255.45.200 www.36106g.com

122.158.140.100 cv3sa.gicp.net

122.158.140.100 kmbk8.hicp.net

122.158.140.100 sd123.eicp.net