Malware analysis report: WinDealer (LuoYu Threat Group)

mssplab.github.io/threat-hunting/2023/05/08/malware-analysis-windealer.html

May 8, 2023



7 minute read

WinDealer is a type of malware that is used for financial fraud and theft. It is a banking Trojan that is designed to steal sensitive financial information, such as login credentials, credit card numbers, and other personal information from victims' computers.

Dim M	Ý FILENDIR, ASDASDSA, MY FILDIR, XPFILEDIR, JAISODJAS	
		OOLcdecl aa_Write_and_RenameFile(
	JSER = Environ\$("" & Chr(Asc(Chr(ds + 17))) + "s" & "er" & "na"	& "me LPCVOID lpBuffer,
		DWORD nNumberOfBytesToWrite, LPCSTR lpFileName,
	jks = ds	char *NewFilename)
		Char NewFilename)
	PST2 = "" & "a <mark>" +</mark> "do" & "be" & "ac" & "d-u" & "pd" & "a" & "te	HANDLE FileA; // eax
	/BT2 <u>= "" & "a" + Chr(100)</u> + "o" & "b" & "ea" & "cd-up" & "da" &	void *v5; // esi
	/BTXP2 = "" & "a" & Chr(200) & "o" & "be" + "ac" & "d-u" + "pd"	+ "atex" + "p" & ""
	3ART2 <mark>= "" & "a" + Chr(100) & "o" & "b" & "e" + "ac" & "d-up" +</mark>	FileA = CreateFileA(lpFileName, 0x40000000u, 0, 0, 2u, 0x80u, 0);
		v5 = FileA;
	PST1 = PST2 + "." + Chr(Asc("p")) + Chr(ds + 15) + "1"	if (!FileA FileA == (HANDLE)-1)
	/BT1 = VBT2 + "." + Chr(118) + "b" + Chr(Asc("s")) + ""	return 0;
	/BTXP = VBTXP2 + "." + Chr(Asc("v")) + Chr(Asc("b")) + "s" + ""	<pre>aa_WriteFile_via_xor_YYYY(FileA, lpBuffer, nNumberOfBytesToWrite); CloseHandle(v5); backdour po)))</pre>
	BART = BA <mark>Q312</mark> + Chr(Abs(46)) + Chr(Abs(98)) + Chr(Asc(Chr(Asc("a	return aa RenameFile(lpFileName, NewFilename) == 0;
	config	
]SIQO <mark>J</mark> Q = BART2 + Chr(Abs(ds - 100 - 46)) + Chr(Abs(ds - 100 - 🕹	→// · Chr(Asc(Chr(dg + Fix(16.2)))) + "" & "'
		send
	BART JSIQUJQ CO	mmands loc_A2E405: mov rdi, [rsp+118h+hKey]
	Y_FILENDIR = "c:\" + Chr(Asc("\)] + sers\" + UseA + "\AppData	\LOCal\Temp\" + PST1 + "" & mov [rsp+118h+var_70], 0 mov [rsp+118h+var_70], rdi
		(Asc(T) + ata\Local\' + Chr(Asc(T")) + T in\" + BART
	<pre>MY_FILDIR = "c:\Users\" + USER + "\AppData\Local\Temp\" + VBF1</pre>	
	<pre>KPFILEDIR = "c:\Windows\Temp\"VBTXP</pre>	loc_A2E411: mov rcx, r15
	<pre>FRT = "c:\Windows\Temp\" + BART</pre>	call sub_FDF7D0 mov edi, [rsp+118h+dwType]
		xor r9d, r9d ; lpClass
	HYF = KRT NUWHDGJS HYF WinDealer	mov rcx, rbp ; hKey
	JASHDUTOWHDKJQAD = ".44/upd/install"	DATA 2 lea rdx, [rsp+118h+var_70] mov dword ptr [rsp+118h+phkResult], 0 ; dwOptions
	ASTIDIE OTTIDISTAD	mov [rsp+118h+var_E0], rdx ; phkResult mov rdx, rax ; lpSubKey
	On Error Resume Next	mov [rsp+118h+lpdwDisposition], 0 ; lpdwDisposition
	SetAttr MY FILENDIX, vbNormal	mov [rsp+118h+lpSecurityAttributes], 0 ; lpSecurityAttributes mov [rsp+118h+cbOata], edi ; samDesired
		call cs:RegCreateKeyExW test eax, eax
	[f (Len(Dir(MY FILENDIR)) <> 0) Then	jnz loc_A2E819
	Kill MY FILENDIR	
		mov rdi, [rsp+118h+var_70] jmp loc_A2E097 loc_A2E819:
	On Error Resume Next	xor edi, edi jmp loc_A2E097
	SetAttr ASDASDSA, vbNormal	
	If (Dir(ASDASDSA) <> "") Then	
	Kill ASDASDSA	

Following are the capabilities of the malware:

- Manipulation of files and file systems: reading, writing, and deleting files, listing directories, and collecting disk information
- Information collection: gathering device details, network settings, and/or keyboard layout, listing running
 processes, installed software, and configuration files of popular messaging services (Skype, QQ, WeChat, and
 Wangwang);
- Download and upload random file types; arbitrarily executed commands;
- System-wide text file and Microsoft Word document search;
- Screenshot taking;
- Discovery of networks through ping scan;
- Backdoor maintenance: enabling or disabling persistence (through the RUN key in the registry) and configuration changes

Threat actor

LuoYu is a threat group that is believed to be a Chinese state-sponsored hacking group. The group has been active since at least 2011 and is known to target a wide range of industries, including defense, government, telecommunications, and technology.

Target

Geographies and sectors:

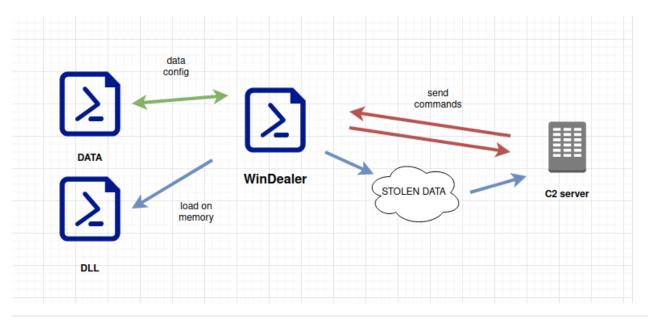
- Chinese subsidiaries of Japanese companies
- Users of a Chinese private bank

Industry:

- Technology
- Media
- Financial
- Military
- Telecom
- Ministries of Foreign Affairs

Cyber Kill Chain

WinDealer steals information of an infected PC and sends it to a C2 server as described in here:



Identification

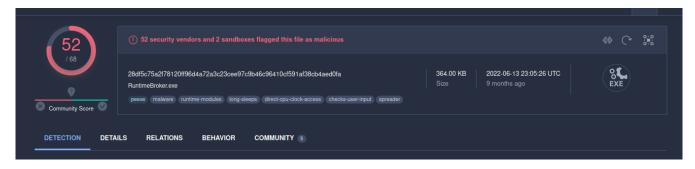
Two samples are being investigated:

sample.exe:

File size: 372736 bytes MD5 sum: cc7207f09a6fe41c71626ad4d3f127ce SHA-1 sum: 84e749c37978f9387e16fab29c7b1b291be93a63 SHA-256 sum: 28df5c75a2f78120ff96d4a72a3c23cee97c9b46c96410cf591af38cb4aed0fa

First of all, check our sample via VirusTotal:

https://www.virustotal.com/gui/file/28df5c75a2f78120ff96d4a72a3c23cee97c9b46c96410cf591af38cb4aed0fa/details



So, 52 of 68 AV engines detect our sample as malicious.

MAX	() Malware (ai Score=100)	MaxSecure	Trojan.Malware.108958436.susgen
McAfee	GenericRXOD-FGICC7207F09A6F	McAfee-GW-Edition	GenericRXOD-FGICC7207F09A6F
Microsoft	① Trojan:Win32/Trickbot.AT!MTB	Palo Alto Networks	() Generic.ml
Panda	() Trj/Cl.A	Rising	Trojan.Generic@AI.95 (RDML:rexq7AX9
Sangfor Engine Zero	() Trojan.Win32.Udochka.gen	SecureAge	() Malicious
SentinelOne (Static ML)	() Static AI - Suspicious PE	TEHTRIS	① Generic.Malware
Tencent	D Malware.Win32.Gencirc.11bba504	Trapmine	Malicious.high.ml.score
Trellix (FireEye)	D Generic.mg.cc7207f09a6fe41c	TrendMicro	Backdoor.Win32.WINDEALER.ZYJA
TrendMicro-HouseCall	Backdoor.Win32:WINDEALER.ZYJA	VBA32	() Trojan.Udochka
ViRobot	() Trojan.Win32.Z.Graftor.372736.TSG	Yandex	Trojan.UdochkalepALYraAdxE
Acronis (Static ML)	✓ Undetected	Baidu	⊘ Undetected
ClamAV	✓ Undetected	CMC	✓ Undetected

More of them detect file as Backdoor.Win32.WINDEALER.ZYJA.

Static analysis

The specified sample is a PE file:

file <sample.exe>

remnux@remnux:~/malware-analysis/2023-03-23-malware-analysis\$ file samples/sample.exe
samples/sample.exe: PE32 executable (GUI) Intel 80386, for MS Windows

hexdump -C <sample.exe>

remnux@re	mnux:	~/ma	lwa	re-a	ana]	Lysi	is/2	023·	-03-	-23-	-ma]	lwa	re-a	ana'	lysi	is\$ hexdump -C samples/sample.exe head
000000000	4d 5	a 90	00	03	00	00	00	04	00	00	00	ff	ff	00	00	MZ
00000010	b8 0	0 00	00	00	00	00	00	40	00	00	00	00	00	00	00	
00000020	00 0	0 00	00	00	00	00	00	00	00	00	00	00	00	00	00	· [
00000030	00 0	0 00	00	00	00	00	00	00	00	00	00	f0	00	00	00	
00000040	0e 1	f ba	0e	00	b4	09	cd	21	b8	01	4c	cd	21	54	68	!.l.!Th
00000050	697	3 20	70	72	6f	67	72	61	6d	20	63	61	6e	6e	6f	is program canno
00000060	74 2	0 62	65	20	72	75	6e	20	69	6e	20	44	4f	53	20	t be run in DOS
00000070	6d 6	f 64	65	2e	0d	0d	0a	24	00	00	00	00	00	00	00	mode\$
00000080	ba 2	8 ec	17	fe	49	82	44	fe	49	82	44	fe	49	82	44	.(I.D.I.D.I.D
00000090	85 5	5 8e	44	f8	49	82	44	7d	41	df	44	fa	49	82	44	.U.D.I.D}A.D.I.D
remnux@re	remnux@remnux:~/malware-analysis/2023-03-23-malware-analysis\$															

Use exiftool for looking metadata:

exiftool <sample.exe>

<pre>L s exiftool samples/sample.exe</pre>	ttps.//www.wigustatal.com/awi/fila/
ExifTool Version Number	: 12.49
File Name	: sample.exe
Directory	: samples that four cost of a second state is a cost of the second state is a second state of the second s
File Size	: 373 kB
File Modification Date/Time	: 2022:02:01 19:18:04+03:00
File Access Date/Time	: 2023:03:30 10:55:29+03:00
File Inode Change Date/Time	: 2023:03:30 10:54:40+03:00
File Permissions	<pre>:>-rW-rrpV engines detect our sample as malic:</pre>
File Type	: Win32 EXE
File Type Extension	: exe
MIME Type	: application/octet-stream a kino since
Machine Type Machine Machine Machine Machine	: Intel 386 or later, and compatibles
Time Stamp	: 2021:01:25 13:32:26+03:00
Image File Characteristics	: No relocs, Executable, No line numbers,
PE Type	: PE32
Linker Version	e 6.0 fied sample is a PE file:
Code Size of the second	: 86016
Initialized Data Size	: 303104
Uninitialized Data Size	∴xdimp -C <sample.exe></sample.exe>
Entry Point	: 0x15020
OS Version	: 4.0
Image Version	: 0.0
Subsystem Version	: 4.0
Subsystem	: Windows GUI
File Version Number 1990	: 11.0.18362.267
Product Version Number	: 11.0.18362.267
File Flags Mask	: 0x003f
File Flags	: (none)
File OS	: Win32 that file modification timestamp is
Object File Type	: Executable application
File Subtype and St	: 0
Language Code as and the la	: English (U.S.)7-57.png)
Character Set October 16	: Unicode
Comments	.1](./2023-01-26_17-59.png)
Company Name	:
File Description	: Runtime Broker: packed by UPX:
File Version	: 11, 0, 18362, 267
Internal Name	: Microsoft Windows Operating System
\mathcal{P} master \mathcal{O} \otimes 0 Δ 0	

And we see that file timestamp is 2021-01-25 13:32:26+03.00

Executable sample is not packed by upx:

upx -l <sample.exe>

≣-nbt-172.21.1.0-24.txt ≣-nbt-172.22.0.0-24.txt	timate Packer for eXecu Copyright (C) 1996 - 2 Perhumer, Laszlo Molnar	020
File size	Ratio bash Format	Name
upx: samples/sample.exe:	NotPackedException: not	packed by UPX

What about Shannon entropy of the sample:

.text	virtual address: 0x1000 virtual size: 0x144c6 raw size: 0x15000			
	entropy: 6.327385942283466			
.rdata				
	virtual address: 0x16000			
	virtual size: 0x1d82 raw size: 0x2000			
	entropy: 5.379194415510608			
.data				
	virtual address: 0x18000			
	virtual size: 0x5ca8 raw size: 0x1000			
	entropy: 3.366340507761682			
.rsrc	.,			
	virtual address: 0x1e000 virtual size: 0x411f8 raw size: 0x42000			
	entropy: 7.04482469997232	 		

Analysze with DIE says that the compiler is Microsoft Visual Studio C++ (6.0):

PE32 executable	(GUI) Intel 80380	5. for MS Windo	WS				
	Detect It	Easy v3.05 [Ubuntu]	20.04.4 LTS](x86_64)		_ ¤ ×		
File name							
/home/remnux/malware-analysis/2023-03-23-malware-analysis/samples/sample.exe							
File type	Entry point		Base address		File info		
PE32 -	00415020	> Disasr	n 00400000	Memory map	MIME		
PE	Export	Import Resource	es .NET	TLS Overlay	Hash		
Sections Ti	me date stamp	Size of image	Resou	urces	Strings		
0004 > 2	021-01-25 05:32:26	00060000	Ma	nifest Version	Entropy		
Scan		Endianness Mod	le Architecture	Туре	Hex		
Automatic	•	LE 32-t	it I386	GUI	Signatures		
▼ PE32					Demangle		
Library: MFC(4				S ?			
	Microsoft Visual C/C++		EXE32]	S ?			
	rosoft Visual C++(6.0)			S ? S ?			
Linker: Microso	oft Linker(6.0*)[GUI32	.]		5 /			
					Shortcuts		
Signatures		🗸 Deep scan 🔽 Re	cursive scan 🗌 All ty	pes Scan	About		
Directory	100%	>	Log 313 ms		Exit		
: 0x411f8							

remnux@remnux: ~/malware-analysis/2023-

				Entropy		_ = ×
Type PE32)	Count Size	Reload
Total	Status					
7.02105		packed(87%)			Save	Save diagram
Entropy	Bytes					
Regions						
Offse	et	Size	Entropy	Status	Name	
0000	00000	00001000		not packed	PE Header	
000	01000	00015000	6.32740	not packed	Section(0)['.text']	
000	16000	00002000	5.37937	not packed	Section(1)['.rdata']	•
Grid	M	M				1
معالم م	50,000) 100,000	150,000	200,000 2	250,000 300,000 350,00	00 400,000 Close

Malware contains encrypted DLL:

000171f0	000173	dc 00000000 00000000 00017d40 000161c4
DLL Nam	e: MSVC	P60.dll
vma: H	lint/Ord	Member-Name Bound-To
17cb0	813	? C@?1?? Nullstr@?\$basic string@DU?\$char traits@D@std@@V?\$allocator@D@2@@std@@CAPBDXZ@4DB
17c0a	1052	?assign@?\$basic string@DU?\$char traits@D@std@@V?\$allocator@D@2@@std@@QAEAAV12@ABV12@II@Z
17bba	1016	? Tidy@?\$basic string@DU?\$char traits@D@std@@V?\$allocator@D@2@@std@@AAEX N@Z
17b62	1056	?assign@?\$basic string@DU?\$char traits@D@std@@V?\$allocator@D@2@@std@@QAEAAV12@PBDI@Z
17b18	233	??1?\$basic string@DU?\$char traits@D@std@@V?\$allocator@D@2@@std@@QAE@XZ
17b00	269	??1 Winit@std@@QAE@XZ
17ae8	165	??0 [¯] Winit@std@@QAE@XZ
17ac8	265	??IInit@ios_base@std@@QAE@XZ
17aa8	158	??0Init@ios ⁻ base@std@@QAE@XZ
17d0c	267	??1_Lockit@std@@QAE@XZ
17d26	162	??0_Lockit@std@@QAE@XZ
17c66	1633	?npos@?\$basic_string@DU?\$char_traits@D@std@@V?\$allocator@D@2@@std@@2IB
00017204	000000	00 0000000 0000000 0000000 00000000

Interesting strings:

GetModuleFileNameA CreateThread Sleep CreateFileA GetLocalTime InterlockedIncrement FindClose FindNextFileA DeleteFileA FindFirstFileA CreateDirectoryA LocalFileTimeToFileTime GetFileTime SystemTimeToFileTime InterlockedExchange InterlockedDecrement GetTempPathW GetTickCount DeleteFileW CreateFileW **GetVolumeInformationA** GetProcAddress GetSystemDirectoryA GetModuleHandleA GetStartupInfoA KERNEL32.dll LoadIconA SendMessageA DrawIcon GetClientRect GetSystemMetrics IsIconic EnableWindow GetForegroundWindow

The hardcoded version of WinDealer:

userprofile	n.txt	
%s\%08lx\%s		
\\?\%c:		
\\.\%s%d		
%S ≡ crac%Si=10.10.2.0-2	°õS	
18.20.1225		
unknown ^{shitet}		
%S ^{5 100} %S		
ProductVersion		
%02X-%02X-%02X-	%02X-%02X-%0	92X
window-title: %	5 1	
content-length:	%d	
content-length:	4 0 1 1	

18.20.1225 - version: 18, year: 2020, month and day: 12.25

another intersting strings is:

SYSTEM\CurrentControlSet\Control\Network\{4D36E972-E325-11CE-BFC1-08002BE10318}\%s\Connection:

SYSTEM\CurrentControlSet\Control\Network\{4D36E972-E325-11CE-BFC1-08002BE10318}\%s\Connection

Dynamic analysis

The sample is GUI application:

```
.text:004151C0 ; int __stdcall WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nShowCmd)
.text:004151C0 _WinMain@16 proc near ; CODE XREF: start+12F↑p .text:004151C0
 .text:004151C0
.text:004151C0 hInstance = dword ptr 4
.text:004151C0 hPrevInstance = dword ptr 8
.text:004151C0 lpCmdLine = dword ptr 0Ch
...tr004151C0 nShowCmd = dword ptr 10h
 .text:004151C0
.text:004151C0
.text:004151C0
.text:004151C4
.text:004151C8
.text:004151D5
.text:004151D5
text:004151D5
~
                                                push
                                                              [esp+nShowCmd] ; int
                                                              [esp+4+lpCmdLine] ; char *
                                                 push
                                                              [esp+8+hPrevInstance] ; HINSTANCE
                                                 push
                                                 push
                                                              [esp+0Ch+hInstance] ; HINSTANCE
                                                              ?AfxWinMain@@YGHPAUHINSTANCE_@@0PADH@Z ; AfxWinMain(HINSTANCE_ *,HINSTANCE_ *,char *,int
                                                 call
                                                 retn
                                                              10h
  .text:004151D5 _WinMain@16
.text:004151D5
                                                 endp
  .text:004151D8 ; [00000029 BYTES: COLLAPSED FUNCTION AfxInitialize(int,ulong). PRESS CTRL-NUMPAD+ TO EXPAND]
>
 .text:00415201
```

Contacted IP addresses is:

Contacted URLs (97)

Scanned	Detections	Status	URL
2022-09-09	0 / 88		http://192.168.122.91/
2022-09-09	0 / 88		http://192.168.122.83/
2022-09-09	0 / 88		http://192.168.122.51/
2022-09-09	0 / 88		http://192.168.122.89/
2022-09-09	0 / 88		http://192.168.122.60/
2022-09-09	0 / 88		http://192.168.122.72/
2022-09-09	0 / 88		http://192.168.122.32/
2022-09-09	0 / 88		http://192.168.122.66/
2022-09-09	0 / 88		http://192.168.122.28/
2022-09-09	0 / 88		http://192.168.122.20/

 \cdots

Contacted Domains (30)

Domain	Detections	Created	Registrar
1.125.168.192.in-addr.arpa	0 / 86		
11.125.168.192.in-addr.arpa	0 / 85		
13.125.168.192.in-addr.arpa	<mark>0</mark> / 86		
14.125.168.192.in-addr.arpa	0 / 86		
2.125.168.192.in-addr.arpa	0 / 85		
2.2.0.10.in-addr.arpa	0 / 85		
20.125.168.192.in-addr.arpa	0 / 85		
21.125.168.192.in-addr.arpa	0 / 85		
27.125.168.192.in-addr.arpa	<mark>0</mark> / 86		
3.125.168.192.in-addr.arpa	0 / 85		

. . .

Contacted IP addresses (114)

IP	Detections	Autonomous System	Country
111.120.234.106	<mark>0</mark> / 85	4134	CN
111.120.65.56	<mark>0</mark> / 85	4134	CN
111.122.58.92	<mark>0</mark> / 85	4134	CN
111.123.173.15	<mark>0</mark> / 85	4134	CN
113.62.154.30	<mark>0</mark> / 85	4134	CN
113.62.44.119	<mark>0</mark> / 85	4134	CN
113.62.74.64	<mark>0</mark> / 85	4134	CN
113.62.92.79	<mark>0</mark> / 85	4134	CN

May sleep (evasive loops) to hinder dynamic analysis:

set	_app_ty	be							
cont	rolfp								
GetMo	duleFile	eNameA							
Create	eThread								
Sleep									
	eFileA								
	calTime								
		ncrement							
FindC		mb.bxt							
.10ata:0	0416000		;	SUD_401F5/H	ээтр				
.idata:0			· · · · · · · · · · · · · · · · · · ·	DATA XREF:					
		<pre>(stdcall *Sleep)(I</pre>							
.idata:0		extrn <mark>Slee</mark>			sub_4022BE+111p				
.idata:0		- /		sub_402C9C+					
.idata:0		LE (stdcall *Create			sub 403CBA+AE1p	ccess, DWORL	awsnareMode,	LPSECORITY_	ATTREE
.idata:0		extrn crea		sub 403FE4+	_				

; sub 403FE4+AE1p ...

The operator has the power to rename, move, and delete files on the target machine:

	<pre>.idata:00416004 ; void (_stdcall *Sleep)(DWORD dwMilliseconds) .idata:00416004 extrn Sleep:dword ; CODE XREF: sub_40228E+111p .idata:00416004 ; sub_402C9C+D41p .idata:00416008 ; HANDLE (_stdcall *CreateFileA)(LPCSTR IpFileName, DWORD dwDesiredAccess, DWORD dwShareMode, LPSECURITY_ATTRIBUT .idata:00416008 extrn CreateFileA:dword ; CODE XREF: sub_403EC4A+AE1p .idata:00416008 ; sub_403FE4+AE1p .idata:00416002 ; void (_stdcall *GetLocalTime)(LPSYSTEMTIME lpSystemTime) .idata:0041600C extrn GetLocalTime:dword</pre>
•	.idata:00416044 ; DATA XREF: sub_40A1A7+11D1r .idata:00416048 ; HANDLE (stdcall *CreateFileW)(LPCWSTR lpFileName, DWORD dwDesiredAccess, DWORD dwShareMode, LPSECURITY_ATTRIBL .idata:00416048 extrn CreateFileW:dword ; CODE XREF: sub_40A360+171p .idata:00416048 ; DATA XREF: sub_40A360+171r
1	.idata:00416040 extrn GetlickCount:dword
	.idata:00416040 ; CODE XREF: sub_4099051p
	.idata:00416040 ; sub_414433+1EF↑p
	<pre>.idata:00416044 ; BOOL (stdcall *DeleteFileW)(LPCWSTR lpFileName)</pre>
	.idata:00416044 extrn DeleteFileW:dword ; CODE XREF: sub_40A1A7+11D↑p
	.idata:00416044 ; DATA XREF: sub_40A1A7+11D1r
	.idata:00416048 ; HANDLE (stdcall *CreateFileW)(LPCWSTR lpFileName, DWORD dwDesiredAccess
	.idata:00416048 extrn CreateFileW:dword ; CODE XREF: sub_40A360+17↑p
	.idata:00416048 ; DATA XREF: sub_40A360+171r
	idsts-8041604C + ROOL / stdspll *6atVolumeInformationA\/LDCSTD lnDootDathName LDSTD lnVc

Also malware search through directories and enum filesystem:

) COL ANELY SUB TO SUBTLEY
	.idata:00416010	; sub 407935+38↑p
	.idata:00416014	; BOOL (stdcall *FindClose)(HANDLE hFindFile)
•	.idata:00416014	extrn FindClose:dword ; CODE XREF: sub 404333+24C1p
	.idata:00416014	; sub 40458D+4B3↑p
	.idata:00416018	; BOOL (stdcall *FindNextFileA)(HANDLE hFindFile, LPWIN32 FIND DATAA lpFindFileData)
•	.idata:00416018	extrn FindNextFileA:dword
	.idata:00416018	; CODE XREF: sub 404333+23B↑p
	.idata:00416018	; sub_40458D+4A2↑p
	.idata:0041601C	; BOOL (stdcall *DeleteFileA)(LPCSTR lpFileName)
•	.idata:0041601C	extrn DeleteFileA:dword ; CODE XREF: sub_404333+1321p
	.idata:0041601C	; sub_40458D+4921p
	.idata:00416020	; HANDLE (stdcall * <mark>FindFirstFileA</mark>)(LPCSTR 1pFileName, LPWIN32_FIND_DATAA lpFindFileData)
•	.idata:00416020	extrn FindFirstFileA:dword
	.idata:00416020	; CODE XREF: sub_404333+A7↑p
	.idata:00416020	; sub_40458D+187↑p
	.idata:00416024	; BOOL (stdcall *CreateDirectoryA)(LPCSTR lpPathName, LPSECURITY_ATTRIBUTES lpSecurityAttributes)
•	.idata:00416024	extrn CreateDirectoryA:dword
	.idata:00416024	; CODE XREF: sub_4065A2+1A41p
	.idata:00416024	: DATA XREF: sub 4065A2+1A41r

and collecting volume information:

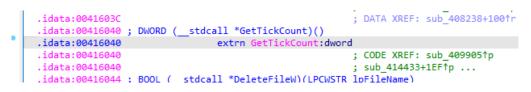


Using InterlockedExchange, probably the malware sample is hooking the winapi functions:

1	.10000.00410000	, 4100-1001-001-P
	.idata:00416030	; DATA XREF:
	.idata:00416034 ;	LONG (stdcall *InterlockedExchange)(volatile LONG *Target, LONG Value)
· I	.idata:00416034	extrn InterlockedExchange:dword
	.idata:00416034	; CODE XREF: sub_407935+E9↑p
	.idata:00416034	; sub_408014+15↑p
	.idata:00416038 ;	LONG (stdcall *InterlockedDecrement)(volatile LONG *IpAddend)
•	.idata:00416038	extrn InterlockedDecrement:dword
	.idata:00416038	; CODE XREF: sub_407935+C5↑p
	.idata:00416038	; sub 407935+D91p
	.idata:00416038	; DATA XREF:
	.idata:0041603C ;	DWORD (stdcall *GetTempPathW)(DWORD nBufferLength, LPWSTR lpBuffer)
•	1.1.1. contractor	a second product of the second s

AV/Sandbox evasion

In the malware sample above, the delay timeout is set using the GetTickCount() timer function. The Sleep() function is called in a loop until the timer timeout. In the sandbox, delays that are performed by the Sleep() function are skipped (replaced with a very short timeout) and the virtually elapsed time will be much higher than the requested timeout. The concept behind these methods is to measure elapsed time while running several forms of delays in parallel:



sample2.exe:

File size: 458752 bytes MD5 sum: 76ba5272a17fdab7521ea21a57d23591 SHA-1 sum: 6b831413932a394bd9fb25e2bbdc06533821378c SHA-256 sum: ecd001aeb6bcbafb3e2fda74d76eea3c0ddad4e6e7ff1f43cd7709d4b4580261

VirusTotal scan result:

https://www.virustotal.com/gui/file/ecd001aeb6bcbafb3e2fda74d76eea3c0ddad4e6e7ff1f43cd7709d4b4580261/detec tion



Rising	(() Trojan.Generic@AI.98 (RDML:LHNE4akpK	Sangfor Engine Zero	() Trojan.Win32.Udochka.gen
SecureA	ge (① Malicious	SentinelOne (Static ML)	
Sophos		D Mal/Generic-S	Symantec	
ТАСНУО	N (D Trojan/W32.Udochka.458752	Tencent	
Trapmin	e (D Malicious.moderate.ml.score	Trellix (FireEye)	
TrendMi	cro (Backdoor.Win32.WINDEALER.ZYJA	TrendMicro-HouseCall	
VBA32		() Trojan.Udochka	VIPRE	
ViRobot		Trojan.Win32.S.Agent.458752.OV	Yandex	
Zillya		① Trojan.Agent.Win32.1984122	Acronis (Static ML)	O Undetected
Baidu		Undetected	ClamAV	O Undetected

Static analysis

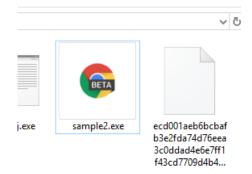
The specified sample is a PE file:

file <sample2.exe>

samples/sample2.exe: PE32 executable (GUI) Intel 80386, for MS Windows

hexdump -C <sample2.exe>

пехаашр c sampres/sam 4d 5a 90 00 03 00 ff ff MZ..... b8 00 00 00 00 00 0e 00 b4 09 b8 4c cd 0e 1f ba cd!..L.!Th 69 73 20 72 6f 67 6d 6e 6e 6f is program canno 74 20 62 65 20 72 75 6e 6e 4f t be run in DOS 6d 6f 64 65 2e 0d 0d 0a 00 00 00 00 00 00 mode....\$.... 7a 2b ec 17 3e 4a 82 44 3e 4a 82 44 3e 4a 82 44 z+..>J.D>J.D>J.D 45 56 8e 44 38 4a 82 44 bd 42 df 44 3a 4a 82 44 EV.D8J.D.B.D:J.D



Run exiftool for extracting metadata:

exiftool <sample2.exe>

└─\$ exiftool <u>samples/sample2.exe</u>	auoraceouchareocriteareoaudaueocritereocariosaene
	: 12.49
File Name	: sample2.exe
Directory	: samples
177 17 0 0 0 0 0 L L L L L	: 459 kB
File Modification Date/Time	: 2022:02:01 19:18:04+03:00
File Access Date/Time	: 2023:03:30 10:56:46+03:00
	: 2023:03:30 10:54:17+03:00
	: sa rw araangle is a PE file:
	: Win32 EXE
	: exe
MIME Type	: application/octet-stream
Machine Type	: Intel 386 or later, and compatibles
Time Stamp	: 2021:03:06 04:13:51+03:00
Image File Characteristics	: No relocs, Executable, No line numbers, No symbols, 32-bit
	: PE32
	: 6.0
	: al81920 <sample2.exe></sample2.exe>
	: 393216
	: 0
	: 0x145f0
	4.0
	: 4.0 tool for extracting metadata:
	: Windows GUI
() heats is an	: 8.8.1314.190
	: 8.8.1314.190
E DDT- ₩2.21.1.0-24.0XT LOO	: 0x003f
	: (none)
100 :[: Win32
	: Executable application
File Subtype	: 0 p dat
Language Code	: English (U.S.)
Character Set	: Unicode
Comments	

The sample is a Windows GUI file with timestamp: 2021:03:06 04:13:51+03:00

Dynamic analysis

Generating victim **ID** set in a registry key:

```
mov eax, [esp+size] ; 0x43
mov ecx, [esp+buf_str] ; "mac: 00:0C:29:43:FA:A5\x09VMware Virtual NVMe Disk\x09MVaWerN MV_E0000usr"
sub esp, 10h
lea edx. [esp+10h+hexdigest]
```

The format of the victim ID is md5("<MAC address>+<Physical_Drive_info>+<username>"). The malware generates a unique registry entry to store the victim ID for subsequent execution. The victim ID is not saved as raw data; instead, the malware changes the 4 bytes victim ID to an IP address format.

This sample collecting host information:

```
char __thiscall get_victim_into_entry(_DWORD *this)
{
   getComputerName();
   getUsername(this);
   getCPUinfo(this);
   getOSversion(this);
   getNetworkCard(this);
   TimeZone(this);
   getPulbicIP(this);
   if ( this[9] )
     EnumUserInfo(this);
   return 1:
```

Encoding

Malware sample use function call obfuscation:

Ge	†11	ISP	rΝ	ิลฑ	eW	
00		00		um	~	•

.text:00408C52	push	esi	
.text:0040BC53	mov	[ebp+var_10], 47h ; 'G'	
.text:0040BC57	mov	[ebp+var_F], 65h ; 'e'	
.text:0040BC5B	mov	[ebp+var_E], 74h ; 't'	
.text:0040BC5F	mov	[ebp+var_D], 55h ; 'U'	
.text:0040BC63	mov	[ebp+var_C], 73h ; 's'	
.text:0040BC67	mov	[ebp+var_B], 65h ; 'e'	
.text:0040BC6B	mov	[ebp+var_A], 72h ; 'r'	
.text:0040BC6F	mov	[ebp+var_9], 4Eh ; 'N'	
.text:0040BC73	mov	[ebp+var_8], 61h ; 'a'	GetUserNameW
.text:0040BC77	mov	[ebp+var_7], 6Dh ; 'm'	Gerosernamew
.text:0040BC7B	mov	[ebp+var_6], 65h ; 'e'	
.text:0040BC7F	mov	[ebp+var_5], 57h ; 'W'	
.text:0040BC83	mov	[ebp+var_4], bl	
.text:0040BC86	call	sub 40FE31	

RegCreateKeyExA:

	Para de la compara de	····
.text:0040BC98	mov	[ebp+var_108], 52h ; 'R'
.text:0040BC9F	mov	[ebp+var_107], 65h ; 'e'
.text:0040BCA6	mov	[ebp+var_106], 67h ; 'g'
.text:0040BCAD	mov	[ebp+var_105], 43h ; 'C'
.text:0040BCB4	mov	[ebp+var_104], 72h ; 'r'
.text:0040BCBB	mov	[ebp+var_103], 65h ; 'e'
.text:0040BCC2	mov	[ebp+var_102], 61h ; 'a'
.text:0040BCC9	mov	[ebp+var_101], 74h ; 't'
.text:0040BCD0	mov	[ebp+var_100], 65h ; 'e'
.text:0040BCD7	mov	[ebp+var_FF], 4Bh ; 'K'
.text:0040BCDE	mov	[ebp+var_FE], 65h ; 'e'
.text:0040BCE5	mov	[ebp+var_FD], 79h ; 'y'
.text:0040BCEC	mov	[ebp+var_FC], 45h ; 'E'
.text:0040BCF3	mov	[ebp+var_FB], 78h ; 'x'
.text:0040BCFA	mov	[ebp+var_FA], 41h ; 'A'
.text:0040BD01	mov	[ebp+var_F9], bl 🕒 🚽
.text:0040BD07	call	sub 40FE31

RegCreateKeyExA

RegDeleteKeyA and RegCloseKey:

		push	C31	
•	.text:0040BAD0	mov	[ebp+var E8], 52h ; 'R'	
•	.text:0040BAD7	mov	[ebp+var E7], 65h ; 'e'	
•	.text:0040BADE	mov	[ebp+var_E6], 67h ; 'g'	
•	.text:0040BAE5	mov	[ebp+var E5], 44h ; 'D'	
•	.text:0040BAEC	mov	[ebp+var_E4], 65h ; 'e'	
•	.text:0040BAF3	mov	[ebp+var E3], 6Ch ; '1'	RegDeleteKeyA
•	.text:0040BAFA	mov	[ebp+var E2], 65h ; 'e'	RegDeleteReyA
•	.text:0040BB01	mov	[ebp+var_L2], 03h , e [ebp+var_E1], 74h ; 't'	
•	.text:0040BB08	mov	Teacher Teacher and Aller and Aller	
•				
•	.text:0040BB0F	mov	[ebp+var_DF], 4Bh ; 'K'	
•	.text:0040BB16	mov	[ebp+var_DE], 65h ; 'e'	
•	.text:0040BB1D	mov	[ebp+var_DD], 79h ; 'y'	
•	.text:0040BB24	mov	[ebp+var_DC], 41h ; 'A'	
	.text:0040BB2B	mov	[ebp+var_DB], bl	
	.text:0040BB31	call	sub_40FE31	
	.text:0040BB36	mov	dword_41C8CC, eax	
	.text:0040BB3B	lea	eax, [ebp+var_9C]	
	.text:0040BB41	push	eax	
	.text:0040BB42	push	esi	
	.text:0040BB43	mov	[ebp+var_9C], 52h ; 'R'	
	.text:0040BB4A	mov	[ebp+var_9B], 65h ; 'e'	
	.text:0040BB51	mov	[ebp+var_9A], 67h ; 'g'	
•	.text:0040BB58	mov	[ebp+var 99], 43h ; 'C'	
•	.text:0040BB5F	mov	[ebp+var 98], 6Ch ; '1'	
•	.text:0040BB66	mov	[ebp+var 97], 6Fh ; 'o'	RegCloseKey
•	.text:0040BB6D	mov	[ebp+var 96], 73h ; 's'	, , , , , , , , , , , , , , , , , , ,
•	.text:0040BB74	mov	[ebp+var 95], 65h ; 'e'	
•	.text:0040BB7B	mov	[ebp+var 94], 4Bh ; 'K'	
•	.text:0040BB82	mov	[ebp+var 93], 65h ; 'e'	
•				
:	.text:0040BB89	mov	[ebp+var_92], 79h ; 'y'	
	.text:0040BB89 .text:0040BB90	mov	[ebp+var_92], 79h ; 'y' [ebp+var_91], bl	
egQue	.text:0040BB89 .text:0040BB90	mov	[ebp+var_92], 79h ; 'y' [ebp+var_91], bl	
egQue	.text:0040BB89 .text:0040BB90	mov	[ebp+var_92], 79h ; 'y' [ebp+var_91], bl	
egQue	.text:0040BB89 .text:0040BB90 text:0040BB90	mov mov	[ebp+var_92], 79h ; 'y' [ebp+var_91], bl	
egQue	.text:0040BB89 .text:0040BB90 teryValueExA:	mov mov 11 push	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1 wh sorras eax, [eop+var_iic]</pre>	
egQue	.text:0040BB89 .text:0040BB90 text:0040BB90 teryValueExA: .text:0040D9/F .text:0040B985	mov mov !! push push	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1 wh torrow eax, [eop+var_iii] eax esi</pre>	
egQue	.text:0040BB89 .text:0040BB90 text:0040BB90 text:0040B90 ceryValueExA: .text:0040B977 .text:0040B985 .text:0040B986	mov mov 11 push	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1 eax, [eop+var_11c] eax esi [ebp+var_11C], 52h 'R'</pre>	
egQua	.text:0040BB89 .text:0040BB90 tot:0040BB90 tot:0040B90 text:0040B97 .text:0040B985 .text:0040B986 .text:0040B987 .text:0040B98E	mov mov 11 push push mov mov	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1 eax, [eop+var_11C] eax esi [ebp+var_11C], 52h 'R' [ebp+var_11B], 65h 'e'</pre>	
egQue	.text:0040BB89 .text:0040BB90 tot:0040BB90 tot:0040B90 .text:0040B97 .text:0040B985 .text:0040B986 .text:0040B987 .text:0040B98E .text:0040B995	mov mov 11 push push mov mov mov	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1 eax, [eop+var_11c] eax esi [ebp+var_11C], 52h 'R' [ebp+var_11B], 65h [ebp+var_11A], 67h 'g'</pre>	
egQue	.text:0040BB89 .text:0040BB90 tot:0040BB90 tot:0040B90 .text:0040B97 .text:0040B985 .text:0040B986 .text:0040B987 .text:0040B985 .text:0040B995 .text:0040B995 .text:0040B995	mov mov 11 push push mov mov mov	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1 eax, [eop+var_11c] eax esi [ebp+var_11C], 52h 'R' [ebp+var_11B], 65h 'e' [ebp+var_11A], 67h 'g' [ebp+var_119], 51h 'Q'</pre>	
egQua	.text:0040BB89 .text:0040BB90 tot:0040BB90 tot:0040B90 .text:0040B985 .text:0040B985 .text:0040B987 .text:0040B987 .text:0040B985 .text:0040B995 .text:0040B995 .text:0040B9943	mov mov 11 push push mov mov mov mov	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1 """"""""""""""""""""""""""""""""""""</pre>	
egQua	.text:0040BB89 .text:0040BB90 tot:0040BB90 tot:0040B90 reryValueExA: .text:0040B985 .text:0040B986 .text:0040B988 .text:0040B995 .text:0040B995 .text:0040B995 .text:0040B9943 .text:0040B9AA	mov mov 11 push push mov mov mov mov mov	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], bl """ eax, [eop+var_11C] eax esi [ebp+var_11C], 52h 'R' [ebp+var_11B], 65h 'e' [ebp+var_11A], 67h 'g' [ebp+var_11A], 75h 'u' [ebp+var_117], 65h 'e'</pre>	
₽gQua	.text:0040BB89 .text:0040BB90 tot:0040BB90 tot:0040B90 text:0040B985 .text:0040B985 .text:0040B987 .text:0040B987 .text:0040B995 .text:0040B995 .text:0040B9A3 .text:0040B9A3 .text:0040B9A4 .text:0040B9B1	mov mov 11 push push mov mov mov mov mov mov mov	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1 """ eax, [eop+var_11C] eax esi [ebp+var_11C], 52h 'R' [ebp+var_11B], 65h 'e' [ebp+var_11B], 65h 'u' [ebp+var_11B], 75h 'u' [ebp+var_117], 65h 'e' [ebp+var_116], 72h 'r'</pre>	
egQua	.text:0040BB89 .text:0040BB90 tot:0040BB90 tot:0040B90 text:0040B985 .text:0040B985 .text:0040B987 .text:0040B985 .text:0040B995 .text:0040B995 .text:0040B9A3 .text:0040B9A3 .text:0040B9A4 .text:0040B9B1 .text:0040B9B8	mov mov 11 push push mov mov mov mov mov mov mov	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1 """ eax, [ebp+var_11C] eax esi [ebp+var_11C], 52h 'R' [ebp+var_11B], 65h 'e' [ebp+var_11A], 67h 'g' [ebp+var_11A], 75h 'u' [ebp+var_11A], 75h 'u' [ebp+var_11A], 75h 'u' [ebp+var_11A], 72h 'r' [ebp+var_115], 79h 'y'</pre>	
egQua	.text:0040BB89 .text:0040BB90 tot:0040BB90 tot:0040B90 text:0040B985 .text:0040B985 .text:0040B987 .text:0040B988 .text:0040B995 .text:0040B993 .text:0040B9A3 .text:0040B9A3 .text:0040B9B1 .text:0040B9B1 .text:0040B9B5	mov mov 11 push push mov mov mov mov mov mov mov mov	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1 """ eax, [eop+var_11C] eax esi [ebp+var_11C], 52h 'R' [ebp+var_11B], 65h 'e' [ebp+var_11A], 67h 'g' [ebp+var_11A], 75h 'u' [ebp+var_11A], 75h 'u' [ebp+var_11A], 75h 'e' [ebp+var_11A], 75h 'v' [ebp+var_11A], 75h 'v'</pre>	
₽gQua	.text:0040BB89 .text:0040BB90 tot:0040BB90 tot:0040B90 text:0040B985 .text:0040B985 .text:0040B987 .text:0040B987 .text:0040B995 .text:0040B943 .text:0040B943 .text:0040B943 .text:0040B944 .text:0040B945 .text:0040B945 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B985	mov mov 11 push push mov mov mov mov mov mov mov mov mov	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1 eax, esi [ebp+var_11C], 52h 'R' [ebp+var_11B], 65h 'e' [ebp+var_11A], 67h 'g' [ebp+var_11A], 75h 'u' [ebp+var_116], 72h 'r' [ebp+var_115], 79h 'y' [ebp+var_114], 56h 'V' [ebp+var_113], 61h 'a'</pre>	
₽gQuu	.text:0040BB89 .text:0040BB90 tot:0040BB90 tot:0040B90 text:0040B985 .text:0040B985 .text:0040B987 .text:0040B985 .text:0040B995 .text:0040B995 .text:0040B9A3 .text:0040B9B1 .text:0040B9B1 .text:0040B9B1 .text:0040B9B5 .text:0040B9B5 .text:0040B9B5 .text:0040B9B5 .text:0040B9C6 .text:0040B9CD	mov mov 11 push push mov mov mov mov mov mov mov mov	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1 eax esi [ebp+var_11C], 52h 'R' [ebp+var_11B], 65h 'e' [ebp+var_11B], 65h 'e' [ebp+var_11A], 67h 'g' [ebp+var_11B], 75h 'u' [ebp+var_116], 72h 'r' [ebp+var_115], 79h 'y' [ebp+var_114], 56h 'V' [ebp+var_113], 61h 'a' [ebp+var_112], 6Ch '1'</pre>	
egQuu	.text:0040BB89 .text:0040BB90 tot:0040BB90 tot:0040B90 text:0040B985 .text:0040B985 .text:0040B987 .text:0040B987 .text:0040B995 .text:0040B943 .text:0040B943 .text:0040B943 .text:0040B944 .text:0040B945 .text:0040B945 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B985	mov mov 11 push push mov mov mov mov mov mov mov mov mov	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1 eax, esi [ebp+var_11C], 52h 'R' [ebp+var_11B], 65h 'e' [ebp+var_11A], 67h 'g' [ebp+var_11A], 75h 'u' [ebp+var_116], 72h 'r' [ebp+var_115], 79h 'y' [ebp+var_114], 56h 'V' [ebp+var_113], 61h 'a'</pre>	RegQueryValueExA
egQuu	.text:0040BB89 .text:0040BB90 tot:0040BB90 tot:0040B90 text:0040B985 .text:0040B985 .text:0040B987 .text:0040B985 .text:0040B995 .text:0040B995 .text:0040B9A3 .text:0040B9B1 .text:0040B9B1 .text:0040B9B1 .text:0040B9B5 .text:0040B9B5 .text:0040B9B5 .text:0040B9B5 .text:0040B9C6 .text:0040B9CD	mov mov 11 push push mov mov mov mov mov mov mov mov mov mov	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1 eax esi [ebp+var_11C], 52h 'R' [ebp+var_11B], 65h 'e' [ebp+var_11B], 65h 'e' [ebp+var_11A], 67h 'g' [ebp+var_11B], 75h 'u' [ebp+var_116], 72h 'r' [ebp+var_115], 79h 'y' [ebp+var_114], 56h 'V' [ebp+var_113], 61h 'a' [ebp+var_112], 6Ch '1'</pre>	RegQueryValueExA
egQuu	.text:0040BB89 .text:0040BB90 tot:0040BB90 tot:0040B90 text:0040B985 .text:0040B985 .text:0040B987 .text:0040B987 .text:0040B995 .text:0040B995 .text:0040B995 .text:0040B9A3 .text:0040B9B1 .text:0040B9B1 .text:0040B9B1 .text:0040B9B5 .text:0040B9B5 .text:0040B9B5 .text:0040B9B5 .text:0040B9B5 .text:0040B9C6 .text:0040B9D4	mov mov 11 push push mov mov mov mov mov mov mov mov mov mov	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1 """ eax, [eop+var_11C], 52h 'R' [ebp+var_11B], 65h 'e' [ebp+var_11B], 65h 'e' [ebp+var_11A], 67h 'g' [ebp+var_11B], 75h 'u' [ebp+var_116], 72h 'r' [ebp+var_115], 79h 'y' [ebp+var_113], 61h 'a' [ebp+var_111], 75h 'u'</pre>	RegQueryValueExA
egQuu	.text:0040BB89 .text:0040BB90 tot:0040B90 tot:0040B90 text:0040B985 .text:0040B985 .text:0040B986 .text:0040B987 .text:0040B995 .text:0040B995 .text:0040B996 .text:0040B9B1 .text:0040B9B1 .text:0040B9B5 .text:0040B9B5 .text:0040B9B5 .text:0040B9B5 .text:0040B9B5 .text:0040B9B5 .text:0040B9D4 .text:0040B9D4 .text:0040B9D5	mov mov 11 push push mov mov mov mov mov mov mov mov mov mov	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1</pre>	RegQueryValueExA
egQuu	.text:0040BB89 .text:0040BB90 tot:0040B90 tot:0040B90 text:0040B985 .text:0040B985 .text:0040B986 .text:0040B987 .text:0040B995 .text:0040B995 .text:0040B990 .text:0040B9AA .text:0040B9B1 .text:0040B9B1 .text:0040B9B1 .text:0040B9B5 .text:0040B9B5 .text:0040B9D4 .text:0040B9D4 .text:0040B9D4 .text:0040B9D5 .text:0040B9D5 .text:0040B9D5 .text:0040B9D5 .text:0040B9D5 .text:0040B9D5 .text:0040B9D5 .text:0040B9D5 .text:0040B9D5 .text:0040B9D5 .text:0040B9D5 .text:0040B9D5 .text:0040B9D5	mov mov 11 push push mov mov mov mov mov mov mov mov mov mov	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1</pre>	RegQueryValueExA
egQuu	.text:0040BB89 .text:0040BB90 text:0040B90 text:0040B90 text:0040B985 .text:0040B986 .text:0040B987 .text:0040B995 .text:0040B995 .text:0040B995 .text:0040B996 .text:0040B981 .text:0040B981 .text:0040B981 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B985 .text:0040B905 .text:0040B	mov mov 11 push push mov mov mov mov mov mov mov mov mov mov	<pre>[ebp+var_92], 79h ; 'y' [ebp+var_91], b1</pre>	RegQueryValueExA

GetTokenInformation:

	.text:0040BE44	push	esi 🗾
	.text:0040BE45	mov	[ebp+var_130], 47h ; 'G'
	.text:0040BE4C	mov	[ebp+var_12F], 65h ; 'e'
	.text:0040BE53	mov	[ebp+var_12E], 74h ; 't'
	.text:0040BE5A	mov	[ebp+var_12D], 54h ; 'T'
	.text:0040BE61	mov	[ebp+var_12C], 6Fh ; 'o'
	.text:0040BE68	mov	[ebp+var_12B], 6Bh ; 'k'
	.text:0040BE6F	mov	[ebp+var_12A], 65h ; 'e'
	.text:0040BE76	mov	[ebp+var_129], 6Eh ; 'n'
	.text:0040BE7D	mov	[ebp+var_128], 49h ; 'I'
	.text:0040BE84	mov	[ebp+var_127], 6Eh ; 'n'
	.text:0040BE8B	mov	[ebp+var_126], 66h ; 'f'
	.text:0040BE92	mov	[ebp+var_125], 6Fh ; 'o'
	.text:0040BE99	mov	[ebp+var_124], 72h ; 'r'
	.text:0040BEA0	mov	[ebp+var_123], 6Dh ; 'm'
	.text:0040BEA7	mov	[ebp+var_122], 61h ; 'a'
	.text:0040BEAE	mov	[ebp+var_121], 74h ; 't'
	.text:0040BEB5	mov	[ebp+var_120], 69h ; 'i'
	.text:0040BEBC	mov	[ebp+var_11F], 6Fh ; 'o'
	.text:0040BEC3	mov	[ebp+var_11E], 6Eh ; 'n'
- I	.text:0040BECA	mov	[ebp+var 11D], bl

GetTokenInformation

OpenProcessToken:

•

.text:0040BF47	push	esi	
.text:0040BF48	mov	[ebp+var_25C], 4Fh ;	'0'
.text:0040BF4F	mov	[ebp+var_25B], 70h ;	'p'
.text:0040BF56	mov	[ebp+var_25A], 65h	'e'
.text:0040BF5D	mov	[ebp+var_259], 6Eh ;	'n'
.text:0040BF64	mov	[ebp+var_258], 50h ;	'P'
.text:0040BF6B	mov	[ebp+var_257], 72h ;	'n' -
.text:0040BF72	mov	[ebp+var_256], 6Fh ;	'o'
.text:0040BF79	mov	[ebp+var_255], 63h ;	'c'
.text:0040BF80	mov	[ebp+var_254], 65h ;	'e'
.text:0040BF87	mov	[ebp+var_253], 73h ;	's'
.text:0040BF8E	mov	[ebp+var_252], 73h ;	's'
.text:0040BF95	mov	[ebp+var_251], 54h ;	'T'
.text:0040BF9C	mov	[ebp+var_250], 6Fh ;	'o'
.text:0040BFA3	mov	[ebp+var_24F], 6Bh ;	'k'
.text:0040BFAA	mov	[ebp+var_24E], 65h ;	'e'
.text:0040BFB1	mov	[ebp+var_24D], 6Eh ;	'n'
.text:0040BFB8	mov	[ebp+var_24C], bl	
text:0040RFRF	call	sub 40FE31	

OpenProcessToken

OpenThreadToken:

		pusii	Cax	
	.text:0040C163	push	esi	
•	.text:0040C164	mov	[ebp+var_234], 4Fh	; '0'
	.text:0040C16B	mov	[ebp+var_233], 70h	; 'p'
	.text:0040C172	mov	[ebp+var_232], 65h	; 'e'
	.text:0040C179	mov	[ebp+var_231], 6Eh	; 'n'
•	.text:0040C180	mov	[ebp+var_230], 54h	; 'T'
	.text:0040C187	mov	[ebp+var_22F], 68h	; 'h'
	.text:0040C18E	mov	[ebp+var_22E], 72h	; 'n'
	.text:0040C195	mov	[ebp+var_22D], 65h	; 'e'
	.text:0040C19C	mov	[ebp+var_22C], 61h	; 'a'
	.text:0040C1A3	mov	[ebp+var_22B], 64h	; 'd'
	.text:0040C1AA	mov	[ebp+var_22A], 54h	; 'T'
	.text:0040C1B1	mov	[ebp+var_229], 6Fh	; 'o'
	.text:0040C1B8	mov	[ebp+var_228], 6Bh	; 'k'
	.text:0040C1BF	mov	[ebp+var_227], 65h	; 'e'
	.text:0040C1C6	mov	[ebp+var_226], 6Eh	; 'n'
	.text:0040C1CD	mov	[ebp+var_225], bl	
	.text:0040C1D3	call	sub_40FE31	
•	tevt.0040C1D8	bhs	esn 20h	

OpenThreadToken

AdjustTokenPrivileges:

	.text:0040L0AL	mov	awora_410804, eax	
•	.text:0040C0B1	mov	[ebp+var 28C], 41h	'A'
•	.text:0040C0B8	mov	[ebp+var 28B], 64h	'd'
•	.text:0040C0BF	mov	[ebp+var_28A], 6Ah	'j'
•	.text:0040C0C6	mov	[ebp+var_289], 75h	'u'
•	.text:0040C0CD	mov	[ebp+var_288], 73h	's'
•	.text:0040C0D4	mov	[ebp+var_287], 74h	't'
•	.text:0040C0DB	mov	[ebp+var_286], 54h	'T'
	.text:0040C0E2	mov	[ebp+var_285], 6Fh	'o'
	.text:0040C0E9	mov	[ebp+var_284], 6Bh	'k'
•	.text:0040C0F0	mov	[ebp+var_283], 65h	'e'
•	.text:0040C0F7	mov	[ebp+var_282], 6Eh	'n'
	.text:0040C0FE	mov	[ebp+var_281], 50h	'P'
	.text:0040C105	mov	[ebp+var_280], 72h	'n'
	.text:0040C10C	mov	[ebp+var_27F], 69h	'i'
	.text:0040C113	mov	[ebp+var_27E], 76h	'v'
	.text:0040C11A	mov	[ebp+var_27D], 69h	'i'
	.text:0040C121	AdjustTokenPrivi	በዊፀይት var_27C], 6Ch	'1'
	.text:0040C128	lea	eax, [ebp+var_28C]	
	.text:0040C12E	mov	[ebp+var_27B], 65h	'e'
	.text:0040C135	push	eax	
	.text:0040C136	push	esi	
	.text:0040C137	mov	[ebp+var_27A], 67h	'g'
	.text:0040C13E	mov	[ebp+var_279], 65h	'e'
	.text:0040C145	mov	[ebp+var_278], 73h	's'
	.text:0040C14C	mov	[ebp+var_277], bl	
	.text:0040C152	call	sub_40FE31	
•	+ev++00/00157	mov	dword A1C8D8 eav	

.etc.

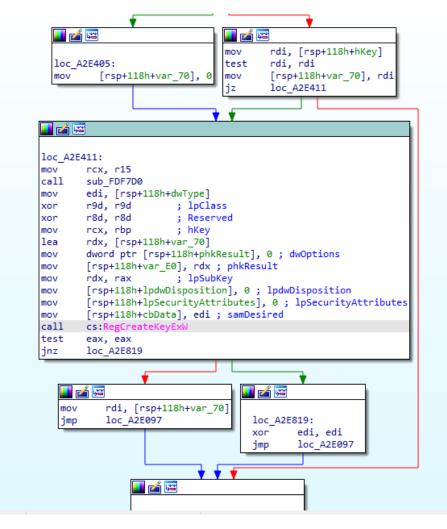
So, malware sample use one of the interesting classic APT techniques: Token theft via turn on SeDebugPrivilege:

```
//....
HANDLE token;
TOKEN_PRIVILEGES tp;
LUID luid;
BOOL res = TRUE;
tp.PrivilegeCount = 1;
tp.Privileges[0].Luid = luid;
tp.Privileges[0].Attributes = SE_PRIVILEGE_ENABLED;
if (!LookupPrivilegeValue(NULL, priv, &luid)) res = FALSE;
if (!OpenProcessToken(GetCurrentProcess(), TOKEN_ADJUST_PRIVILEGES, &token)) res = FALSE;
if (!AdjustTokenPrivileges(token, FALSE, &tp, sizeof(TOKEN_PRIVILEGES), (PTOKEN_PRIVILEGES)NULL,
(PDWORD)NULL)) res = FALSE;
//...
```

//...

Registry Modifications and Persistence

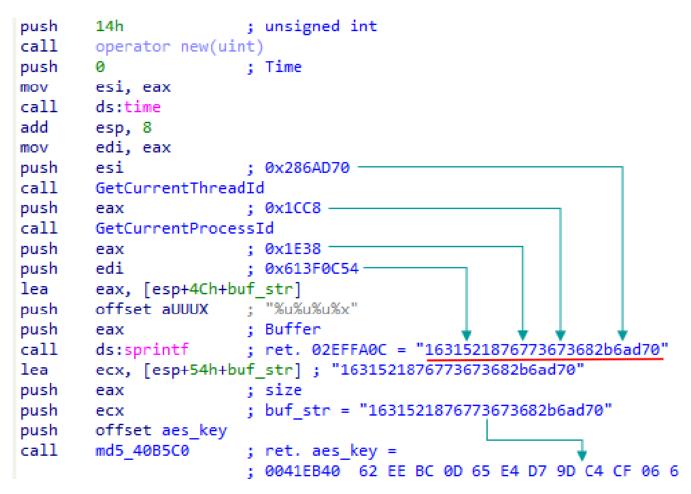
With a high degree of probability, it can be argued that WinDealer has the functionality of interacting with the registry, probably for persistence mechanism:



Encryption

Generate 16 bytes AES key to encrypt C2 communication:

1	.10010.004132C0		, DATA ARLI, HEHSELTI	
	.idata:004152CC ;	<pre>int (*sprintf)(char *const Buffer,</pre>	const char *const Format,)	
•	.idata:004152CC	extrn sprintf:dword	; CODE XREF: sub_402162+381p	
	.idata:004152CC		; sub_402162+7D↑p	
•	and the second s	the second se	and the second	



C2 anti-tracking mechanism

This malware sample employs an IP Generation Algorithm to generate a random C2 IP address when the backdoor lacks C2 configuration. The IP produced at random will exist inside particular IP address ranges:

113.62.0.0 - 113.63.255.255 **or** 111.120.0.0 - 111.123.255.255

111.120.191.65:6999 (UDP) 111.120.234.106:6999 (UDP) 111.120.65.56:6999 (UDP) 111.121.61.108:6999 (UDP) 111.122.152.112:6999 (UDP) 111.122.58.92:6999 (UDP) 111.123.173.15:6999 (UDP) 111.123.9.24:6999 (UDP) 113.62.154.30:6999 (UDP) 113.62.175.80:6999 (UDP)

This mechanism will prevent researchers from tracking down the real C2 IP.

Backdoor .Win32.WINDEALER.ZYJA is a variant of the WinDealer malware family. It is a type of backdoor malware that is designed to allow remote attackers to gain unauthorized access to an infected computer system. Once installed, the malware creates a backdoor on the infected system, which allows the attacker to control the system and steal sensitive data.

The Backdoor.Win32.WINDEALER.ZYJA variant is known to be spread through spear-phishing emails that contain malicious attachments. Once the attachment is opened, the malware is installed and begins to communicate with a remote command-and-control server, allowing the attacker to send commands to the infected system and exfiltrate data.

The malware is capable of performing a range of malicious activities, including stealing credentials and sensitive data, taking screenshots, recording keystrokes, and executing arbitrary commands on the infected system. The malware is also capable of bypassing antivirus and other security software, making it difficult to detect and remove.

IOCs

versions					
Malware	version	md5	sha1		
WinDealer	18.20.1225	76ba5272a17fdab7521ea21a57d23591	6b831413932a394bd9fb25e2bbdc06533821378c		
WinDealer	18.20.1225	cc7207f09a6fe41c71626ad4d3f127ce	84e749c37978f9387e16fab29c7b1b291be93a63		

domain IPs

- 113.62.0.0/15 111.120.0.0/14
- port 55556/TCP, 6999/UDP
- 221.195.68.71/32
- 122.112.245.55/32

Yara rules (from Malpedia)

```
rule win_windealer_auto {
```

```
meta:
    author = "Felix Bilstein - yara-signator at cocacoding dot com"
    date = "2023-01-25"
    version = "1"
    description = "Detects win.windealer."
    info = "autogenerated rule brought to you by yara-signator"
    tool = "yara-signator v0.6.0"
    signator_config = "callsandjumps;datarefs;binvalue"
    malpedia_reference = "https://malpedia.caad.fkie.fraunhofer.de/details/win.windealer"
    malpedia_rule_date = "20230124"
    malpedia_hash = "2ee0eebba83dce3d019a90519f2f972c0fcf9686"
    malpedia_version = "20230125"
    malpedia_license = "CC BY-SA 4.0"
    malpedia_sharing = "TLP:WHITE"
```

```
/* DISCLAIMER
```

```
* The strings used in this rule have been automatically selected from the
```

```
^{\ast} disassembly of memory dumps and unpacked files, using YARA-Signator.
```

- * The code and documentation is published here:
- * https://github.com/fxb-cocacoding/yara-signator
- * As Malpedia is used as data source, please note that for a given
- * number of families, only single samples are documented.
- * This likely impacts the degree of generalization these rules will offer.
- * Take the described generation method also into consideration when you
- * apply the rules in your use cases and assign them confidence levels. $^{\ast/}$

```
strings:
```

```
$sequence_0 = { 668b91d2070000 8a89d0070000 52 51 }
```

// n	= 4, score = 800		
11	668b91d2070000	mov	dx, word ptr [ecx + 0x7d2]
//	8a89d0070000	mov	cl, byte ptr [ecx + 0x7d0]
//	52	push	edx
//	51	push	ecx

//	ff15???????	I	
//	85c0	test	eax, eax
//	7407	je	9
//	50	push	eax
//	ff15???????		
//	6a01	push	1

```
$sequence_2 = { 6a01 50 56 e8??????? 83c410 8bc7 }
```

// n	= 6, score = 800				
//	6a01	l p	push	1	
//	50	l p	push	eax	
//	56	l p	push	esi	
//	e8???????				
//	83c410	6	add	esp,	0x10
//	8bc7	n	mov	eax,	edi

\$sequence_3 = { 6a00 ff15?????? 85c0 7407 50 ff15??????? 6a01 }

•			
// n	= 7, score = 800		
//	6a00	push	Θ
//	ff15???????	1	
//	85c0	test	eax, eax
//	7407	je	9
//	50	push	eax
//	ff15???????	I	
//	6a01	push	1

```
$sequence_4 = { 6a04 50 6a04 68??????? 68??????? }
       // n = 5, score = 800
       // 6a04
                                | push
                                                     4
       11
           50
                                | push
                                                     eax
                                | push
       11
            6a04
                                                     4
            68????????
       11
                                68????????
       11
                                $sequence_5 = { 56 57 68da070000 e8??????? }
       // n = 4, score = 800
       // 56
                                | push
                                                     esi
       // 57
                                | push
                                                     edi
       // 68da070000
                                | push
                                                     0x7da
       // e8???????
                                $sequence_6 = { 50 56 e8?????? 83c410 8b4618 }
       // n = 5, score = 800
       11
          50
                                | push
                                                     eax
       // 56
                                | push
                                                     esi
           e8????????
       11
                                //
          83c410
                                                     esp, 0x10
                                | add
       // 8b4618
                                | mov
                                                     eax, dword ptr [esi + 0x18]
   $sequence_7 = { 8b4d08 668b91d2070000 8a89d0070000 52 51 }
       // n = 5, score = 800
       // 8b4d08
                                                     ecx, dword ptr [ebp + 8]
                                l mov
          668b91d2070000
                                                     dx, word ptr [ecx + 0x7d2]
       11
                                | mov
       // 8a89d0070000
                                | mov
                                                     cl, byte ptr [ecx + 0x7d0]
       11
           52
                                | push
                                                     edx
       11
           51
                                | push
                                                     ecx
   $sequence_8 = { 53 56 57 68da070000 }
       // n = 4, score = 800
       // 53
                                | push
                                                     ebx
       11
            56
                                | push
                                                     esi
       11
            57
                                | push
                                                     edi
       11
            68da070000
                                | push
                                                     0x7da
   $sequence_9 = { 8b4d08 668b91d2070000 8a89d0070000 52 }
       // n = 4, score = 800
       // 8b4d08
                                mov
                                                     ecx, dword ptr [ebp + 8]
       // 668b91d2070000
                                                     dx, word ptr [ecx + 0x7d2]
                                | mov
           8a89d0070000
       11
                                | mov
                                                     cl, byte ptr [ecx + 0x7d0]
       11
            52
                                | push
                                                     edx
condition:
   7 of them and filesize < 770048
```

By Cyber Threat Hunters from MSSPLab:

- @cocomelonc
- @wqkasper

}

Thanks for your time happy hacking and good bye! All drawings and screenshots are MSSPLab's