



WANNACRY RANSOMWARE REPORT

REPORT BY:

Jonathan S.



TABLE OF CONTENTS

Executive Summary	3
High-Level Technical Summary	4
Malware Composition	6
Static Analysis	7
Dynamic Analysis	12
Indicators of Compromise	21
Yara Rules	22
Contact Information	23

EXECUTIVE SUMMARY

SHA-256

24d004a104d4d54034dbcffc2a4b19a11f39008
a575aa614ea04703480b1022c

Wannacry is ransomware that utilized the EternalBlue exploit to propagate through the targets network and attacked outdated Windows computers globally in May of 2017. WannaCry was a multistage attack starting with a dropper that unpacked a payload onto the targets system under the right conditions. Once the files were encrypted, the threat actors demanded a ransom of \$300 worth of Bitcoin. If the ransom is not paid in a specified amount of time, the ransom is increased to \$600. This attack infected around 230,000 computers across 150 countries. Marcus Hutchins later discovered a kill switch that stalled the spread of the attack. [Click here to view the full analysis report.](#)

HIGH-LEVEL TECHNICAL SUMMARY

For the 12 months to December 1, 2025

WannaCry consist of 2 stages, the first stage being a dropper that tries to make contact with a suspicious URL that can be found in the strings `hxxp[:]//]iuqerfsodp9ifjaposdfjhgosu rijfaewrwergrwea[.]com` if a connection is established the program exits, if a connection is not established the program proceeds with the rest of the execution. Once the program proceeds with execution a service is created by the program `mssecsvc2.0` and has the display name Microsoft Security Center (2.0) Service. The service also contains a path to the executable `<PATH_TO_WANNACRY>\wannacr y.exe -m security`. During this stage the program will attempt to propagate by reaching out to a large range of IPv4 addresses.

Stage two the payload is unpacked from the dropper and proceeds to create persistence mechanisms such as creating a folder in the `C:\ProgramData\<GENERATED_STRING>\` directory and creating a file named `tasksche.exe` in the `C:\Windows\` path and copying itself to the newly created directory. Once the file has been copied to the directory, a service is created and is named after the same generated string as the newly created folder and contains a path leading to the payload `C:\ProgramData\<GENERATED_STRING>\tasksche.exe`. After the service is created and the payload is executed the encryption process starts which changes the background image, drops instructions on how to decrypt the files and more in the generated directory.

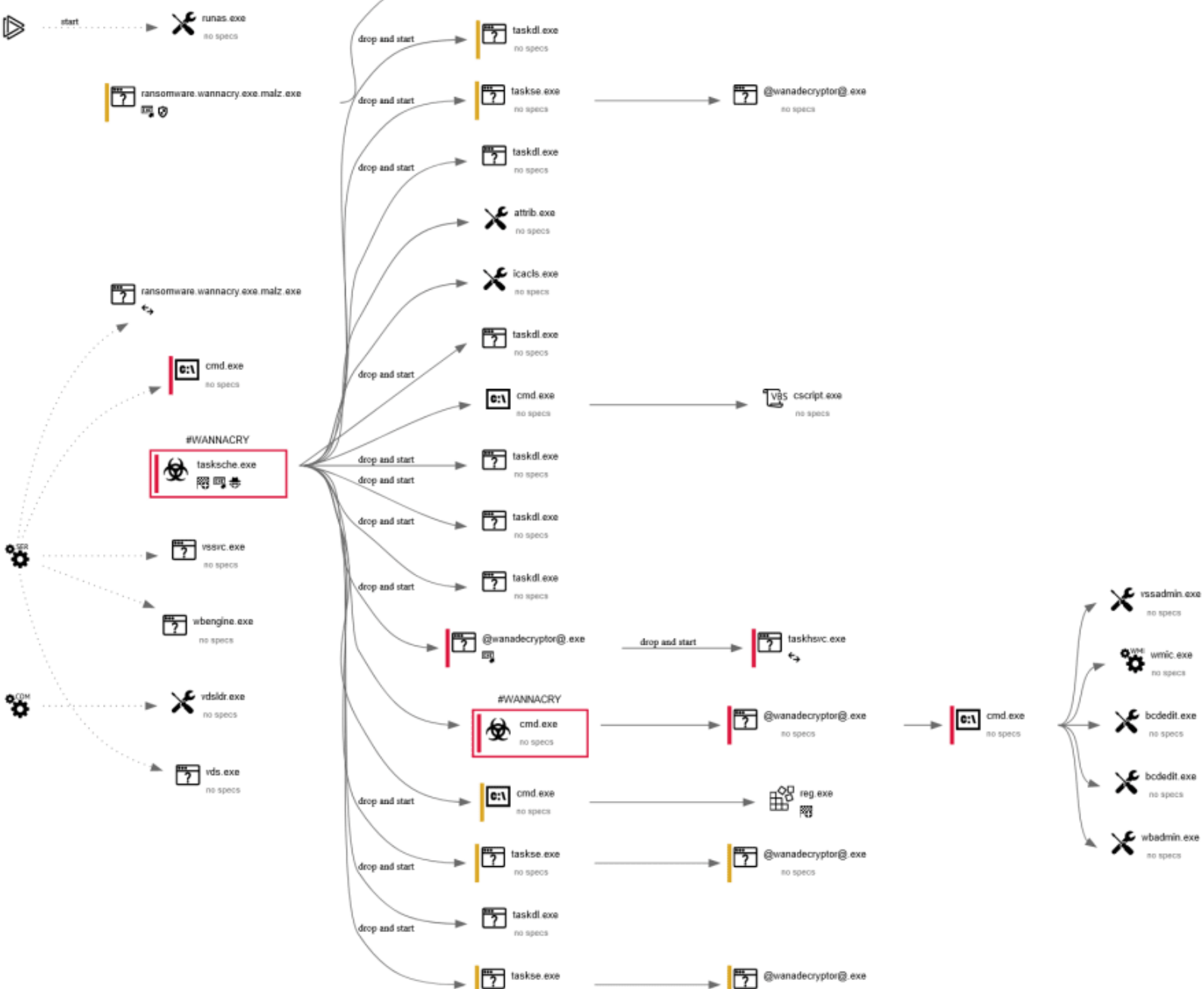


Fig.1 Execution flow graph

MALWARE COMPOSITION

ITEM	Description	SHA-256
Ransomware.wannacry.exe	Initial file detonated	24d004a104d4d54034dbcffc2a4b 19a11f39008a575aa614ea047034 80b1022c
tasksche.exe	The payload unpacked from the dropper	ed01ebfbc9eb5bbea545af4d01bf5 f1071661840480439c6e5babe8e0 80e41aa
@WanaDecryptor@[.].exe	The GUI application that is executed by tasksche after all files have been encrypted and handles ransom payment	b9c5d4339809e0ad9a00d4d3dd2 6fdf44a32819a54abf846bb9b560 d81391c25
taskdl.exe	SQL Client Configuration Utility EXE	4a468603fdcb7a2eb5770705898cf9ef37a ade532a7964642ecd705a74794b79
taskhsvc.exe	Handles communication to TOR URL and other TOR activites	e48673680746fbe027e8982f62a8 3c298d6fb46ad9243de8e79b7e5a 24dcd4eb
taskse.exe	Waitfor - Wait/send a signal over a network	2ca2d550e603d74dedda03156023 135b38da3630cb014e3d00b12633 58c5f00d

STATIC ANALYSIS

Summary

Analyzing the strings reveals a suspicious URL, later on during the advanced static analysis phase, we can see that the URL is moved to the ESI register and later pushed to the stack as it is used as a parameter in the InternetOpenUrlA function. After the InternetOpenUrlA function has been called, the dropper checks to see if the connection to the URL was successful or not. If the connection succeeds the program exits, otherwise the program continues with execution as seen in (Fig.2). The original filename (*lhdfrgui.exe*) of the dropper can be found in the "Version" tab of PE Studio (Fig.3). Upon further inspection of the dropper, the date the executable was compiled was also spotted (Fig.4). Heading over to the "Imports" tab of PE Studio, we can see that the dropper utilizes a few network, cryptography, and services functions such as

- InternetOpenA
- InternetOpenUrlA
- CryptGenRandom
- CreateServiceA
- StartServiceCtrlDispatcherA
- connect
- socket

The full list of imports can be found in (Fig.7). While inspecting the droppers' headers, an executable was spotted in the .rsrc header as shown in (Fig.5).

While analyzing the dropper in Cutter, a reference to the payload (*tasksche.exe*) is seen being pushed to the stack along with the location the payload will be dropped which is the *C:\Windows* location (Fig.6). There is a reference to a file named "*qeriuwjhrf*" in the same location the payload will be dropped, upon further investigation there was no such file created (Fig.6). Bitcoin addresses were also found in the payload (Fig.8).

STATIC ANALYSIS

Images



FIG.2 REFERENCE TO DNS QUERY URL STRING.

STATIC ANALYSIS

Images

FileDescription	Microsoft® Disk Defragmenter
FileVersion	6.1.7601.17514 (win7sp1_rtm.101119-1850)
InternalName	lhdfrgui.exe
LegalCopyright	© Microsoft Corporation. All rights reserved.
OriginalFilename	lhdfrgui.exe
ProductName	Microsoft® Windows® Operating System
ProductVersion	6.1.7601.17514

FIG.3 ORIGINAL NAME OF DROPPER.

compiler-stamp	0x4CE78ECC	Sat Nov 20 09:03:08 2010 UTC
----------------	------------	--------------------------------

FIG.4 TIMESTAMP THE DROPPER WAS COMPILED.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
32090	A4	A0	66	00	B0	03	00	00	E4	04	00	00	00	00	00	00		н	.	ф	.	*
320A0	01	00	52	00	4D	5A	90	00	03	00	00	00	00	04	00	00		.	.	Р	.	М	З	
320B0	FF	FF	00	00	B8	00	00	00	00	00	00	00	00	40	00	00		у	у	
320C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
320D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
320E0	F8	00	00	00	0E	1F	BA	0E	00	B4	05	CD	21	B8	01	4C		е	
320F0	CD	21	54	68	69	73	20	70	72	6F	67	72	61	6D	20	63		і	!	T	h	i	s	.	p	r	o	g	r	a	m	.	c
32100	61	6E	6E	6F	74	20	62	65	20	72	75	6E	20	69	6E	20		a	n	n	o	t	.	b	e	.	r	u	n	.	i	n	.
32110	44	4F	53	20	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00		D	O	S	.	m	o	d	e	
32120	00	00	00	00	E0	C5	3A	D1	A4	A4	54	82	A4	A4	54	82		
32130	A4	A4	54	82	DF	B8	58	82	A6	A4	54	82	CB	BB	5F	82		
32140	A5	A4	54	82	27	B8	5A	82	A0	A4	54	82	CB	BB	5E	82		н	м	т	.	а	.	А	:	Н	м	т	.	м	т	.	
32150	AF	A4	54	82	CB	BB	50	82	A0	A4	54	82	67	AB	09	82		н	м	т	.	а	.	Х	:	!	м	т	.	Е	»	.	
32160	A9	A4	54	82	A4	A4	55	82	07	A4	54	82	92	92	5F	82		У	м	т	
32170	A3	A4	54	82	63	A2	52	82	A5	A4	54	82	52	69	63	68		—	м	т	.	Е	»	Р	
32180	A4	A4	54	82	00	00	00	00	00	00	00	00	00	00	00	00		©	м	т	.	м	м	У	
32190	00	00	00	00	00	00	00	00	00	00	00	00	00	50	45	00		н	м	т	.	с	с	Р	
321A0	4C	01	04	00	41	8F	E7	4C	00	00	00	00	00	00	00	00		н	м	т	
321B0	E0	00	0F	01	0B	01	06	00	70	00	00	00	20	35	00		а	

FIG.5 IMAGE OF EXECUTABLE IN THE .RSRC HEADER OF THE DROPPER.

STATIC ANALYSIS

Images

```

push    str.WINDOWS          ; 0x431364
lea     eax, [lpExistingFileName]
push    str.C:___s___s      ; 0x431358 ; tasksche.exe payload
push    eax
call    esi
add     esp, 0x10
lea     ecx, [lpNewFileName]
push    str.WINDOWS          ; 0x431364
push    str.C:___s___qeriuwjhrf ; 0x431344
push    ecx
call    esi
add     esp, 0xc
lea     edx, [lpNewFileName]
lea     eax, [lpExistingFileName]
push    1                    ; 1 ; DWORD dwFlags
push    edx                  ; LPCSTR lpNewFileName
push    eax                  ; LPCSTR lpExistingFileName
call    dword [MoveFileExA]   ; 0x40a04c ; BOOL MoveFileExA(LPCSTR lpExistingFileName, LPCST...
push    ebx
push    4                    ; 4
push    2                    ; 2
push    ebx
push    ebx
lea     ecx, [var_7ch]
push    0x40000000
push    ecx
call    dword [0x431458]      ; CreateFileA
mov     esi, eax
cmp     esi, 0xffffffff
je      0x407f08

```

[0x00407e54]

```

mov     eax, dword [var_10h_2]
lea     edx, [var_10h_2]
push    ebx
push    edx
push    ebp
push    eax
push    esi
call    dword [0x431460]      ; WriteFile
push    esi
call    dword [0x43144c]      ; CloseHandle

```

FIG.6 REFERENCE TO PACKED PAYLOAD IN CUTTER.

STATIC ANALYSIS

Images

functions (91)	blacklist (29)	anonymous (13)	library (7)
GetCurrentThreadId	x	-	kernel32.dll
GetCurrentThread	x	-	kernel32.dll
MoveFileExA	x	-	kernel32.dll
TerminateThread	x	-	kernel32.dll
QueryPerformanceFrequency	x	-	kernel32.dll
StartServiceCtrlDispatcherA	x	-	advapi32.dll
ChangeServiceConfig2A	x	-	advapi32.dll
CreateServiceA	x	-	advapi32.dll
CryptGenRandom	x	-	advapi32.dll
CryptAcquireContextA	x	-	advapi32.dll
3 (closesocket)	x	x	ws2_32.dll
16 (recv)	x	x	ws2_32.dll
19 (send)	x	x	ws2_32.dll
8 (htonl)	x	x	ws2_32.dll
14 (ntohl)	x	x	ws2_32.dll
115 (WSAStartup)	x	x	ws2_32.dll
12 (inet_ntoa)	x	x	ws2_32.dll
10 (ioctlsocket)	x	x	ws2_32.dll
18 (select)	x	x	ws2_32.dll
9 (htons)	x	x	ws2_32.dll
23 (socket)	x	x	ws2_32.dll
4 (connect)	x	x	ws2_32.dll
11 (inet_addr)	x	x	ws2_32.dll
GetAdaptersInfo	x	-	iphlpapi.dll
InternetOpenA	x	-	wininet.dll
InternetOpenUrlA	x	-	wininet.dll
InternetCloseHandle	x	-	wininet.dll
rand	x	-	msvcrt.dll
srand	x	-	msvcrt.dll

FIG.7 IMPORTS OF INTEREST IN THE DROPPER.

```

mov dword ptr ss:[ebp-c],tasksche.40F48 40F488:"13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94"
mov dword ptr ss:[ebp-8],tasksche.40F46 40F464:"12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw"
mov dword ptr ss:[ebp-4],tasksche.40F44 40F440:"115p7UMMngo1pMvvpHijcRdfJNXj6LrLn"

```

FIG.8 BITCOIN ADDRESSES FOUND IN THE PAYLOAD

DYNAMIC ANALYSIS

Summary

Once the dropper is executed as administrator, a DNS query is made to the suspicious (*hxxp[://]iuqerfsodp9ifjaposdfjhgosurijfaewrwergweal[.]com*) URL mentioned in (Fig. 2). As stated in the static analysis section, if the dropper receives an HTTP 200 response, the program exits. If the program does not receive a response from the DNS query the program proceeds with the rest of the execution. We can see the dropper making the DNS query in Wireshark shown in (Fig. 9). After the DNS query the dropper proceeds and pushes two arguments to the stack *<PATH_TO_WANNACRY>* and *-m security* which are then passed as parameters to the `CreateServiceA` function. The strings *mssecsvc2.0* and *Microsoft Security Center (2.0) Service* are also pushed to the stack in preparation for the creation of the service. The program proceeds to create a service named *mssecsvc2.0* with the display name of *Microsoft Security Center (2.0) Service* as seen in (Fig. 10). After the service is created and executed, the dropper attempts to connect to a range of IPv4 addresses on port 445 (SMB) using the EternalBlue exploit (Fig. 11). As the dropper attempts to connect to the range of IPv4 addresses, the payload is being unpacked from the dropper and is executed (Fig. 12). The payload generates a string based on the hostname of the system and creates a folder named after the generated string in the *C:\ProgramData* directory.

After the creation of the directory, a copy of the payload is moved to the directory (Fig. 13). Along with the creation of the new directory, a service is also created with the same generated name as the directory which uses `cmd` to execute *tasksche* as a persistence mechanism (Fig. 14). Once the service is created, a registry key named *WanaCrypt0r* and registry key value named *wd* is created with the key-value set to the newly created directory in *C:\ProgramData\<RANDOMLY_GENERATED_STRING>* (Fig. 15). After the payload has been executed by `cmd`, the encryption process begins. An executable named *WanaDecryptor@.exe* is dropped along with various other files in the same directory as the payloads' execution and creates a shortcut to the *@WanaDecryptor* executable on the Desktop (Fig. 16). Lastly, the system background is changed and a GUI of the *@WanaDecryptor@.exe* executable is displayed (Fig. 17).

DYNAMIC ANALYSIS

Images

3	3.765142	11.14.56.128	11.14.56.128	ICMP	260 Destination unreachable (Host unreachable)
4	7.764941	11.14.56.128	11.14.56.128	ICMP	260 Destination unreachable (Host unreachable)
5	11.765132	11.14.56.128	11.14.56.128	ICMP	260 Destination unreachable (Host unreachable)


```

> Frame 4: 260 bytes on wire (2080 bits), 137 bytes captured (1096 bits) on interface \Device\NPF_{B5D36D2C-DDB0-45E8-A0E0-0AF805857B22}, id 0
> Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 11.14.56.128, Dst: 11.14.56.128
> Internet Control Message Protocol
  > Domain Name System (query)
    Transaction ID: 0x49d0
    > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  > Queries
    > www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com: type A, class IN
  
```


0000	00 00 00 00 00 00 00 00	00 00 00 00 08 00 45 00E..
0010	00 7b de 9e 00 00 00 01	00 00 0b 0e 38 80 0b 0e	..{.....8...
0020	38 80 03 01 4a 75 00 00	00 00 45 00 00 5f 2f 8e	8...Ju...E.../_
0030	00 00 80 11 00 00 0b 0e	38 80 0b 0e 38 81 e4 2b8...8...+
0040	00 35 00 4b 87 79 49 d0	01 00 00 01 00 00 00 00	..5.K.yI.....
0050	00 00 03 77 77 77 29 69	75 71 65 72 66 73 6f 64	...www)i uqerfsod
0060	70 39 69 66 6a 61 70 6f	73 64 66 6a 68 67 6f 73	p9ifjapo sdfjhgos
0070	75 72 69 6a 66 61 65 77	72 77 65 72 67 77 65 61	urijfaew rwergwea
0080	03 63 6f 6d 00 00 01 00	01	.com.....

FIG.9 DROPPER MAKING DNS QUERY TO SUSPICIOUS URL.

DYNAMIC ANALYSIS

Images

00407C4B	68 60F77000	push ransomware.wannacry.70F760	70F760: "C:\\Users\\Lab\\Desktop\\Ransomware.wannacry.exe"
00407C50	68 30134300	push ransomware.wannacry.431330	431330: "%s -m security"
00407C55	50	push eax	
00407C56	FF15 0CA14000	call dword ptr ds:[<&sprintf>]	
00407C5C	83C4 0C	add esp,C	
00407C5F	68 3F000F00	push F003F	
00407C64	6A 00	push 0	
00407C66	6A 00	push 0	
00407C68	FF15 10A04000	call dword ptr ds:[<&OpenSCManager>]	
00407C6E	8BF8	mov edi,eax	
00407C70	85FF	test edi,edi	
00407C72	74 56	je ransomware.wannacry.407CCA	
00407C74	53	push ebx	
00407C75	56	push esi	
00407C76	6A 00	push 0	
00407C78	6A 00	push 0	
00407C7A	6A 00	push 0	
00407C7C	6A 00	push 0	
00407C7E	8D4C24 1C	lea ecx,dword ptr ss:[esp+1C]	
00407C82	6A 00	push 0	
00407C84	51	push ecx	ecx: "C:\\Users\\Lab\\Desktop\\Ransomware.wannacry.exe -m security"
00407C85	6A 01	push 1	
00407C87	6A 02	push 2	
00407C89	6A 10	push 10	
00407C8B	68 FF010F00	push F01FF	
00407C90	68 08134300	push ransomware.wannacry.431308	431308: "Microsoft Security Center (2.0) Service"
00407C95	68 FC124300	push ransomware.wannacry.4312FC	4312FC: "mssecsvc2.0"
00407C9A	57	push edi	
00407C9B	FF15 14A04000	call dword ptr ds:[<&CreateServiceA>]	
00407CA1	8B1D 18A04000	mov ebx,dword ptr ds:[<&CloseServiceHan	
00407CA7	8BF0	mov esi,eax	
00407CA9	85F6	test esi,esi	
00407CAB	74 0E	je ransomware.wannacry.407CBB	
00407CAD	6A 00	push 0	
00407CAF	6A 00	push 0	
00407CB1	56	push esi	
00407CB2	FF15 1CA04000	call dword ptr ds:[<&StartServiceA>]	

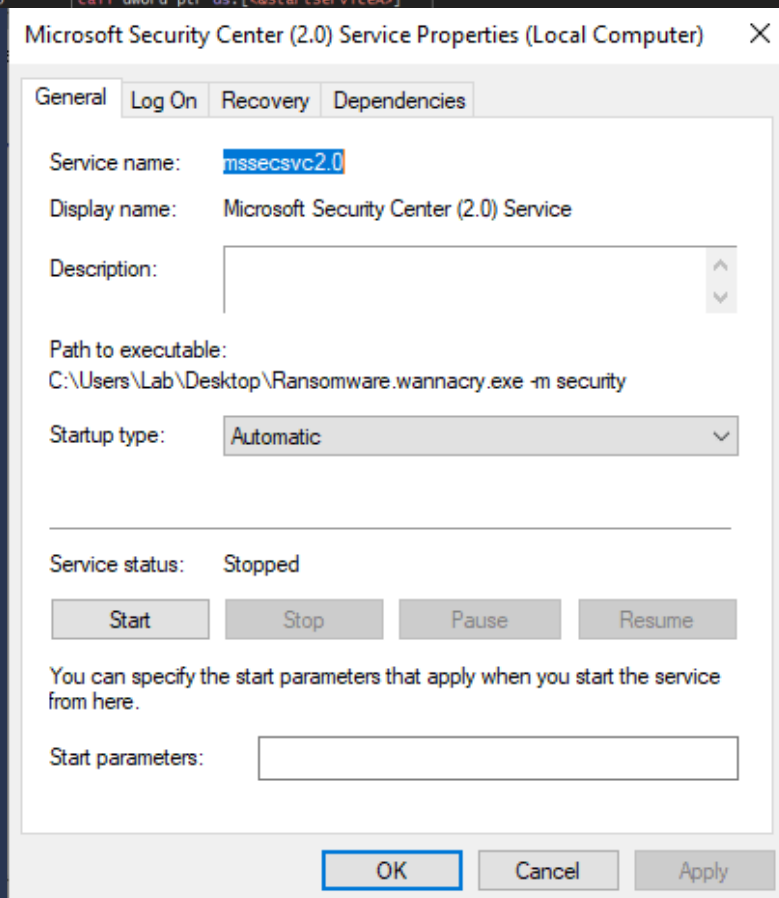


FIG.10 DROPPER CREATES SERVICE AS A
PERSISTENCE MECHANISM

DYNAMIC ANALYSIS

Images

Ransomware.wannacr...	3932	TCP	Syn Sent	169.254.185.116	49892	169.254.140.1	445	4/9/2022 1:13:41 AM	mssecsvc2.0
Ransomware.wannacr...	3932	TCP	Syn Sent	169.254.185.116	49893	169.254.141.1	445	4/9/2022 1:13:41 AM	mssecsvc2.0
Ransomware.wannacr...	3932	TCP	Syn Sent	169.254.185.116	49894	169.254.142.1	445	4/9/2022 1:13:41 AM	mssecsvc2.0
Ransomware.wannacr...	3932	TCP	Syn Sent	169.254.185.116	49896	169.254.143.1	445	4/9/2022 1:13:41 AM	mssecsvc2.0
Ransomware.wannacr...	3932	TCP	Syn Sent	169.254.185.116	49897	169.254.144.1	445	4/9/2022 1:13:41 AM	mssecsvc2.0
Ransomware.wannacr...	3932	TCP	Syn Sent	169.254.185.116	49898	169.254.145.1	445	4/9/2022 1:13:41 AM	mssecsvc2.0
Ransomware.wannacr...	3932	TCP	Syn Sent	169.254.185.116	49900	169.254.146.1	445	4/9/2022 1:13:41 AM	mssecsvc2.0
Ransomware.wannacr...	3932	TCP	Syn Sent	169.254.185.116	49901	169.254.147.1	445	4/9/2022 1:13:41 AM	mssecsvc2.0
Ransomware.wannacr...	3932	TCP	Syn Sent	169.254.185.116	49904	169.254.148.1	445	4/9/2022 1:13:41 AM	mssecsvc2.0
Ransomware.wannacr...	3932	TCP	Syn Sent	169.254.185.116	49906	169.254.149.1	445	4/9/2022 1:13:42 AM	mssecsvc2.0
Ransomware.wannacr...	3932	TCP	Syn Sent	169.254.185.116	49908	169.254.150.1	445	4/9/2022 1:13:42 AM	mssecsvc2.0

FIG.11 SERVICE ATTEMPTS TO REACH OUT TO A RANGE OF IPV4.

00407ED1	52	push edx	edx: "C:\\WINDOWS\\tasksche.exe /i"
00407ED2	53	push ebx	
00407ED3	C74424 4C 44000000	mov dword ptr ss:[esp+4C],44	44: 'D'
00407ED8	66:895C24 7C	mov word ptr ss:[esp+7C],bx	
00407EE0	C74424 78 81000000	mov dword ptr ss:[esp+78],81	
00407EE8	FF15 78144300	call dword ptr ds:[<&CreateProcessA>]	

tasksche.exe	5308	Running	SYSTEM	23	15,656 K	Yes	Not allowed
--------------	------	---------	--------	----	----------	-----	-------------

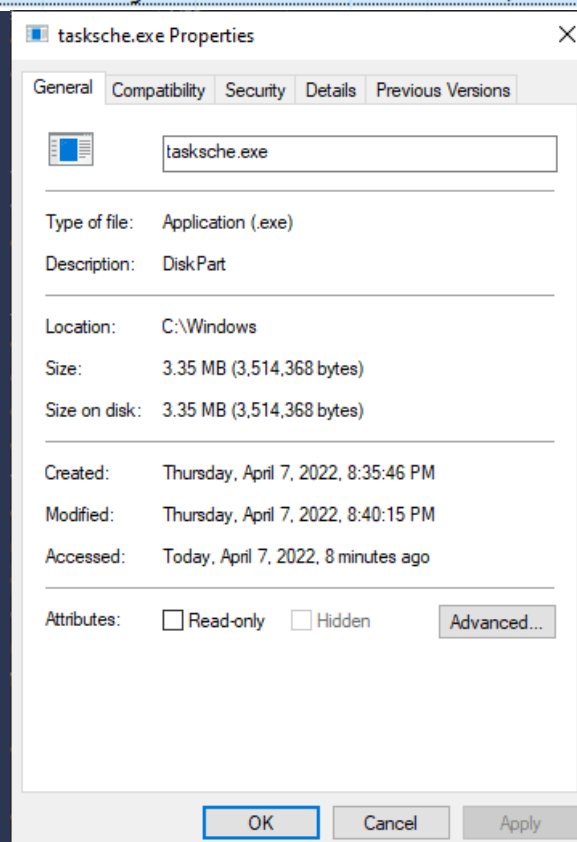
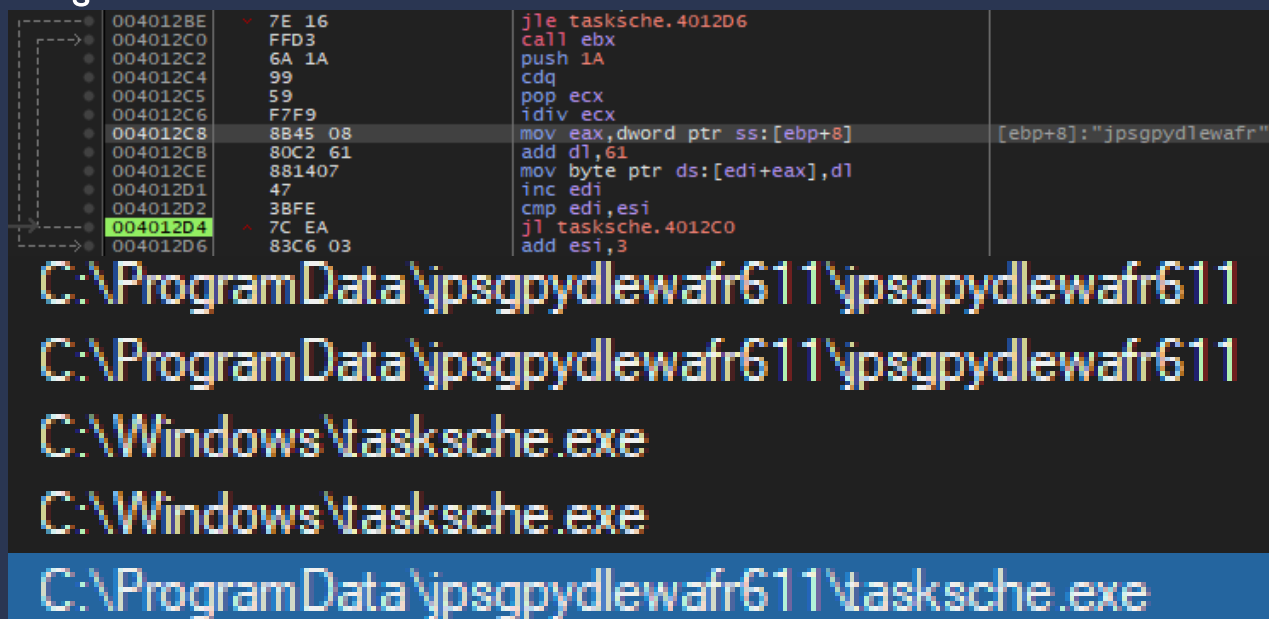


FIG.12 PAYLOAD IS UNPACKED
FROM DROPPER.

DYNAMIC ANALYSIS

Images



The screenshot displays a debugger window with assembly code on the left and a list of file paths on the right. The assembly code is as follows:

Address	Disassembly
004012BE	7E 16 jle taskscche.4012D6
004012C0	FFD3 call ebx
004012C2	6A 1A push 1A
004012C4	99 cdq
004012C5	59 pop ecx
004012C6	F7F9 idiv ecx
004012C8	8B45 08 mov eax,dword ptr ss:[ebp+8]
004012CB	80C2 61 add dl,61
004012CE	8B1407 mov byte ptr ds:[edi+eax],dl
004012D1	47 inc edi
004012D2	3BFE cmp edi,esi
004012D4	7C EA jl taskscche.4012C0
004012D6	83C6 03 add esi,3

The file paths listed on the right are:

- C:\ProgramData\jpsgpydlewafr611\jpsgpydlewafr611
- C:\ProgramData\jpsgpydlewafr611\jpsgpydlewafr611
- C:\Windows\tasksche.exe
- C:\Windows\tasksche.exe
- C:\ProgramData\jpsgpydlewafr611\tasksche.exe

FIG.13 PAYLOAD GENERATES RANDOM STRING BASED ON THE SYSTEM NAME, CREATES A FOLDER IN THE C:\PROGRAMDATA DIRECTORY WITH THE GENERATED NAME AND COPIES THE PAYLOAD TO THE GENERATED DIRECTORY.

DYNAMIC ANALYSIS

Images

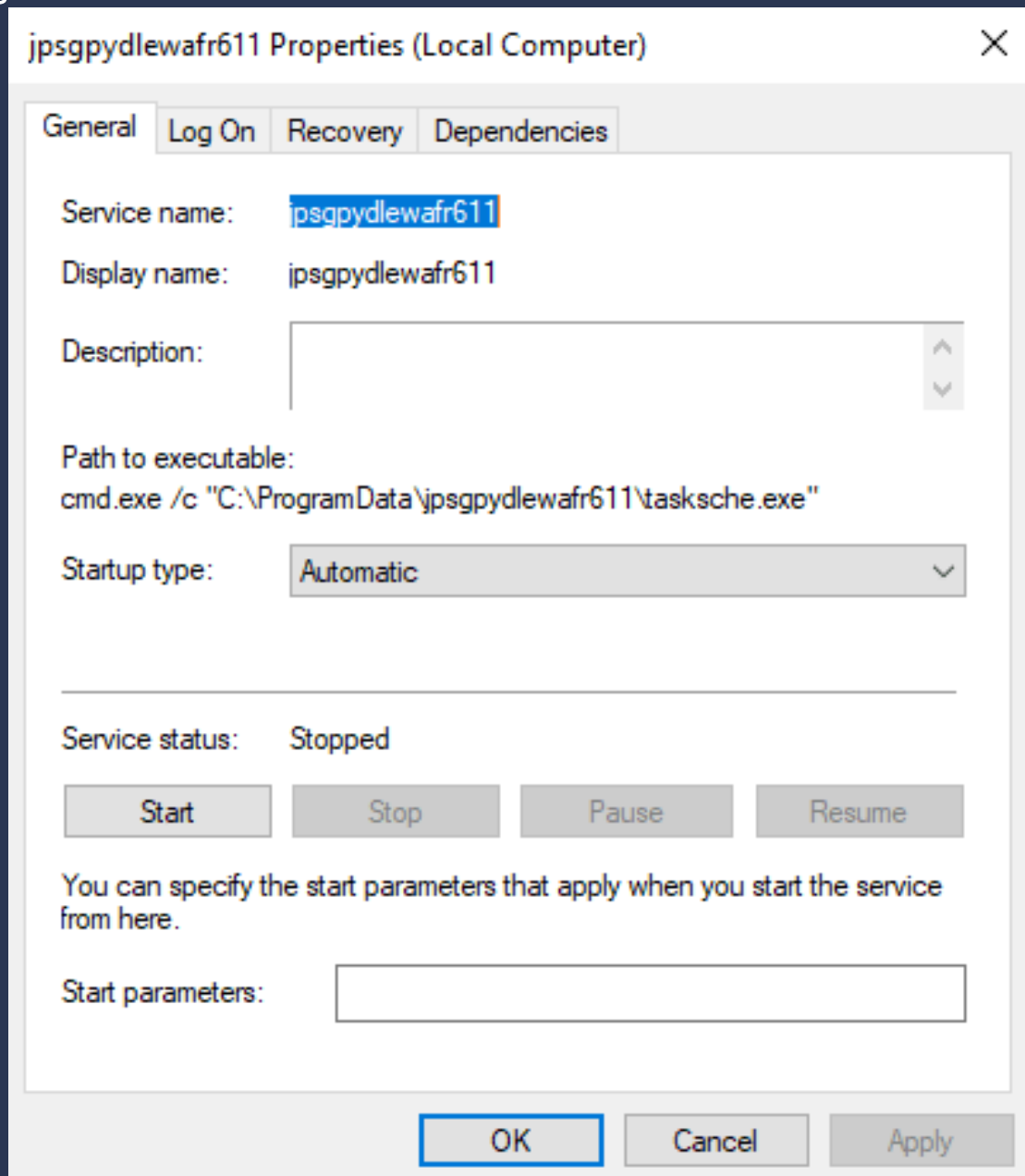


FIG.14 SERVICE IS CREATED WITH THE SAME NAME AS THE GENERATED STRING.

DYNAMIC ANALYSIS

Images

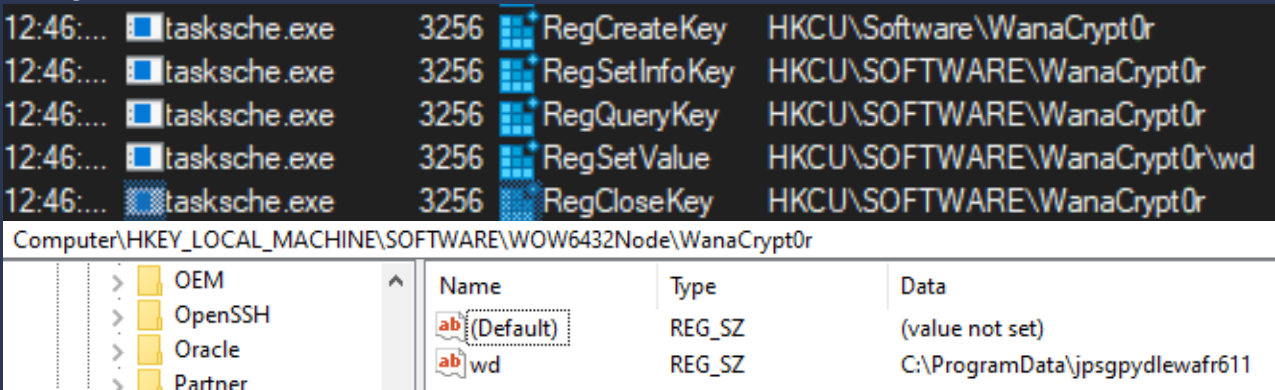


FIG.15 REGISTRY KEY CREATED BY PAYLOAD SERVICE.

DYNAMIC ANALYSIS

Images


















 msg	4/9/2022 6:31 PM	File folder	
 @Please_Read_Me@.txt	4/9/2022 6:30 PM	Text Document	1 KB
 @WanaDecryptor@.exe	5/12/2017 2:22 AM	Application	240 KB
 @WanaDecryptor@.exe	4/9/2022 6:30 PM	Shortcut	1 KB
 00000000.eky	4/9/2022 6:30 PM	EKY File	0 KB
 00000000.pky	4/9/2022 6:30 PM	PKY File	1 KB
 00000000.res	4/9/2022 6:38 PM	RES File	1 KB
 b.wnry	5/11/2017 8:13 PM	WNRy File	1,407 KB
 c.wnry	4/9/2022 6:30 PM	WNRy File	1 KB
 f.wnry	4/9/2022 6:31 PM	WNRy File	1 KB
 r.wnry	5/11/2017 3:59 PM	WNRy File	1 KB
 s.wnry	5/9/2017 4:58 PM	WNRy File	2,968 KB
 t.wnry	5/12/2017 2:22 AM	WNRy File	65 KB
 taskdl.exe	5/12/2017 2:22 AM	Application	20 KB
 tasksche.exe	4/9/2022 6:30 PM	Application	3,432 KB
 taskse.exe	5/12/2017 2:22 AM	Application	20 KB
 u.wnry	5/12/2017 2:22 AM	WNRy File	240 KB

FIG.16 FILES DROPPED FROM PAYLOAD AFTER ENCRYPTION PROCESS HAS
BEGUN.

DYNAMIC ANALYSIS

Images

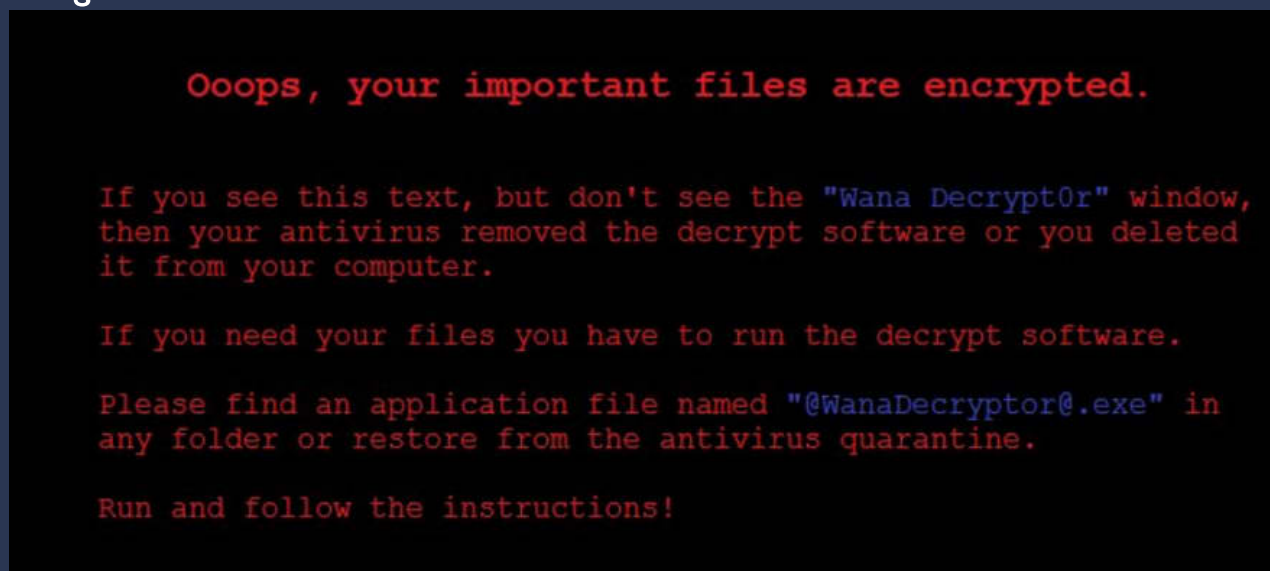


FIG.17 BACKGROUND CHANGED AND GUI APPLICATION
DISPLAYED

INDICATORS OF COMPROMISE

Network Indicators

- Dropper observed making DNS Query to a suspicious domain (Fig. 9).
(*hxxp[:]//]iuqerfsodp9ifjaposdfjhgosurijfaewr
wergwea[.]com*)
- Payload attempts to establish contact with a range of IPv4 addresses (Fig. 11).

INDICATORS OF COMPROMISE

Host-Based Indicators

- Payload is unpacked onto the system in C:\Windows. (Note) During the debugging process, there was a mention of a file in the directory C:\Windows named qeriuwjhrf but the file was never created (Fig. 12).
- Creation of services *mssecsvc2.0* (Fig. 10) and a service with a name randomly generated based on the system name (Fig. 14).
- Creation of registry key *HKCU\SOFTWARE\WanaCrypt0r\wd* (Fig. 15).
- Creation of files following the execution of the payload in the same directory as the execution. Along with files ending in the .WNCRY extension (Fig. 16).
- Background change and appearance of GUI application. (Fig. 17).

YARA RULES

```
rule wannacry_ruleset {
  meta:
    last_updated = "04-09-2022"
    author = "IAANSEC"
    description = "Yara rule to detect wannacry ransomware."
    hash256 = "24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c"

  strings:
    $MZ_byte = "MZ"
    $querydomain_killswitch = "iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea" ascii
    $weird_windows_dir_str = "qeriuwjhrf" ascii
    $reg_name = "WanaCrypt0r" ascii
    $service = "Microsoft Security Center (2.0) Service" ascii
    $payload = "tasksche" ascii
    $exe1 = "taskdl" ascii
    $exe2 = "taskse" ascii
    $import = "Crypt" ascii
    $str = "WNcry@2017" ascii
    $decrypt_exe = "@WanaDecryptor@.exe" ascii
    $wnry = "wnry" ascii
    $decrypt = "decrypt" ascii
    $bitcoin = "bitcoin" ascii
    $btc_wallet1 = "115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn" fullword ascii
    $btc_wallet2 = "13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94" fullword ascii
    $btc_wallet3 = "12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw" fullword ascii

  condition:
    $MZ_byte at 0 and
    5 of them
}
```

CONTACT.



IAANSEC

<https://iaansec.com>