WANNACRY RANSOMWARE REPORT

REPORT BY:

Jonathan S.

TABLE OF CONTENTS

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Wannacry is ransomware that utilized the EternalBlue exploit to propagate through the targets network and attacked outdated Windows computers globally in May of 2017.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 or scan the QR code above or visit the link below.

https://bit.ly/3JnXpd3

HIGH-LEVEL TECHNICAL SUMMARY

For the 12 months to December 1, 2025

WannaCry consists 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[://]iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]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>\wannacry.exe -m security*. During this stage, the program will attempt to propagate by reaching out to a large range of IPv4 addresses.

In 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.
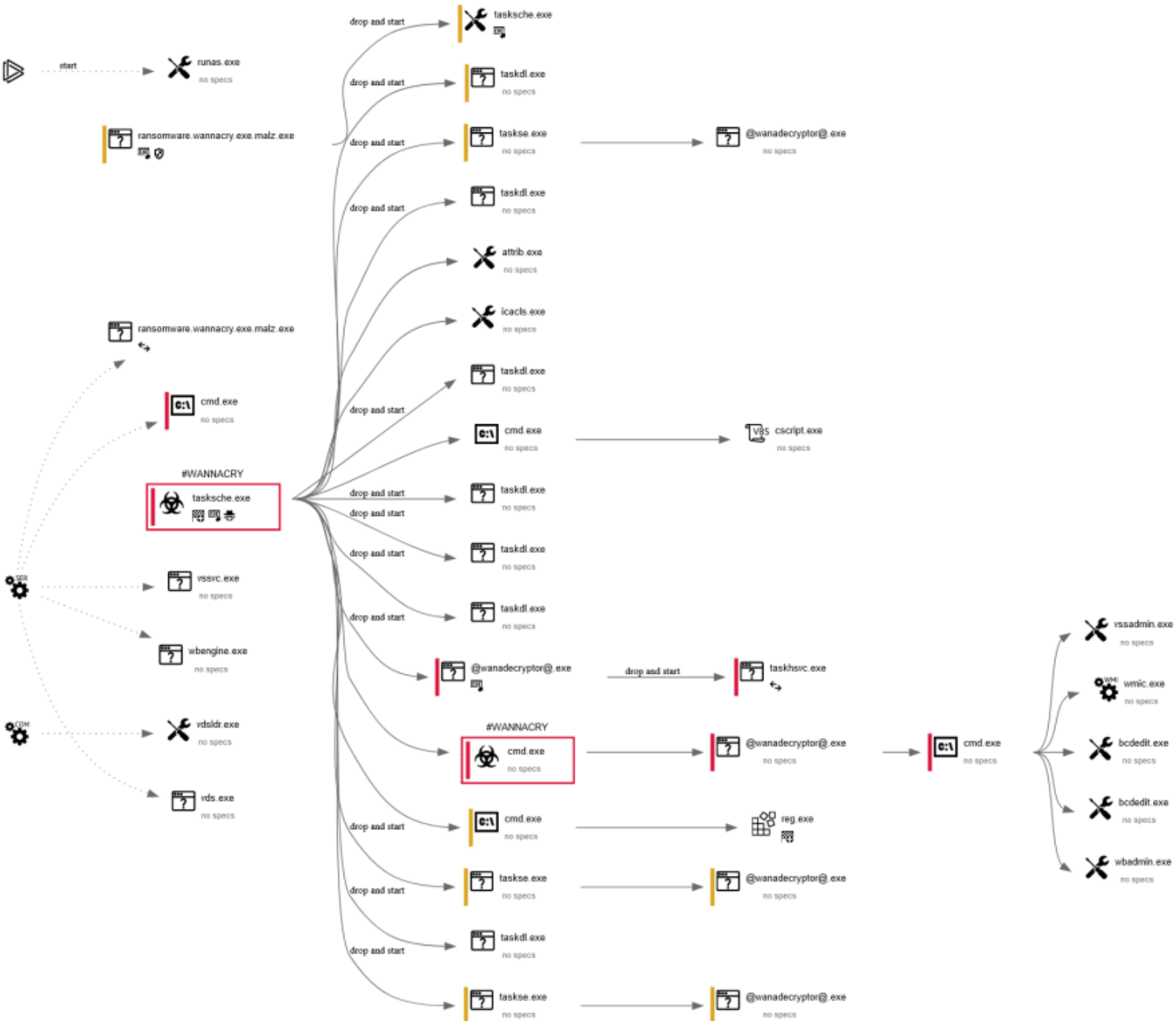
Fig.1 Execution flow graph

MALWARE COMPOSITION

ITEM	Description	SHA-256
Ransomware.wannacry.exe	Initial file detonated	24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c
tasksche.exe	The payload unpacked from the dropper	ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa
@WanaDecryptor@[.]exe	The GUI application that is executed by tasksche after all files have been encrypted and handles ransom payment	b9c5d4339809e0ad9a00d4d3dd26fdf44a32819a54abf846bb9b560d81391c25
taskdl.exe	SQL Client Configuration Utility EXE	4a468603fdcb7a2eb5770705898cf9ef37aade532a7964642ecd705a74794b79
taskhsvc.exe	Handles communication to TOR URL and other TOR activites	e48673680746fbe027e8982f62a83c298d6fb46ad9243de8e79b7e5a24dcd4eb
taskse.exe	Waitfor - Wait/send a signal over a network	2ca2d550e603d74dedda03156023135b38da3630cb014e3d00b126 33 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 the imports shown in the imports section to the right. 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)*.

IMPORTS
- InternetOpenA: InternetOpenA is used to initialize the use of WinINet functions.
- InternetOpenUrlA: InternetOpenUrlA is used to open a resource specified by a complete FTP or HTTP URL.
- CryptGenRandom: CryptGenRandom is used to fill a buffer with cryptographically random bytes.
- CreateServiceA: CreateServiceA is used to create a service object and adds it to the specified service control manager database. This function is commonly used by malware for persistence.
- StartServiceCtrlDispatcherA: StartServiceCtrlDispatcherA is used by a service to connect the main thread of the process to the service control manager.
- connect: Connect is used to establish a connection to a specified socket.
- socket: Socket is used to create a socket that is bound to a specific transport service provider.

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.

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

STATIC ANALYSIS

Images

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:"115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn"
```

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[://]iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]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 and executed along with *attrib +h .* to hide the current directory the payload was copied along with *icacls . /grant Everyone:F /T /C /Q* to grant full permissions 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* are 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). (Note) The following strings can be found in the c.wnry file dropped by the payload:*

- *gx7ekbenv2riucmf.onion*
- *57g7spgrzlojinas.onion*
- *xxlvbrloxvriy2c5.onion*
- *76jdd2ir2embyv47.onion*
- *cwwnhwhlz52maqm7.onion*
- *https://dist.torproject.org/torbrowser/6.5. 1/tor-win32-0.2.9.10.zip*

DYNAMIC ANALYSIS

Images

FIG.9 DROPPER MAKING DNS QUERY TO SUSPICIOUS URL.

DYNAMIC ANALYSIS

Images

*FIG.10 DROPPER CREATES SERVICE AS A
PERSISTENCE MECHANISM*

DYNAMIC ANALYSIS

Images

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

*FIG.12 PAYLOAD IS UNPACKED
FROM DROPPER.*

DYNAMIC ANALYSIS

Images

```
004012BE    7E 16        jle taksche.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]     [ebp+8]:"jpsgpydlewafr"
004012CB    80C2 61      add dl,61
004012CE    881407       mov byte ptr ds:[edi+eax],dl
004012D1    47           inc edi
004012D2    3BFE         cmp edi,esi
004012D4    7C EA        jl taksche.4012C0
004012D6    83C6 03      add esi,3
```

```
C:\ProgramData\jpsgpydlewafr611\jpsgpydlewafr611

C:\ProgramData\jpsgpydlewafr611\jpsgpydlewafr611

C:\Windows\tasksche.exe

C:\Windows\tasksche.exe

C:\ProgramData\jpsgpydlewafr611\tasksche.exe
```

```
[0x004020b4]
lea eax, [lpExistingFileName]
push eax                          ; LPCSTR lpPathName
call dword [SetCurrentDirectoryA] ; 0x4080d8 ; BOOL SetCurrentDirectoryA(LPCSTR lpPathName)
push 1                            ; 1 ; uint32_t arg_8h
call fcn.004010fd
mov dword [esp], str.WNcry_2ol7   ; 0x40f52c ; int32_t arg_ch
push ebx                          ; HMODULE hModule
call fcn.00401dab
call fcn.00401e9e
push ebx
push ebx
push str.attrib__h_.              ; 0x40f520 ; LPSTR lpCommandLine ; attrib +h . (Payload execut...
call fcn.00401064
push ebx
push ebx
push str.icacls_.__grant_Everyone:F__T__C__Q ; 0x40f4fc ; LPSTR lpCommandLine ; icacls . /grant E...
call fcn.00401064    push str.icacls_.__grant_Everyone:F_T_C_Q;0x40f4fc; LPSTR lpCommandLine; icacls ./grant Everyone:F /T /C /Q
add esp, 0x20
call fcn.0040170a
test eax, eax
je 0x402165
```
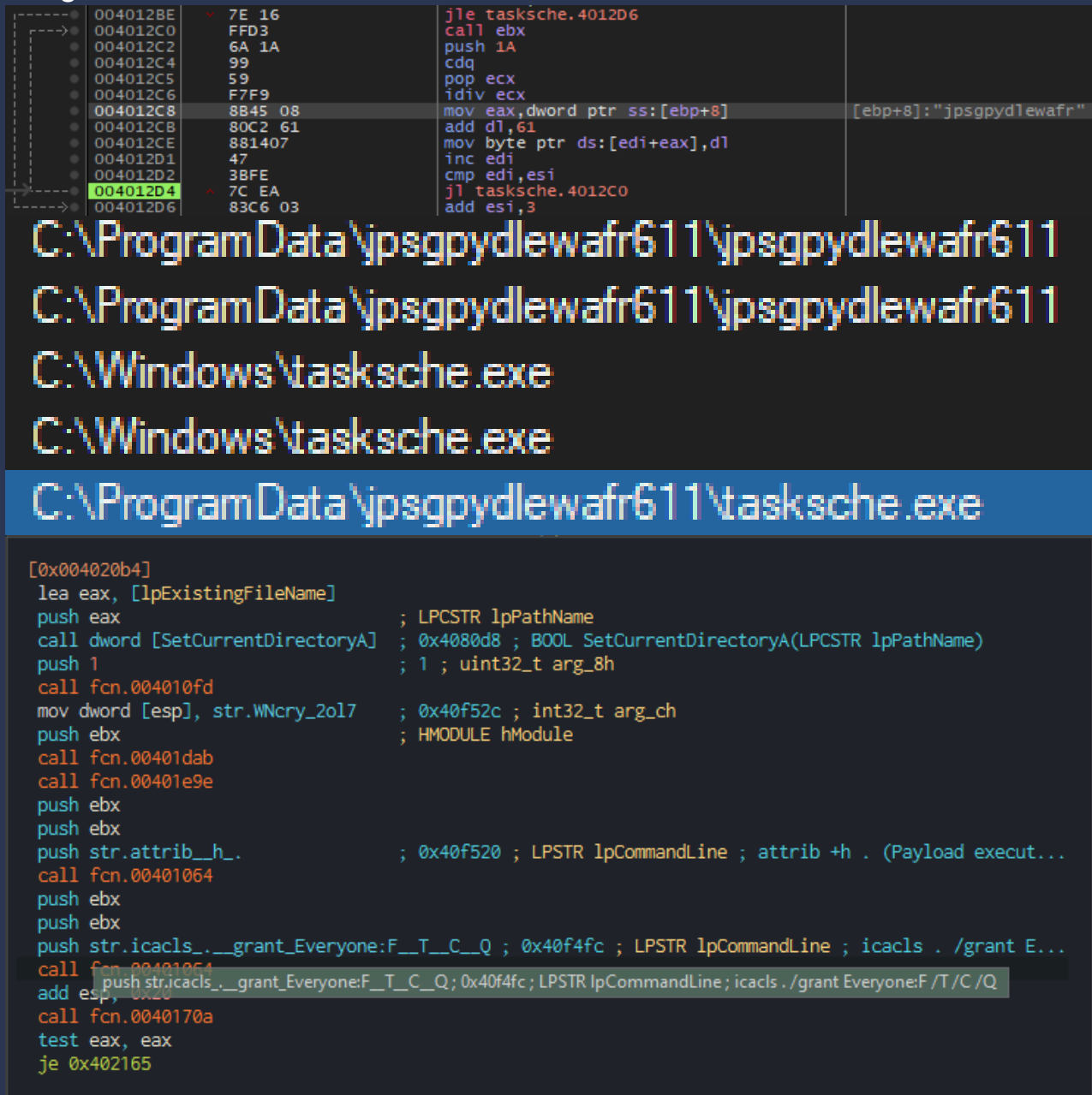
FIG.13 PAYLOAD CREATES DIRECTORY BASED ON SYSTEM NAME AND ADDS HIDDEN ATTRIBUTE TO FOLDER AND GRANTS PERMISSIONS TO SAID DIRECTORY
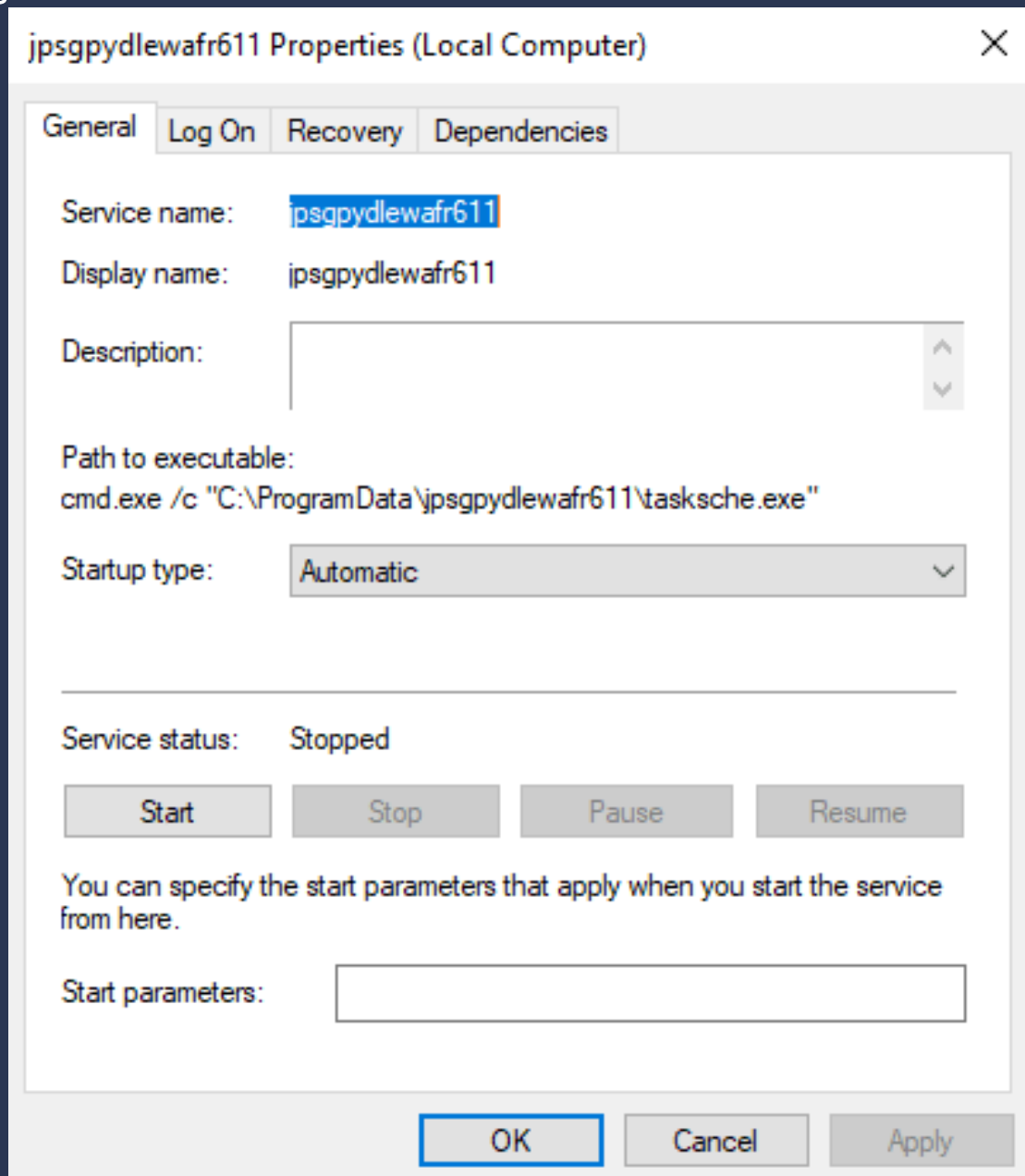
DYNAMIC ANALYSIS

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
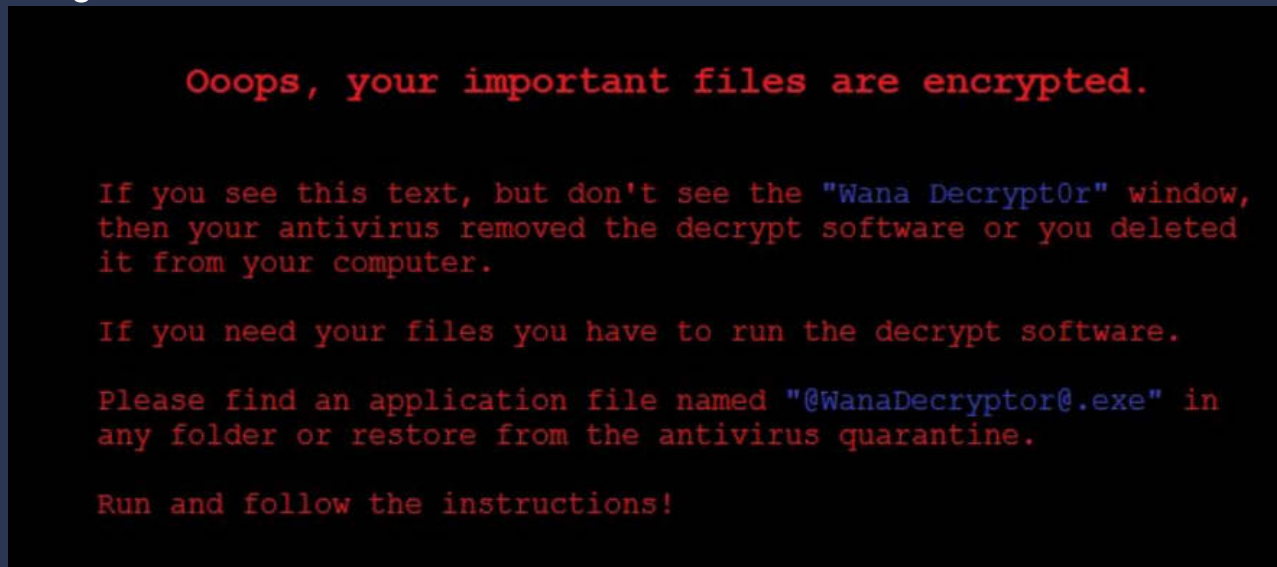
Network Indicators

- *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.