Malware Analysis Report

PMAT - Practical Malware Analysis & Triage

Course Final

Ransomware.wannacry.exe

Jul 2023 | Analysis by Harpocrat3s | v1.1

Table of Contents

Executive Summary

WannaCry is a ransomware worm that targets Windows OS. Once launched, it encrypts files on the computer, changes the desktop wallpaper to indicate the successful infection of the machine, and presents a software interface displaying a ransom note. The ransom note provides instructions on where to send the Bitcoin payment in order to regain access to the encrypted files.

Additionally, WannaCry possesses persistence mechanisms, enabling it to remain on the target machine even after a reboot, and worm capabilities, allowing it to attempt the spread to other IP addresses.

It is important to note that this malware includes a killswitch mechanism. When successfully connected to a specific URL, the malware doesn't execute its malicious actions.

YARA signature rules are attached in Appendix A.

High-Level Technical Summary

File name	Ransomware.wannacry.exe
md5	db349b97c37d22f5ea1d1841e3c89eb4
sha1	e889544aff85ffaf8b0d0da705105dee7c97fe26
sha256	24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c

This particular sample of WannaCry is distributed as a 32-bit executable file and requires administrative privileges to execute its malicious actions.

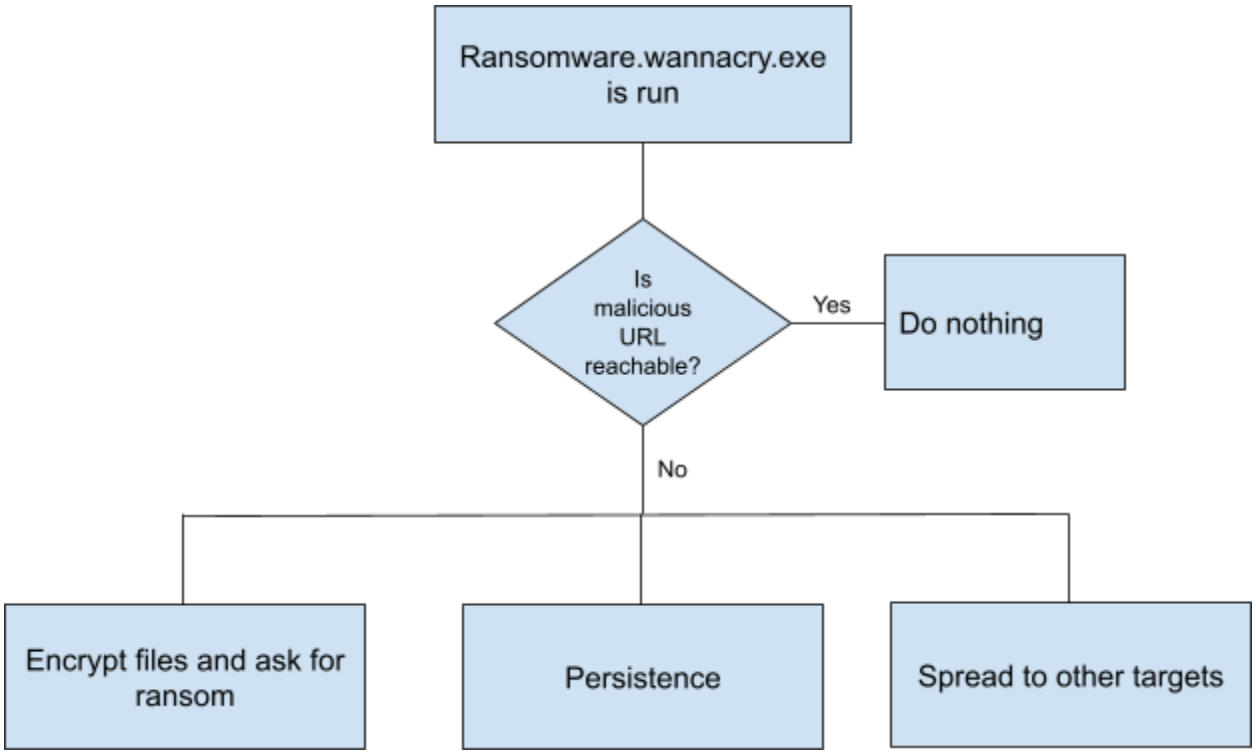
As a first step, it attempts to establish a connection with the URL "**hxxp[://]www[.]iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]com**". If it receives a response from this domain, the malware terminates its routine and remains inactive. Essentially, it acts as a kill-switch.

If the domain is unreachable. (e.g. it doesn't exist), the program proceeds to execute its main function. This involves unpacking additional executables hidden within the file, one of which is an encryptor. Utilizing the Microsoft Enhanced RSA and AES Cryptographic Provider libraries, this encryptor encrypts the files on the targeted machine.

Subsequently, the malware changes the desktop wallpaper, replacing it with an image containing a message that explains the successful infection and provides instructions for recovery.

Additionally, it installs a decryptor program on the desktop, which is executed shortly after the infection. The decryptor program includes the ransom note and instructions on how to make the ransom payment and decrypt the files on the system. The program also displays two timers, indicating the deadlines for the ransom amount to double and the final deadline to pay the ransom and recover the files.

In terms of its worm capabilities, this malware initiates multiple connections to various IP addresses on port 445, which is commonly used for the SMB protocol. It exploits the EternalBlue vulnerability to propagate its infection.


Ransomware.wannacry.exe is run
Is malicious URL reachable?
Yes
Do nothing
No
Encrypt files and ask for ransom
Persistence
Spread to other targets

Malware Composition

This WannaCry sample consists of the following components:

Ransomware.wannacry.exe

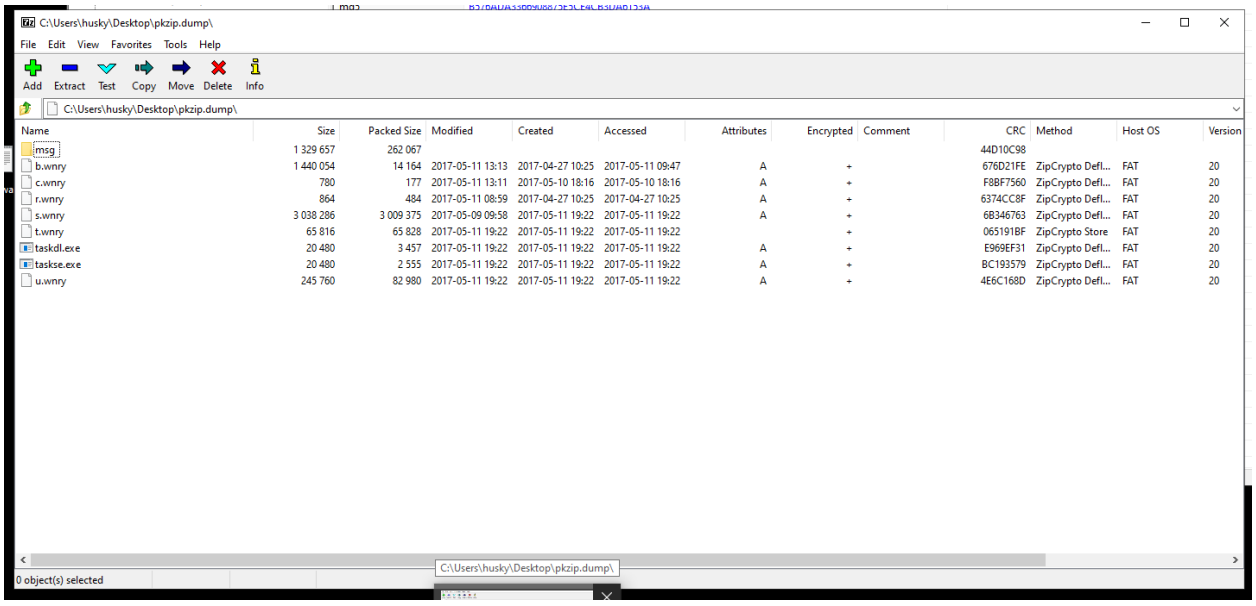
This is the initial executable that is executed when clicked or launched.

Unpacked executables

There's an executable found inside the resources section of the original executable, and additional executables within it.

PKZIP file

Contained inside the unpacked executable, there's a password protected zip file. It contains a folder named "msg" that appears to contain translations of the ransom note. Additionally, the zip file includes two executables (taskdl.exe and taskse.exe) along with supplementary files..



Basic Static Analysis

The basic static analysis of the sample was conducted using FLOSS, capa, and PEStudio. Each program's unique characteristics were utilized to gather comprehensive information about the sample.

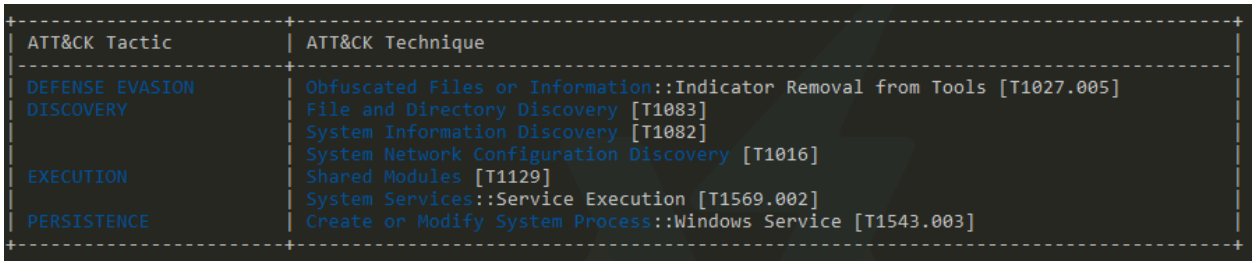
During the analysis, several strings were identified that exhibited indications of potentially malicious activity:

WanaCrypt0r	name of the malware itself
icacls . /grant Everyone:F /T /C /Q	command-line command used to change control access to files and folders
attrib +h .	command used to set the current folder as “hidden”
Microsoft Enhanced RSA and AES Cryptographic Provider	cryptography capabilities
hxxp[://]www[.]iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]com	the (defanged) URL
taskdl.exe taskse.exe diskpart.exe lhdfrgui.exe	some filenames
C:\%s\qeriuwjhrf	suspicious folder name
WriteFile CreateFileA CreateProcessA	examples of API calls found in the sample
str_%s -m securit_00431330r	command (part of it)
u.wnry	one of many .wnry file names
13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94	The bitcoin wallet used in this sample, found by searching through the strings after getting the wallet address during the basic dynamic analysis

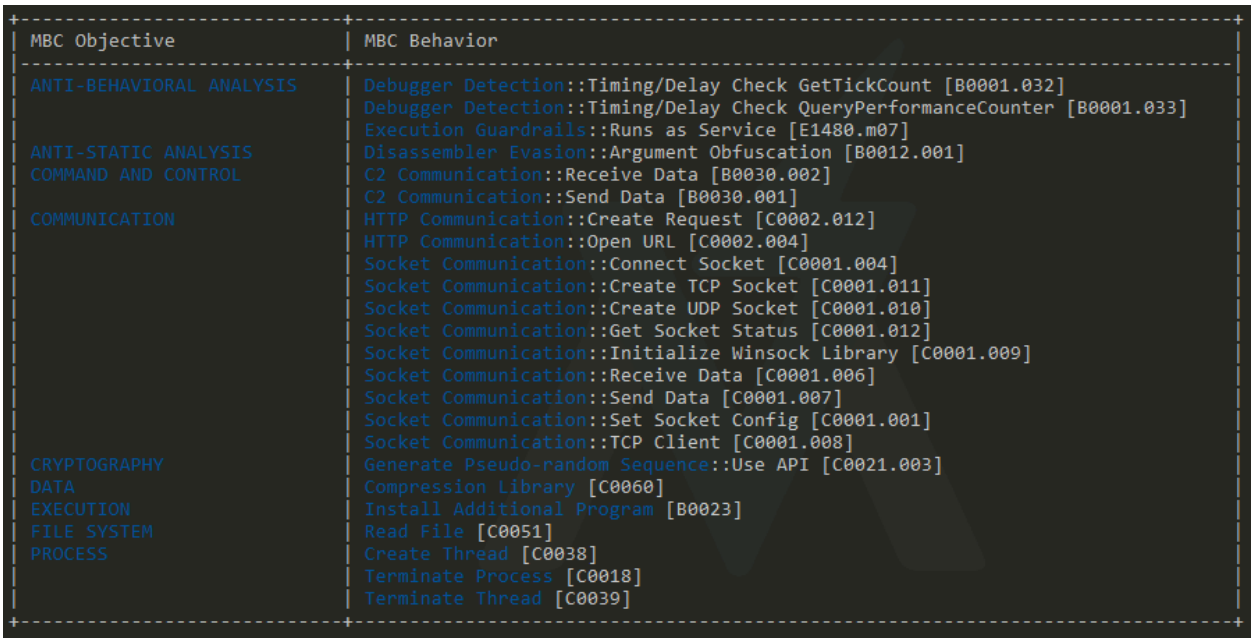
115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn 12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw	Two other bitcoin wallets shown in strings and floss together with the main one, above. These 2 weren't used in the sample. Please check the appendices.
\\172.16.99.5\IPC$ \\192.168.56.20\IPC$	SMB shares
cmd.exe /c "%s"	a command

CAPA found some indicators that would prove correct during the dynamic analysis, like the communication and cryptography abilities, installation of files and creation of processes for persistence.

Here are the screenshots of the results:



ATT&CK Tactic	ATT&CK Technique
DEFENSE EVASION	Obfuscated Files or Information::Indicator Removal from Tools [T1027.005]
DISCOVERY	File and Directory Discovery [T1083]
	System Information Discovery [T1082]
	System Network Configuration Discovery [T1016]
EXECUTION	Shared Modules [T1129]
	System Services::Service Execution [T1569.002]
PERSISTENCE	Create or Modify System Process::Windows Service [T1543.003]



MBC Objective	MBC Behavior
ANTI-BEHAVIORAL ANALYSIS	Debugger Detection::Timing/Delay Check GetTickCount [B0001.032]
	Debugger Detection::Timing/Delay Check QueryPerformanceCounter [B0001.033]
	Execution Guardrails::Runs as Service [E1480.m07]
ANTI-STATIC ANALYSIS	Disassembler Evasion::Argument Obfuscation [B0012.001]
COMMAND AND CONTROL	C2 Communication::Receive Data [B0030.002]
	C2 Communication::Send Data [B0030.001]
COMMUNICATION	HTTP Communication::Create Request [C0002.012]
	HTTP Communication::Open URL [C0002.004]
	Socket Communication::Connect Socket [C0001.004]
	Socket Communication::Create TCP Socket [C0001.011]
	Socket Communication::Create UDP Socket [C0001.010]
	Socket Communication::Get Socket Status [C0001.012]
	Socket Communication::Initialize Winsock Library [C0001.009]
	Socket Communication::Receive Data [C0001.006]
	Socket Communication::Send Data [C0001.007]
	Socket Communication::Set Socket Config [C0001.001]
	Socket Communication::TCP Client [C0001.008]
CRYPTOGRAPHY	Generate Pseudo-random Sequence::Use API [C0021.003]
DATA	Compression Library [C0060]
EXECUTION	Install Additional Program [B0023]
FILE SYSTEM	Read File [C0051]
PROCESS	Create Thread [C0038]
	Terminate Process [C0018]
	Terminate Thread [C0039]

CAPABILITY	NAMESPACE
check for time delay via GetTickCount	anti-analysis/anti-debugging/debugger-detection
check for time delay via QueryPerformanceCounter	anti-analysis/anti-debugging/debugger-detection
contain obfuscated stackstrings	anti-analysis/obfuscation/string/stackstring
receive data (5 matches)	communication
send data (5 matches)	communication
connect to URL	communication/http/client
get socket status	communication/socket
initialize Winsock library	communication/socket
set socket configuration	communication/socket
create UDP socket (4 matches)	communication/socket/udp/send
act as TCP client	communication/tcp/client
generate random numbers via WinAPI	data-manipulation/prng
contain a resource (.rsrc) section	executable/pe/section/rsrc
extract resource via kernel32 functions	executable/resource
contain an embedded PE file	executable/subfile/pe
get file size	host-interaction/file-system/meta
move file	host-interaction/file-system/move
read file	host-interaction/file-system/read
get number of processors	host-interaction/hardware/cpu
get networking interfaces	host-interaction/network/interface
terminate process	host-interaction/process/terminate
run as service	host-interaction/service
create service	host-interaction/service/create
modify service	host-interaction/service/modify
start service	host-interaction/service/start
create thread (4 matches)	host-interaction/thread/create
terminate thread	host-interaction/thread/terminate
link function at runtime	linking/runtime-linking
linked against ZLIB	linking/static/zlib
inspect section memory permissions	load-code/pe
parse PE exports	load-code/pe
parse PE header	load-code/pe
persist via Windows service	persistence/service

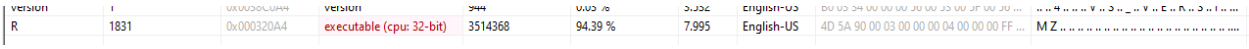
Some findings from PEStudio:

0x0000DDFB	x	-	cryptography	CryptAcquireContext
0x0000DE14	x	-	cryptography	CryptGenRandom
0x0000DFF2	x	-	cryptography	rand
0x0000E022	x	-	cryptography	srand
0x0032B15D	x	-	cryptography	CryptReleaseContext
0x0032B233	x	-	cryptography	rand
0x0032B23C	x	-	cryptography	srand
0x0032C982	x	-	cryptography	CryptGenKey
0x0032C98E	x	-	cryptography	CryptDecrypt
0x0032C99E	x	-	cryptography	CryptEncrypt
0x0032C9AE	x	-	cryptography	CryptDestroyKey
0x0032C9BE	x	-	cryptography	CryptImportKey
0x0032C9CE	x	-	cryptography	CryptAcquireContext
0x0001F647	-	-	windowing	GetLastActivePopup

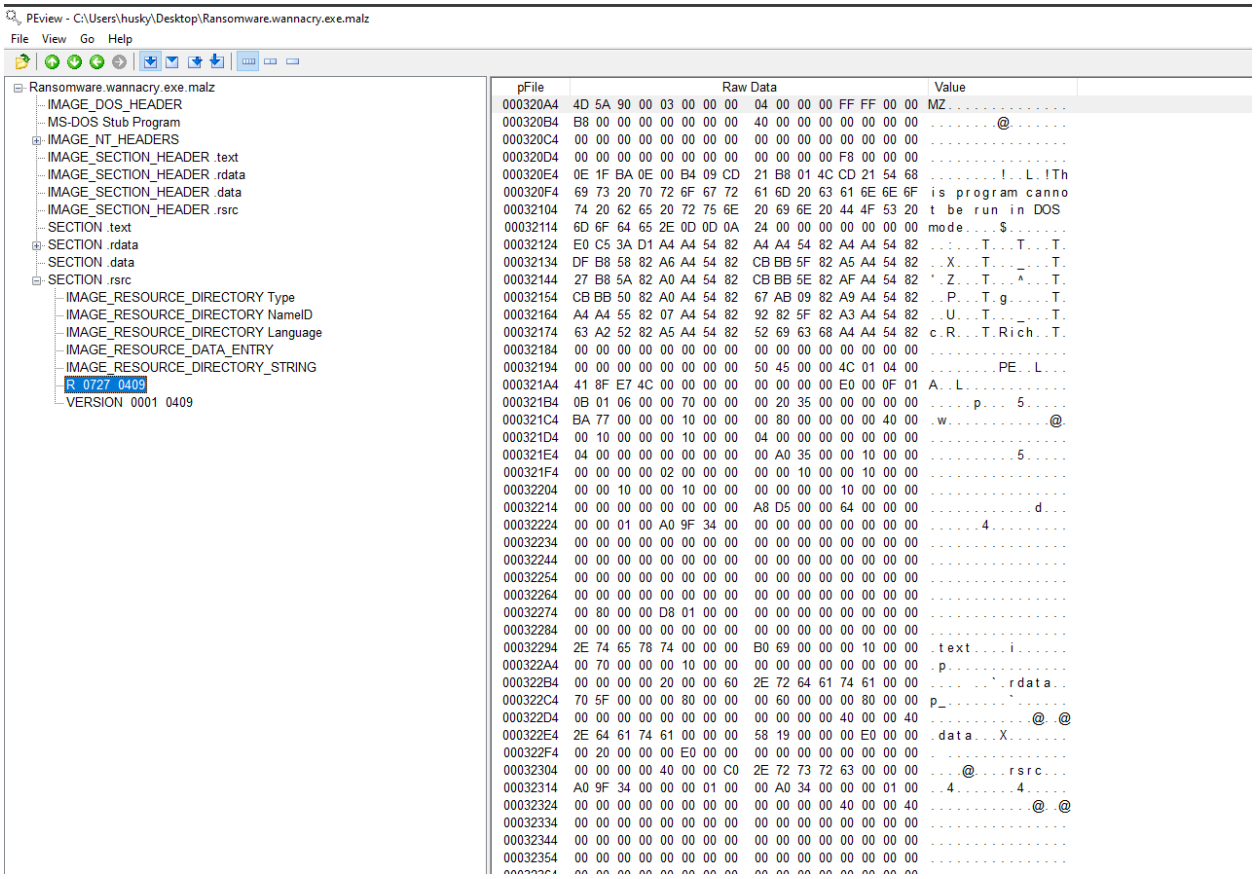
API calls showing cryptographic abilities

property	value
md5	1EBDC36976DD611E1A9E221A88E6858E
sha1	7B5A93CD7DB3DDC7FF48C6E3C7EEFCA46807462E
sha256	2F3FC51546ADA848DFC8E775554C0DE3689D6FAE7BA4BF3D40E3C8DEC68B277B
file-type	executable
date	empty
language	English-US
code-page	Unicode UTF-16, little endian
CompanyName	Microsoft Corporation
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

One of the strings found using FLOSS seems to be the original filename of the sample

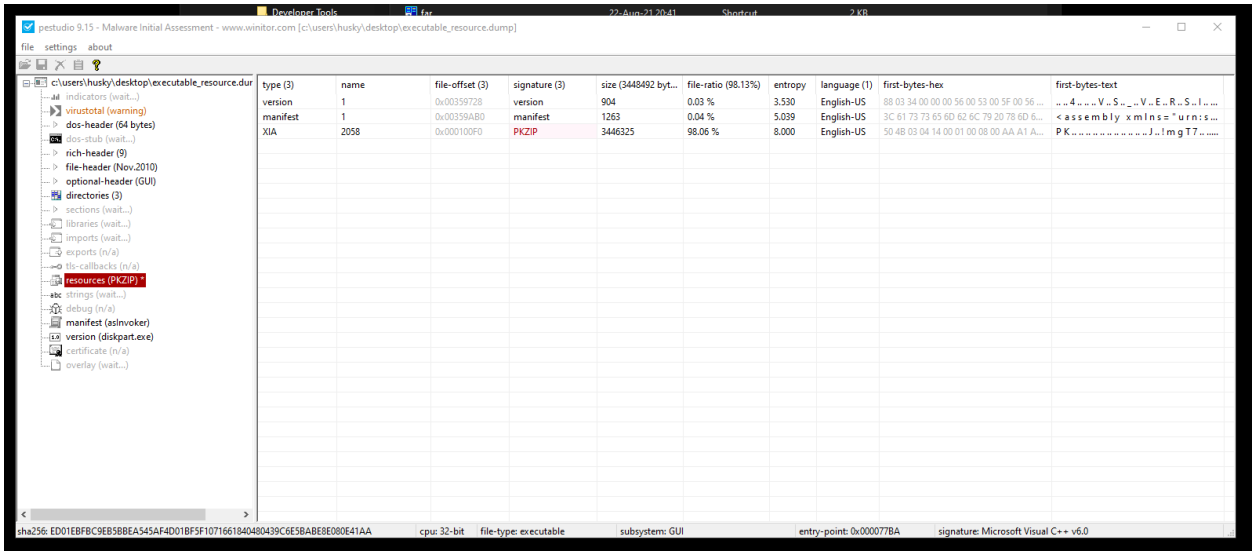


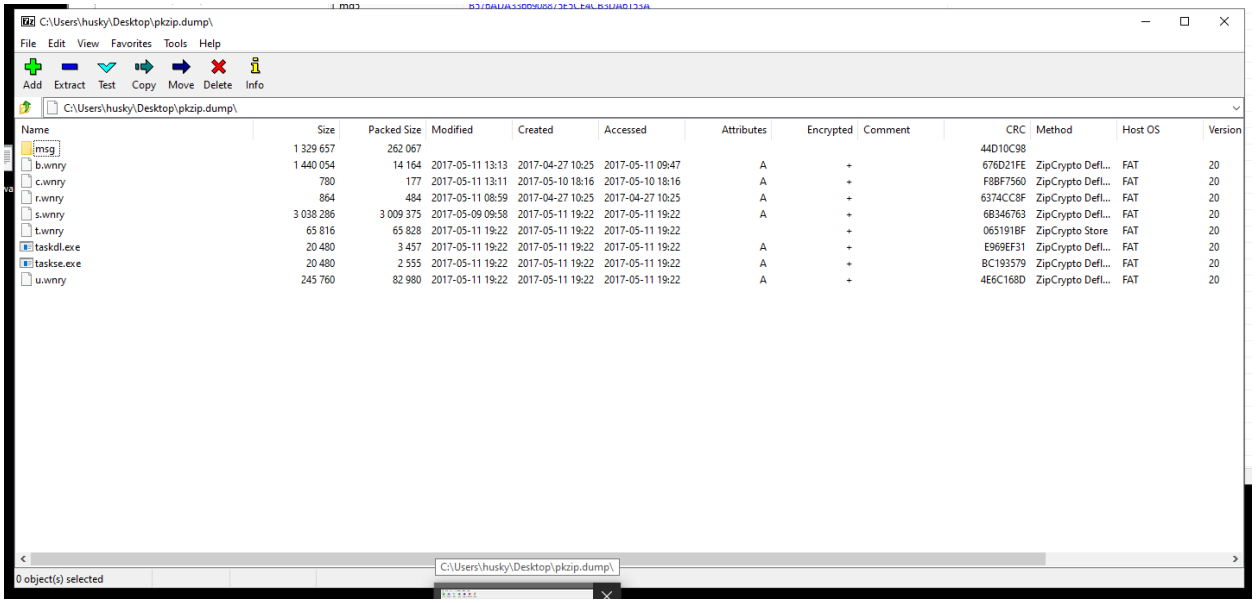
PEStudio shows an executable inside the .rsrc section of the file, called R.1831



Further confirmation of the executable inside the resources section, this time shown in the PEView tool

The following screenshots show the executable inside R.1831 (that had a PKZIP file inside itself) and the content of that ZIP file:





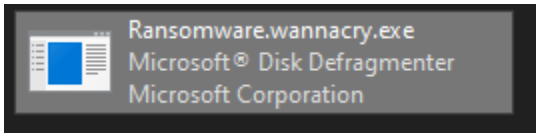
property	value	value	value	value
name	.text	.rdata	.data	.rsrc
md5	C7613102E2ECEC5DCEFC14...	D8037D744B539326C06E897...	22A8598DC29CAD7078C291E94612CE26	12E1BD7375D82CCA3A51CA48FE22D1A9
entropy	6.135	3.504	6.100	7.995
file-ratio (99.89%)	0.99 %	0.11 %	4.29 %	94.50 %
raw-address	0x00001000	0x0000A000	0x0000B000	0x00032000
raw-size (3719168 bytes)	0x00009000 (36864 bytes)	0x00001000 (4096 bytes)	0x00027000 (159744 bytes)	0x0035B000 (3518464 bytes)
virtual-address	0x00401000	0x0040A000	0x0040B000	0x00710000
virtual-size (6718034 bytes)	0x00008BCA (35786 bytes)	0x00000998 (2456 bytes)	0x0030489C (3164316 bytes)	0x0035A454 (3515476 bytes)
entry-point	**0x00009A16**	-	-	-
characteristics	0x60000020	0x40000040	0xC0000040	0x40000040
writable	-	-	**x**	-
executable	**x**	-	-	-
shareable	-	-	-	-
discardable	-	-	-	-
initialized-data	-	x	x	x
uninitialized-data	-	-	-	-
unreadable	-	-	-	-
self-modifying	-	-	-	-
virtualized	-	-	-	-
file	-	-	executable, offset: 0x0000B020, size: 5263716	executable, offset: 0x000320A4, size: 3514368
file	-	-	executable, offset: 0x0000F080, size: 5297524	n/a
file	-	-	executable, offset: 0x00013F94, size: 159744	n/a

ascii	42	0x000312C4	-	-	cryptography	Microsoft Base Cryptographic Provider v1.0
unicode	41	0x00017E18	-	-	-	R6025\r\n- pure virtual function call\r\n
ascii	40	0x0000004D	-	dos-message	-	!This program cannot be run in DOS mode.
ascii	40	0x0000B06D	-	dos-message	-	!This program cannot be run in DOS mode.
ascii	40	0x0000F0CD	-	dos-message	-	!This program cannot be run in DOS mode.
ascii	40	0x000320F1	-	dos-message	-	!This program cannot be run in DOS mode.
unicode	40	0x00017CE8	-	-	-	R6028\r\n- unable to initialize heap\r\n
unicode	40	0x0038B90C	-	-	-	6.1.7601.17514 (win7sp1_rtm.101119-1850)
unicode	40	0x0038C20C	-	-	-	6.1.7601.17514 (win7sp1_rtm.101119-1850)

Additional screenshots from the basic static analysis:

encoding (2)	size (bytes)	file-offset	blacklist (54)	hint (416)	group (18)	value (43891)
ascii	4	0x0000A85C	-	utility	-	time
ascii	46	0x0003EEE0	-	utility	-	inflate 1.1.3 Copyright 1995-1998 Mark Adler
ascii	35	0x000415A0	-	utility	-	icacls . /grant Everyone:F /T /C /Q
ascii	11	0x000415C4	-	utility	-	attrib +h .
ascii	4	0x0034A0DD	-	utility	-	AT 8
ascii	4	0x00380703	-	utility	-	At 1
unicode	8	0x0038B8D8	-	utility	-	DiskPart
unicode	12	0x0038B980	-	utility	-	diskpart.exe
unicode	12	0x0038BA44	-	utility	-	diskpart.exe
ascii	56	0x000313D0	-	url-pattern	-	http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
ascii	2039	0x0001BBBE	-	size	-	h6agLCqPqVyXi2VSQ8O6Yb9ijBX54jY6KM+sz33NmS6TK8XIOk920s0E0aajOV
ascii	1403	0x0001C41B	-	size	-	h54WfF9cGigWFEx92bzmOd0UOaZIMDdU2F4F2+6qn9/ZDSqJksnLIfbdOiMA

Example of interesting strings, some of which already mentioned previously



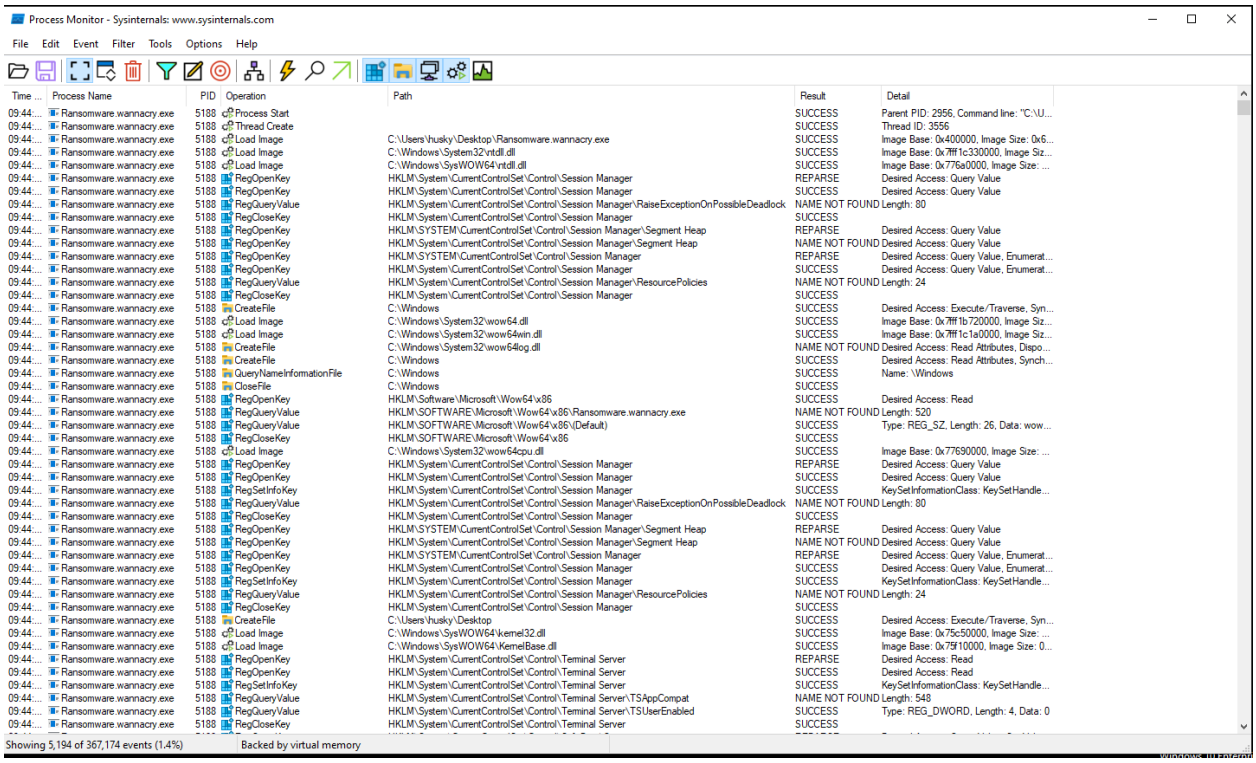
Malware disguises itself as a Disk utility in Windows

This malware sample is flagged on VirusTotal as malicious:

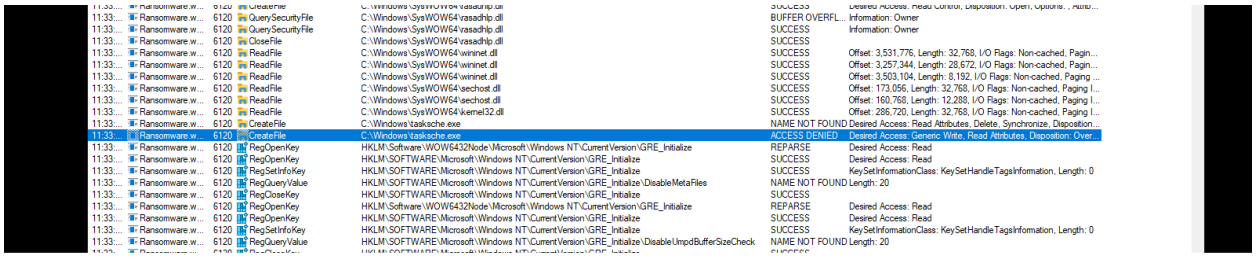
Basic Dynamic Analysis

Running the malware without administrator privileges will not activate the malicious payload, which includes file encryption and spreading to other targets.

In that case, the process performs basic operations but will not lead to the spreading of the infection when run without admin privileges:

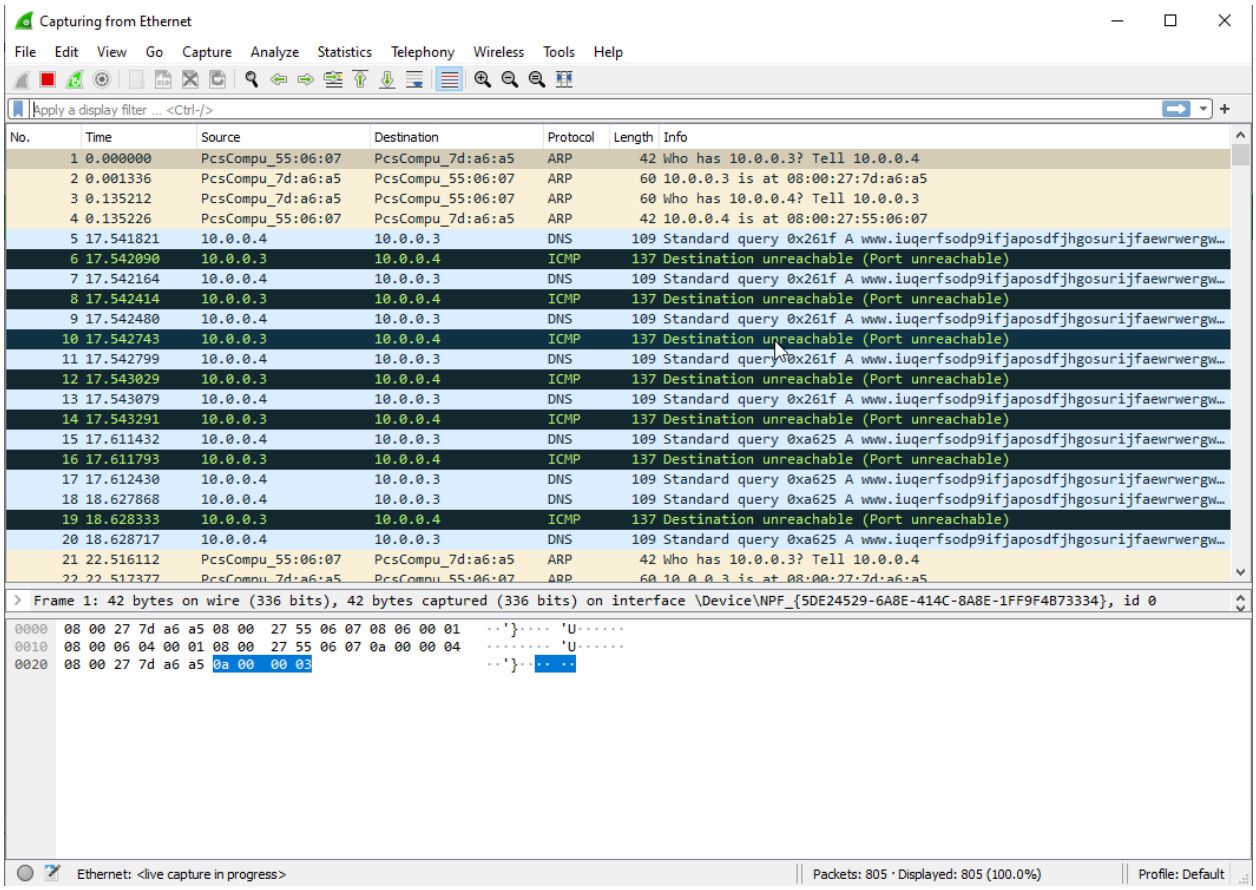


For example we can see in the next screenshot that it wasn't capable to deploy the tasksche.exe file in the Windows folder (which instead is used during the infection to run its operations):

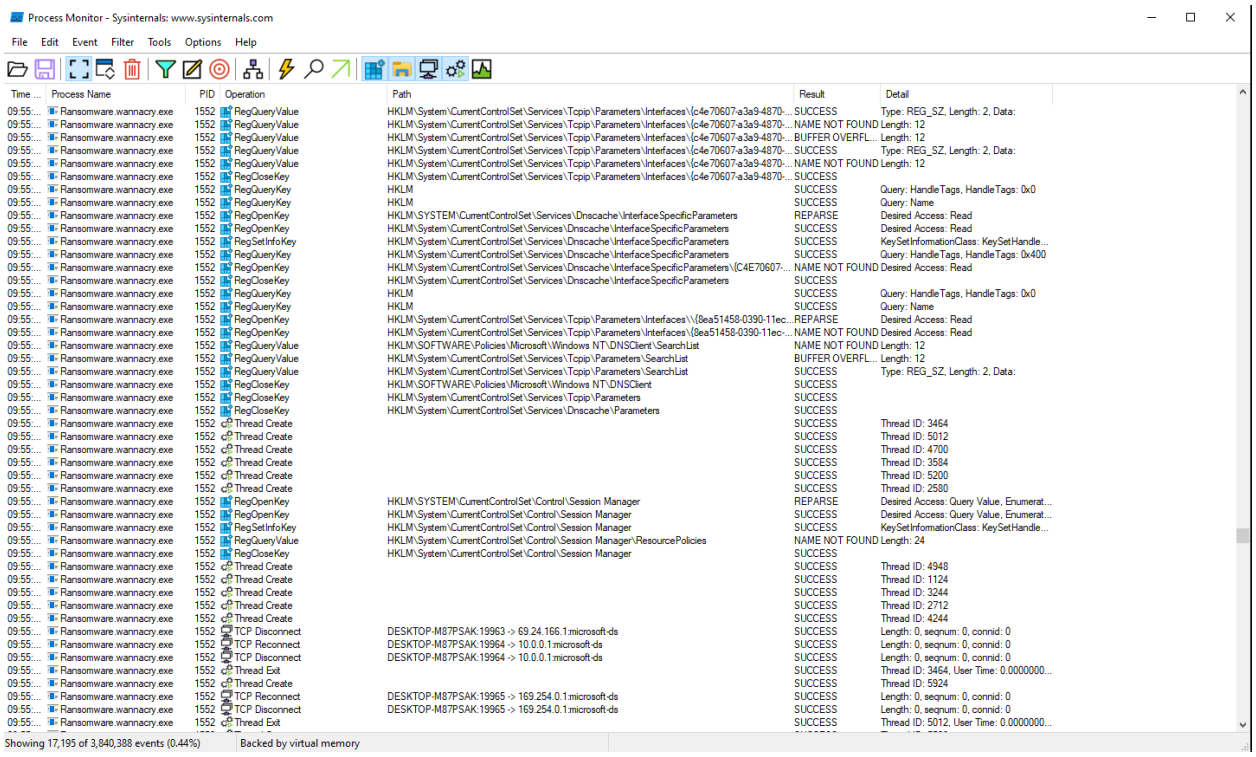


Running the sample with administrator privileges would activate the payload only if there is no internet connection available (i.e. in the forensic environment inetsim is not active).

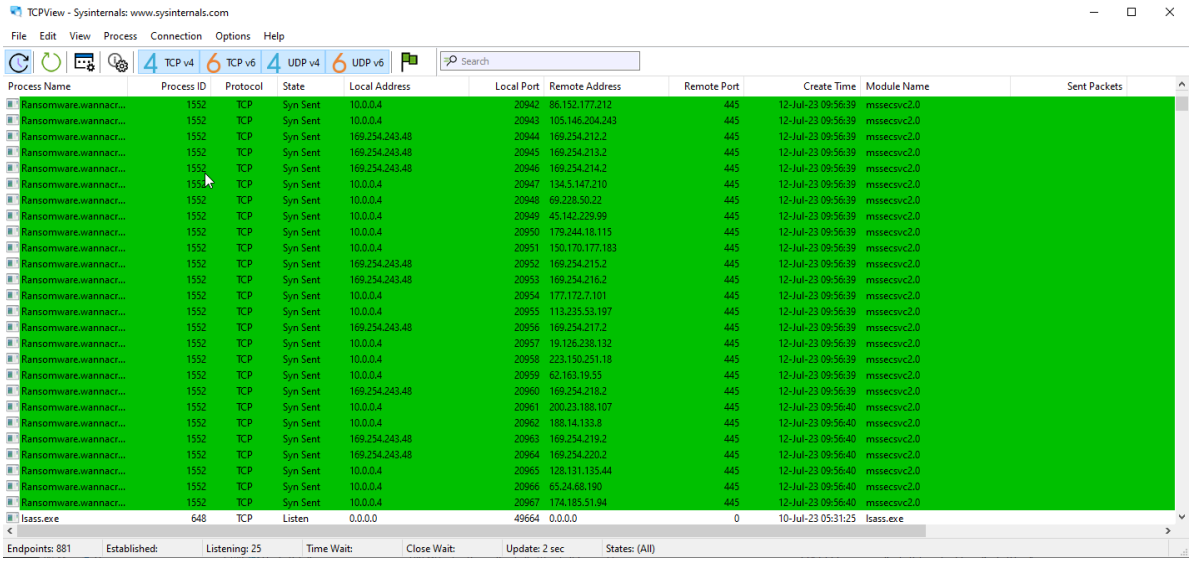
Wireshark and TCPView would show network activity. First, we see in Wireshark the attempts to connect to the killswitch domain:



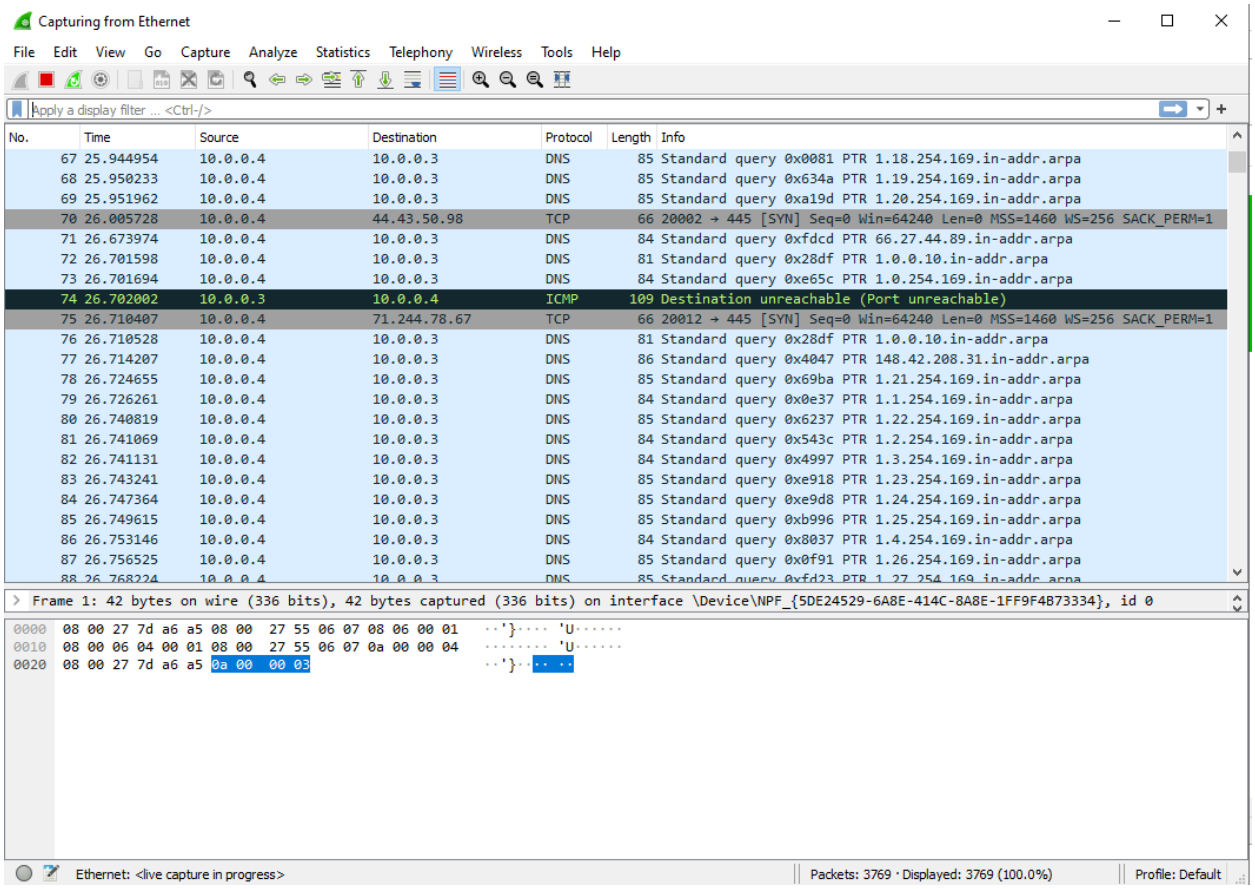
Then, in ProcMon we see further operations (registry queries and modifications, DLL loading, threads creation, files and services creation, etc.). These operations eventually lead to the establishment of TCP connections for the purpose of spreading to other targets:



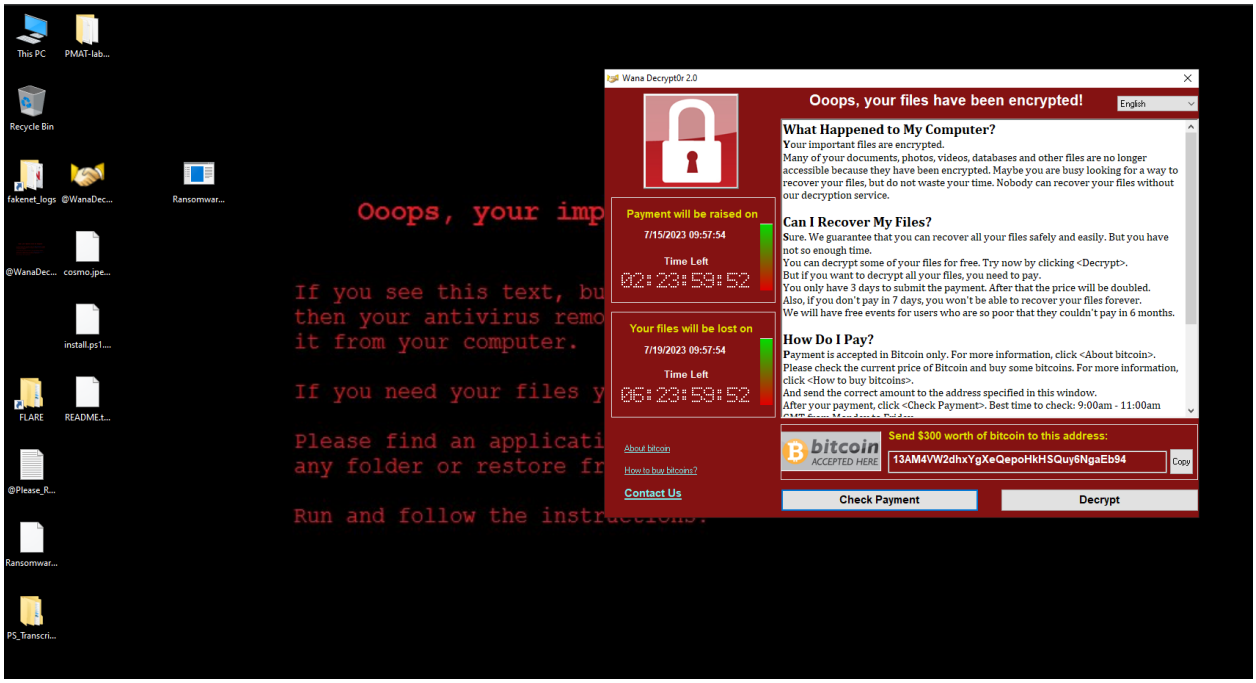
TCPView would also show the TCP connections to port 445, used by the SMB protocol:



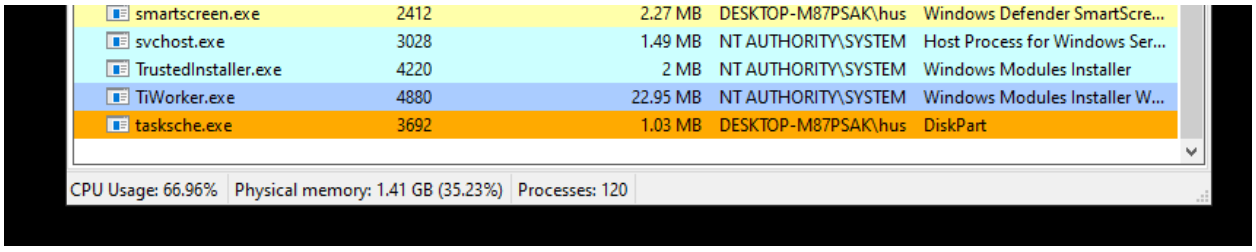
TCP connections also caught in Wireshark:



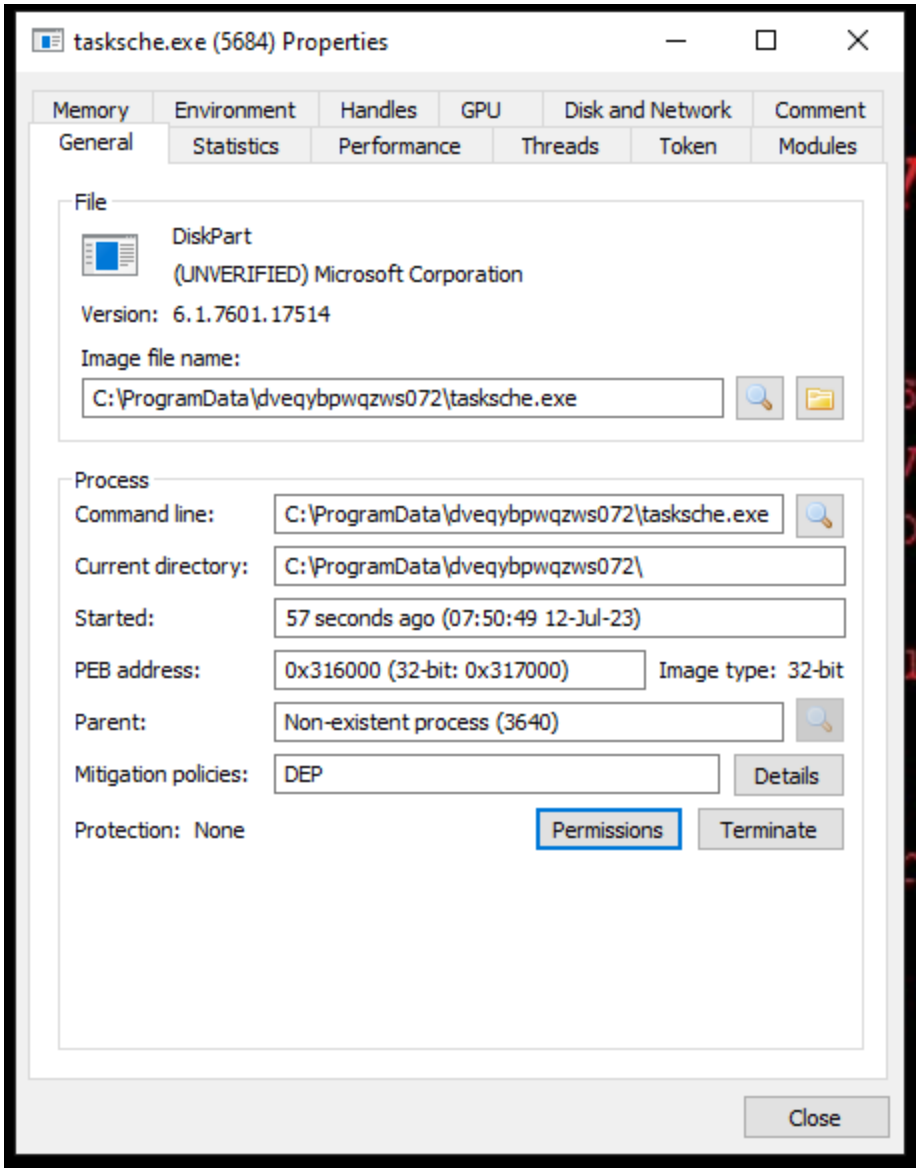
Then, we see files disappearing from the desktop, reappearing after a while with a different extension .wncry; we see the appearance of new files, the desktop wallpaper changing and also the GUI of the decryptor typical of this malware infection:



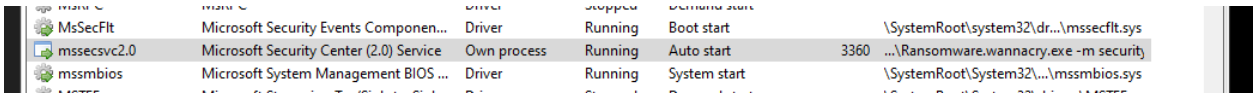
Here are some additional findings, after observing the malware during the dynamic analysis, using additional tools like Process Hacker 2 and the standard Windows Explorer:



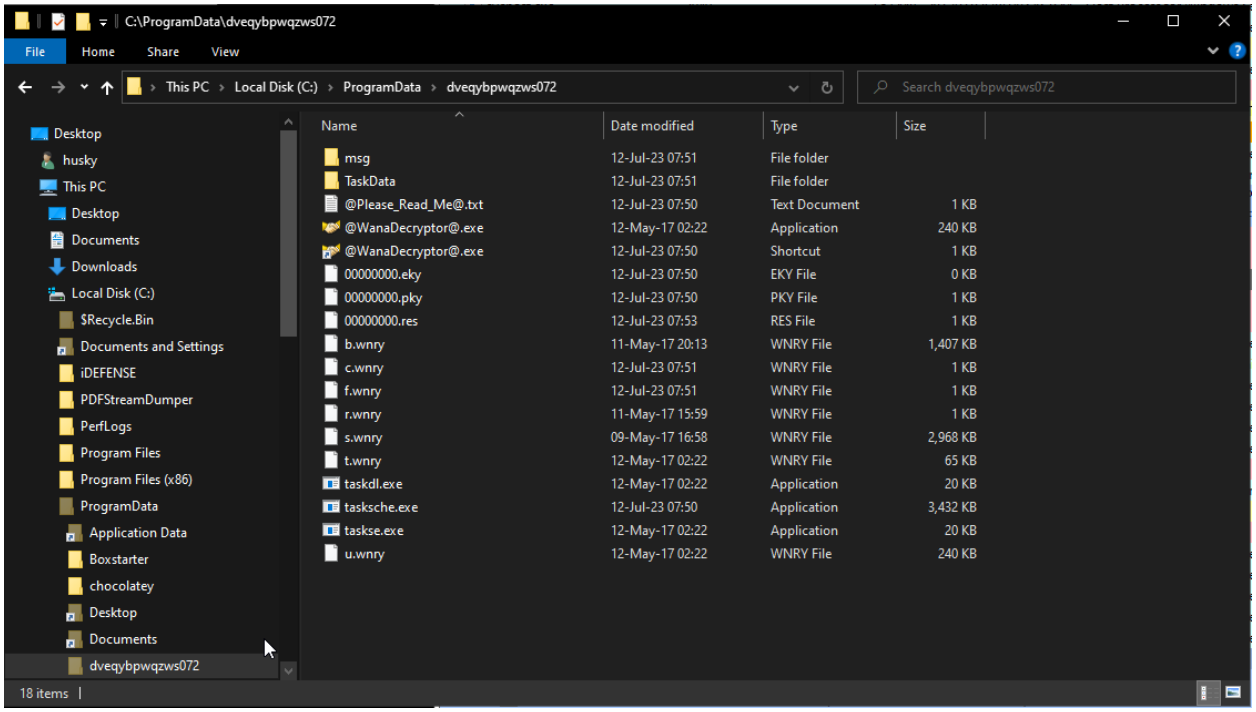
ProcessHacker: Here the malware poses as the DiskPart utility



ProcessHacker: We can see the hidden folder where the malware unpacked the additional executables and files



Process Hacker: service started by the malware, showing the command used to run it (part of it was found during the strings analysis)

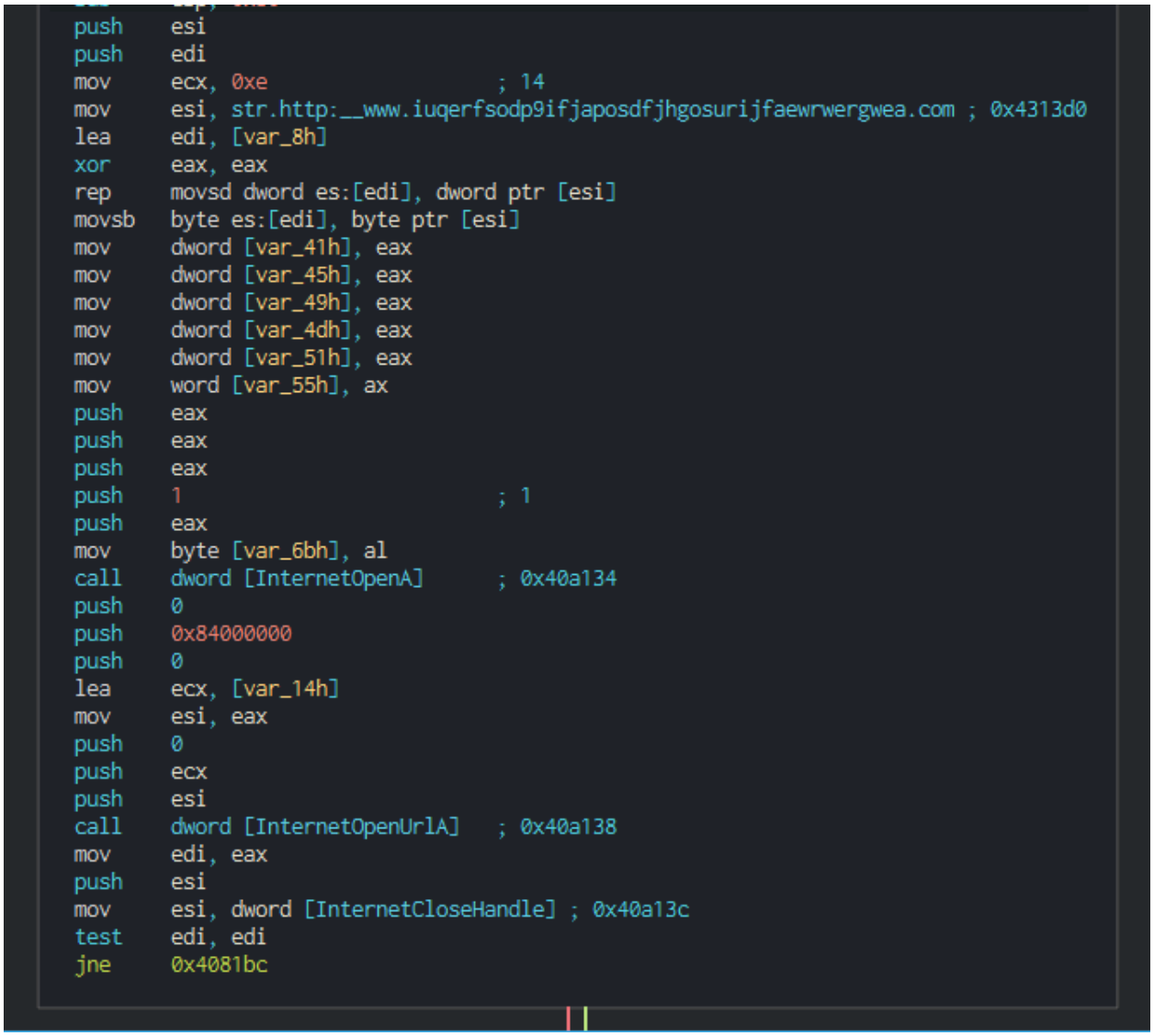


The hidden folder on disk

Advanced Static Analysis

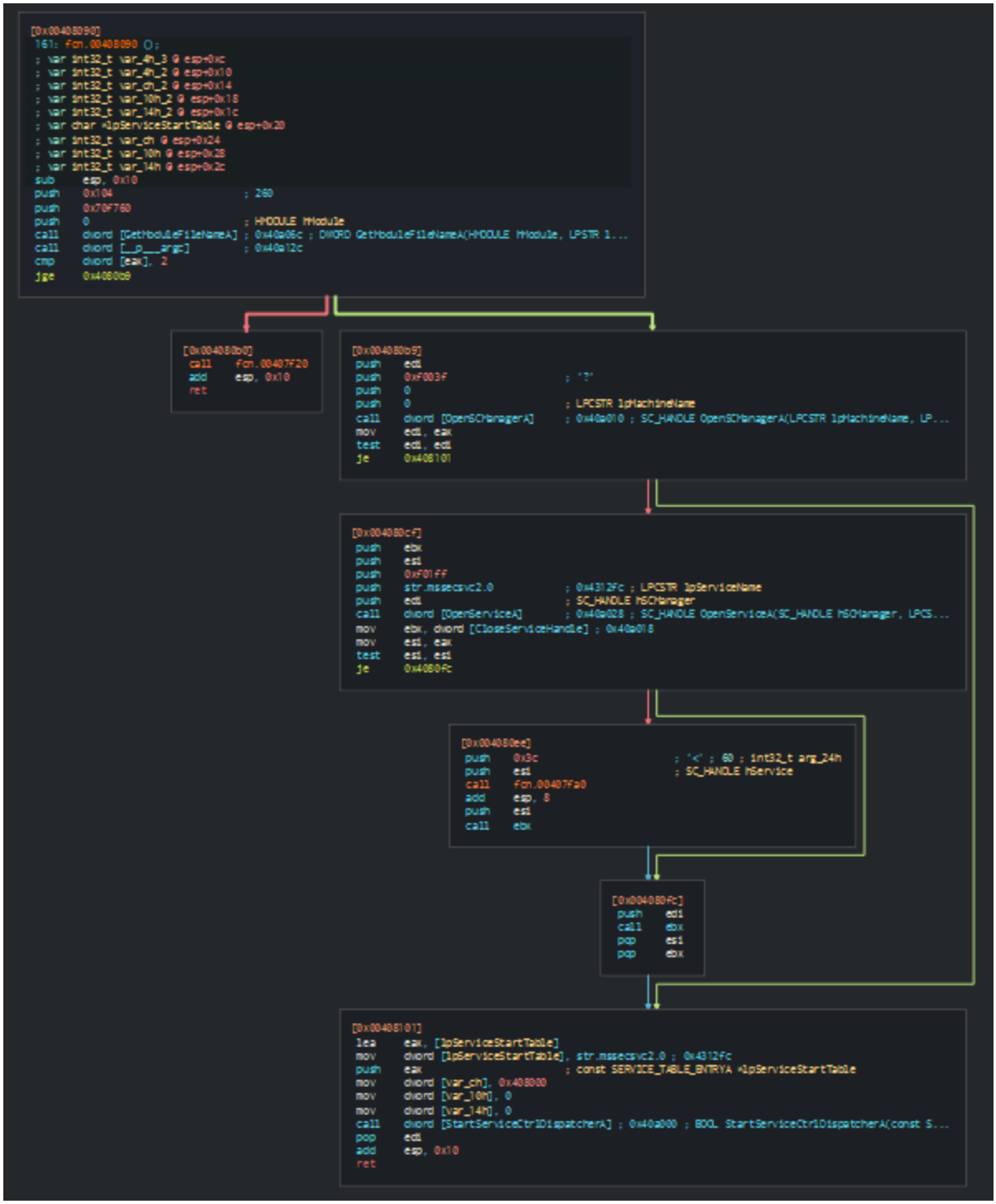
The advanced static analysis was conducted using Cutter, which at a first glance further confirmed the finding from the basic dynamic analysis. The initial step of the program involves checking the availability of the malicious URL identified in the strings and Wireshark logs.
Depending on the outcome of this check, the program would either remain inactive or initiate the malicious routine responsible for infecting the target.

Overview of the main function

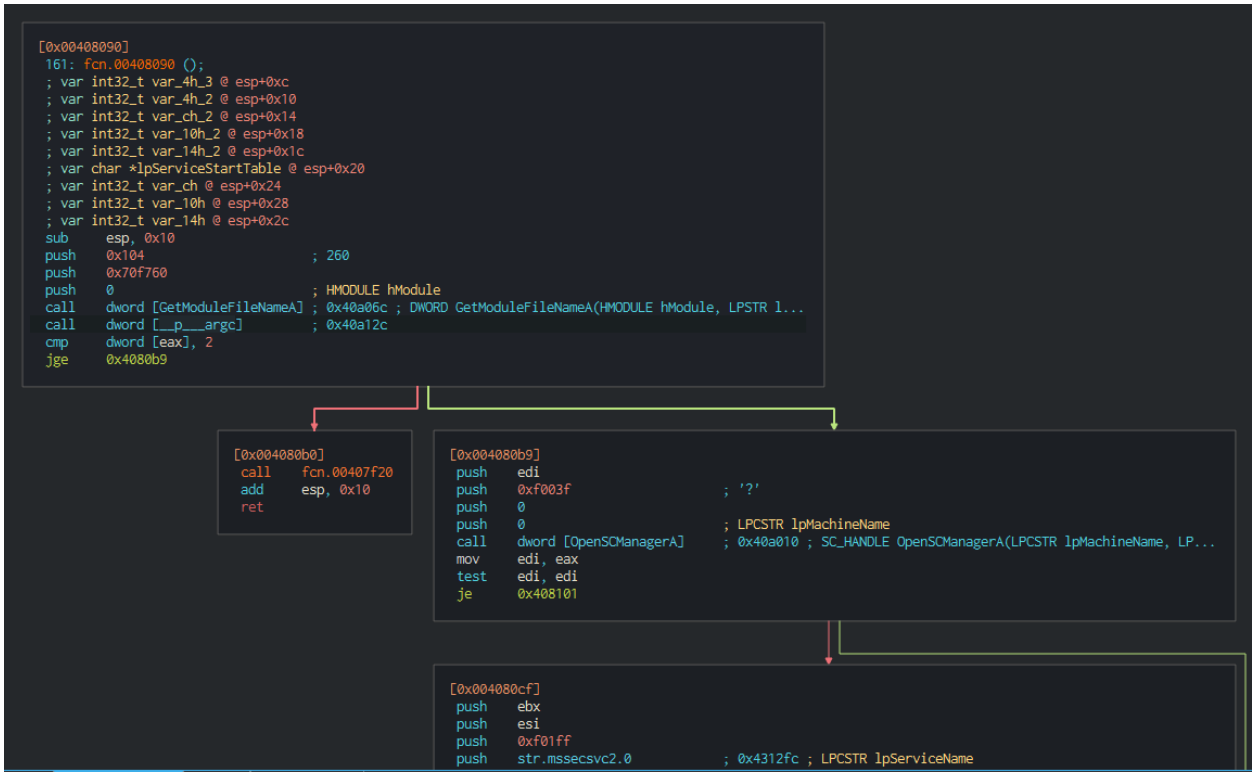


```
push    esi
push    edi
mov     ecx, 0xe                    ; 14
mov     esi, str.http:__www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com ; 0x4313d0
lea     edi, [var_8h]
xor     eax, eax
rep     movsd dword es:[edi], dword ptr [esi]
movsb   byte es:[edi], byte ptr [esi]
mov     dword [var_41h], eax
mov     dword [var_45h], eax
mov     dword [var_49h], eax
mov     dword [var_4dh], eax
mov     dword [var_51h], eax
mov     word [var_55h], ax
push    eax
push    eax
push    eax
push    1                           ; 1
push    eax
mov     byte [var_6bh], al
call    dword [InternetOpenA]       ; 0x40a134
push    0
push    0x84000000
push    0
lea     ecx, [var_14h]
mov     esi, eax
push    0
push    ecx
push    esi
call    dword [InternetOpenUrlA]    ; 0x40a138
mov     edi, eax
push    esi
mov     esi, dword [InternetCloseHandle] ; 0x40a13c
test    edi, edi
jne     0x4081bc
```

Details of the main function, showing the three APIs used to connect to the malicious URL, the arguments passed to the APIs, including the URL itself.



Overview of the malicious routines.

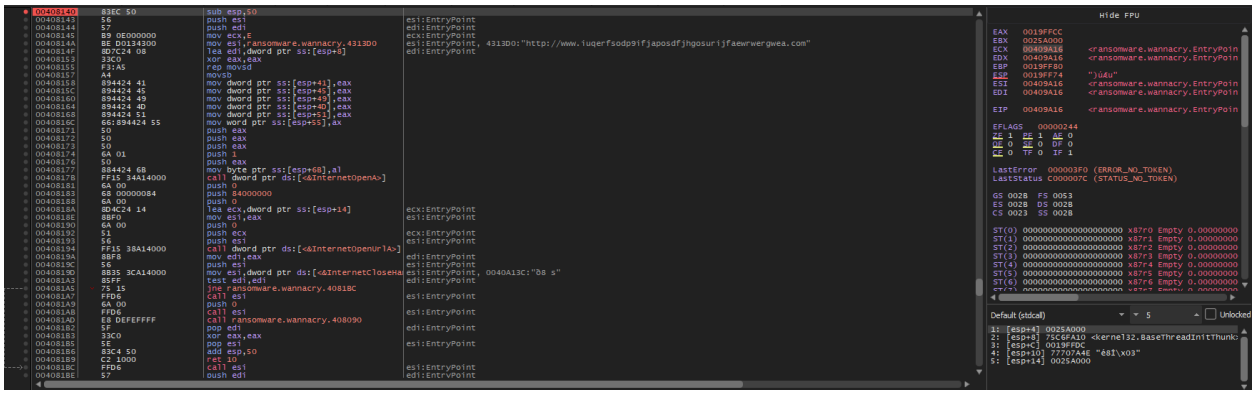


Detail of the malicious function.

Advanced Dynamic Analysis

During the advanced dynamic analysis, there was additional confirmation and deeper understanding of the killswitch domain.

Using x32dbg, a breakpoint was set at address 00408140 (obtained from Cutter), which corresponds to the main section of the program where the killswitch URL was checked before proceeding further..



From that point, it was possible to meticulously follow the execution and validate the program's behavior, which can be summarized as follows:

- The malicious URL is inaccessible (either due to the non-existence of the domain in the wild or the absence of an active inetsim service in the forensic environment).
- ZF (Zero Flag) is already set to 1.
- The program establishes a connection to the URL.
- No response is received from the URL, resulting in EAX being set to 0.
- EAX is then copied to EDI, making EDI also 0.
- The program performs a test on EDI against EDI, resulting in 0 (since 0 and 0 = 0).
- ZF remains set to 1 since the result of the test function is 0.
- JNE (Jump if Not Equal) will not jump to the end of the program because ZF is not 0. This means that it will instead go through the malicious routine as explained in the Advanced Static Analysis section.

Note: For this summary, I have chosen to omit the details of the API calls shown in the screenshot and focus solely on the specific set of instructions

relevant to this particular killswitch behavior.

The following screenshot highlights the points of interest, at the moment where the jump is about to (not) be taken:

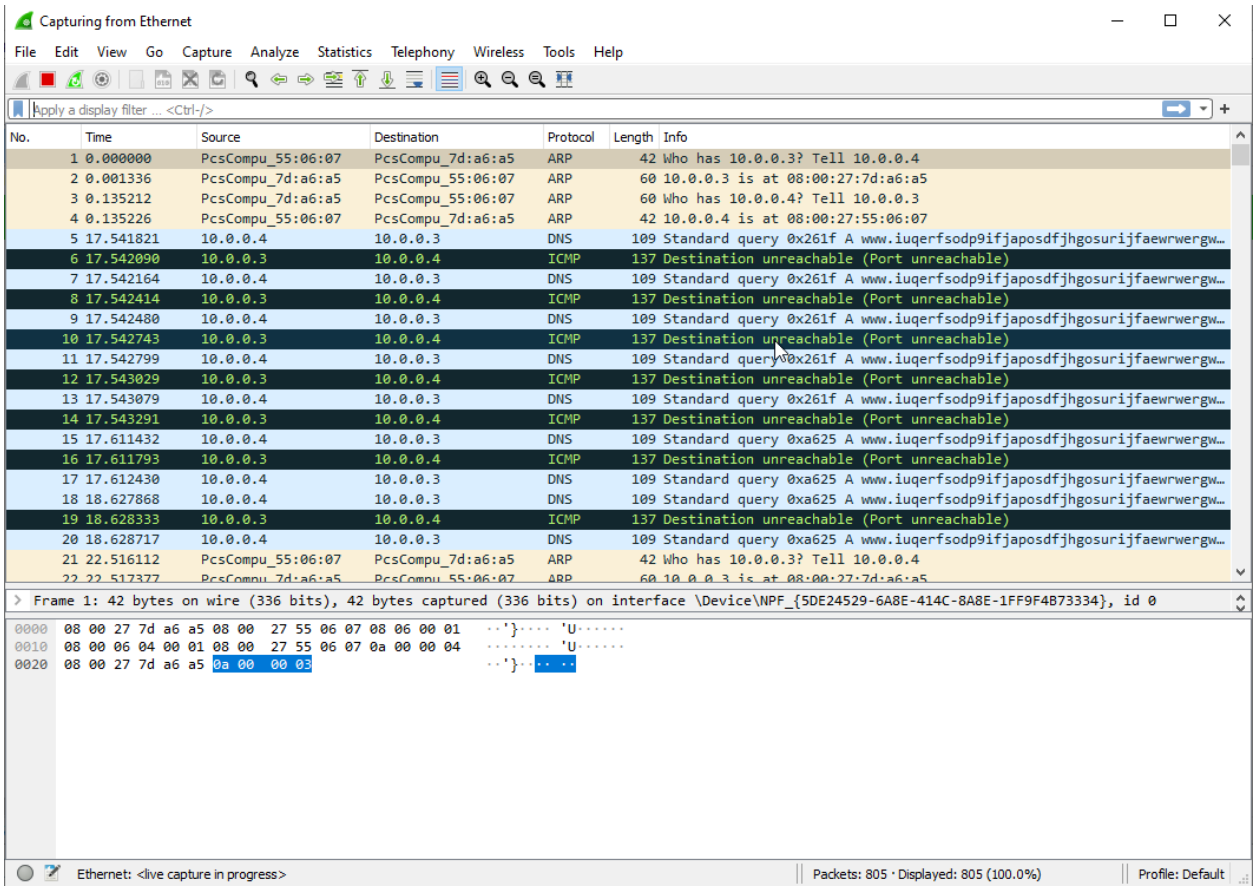
In this case, the program will proceed with the remaining instructions and eventually reach the call at address 004081AD, which points to the malicious function located at 00408090 (as indicated in the previous screenshot and also in the Cutter screenshot provided earlier in this report).

It is important to note that modifying the JNE instruction to JE would allow the malware to execute its malicious payload even if the URL is unreachable (for example, by patching the binary using Cutter). Similarly, during debugging with x32dbg, changing the ZF to 0 could lead to a similar outcome, enabling the execution of the malicious payload.

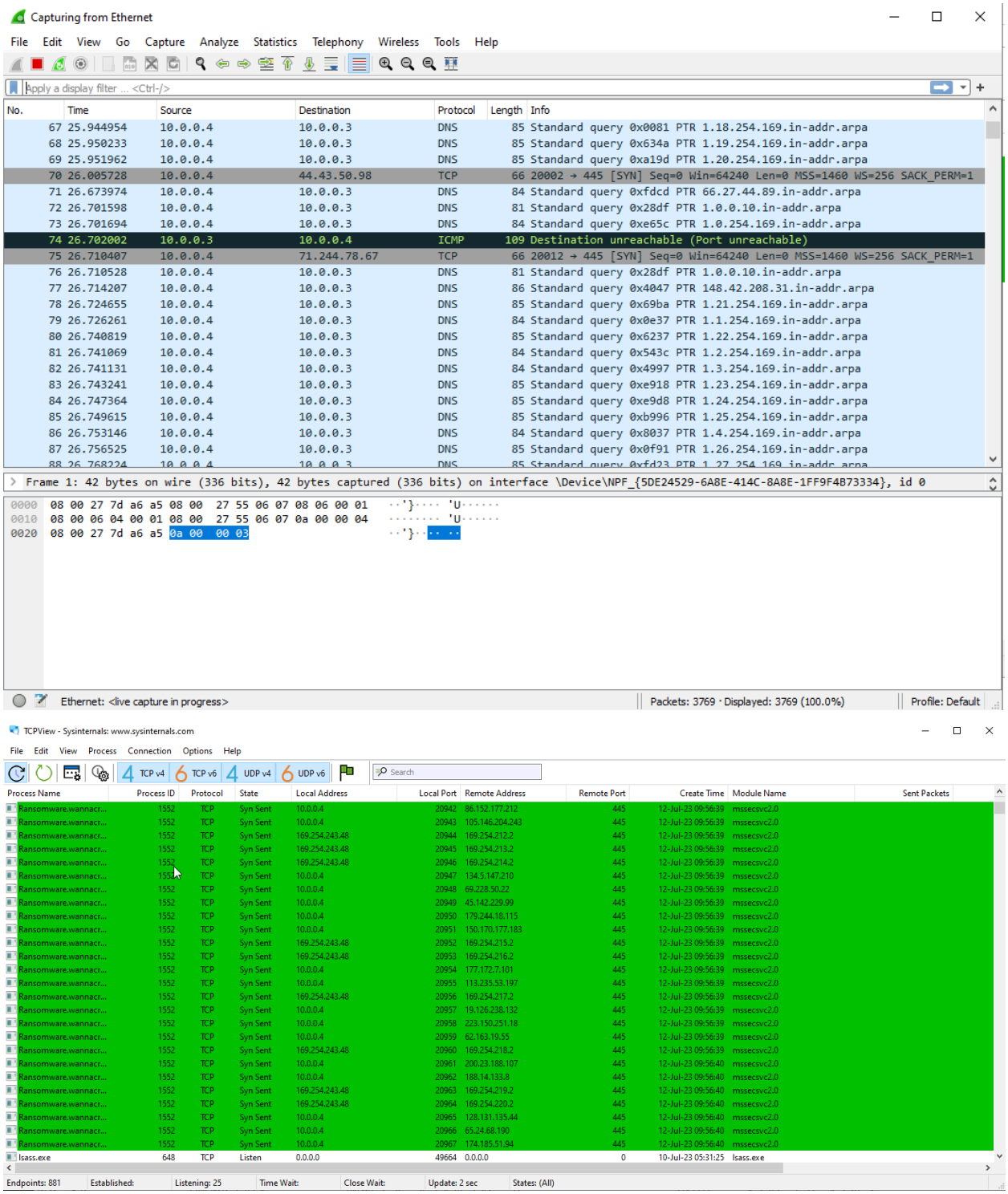
Indicators of Compromise

Network Indicators

Connection to the URL:
hxxp[://]www[.]iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]com

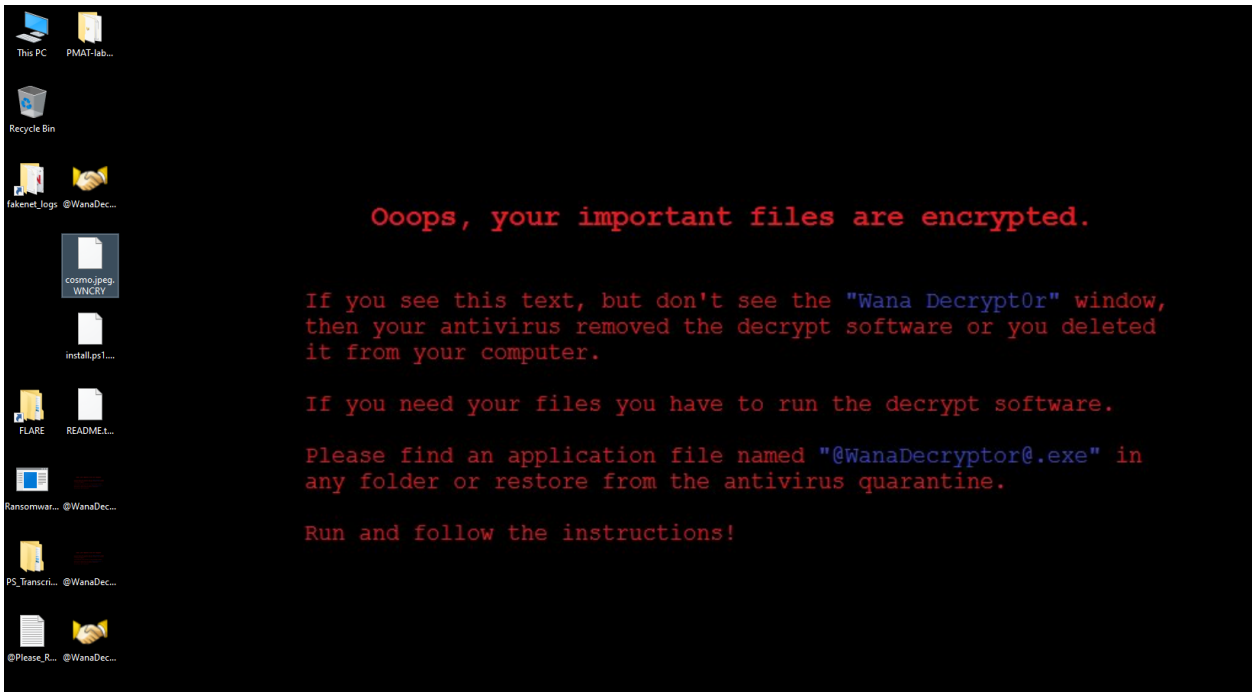


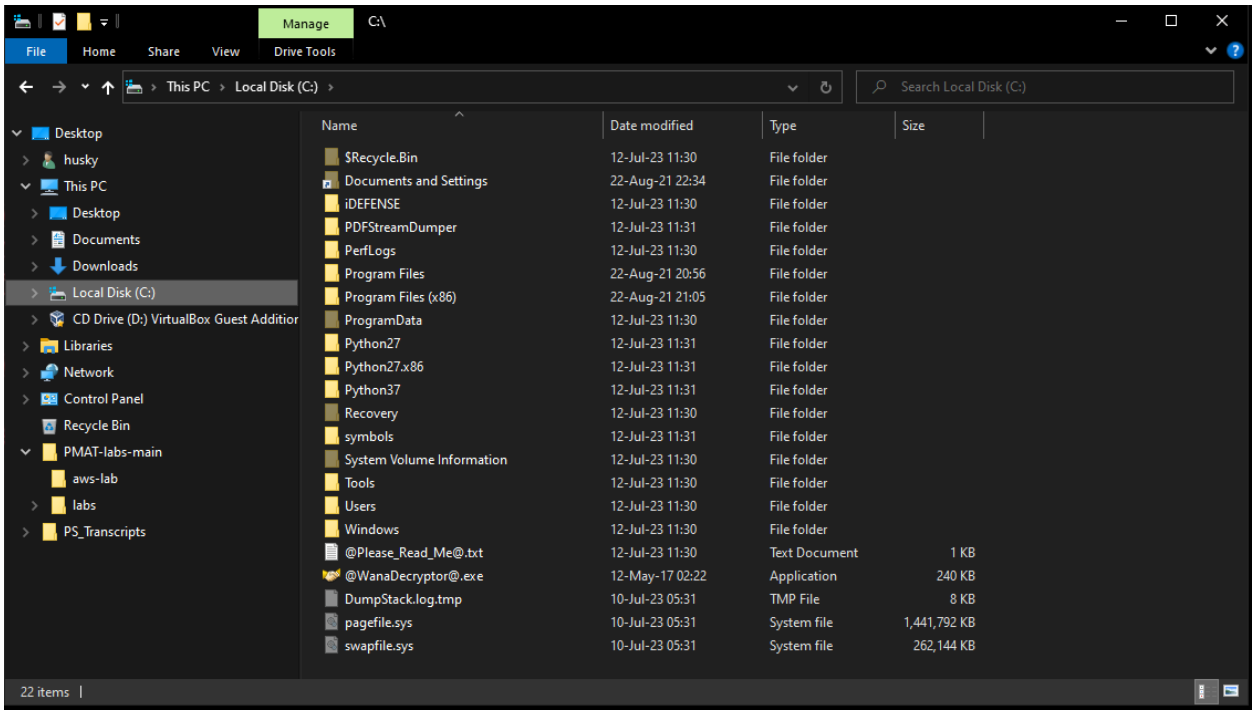
Connections to several IP addresses on port 445:



Host-based Indicators

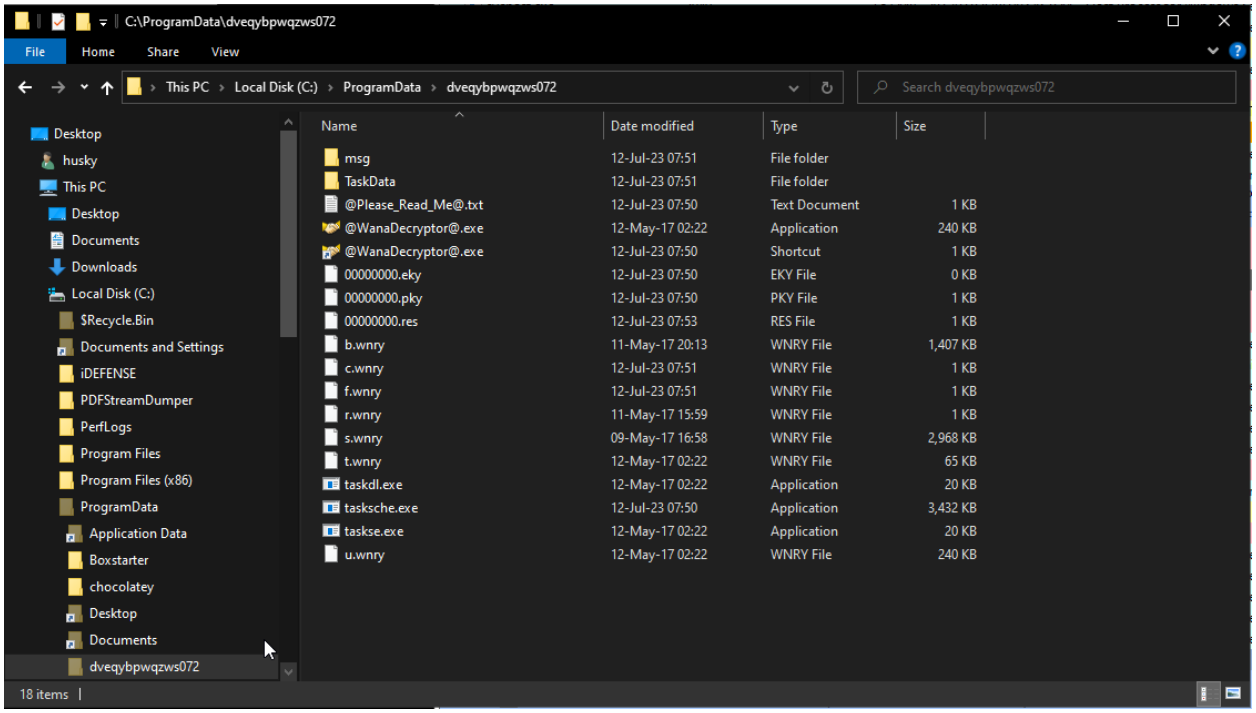
Desktop wallpaper changed, local files encrypted with extension .wncry, presence of new files on the desktop (and, in time, inside other folders as well):





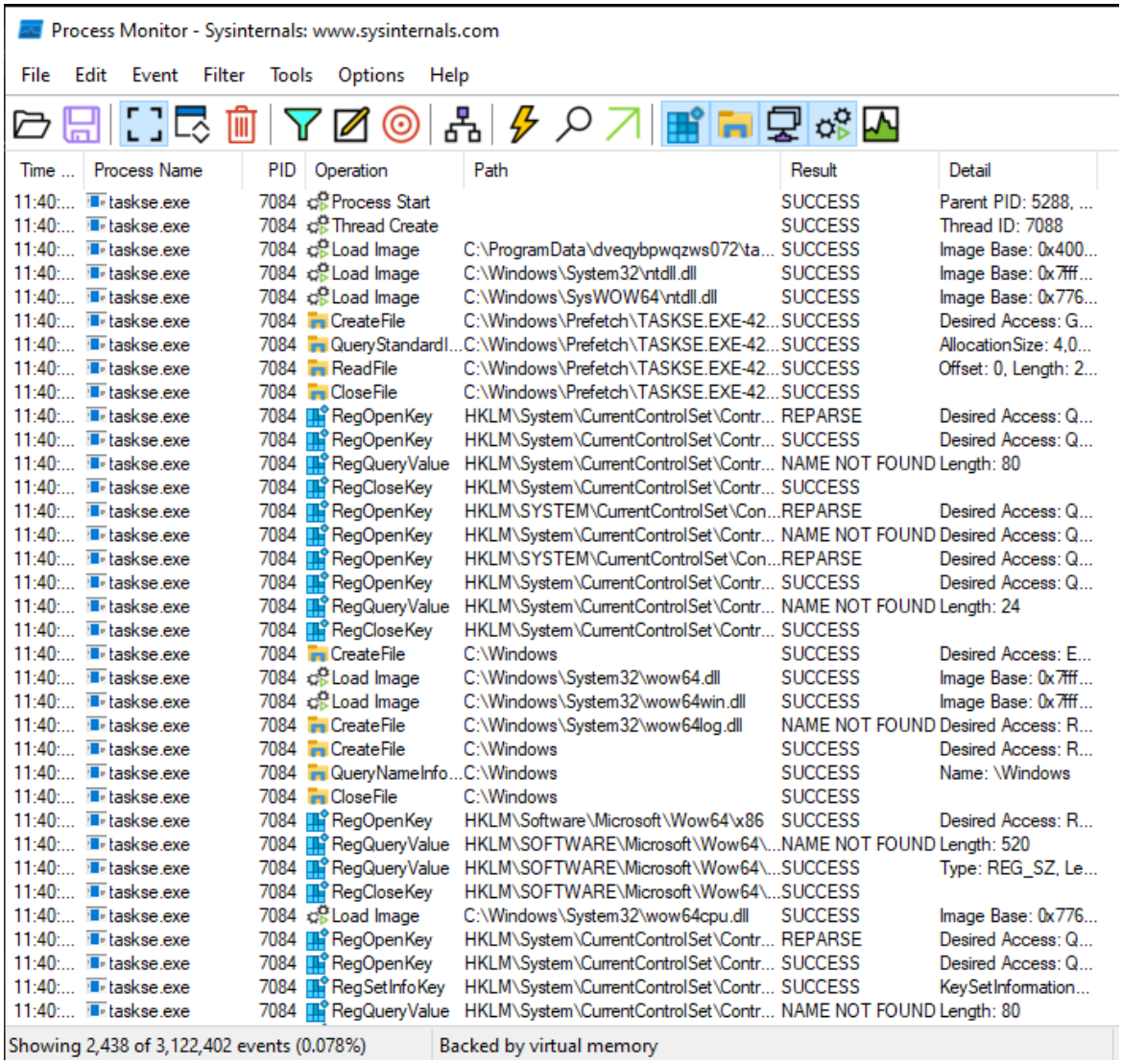
Program loading on the foreground every few seconds:

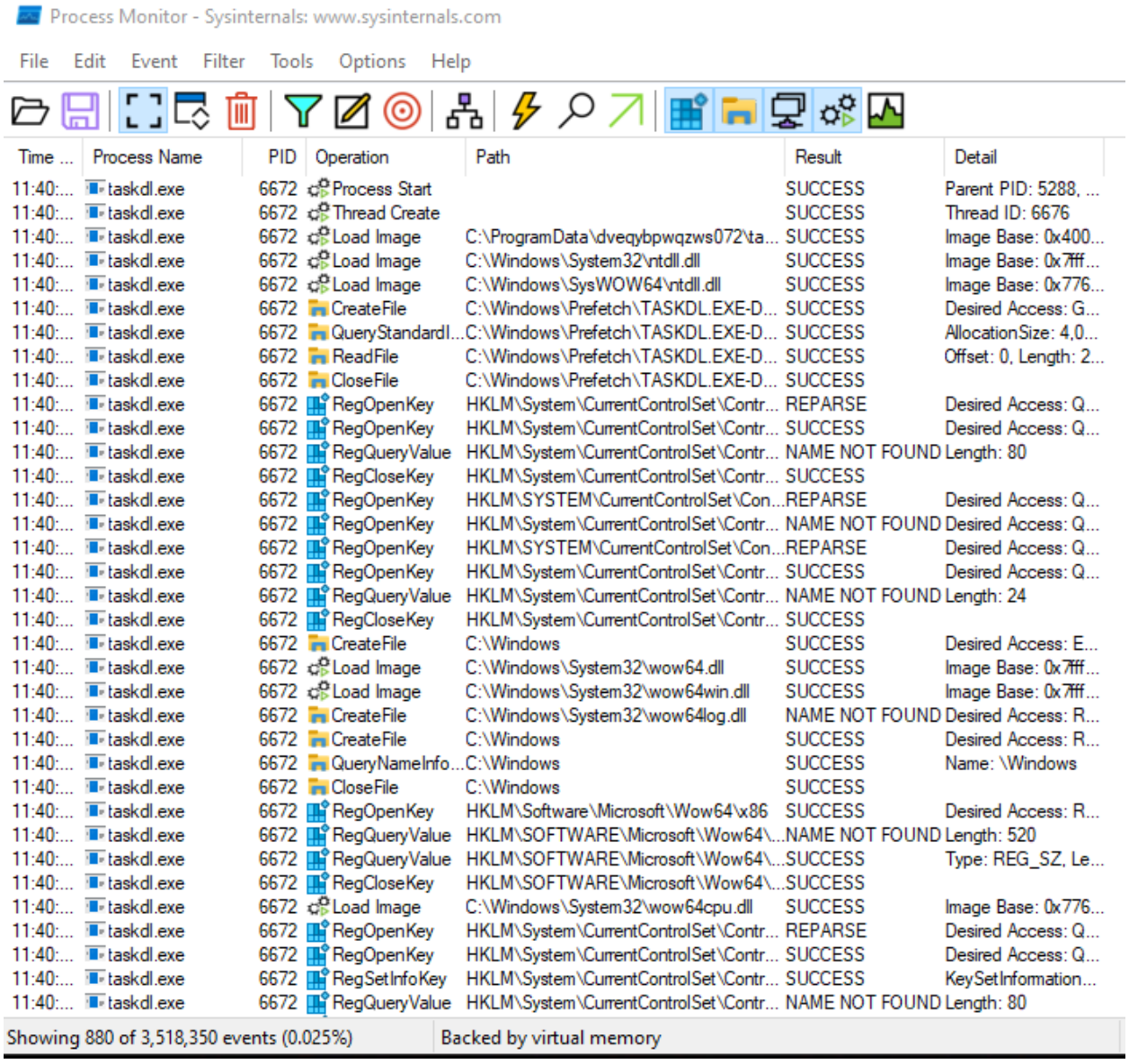
Creation of a new folder named “dveqybpwqzws072” inside C:\ProgramData\ containing malicious files. The folder and its content is set to Hidden:



New processes:

tasksche.exe / taskdl.exe / taskse.exe




Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help
Time ... Process Name PID Operation Path Result Detail
11:40:... taskdl.exe 6672 Process Start SUCCESS Parent PID: 5288, ...
11:40:... taskdl.exe 6672 Thread Create SUCCESS Thread ID: 6676
11:40:... taskdl.exe 6672 Load Image C:\ProgramData\dveqybpwqzws072\ta... SUCCESS Image Base: 0x400...
11:40:... taskdl.exe 6672 Load Image C:\Windows\System32\ntdll.dll SUCCESS Image Base: 0x7fff...
11:40:... taskdl.exe 6672 Load Image C:\Windows\SysWOW64\ntdll.dll SUCCESS Image Base: 0x776...
11:40:... taskdl.exe 6672 CreateFile C:\Windows\Prefetch\TASKDL.EXE-D... SUCCESS Desired Access: G...
11:40:... taskdl.exe 6672 QueryStandardI... C:\Windows\Prefetch\TASKDL.EXE-D... SUCCESS AllocationSize: 4,0...
11:40:... taskdl.exe 6672 ReadFile C:\Windows\Prefetch\TASKDL.EXE-D... SUCCESS Offset: 0, Length: 2...
11:40:... taskdl.exe 6672 CloseFile C:\Windows\Prefetch\TASKDL.EXE-D... SUCCESS
11:40:... taskdl.exe 6672 RegOpenKey HKLM\System\CurrentControlSet\Contr... REPARSE Desired Access: Q...
11:40:... taskdl.exe 6672 RegOpenKey HKLM\System\CurrentControlSet\Contr... SUCCESS Desired Access: Q...
11:40:... taskdl.exe 6672 RegQueryValue HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND Length: 80
11:40:... taskdl.exe 6672 RegCloseKey HKLM\System\CurrentControlSet\Contr... SUCCESS
11:40:... taskdl.exe 6672 RegOpenKey HKLM\SYSTEM\CurrentControlSet\Con... REPARSE Desired Access: Q...
11:40:... taskdl.exe 6672 RegOpenKey HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND Desired Access: Q...
11:40:... taskdl.exe 6672 RegOpenKey HKLM\SYSTEM\CurrentControlSet\Con... REPARSE Desired Access: Q...
11:40:... taskdl.exe 6672 RegOpenKey HKLM\System\CurrentControlSet\Contr... SUCCESS Desired Access: Q...
11:40:... taskdl.exe 6672 RegQueryValue HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND Length: 24
11:40:... taskdl.exe 6672 RegCloseKey HKLM\System\CurrentControlSet\Contr... SUCCESS
11:40:... taskdl.exe 6672 CreateFile C:\Windows SUCCESS Desired Access: E...
11:40:... taskdl.exe 6672 Load Image C:\Windows\System32\wow64.dll SUCCESS Image Base: 0x7fff...
11:40:... taskdl.exe 6672 Load Image C:\Windows\System32\wow64win.dll SUCCESS Image Base: 0x7fff...
11:40:... taskdl.exe 6672 CreateFile C:\Windows\System32\wow64log.dll NAME NOT FOUND Desired Access: R...
11:40:... taskdl.exe 6672 CreateFile C:\Windows SUCCESS Desired Access: R...
11:40:... taskdl.exe 6672 QueryNameInfo... C:\Windows SUCCESS Name: \Windows
11:40:... taskdl.exe 6672 CloseFile C:\Windows SUCCESS
11:40:... taskdl.exe 6672 RegOpenKey HKLM\Software\Microsoft\Wow64\x86 SUCCESS Desired Access: R...
11:40:... taskdl.exe 6672 RegQueryValue HKLM\SOFTWARE\Microsoft\Wow64\... NAME NOT FOUND Length: 520
11:40:... taskdl.exe 6672 RegQueryValue HKLM\SOFTWARE\Microsoft\Wow64\... SUCCESS Type: REG_SZ, Le...
11:40:... taskdl.exe 6672 RegCloseKey HKLM\SOFTWARE\Microsoft\Wow64\... SUCCESS
11:40:... taskdl.exe 6672 Load Image C:\Windows\System32\wow64cpu.dll SUCCESS Image Base: 0x776...
11:40:... taskdl.exe 6672 RegOpenKey HKLM\System\CurrentControlSet\Contr... REPARSE Desired Access: Q...
11:40:... taskdl.exe 6672 RegOpenKey HKLM\System\CurrentControlSet\Contr... SUCCESS Desired Access: Q...
11:40:... taskdl.exe 6672 RegSetInfoKey HKLM\System\CurrentControlSet\Contr... SUCCESS KeySetInformation...
11:40:... taskdl.exe 6672 RegQueryValue HKLM\System\CurrentControlSet\Contr... NAME NOT FOUND Length: 80
Showing 880 of 3,518,350 events (0.025%)
Backed by virtual memory

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time ...	Process Name	PID	Operation	Path	Result	Detail
11:40:...	tasksche.exe	5288	CreateFile	C:\ProgramData\dveqybpwqzws072\00...	SUCCESS	Desired Access: G...
11:40:...	tasksche.exe	5288	WriteFile	C:\ProgramData\dveqybpwqzws072\00...	SUCCESS	Offset: 0, Length: 1...
11:40:...	tasksche.exe	5288	CloseFile	C:\ProgramData\dveqybpwqzws072\00...	SUCCESS	
11:40:...	tasksche.exe	5288	CreateFile	C:\	SUCCESS	Desired Access: S...
11:40:...	tasksche.exe	5288	QueryFullSizeIn...	C:\	SUCCESS	TotalAllocationUnit...
11:40:...	tasksche.exe	5288	CloseFile	C:\	SUCCESS	
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,276,044,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,286,530,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,297,016,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,307,502,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,317,987,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,328,473,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,338,959,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,349,445,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,359,930,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,370,416,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,380,902,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,391,388,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,401,873,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,412,359,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,422,845,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,433,331,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,443,816,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,454,302,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,464,788,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,475,274,...
11:40:...	tasksche.exe	5288	CreateFile	C:\ProgramData\dveqybpwqzws072\00...	NAME NOT FOUND	Desired Access: R...
11:40:...	tasksche.exe	5288	CreateFile	C:\ProgramData\dveqybpwqzws072\00...	NAME NOT FOUND	Desired Access: R...
11:40:...	tasksche.exe	5288	CreateFile	C:\	SUCCESS	Desired Access: S...
11:40:...	tasksche.exe	5288	QueryFullSizeIn...	C:\	SUCCESS	TotalAllocationUnit...
11:40:...	tasksche.exe	5288	CloseFile	C:\	SUCCESS	
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,485,760,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,496,245,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,506,731,...
11:40:...	tasksche.exe	5288	WriteFile	C:\Windows\Temp\hibsys.WNCRYT	SUCCESS	Offset: 10,517,217,...

Showing 755 of 3,800,541 events (0.019%) Backed by virtual memory

New service:

mssecsvc2.0

MsSecFlt	Microsoft Security Events Componen...	Driver	Running	Boot start		\SystemRoot\system32\dr...\mssecflt.sys
mssecsvc2.0	Microsoft Security Center (2.0) Service	Own process	Running	Auto start	3360	...\Ransomware.wannacry.exe -m security
mssmbios	Microsoft System Management BIOS ...	Driver	Running	System start		\SystemRoot\System32\...\mssmbios.sys

Rules & Signatures

A full set of YARA rules is included in Appendix A.

URL:

hxxp[://]www[.]iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]com

Strings:

WanaCrypt0r

.wnry

C:\%s\qeriuwjhrf

taskdl.exe

taskse.exe

tasksche.exe

115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn

12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw

13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94

WANACRY!

\\172.16.99.5\IPC$

\\192.168.56.20\IPC$

Appendices

A. Yara Rules

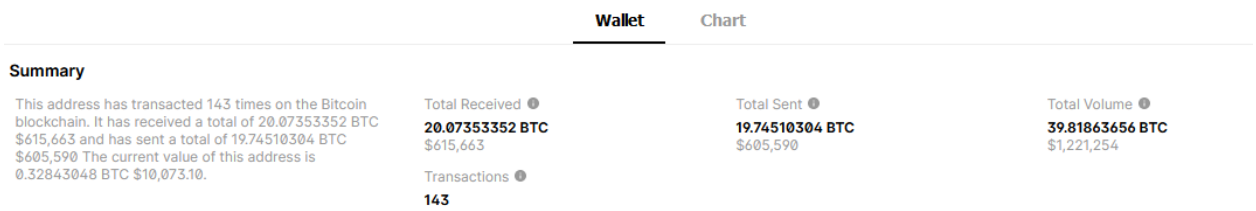
```
rule WannaCry {

      meta:
      last_updated = "2023-07-12"
      author = "Harpocrat3s"
      description = "Detects WannaCry"
      sha256 =
"24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c"
      // Providing the file's hash to pinpoint which strain of the
malware this rules works on

      strings:
      $string1 = "www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com"
ascii
      $string2 = "WanaCrypt0r" ascii
      $string3 = "13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94" ascii
      $string4 = "115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn" ascii
      $string5 = "12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw" ascii
      $string6 = "C:\\%s\\qeriuwjhrf" ascii
      $string7 = "taskdl.exe" ascii
      $string8 = "taskse.exe" ascii
      $string9 = "tasksche.exe" ascii

      condition:
      any of ($string*)
}
```

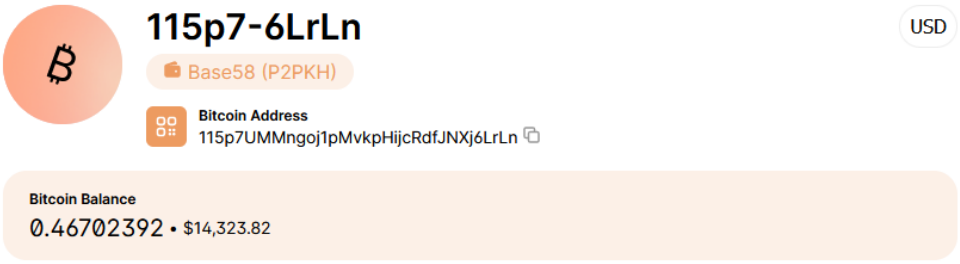
B. Bitcoin Wallet analysis

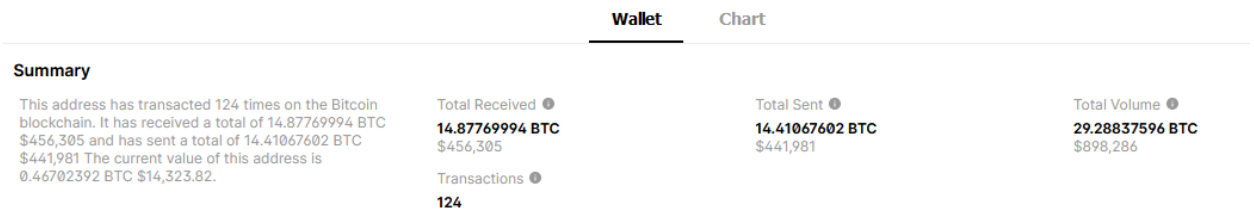
As every operation on the bitcoin blockchain is recorded, it was possible to check the history of the wallet used on this sample, but also to confirm that the other two strings found in the sample were indeed bitcoin wallets that also received funds in a similar fashion of the main wallet.
This further confirms the suspicion that other malware samples of WannaCry could use the other two wallets for a similar operation.

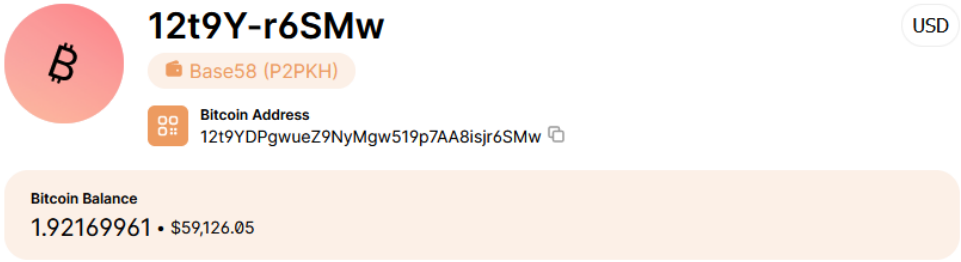
Here are the screenshots of the summary for the 3 bitcoin wallets found in the strings.
Further analysis could be done investigating where the funds were sent from these wallets, and investigate and correlate the other wallets with other strains of malwares (WannaCry or others), potentially finding out additional operations of the threat actor that originally developed this malware.

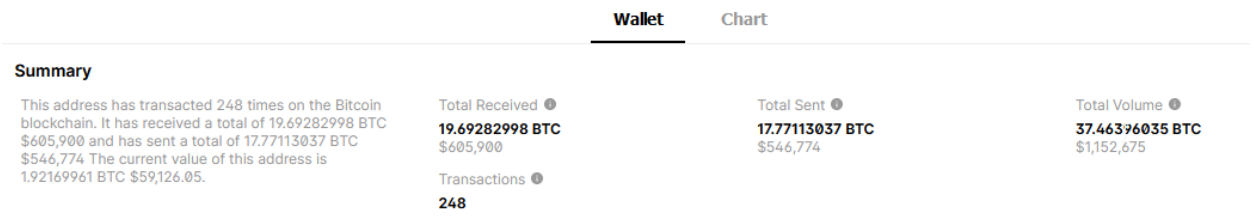
We can see that the total sum of money sent on the 3 wallets is 1,677,868 USD. It's not sure at this point if all the funds were due to WannaCry or if the wallets were used in other malware campaigns. Further investigation is required, but it's outside the scope of this analysis.




115p7-6LrLn
USD
Base58 (P2PKH)
Bitcoin Address
115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn
Bitcoin Balance
0.46702392 • $14,323.82


Wallet
Chart
Summary
This address has transacted 124 times on the Bitcoin blockchain. It has received a total of 14.87769994 BTC $456,305 and has sent a total of 14.41067602 BTC $441,981 The current value of this address is 0.46702392 BTC $14,323.82.
Total Received
14.87769994 BTC
$456,305
Total Sent
14.41067602 BTC
$441,981
Total Volume
29.28837596 BTC
$898,286
Transactions
124


12t9Y-r6SMw
USD
Base58 (P2PKH)
Bitcoin Address
12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
Bitcoin Balance
1.92169961 • $59,126.05


Wallet
Chart
Summary
This address has transacted 248 times on the Bitcoin blockchain. It has received a total of 19.69282998 BTC $605,900 and has sent a total of 17.77113037 BTC $546,774 The current value of this address is 1.92169961 BTC $59,126.05.
Total Received
19.69282998 BTC
$605,900
Total Sent
17.77113037 BTC
$546,774
Total Volume
37.46396035 BTC
$1,152,675
Transactions
248

C. List of file extensions

In the strings there's what looks like a list of file extensions that are encrypted by the malware:

.der
.pfx
.key
.crt
.csr
.p12
.pem
.odt
.ott
.sxw
.stw
.uot
.3ds
.max
.3dm
.ods
.ots
.sxc
.stc
.dif
.slk
.wb2
.odp
.otp
.sxd
.std
.uop
.odg
.otg
.sxm
.mml
.lay
.lay6
.asc
.sqlite3
.sqlitedb
.sql
.accdb
.mdb
.dbf
.odb
.frm
.myd
.myi
.ibd
.mdf
.ldf
.sln
.suo
.cpp
.pas
.asm
.cmd
.bat
.ps1
.vbs
.dip
.dch
.sch
.brd
.jsp
.php
.asp
.java
.jar
.class
.mp3
.wav
.swf
.fla
.wmv
.mpg
.vob
.mpeg
.asf
.avi
.mov
.mp4
.3gp
.mkv
.3g2
.flv
.wma
.mid
.m3u
.m4u
.djvu
.svg
.psd
.nef
.tiff
.tif
.cgm
.raw
.gif
.png
.bmp
.jpg
.jpeg
.vcd
.iso
.backup
.zip
.rar
.tgz
.tar
.bak
.tbk
.bz2
.PAQ
.ARC

.aes
.gpg
.vmx
.vmdk
.vdi
.sldm
.sldx
.sti
.sxi
.602
.hwp
.snt
.onetoc2
.dwg
.pdf
.wk1
.wks
.123
.rtf
.csv
.txt
.vsdx
.vsd
.edb
.eml
.msg
.ost
.pst
.potm
.potx
.ppam
.ppsx
.ppsm
.pps
.pot
.pptm
.pptx
.ppt
.xltm
.xltx
.xlc
.xlm
.xlt
.xlw
.xlsb
.xlsm
.xlsx
.xls
.dotx
.dotm
.dot
.docm
.docb
.docx
.doc