Practical Malware Analysis & Triage

Malware Analysis Report

WannaCry - RansomWare

May 2024 | AnikshaShetty | v1.0

Table of Contents

Executive Summary

SHA256 hash	24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c

WannaCry is a Ransomware cryptoworm sample first identified on May 2017. It is a 32 bit C++ compiled program that runs on both x32 and x64 bit windows operating system. It consists of multiple payloads that encrypts files, and exploits the EternalBlue vulnerability to spread across networks. Symptoms of infection include file encryption with ".wncry" extensions, a changed desktop background indicating encryption, and prompts for Bitcoin ransom payment via a decryptor program.

YARA signature rules are attached in Appendix A. Malware sample and hashes have been submitted to VirusTotal for further examination.

High-Level Technical Summary

WannaCry.exe incorporates a unique safeguard: a killswitch mechanism. When executed, the program attempts to connect to a non-existent domain. If the connection succeeds, the program terminates without further action. However, if the connection fails, it continues with its intended functions. This ingenious tactic is believed to thwart sandbox environments and halt execution when under analysis.

Persistence is achieved by creating service called Microsoft security center 2.0 (mssecsvc) which launches original wannacry.exe on reboot. And then writes the next stage payload tasksche.exe which drops further resources for enumeration and encryption routines.

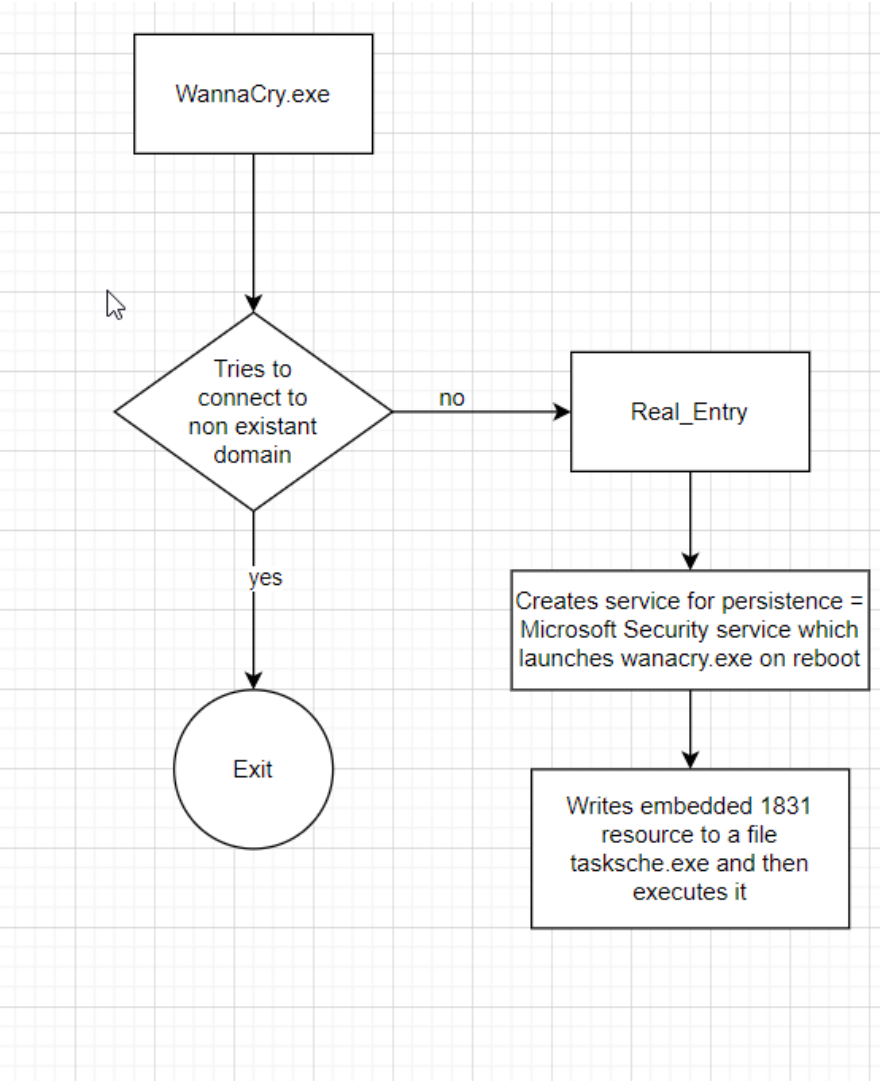


Figure 1 Initial payload wannacry execution flow

Tasksche.exe initially creates a random alpha numeric string based on computer name which is later used as a staging folder and then checks for any arguments passed to the program. If there is an argument, then creates hidden staged folder in program data directory with the random name obtained from previous function and then copies itself to this directory and creates a service with the same name to run tasksche.exe and then starts this service.

If no argument is passed then tasksche.exe, creates a reg key in HKLM/HKCU Software hive with wd value set to current working directory. And then loads and unzips password protected zip file from resource 2058 consisting of config files, encryption routine, decryption routine, background images and the text files containing the msg to be displayed in different languages. After which it writes a bitcoin address to the config file – c.wnry created in previous step. Then executes two commands to make the directory hidden and grant all access to everyone on all subdirectories.

Then uses an embedded rsa and aes key to decrypt the packed t.wnry sample and then executes it. T.wnry was found to be a dll file constituting the encryption routine for the malware. It searches for all the files of interest and encrypts it with RSA AES algorithm either using an embedded key or key file(0000000.pky / 0000000.eky).

There are additional support programs that are dropped which provides different functionality like – deletion of programs, enumeration and propagation through network, config files containing the c2 and decryptor program to show the users that the files are decryptable. See fig2. for execution flow of tasksche.exe

All the details containing the files dropped and their functionality has been detailed in the next section.

Execution flow of tasksche.exe

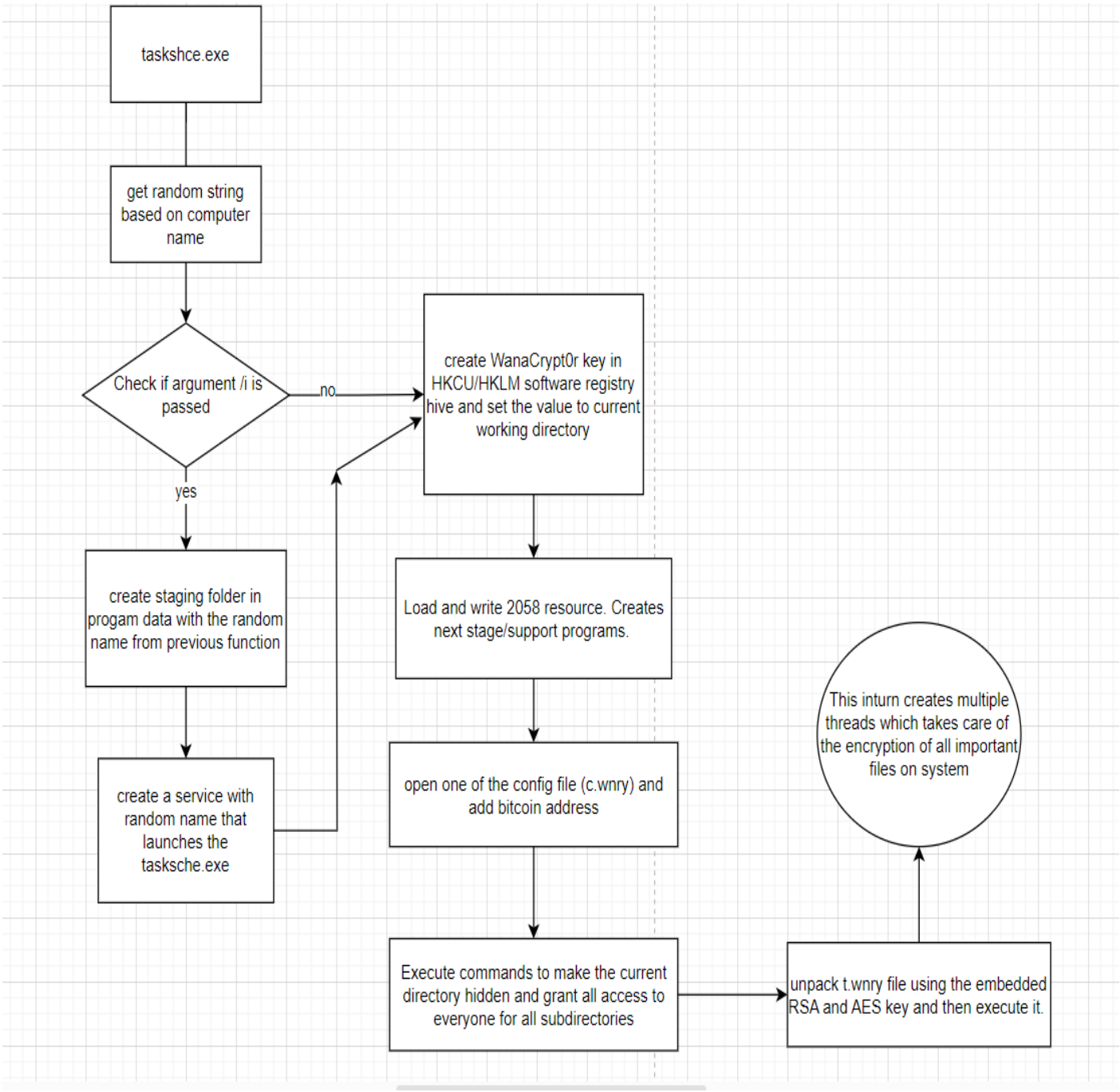


Figure 2 Second stage tasksche.exe execution flow

Malware Composition

Ransomware consists of the following components :

File Name	FileType	SHA256 Hash
Ransomware.wannacry.exe	PE32 Windows	24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c
***tasksche.exe**	PE32 Windows	ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa
***taskdl.exe**	PE32 Windows	4a468603fdcb7a2eb5770705898cf9ef37aade532a7964642ecd705a74794b79
***taskse.exe**	PE32 Windows	2ca2d550e603d74dedda03156023135b38da3630cb014e3d00b1263358c5f00d
***t.wnry**	Packed dll file	97ebce49b14c46bebc9ec2448d00e1e397123b256e2be9eba5140688e7bc0ae6
***b.wnry**	Bit map image	d5e0e8694ddc0548d8e6b87c83d50f4ab85c1debadb106d6a6a794c3e746f4fa
***c.wnry**	Config file	055c7760512c98c8d51e4427227fe2a7ea3b34ee63178fe78631fa8aa6d15622
***r.wnry**	Ransom note	402751fa49e0cb68fe052cb3db87b05e71c1d950984d339940cf6b29409f2a7c
***s.wnry**	Tor software zip archive	e18fdd912dfe5b45776e68d578c3af3547886cf1353d7086c8bee037436dff4b
***u.wnry**	PE32 Windows	b9c5d4339809e0ad9a00d4d3dd26fdf44a32819a54abf846bb9b560d81391c25
***t.wnry_unpacked**	Decrypted t.wnry file – dll	1be0b96d502c268cb40da97a16952d89674a9329cb60bac81a96e01cf7356830
***msg/**	Directory containing msg file in different languages	

Ransomware.wannacry.exe

The initial executable that checks the killswitch and then drops the next stage payload and creates a service for persistence.

*tasksche.exe

Second stage payload with an embedded resource containing a password protected archive consisting of next stage support programs that helps in encryptor program, background image, decryptor program, config files containing the command and control servers, bitcoin addresses, msg files and the ransom note.

*taskdl.exe

Support program to delete all the temporary files.

*taskse.exe:

Copy of tor.exe

*t,wnry

Encrypted dll containing the encryption routine

*b.wnry

Bitmap image containing the desktop background that displays the ransom note.

*c.wnry

Config file consisting the c2 servers (tor/.onion domains) and the bitcoin address

*r.wnry

Ransom note

*s.wnry

Zip file containing the Tor software and its dependency files

*u.wnry

Unknown

*msg/

Directory containing msg files displaying the instructions to recover the files in different languages.

Detailed view of these files can be found in extracted files section in this document.

Basic Static Analysis

{Screenshots and description about basic static artifacts and methods}

Floss – String Analysis

- There are multiple DOS headers. A domain that it connects to.
- Tries to access a network IPC share
- Multiple blocks of base 64 encoded /encrypted strings
- Message to be displayed in different languages
- Looks for various types of file extension

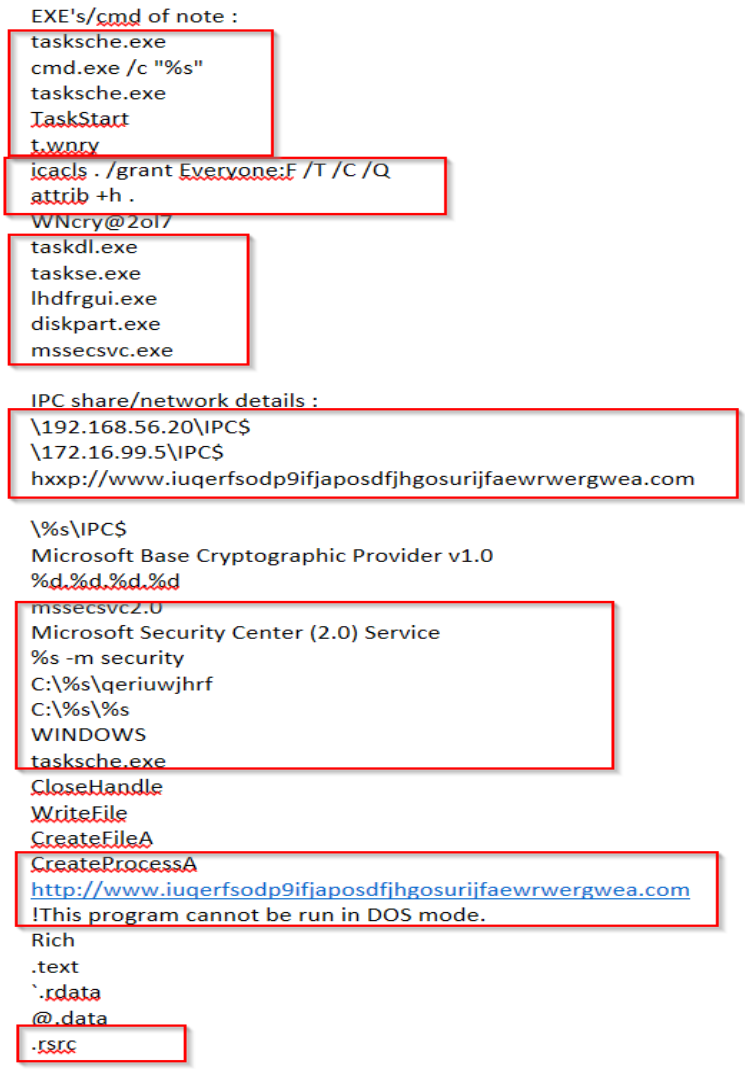


Figure 3 Floss Output - interesting strings capturing embedded resources and commands and killswitch domain

```
.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
```

Figure 4 File Extensions

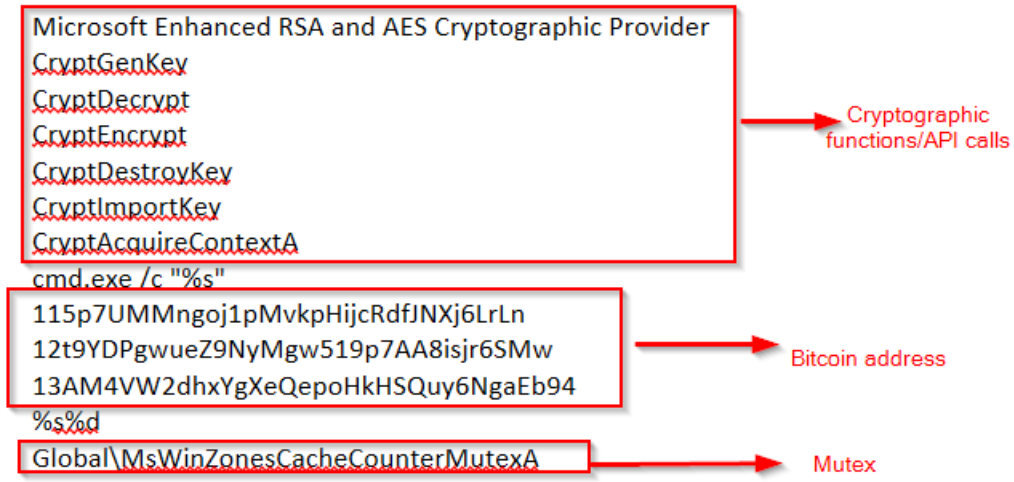


Figure 5 Floss Output interesting strings capturing bitcoin address, cryptographic calls and mutex

PEStudio

It is a 32 bit program compiled on 20th Nov 2010 written in c++ language.

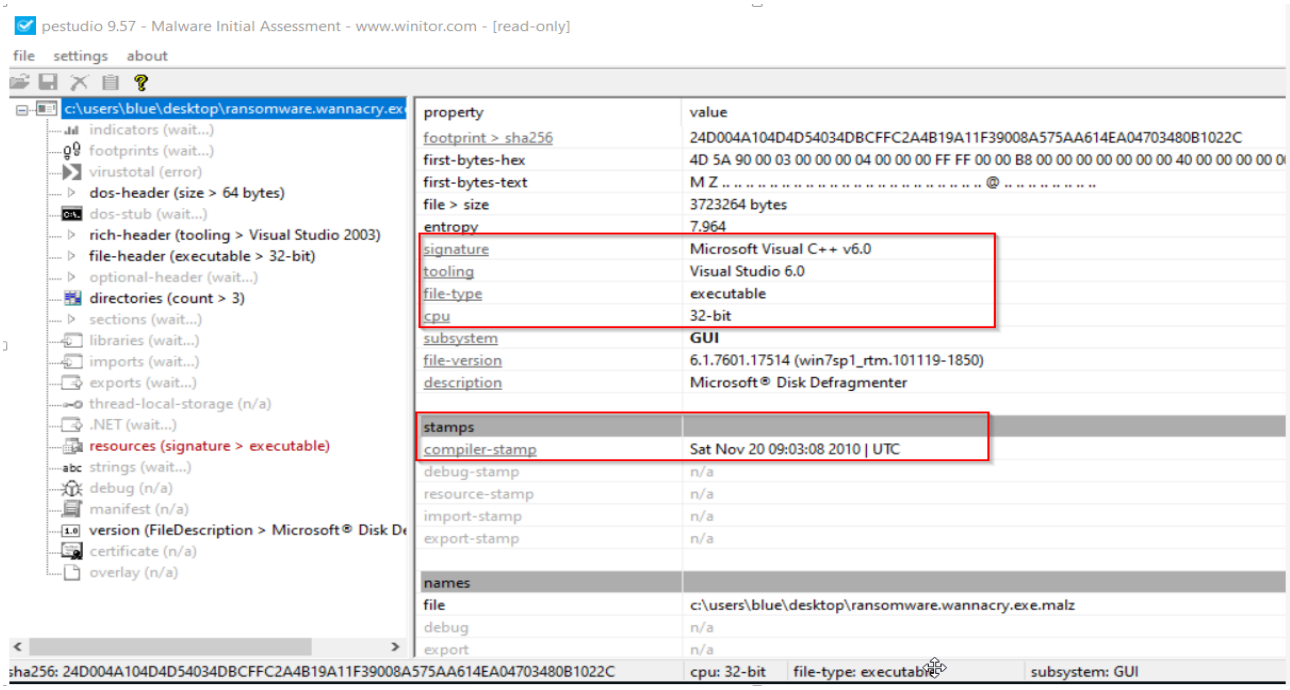


Figure 6 PE Studio Output

Not much difference between raw size and virtual size of the binary ---> not packed.

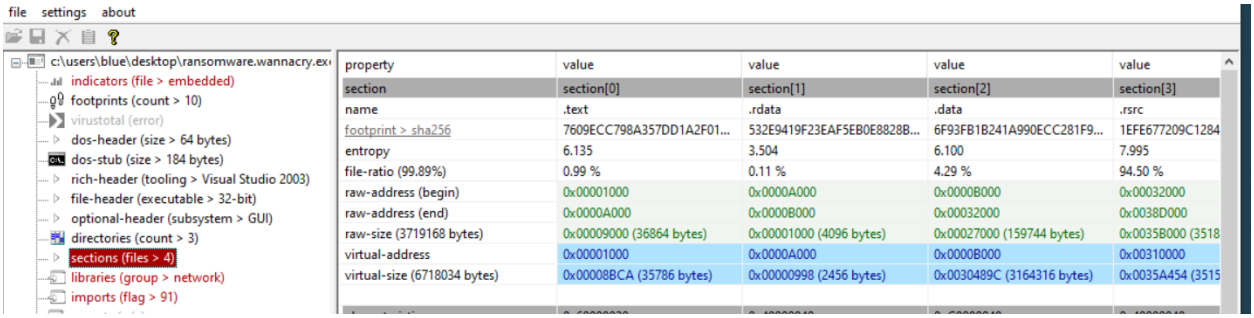
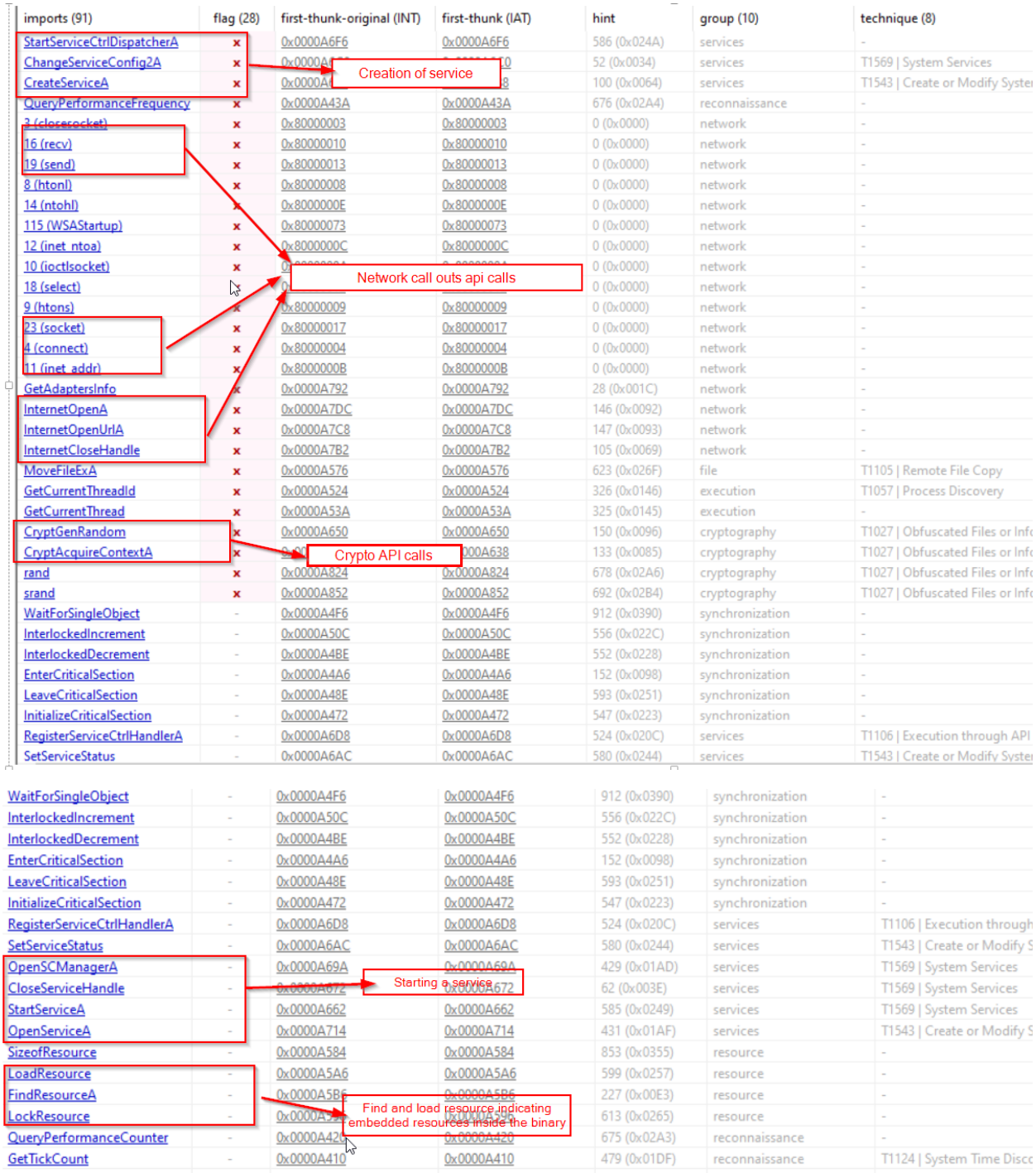


Figure 7PE Studio Output showing Virtual size and raw size

Interesting API calls showing the binary has embedded resources and writes new files into the system and also encryption functionality.



imports (91)	flag (28)	first-thunk-original (INT)	first-thunk (IAT)	hint	group (10)	technique (8)
StartServiceCtrlDispatcherA	x	0x0000A6F6	0x0000A6F6	586 (0x024A)	services	-
ChangeServiceConfig2A	x	0x0000A[illegible]10		52 (0x0034)	services	T1569 | System Services
CreateServiceA	x	0x0000A[illegible]8		100 (0x0064)	services	T1543 | Create or Modify Syste
QueryPerformanceFrequency	x	0x0000A43A	0x0000A43A	676 (0x02A4)	reconnaissance	-
3 (closesocket)	x	0x80000003	0x80000003	0 (0x0000)	network	-
16 (recv)	x	0x80000010	0x80000010	0 (0x0000)	network	-
19 (send)	x	0x80000013	0x80000013	0 (0x0000)	network	-
8 (htonl)	x	0x80000008	0x80000008	0 (0x0000)	network	-
14 (ntohl)	x	0x8000000E	0x8000000E	0 (0x0000)	network	-
115 (WSAStartup)	x	0x80000073	0x80000073	0 (0x0000)	network	-
12 (inet_ntoa)	x	0x8000000C	0x8000000C	0 (0x0000)	network	-
10 (ioctlsocket)	x	0[illegible]		0 (0x0000)	network	-
18 (select)	x	0[illegible]		0 (0x0000)	network	-
9 (htons)	x	0x80000009	0x80000009	0 (0x0000)	network	-
23 (socket)	x	0x80000017	0x80000017	0 (0x0000)	network	-
4 (connect)	x	0x80000004	0x80000004	0 (0x0000)	network	-
11 (inet_addr)	x	0x8000000B	0x8000000B	0 (0x0000)	network	-
GetAdaptersInfo	x	0x0000A792	0x0000A792	28 (0x001C)	network	-
InternetOpenA	x	0x0000A7DC	0x0000A7DC	146 (0x0092)	network	-
InternetOpenUrlA	x	0x0000A7C8	0x0000A7C8	147 (0x0093)	network	-
InternetCloseHandle	x	0x0000A7B2	0x0000A7B2	105 (0x0069)	network	-
MoveFileExA	x	0x0000A576	0x0000A576	623 (0x026F)	file	T1105 | Remote File Copy
GetCurrentThreadId	x	0x0000A524	0x0000A524	326 (0x0146)	execution	T1057 | Process Discovery
GetCurrentThread	x	0x0000A53A	0x0000A53A	325 (0x0145)	execution	-
CryptGenRandom	x	0x0000A650	0x0000A650	150 (0x0096)	cryptography	T1027 | Obfuscated Files or Inf
CryptAcquireContextA	x	0x[illegible]	0x0000A638	133 (0x0085)	cryptography	T1027 | Obfuscated Files or Inf
rand	x	0x0000A824	0x0000A824	678 (0x02A6)	cryptography	T1027 | Obfuscated Files or Inf
srand	x	0x0000A852	0x0000A852	692 (0x02B4)	cryptography	T1027 | Obfuscated Files or Inf
WaitForSingleObject	-	0x0000A4F6	0x0000A4F6	912 (0x0390)	synchronization	-
InterlockedIncrement	-	0x0000A50C	0x0000A50C	556 (0x022C)	synchronization	-
InterlockedDecrement	-	0x0000A4BE	0x0000A4BE	552 (0x0228)	synchronization	-
EnterCriticalSection	-	0x0000A4A6	0x0000A4A6	152 (0x0098)	synchronization	-
LeaveCriticalSection	-	0x0000A48E	0x0000A48E	593 (0x0251)	synchronization	-
InitializeCriticalSection	-	0x0000A472	0x0000A472	547 (0x0223)	synchronization	-
RegisterServiceCtrlHandlerA	-	0x0000A6D8	0x0000A6D8	524 (0x020C)	services	T1106 | Execution through API
SetServiceStatus	-	0x0000A6AC	0x0000A6AC	580 (0x0244)	services	T1543 | Create or Modify Syste
WaitForSingleObject	-	0x0000A4F6	0x0000A4F6	912 (0x0390)	synchronization	-
InterlockedIncrement	-	0x0000A50C	0x0000A50C	556 (0x022C)	synchronization	-
InterlockedDecrement	-	0x0000A4BE	0x0000A4BE	552 (0x0228)	synchronization	-
EnterCriticalSection	-	0x0000A4A6	0x0000A4A6	152 (0x0098)	synchronization	-
LeaveCriticalSection	-	0x0000A48E	0x0000A48E	593 (0x0251)	synchronization	-
InitializeCriticalSection	-	0x0000A472	0x0000A472	547 (0x0223)	synchronization	-
RegisterServiceCtrlHandlerA	-	0x0000A6D8	0x0000A6D8	524 (0x020C)	services	T1106 | Execution through
SetServiceStatus	-	0x0000A6AC	0x0000A6AC	580 (0x0244)	services	T1543 | Create or Modify S
OpenSCManagerA	-	0x0000A69A	0x0000A69A	429 (0x01AD)	services	T1569 | System Services
CloseServiceHandle	-	0x0000A672	0x0000A672	62 (0x003E)	services	T1569 | System Services
StartServiceA	-	0x0000A662	0x0000A662	585 (0x0249)	services	T1569 | System Services
OpenServiceA	-	0x0000A714	0x0000A714	431 (0x01AF)	services	T1543 | Create or Modify S
SizeofResource	-	0x0000A584	0x0000A584	853 (0x0355)	resource	-
LoadResource	-	0x0000A5A6	0x0000A5A6	599 (0x0257)	resource	-
FindResourceA	-	0x0000A5B6	0x0000A5B6	227 (0x00E3)	resource	-
LockResource	-	0x0000A5[illegible]	0x0000A5[illegible]	613 (0x0265)	resource	-
QueryPerformanceCounter	-	0x0000A42[illegible]	0x0000A420	675 (0x02A3)	reconnaissance	-
GetTickCount	-	0x0000A410	0x0000A410	479 (0x01DF)	reconnaissance	T1124 | System Time Disco

Figure 8 PE Studio - Interesting API calls

Basic Dynamic Analysis

{Screenshots and description about basic dynamic artifacts and methods}

Initial Detonation with InetSim

Does nothing, the process exits as soon as it started.

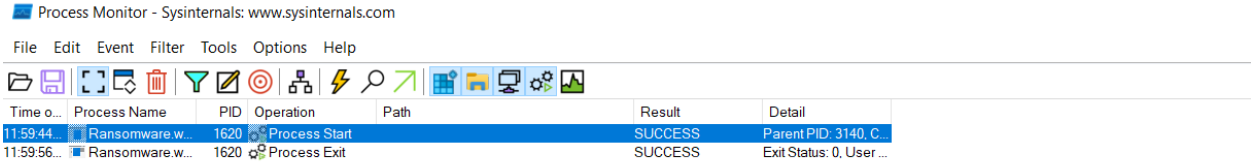


Figure 9 Initial detonation with InetSim running

Detonation without inetsim

Detonating the malware with procmon running in the background, captures many interesting details.

Firstly all the important documents/files in our system is encrypted and the desktop wallpaper is changed and decryptor window pops up on the screen.The decryptor window states that our files are encrypted and demands to pay a ransom amount to a bitcoin address in order to recover the files. It also starts a timer to make the payment.

Figure 10 Desktop wallpaper change and decryptor window

Clicking on the decrypt button, decrypts some of the sample encrypted files to show that these are recoverable and asks to pay the ransom in order to recover all the files

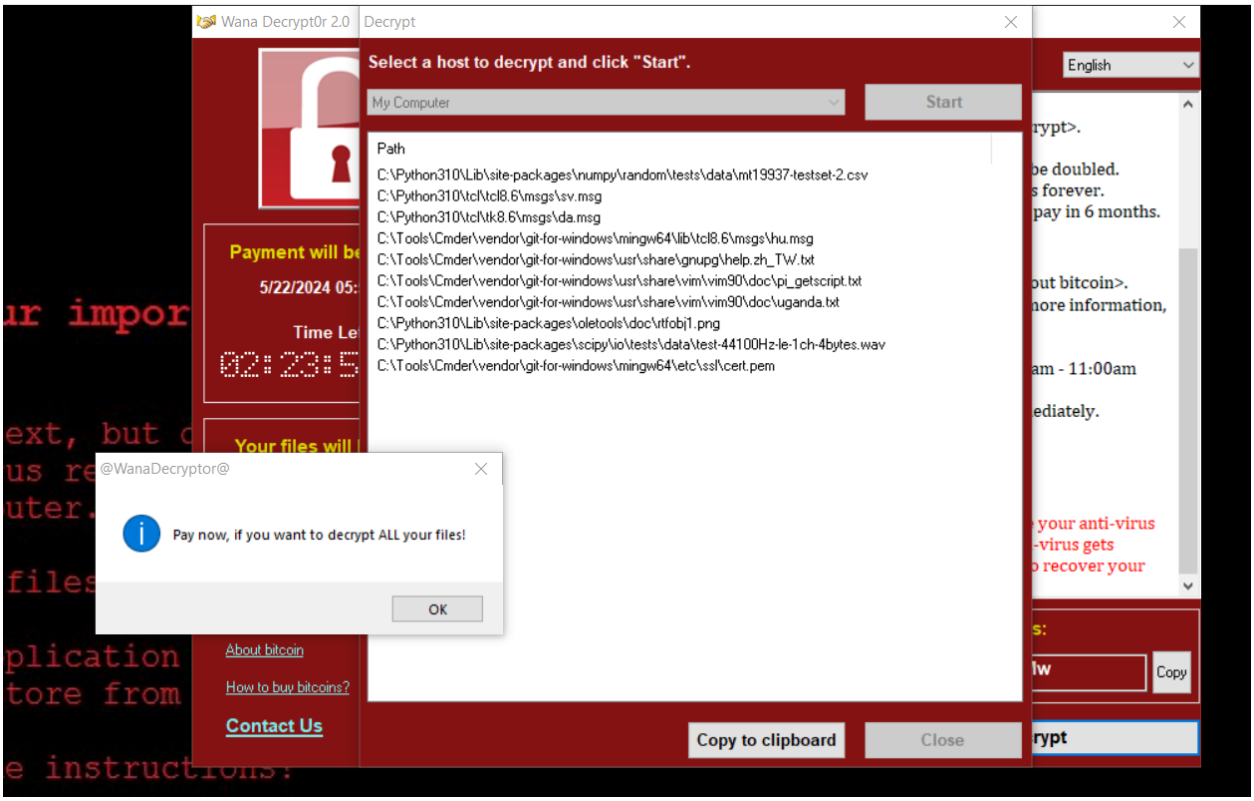
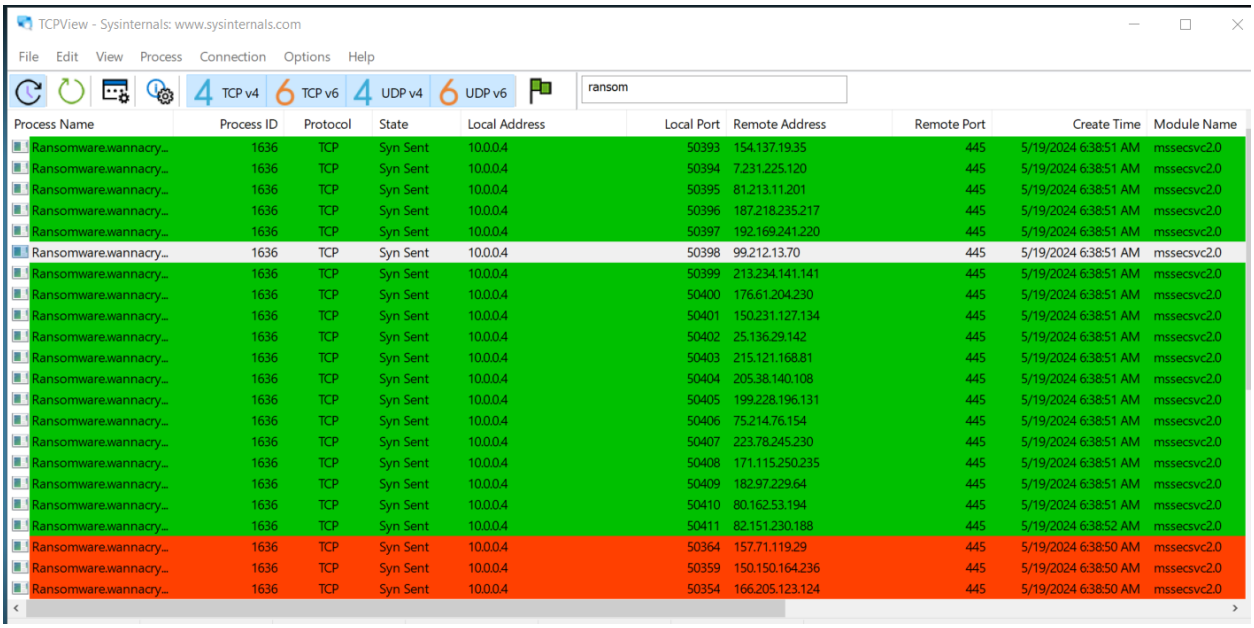


Figure 11 Decryptor program performing decryption operation on certain files in system

Procmon & TCPView analysis

Now looking at the procmon logs – we can see the program tries attempts to connect over different IP's. Likely scanning the local network to spread through smb (port 445).



Process Name	Process ID	Protocol	State	Local Address	Local Port	Remote Address	Remote Port	Create Time	Module Name
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50393	154.137.19.35	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50394	7.231.225.120	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50395	81.213.11.201	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50396	187.218.235.217	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50397	192.169.241.220	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50398	99.212.13.70	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50399	213.234.141.141	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50400	176.61.204.230	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50401	150.231.127.134	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50402	25.136.29.142	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50403	215.121.168.81	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50404	205.38.140.108	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50405	199.228.196.131	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50406	75.214.76.154	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50407	223.78.245.230	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50408	171.115.250.235	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50409	182.97.229.64	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50410	80.162.53.194	445	5/19/2024 6:38:51 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50411	82.151.230.188	445	5/19/2024 6:38:52 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50364	157.71.119.29	445	5/19/2024 6:38:50 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50359	150.150.164.236	445	5/19/2024 6:38:50 AM	mssecsvc2.0
Ransomware.wannacry...	1636	TCP	Syn Sent	10.0.0.4	50354	166.205.123.124	445	5/19/2024 6:38:50 AM	mssecsvc2.0

Figure 12 TCP view of malware scanning the network to exploit SMB vuln

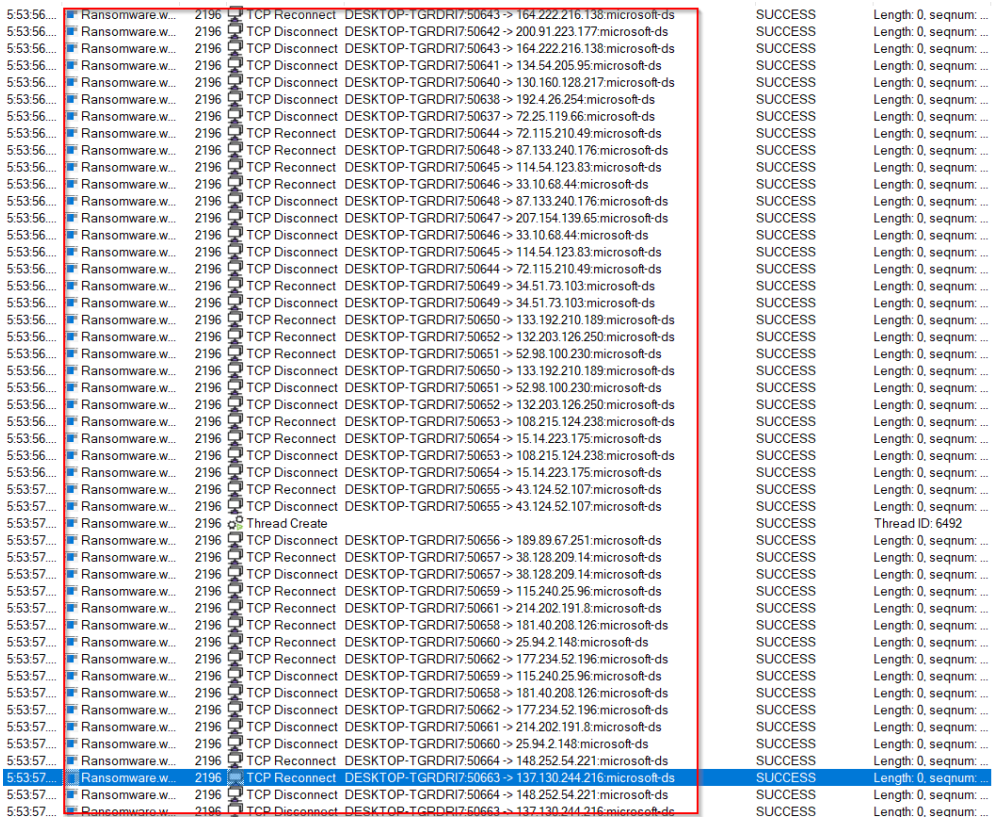


Figure 13 Procmon logs showing TCP connection over local network

Filtering for process creation events, we can see the initial binary spawns tasksche.exe from windows directory

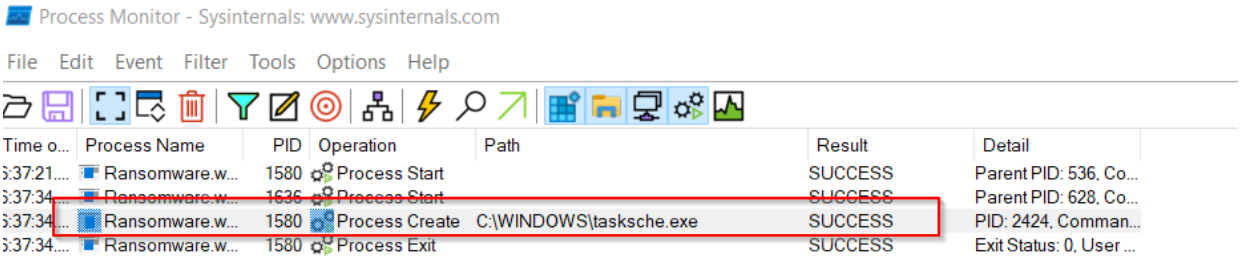
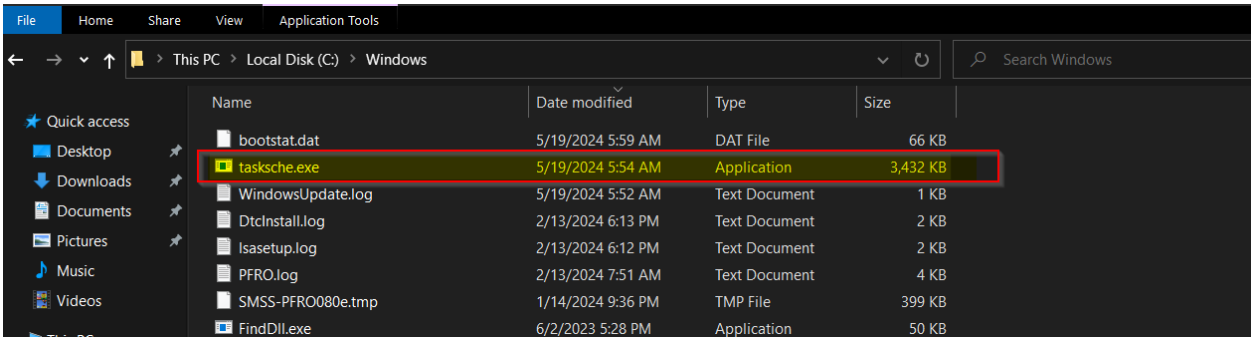
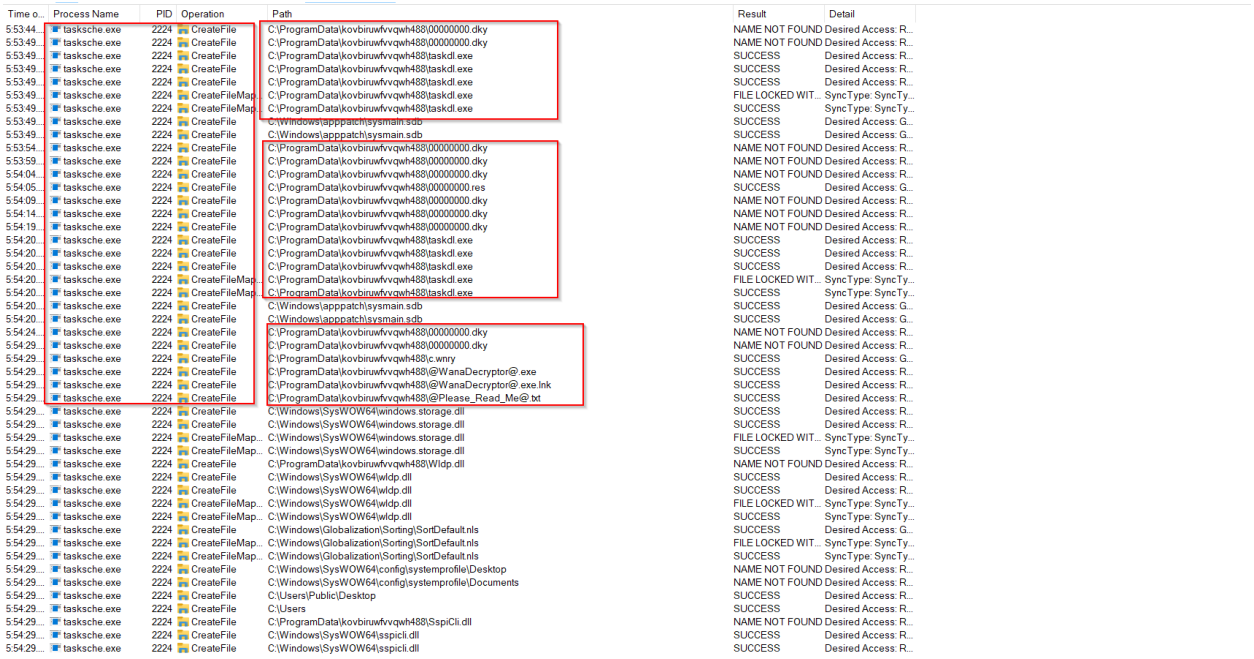


Figure 14 Child process tasksche.exe being created

Reviewing windows directory we can see tasksche.exe has been modified.



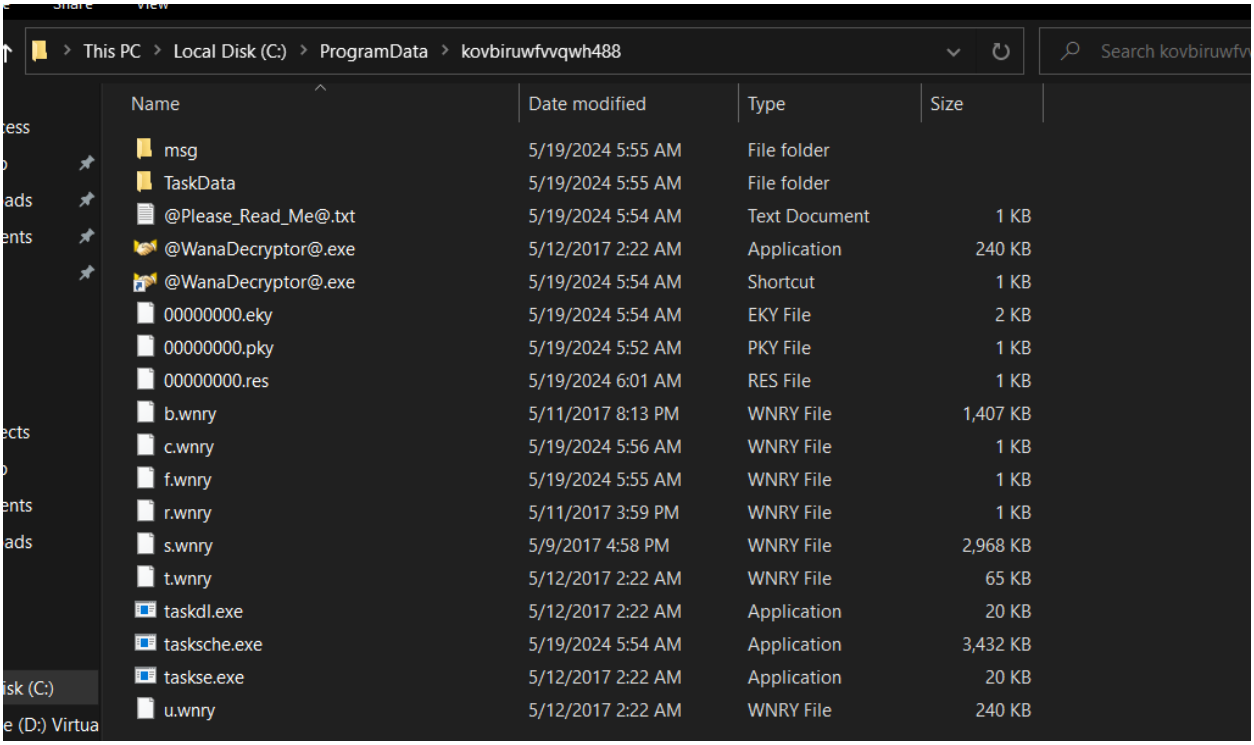
Filtering for tasksche.exe, we can see this too creates multiple next stage files in randomly named folder programdata directory. This appears to be staging area for the program.



Time o...	Process Name	PID	Operation	Path	Result	Detail
5:53:44...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\00000000.dky	NAME NOT FOUND	Desired Access: R...
5:53:49...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\00000000.dky	NAME NOT FOUND	Desired Access: R...
5:53:49...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe	SUCCESS	Desired Access: R...
5:53:49...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe	SUCCESS	Desired Access: R...
5:53:49...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe	SUCCESS	Desired Access: R...
5:53:49...	tasksche.exe	2224	CreateFileMap...	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe	FILE LOCKED WIT...	SyncType: SyncTy...
5:53:49...	tasksche.exe	2224	CreateFileMap...	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe	SUCCESS	SyncType: SyncTy...
5:53:49...	tasksche.exe	2224	CreateFile	C:\Windows\apppatch\sysmain.sdb	SUCCESS	Desired Access: G...
5:53:49...	tasksche.exe	2224	CreateFile	C:\Windows\apppatch\sysmain.sdb	SUCCESS	Desired Access: G...
5:53:54...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\00000000.dky	NAME NOT FOUND	Desired Access: R...
5:53:59...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\00000000.dky	NAME NOT FOUND	Desired Access: R...
5:54:04...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\00000000.dky	NAME NOT FOUND	Desired Access: R...
5:54:05...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\00000000.res	SUCCESS	Desired Access: G...
5:54:09...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\00000000.dky	NAME NOT FOUND	Desired Access: R...
5:54:14...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\00000000.dky	NAME NOT FOUND	Desired Access: R...
5:54:19...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\00000000.dky	NAME NOT FOUND	Desired Access: R...
5:54:20...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe	SUCCESS	Desired Access: R...
5:54:20...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe	SUCCESS	Desired Access: R...
5:54:20...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe	SUCCESS	Desired Access: R...
5:54:20...	tasksche.exe	2224	CreateFileMap...	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe	FILE LOCKED WIT...	SyncType: SyncTy...
5:54:20...	tasksche.exe	2224	CreateFileMap...	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe	SUCCESS	SyncType: SyncTy...
5:54:20...	tasksche.exe	2224	CreateFile	C:\Windows\apppatch\sysmain.sdb	SUCCESS	Desired Access: G...
5:54:20...	tasksche.exe	2224	CreateFile	C:\Windows\apppatch\sysmain.sdb	SUCCESS	Desired Access: G...
5:54:24...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\00000000.dky	NAME NOT FOUND	Desired Access: R...
5:54:29...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\00000000.dky	NAME NOT FOUND	Desired Access: R...
5:54:29...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\c.wnry	SUCCESS	Desired Access: G...
5:54:29...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\@WanaDecryptor@.exe	SUCCESS	Desired Access: R...
5:54:29...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\@WanaDecryptor@.exe.lnk	SUCCESS	Desired Access: R...
5:54:29...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\@Please_Read_Me@.txt	SUCCESS	Desired Access: R...
5:54:29...	tasksche.exe	2224	CreateFile	C:\Windows\SysWOW64\windows.storage.dll	SUCCESS	Desired Access: R...
5:54:29...	tasksche.exe	2224	CreateFile	C:\Windows\SysWOW64\windows.storage.dll	SUCCESS	Desired Access: R...
5:54:29...	tasksche.exe	2224	CreateFileMap...	C:\Windows\SysWOW64\windows.storage.dll	FILE LOCKED WIT...	SyncType: SyncTy...
5:54:29...	tasksche.exe	2224	CreateFileMap...	C:\Windows\SysWOW64\windows.storage.dll	SUCCESS	SyncType: SyncTy...
5:54:29...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\Wldp.dll	NAME NOT FOUND	Desired Access: R...
5:54:29...	tasksche.exe	2224	CreateFile	C:\Windows\SysWOW64\wldp.dll	SUCCESS	Desired Access: R...
5:54:29...	tasksche.exe	2224	CreateFile	C:\Windows\SysWOW64\wldp.dll	SUCCESS	Desired Access: R...
5:54:29...	tasksche.exe	2224	CreateFileMap...	C:\Windows\SysWOW64\wldp.dll	FILE LOCKED WIT...	SyncType: SyncTy...
5:54:29...	tasksche.exe	2224	CreateFileMap...	C:\Windows\SysWOW64\wldp.dll	SUCCESS	SyncType: SyncTy...
5:54:29...	tasksche.exe	2224	CreateFile	C:\Windows\Globalization\Sorting\SortDefault.nls	SUCCESS	Desired Access: G...
5:54:29...	tasksche.exe	2224	CreateFileMap...	C:\Windows\Globalization\Sorting\SortDefault.nls	FILE LOCKED WIT...	SyncType: SyncTy...
5:54:29...	tasksche.exe	2224	CreateFileMap...	C:\Windows\Globalization\Sorting\SortDefault.nls	SUCCESS	SyncType: SyncTy...
5:54:29...	tasksche.exe	2224	CreateFile	C:\Windows\SysWOW64\config\systemprofile\Desktop	NAME NOT FOUND	Desired Access: R...
5:54:29...	tasksche.exe	2224	CreateFile	C:\Windows\SysWOW64\config\systemprofile\Documents	NAME NOT FOUND	Desired Access: R...
5:54:29...	tasksche.exe	2224	CreateFile	C:\Users\Public\Desktop	SUCCESS	Desired Access: R...
5:54:29...	tasksche.exe	2224	CreateFile	C:\Users	SUCCESS	Desired Access: R...
5:54:29...	tasksche.exe	2224	CreateFile	C:\ProgramData\kovbiruwfvvqwh488\SspiCli.dll	NAME NOT FOUND	Desired Access: R...
5:54:29...	tasksche.exe	2224	CreateFile	C:\Windows\SysWOW64\sspicli.dll	SUCCESS	Desired Access: R...
5:54:29...	tasksche.exe	2224	CreateFile	C:\Windows\SysWOW64\sspicli.dll	SUCCESS	Desired Access: R...

Figure 15 Procmon logs showing creation of next stage files

Programdata directory with the next stage files



Taksche.exe also executes multiple files that were created in previous step

Time o...	Process Name	PID	Operation	Path
5:53:49....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe
5:54:20....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe
5:54:50....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe
5:55:20....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe
5:55:39....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\@WanaDecryptor@.exe
5:55:39....	tasksche.exe	2224	Process Create	C:\Windows\SysWOW64\cmd.exe
5:55:49....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskse.exe
5:55:49....	tasksche.exe	2224	Process Create	C:\Windows\SysWOW64\cmd.exe
5:55:51....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe
5:56:19....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskse.exe
5:56:21....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe
5:56:49....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskse.exe
5:56:52....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe
5:57:19....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskse.exe
5:57:22....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe
5:57:50....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskse.exe
5:57:52....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe
5:58:20....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskse.exe
5:58:23....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe
5:58:50....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskse.exe
5:58:53....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe
5:59:20....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskse.exe
5:59:23....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe
5:59:50....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskse.exe
5:59:53....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe
6:00:20....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskse.exe
6:00:23....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe
6:00:50....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskse.exe
6:00:53....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe
6:01:20....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskse.exe
6:01:23....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe
6:01:50....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskse.exe
6:01:53....	tasksche.exe	2224	Process Create	C:\ProgramData\kovbiruwfvvqwh488\taskdl.exe

Advanced Static Analysis

{Screenshots and description about findings during advanced static analysis}

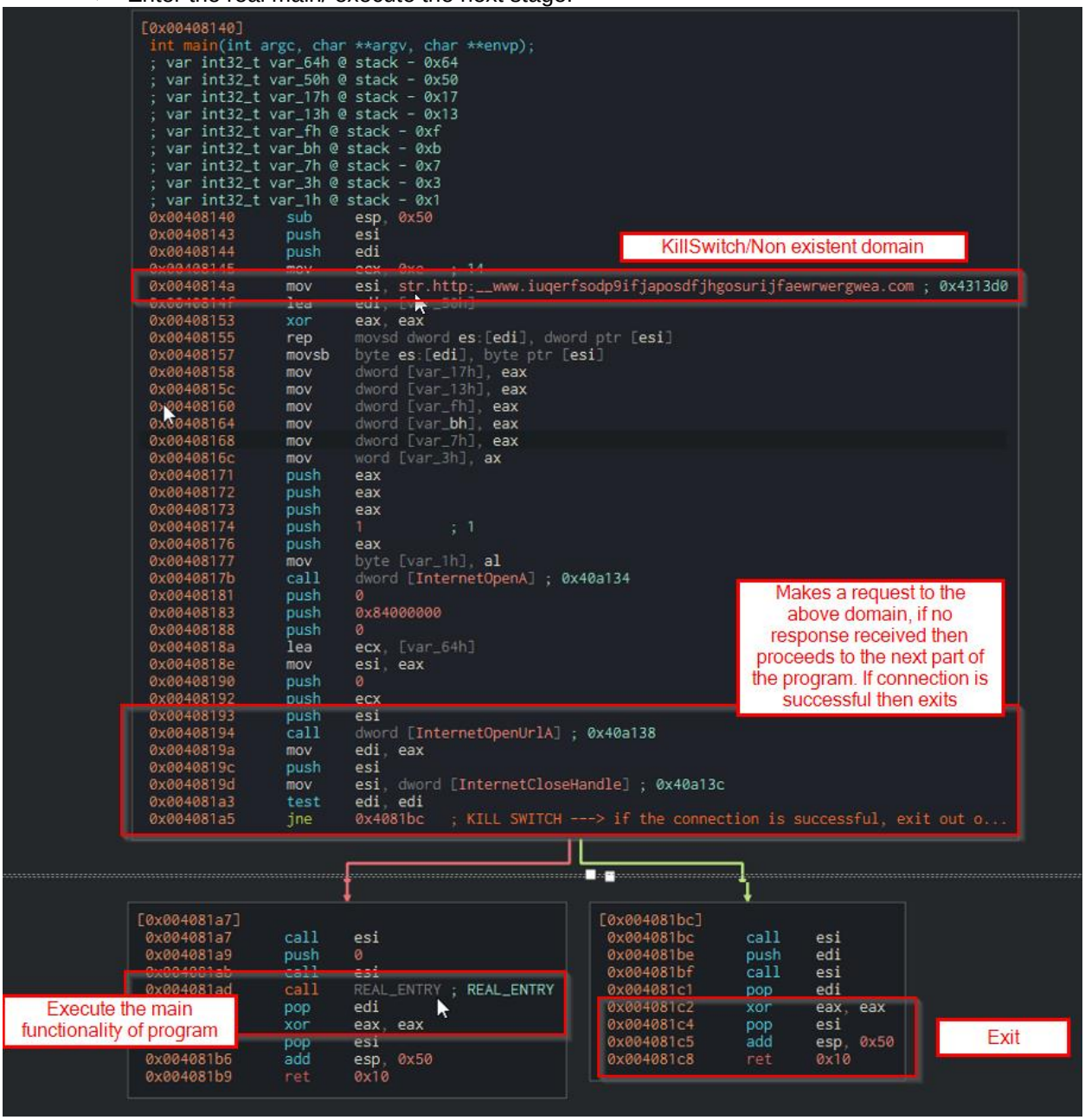
Initial Kill Switch

Attempts to reach out to “www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]com”.
If connection is successful :

- ⇨ Do nothing and exit the program

If connection fails :

- ⇨ Enter the real main/ execute the next stage.



Real Entry/ Main execution call

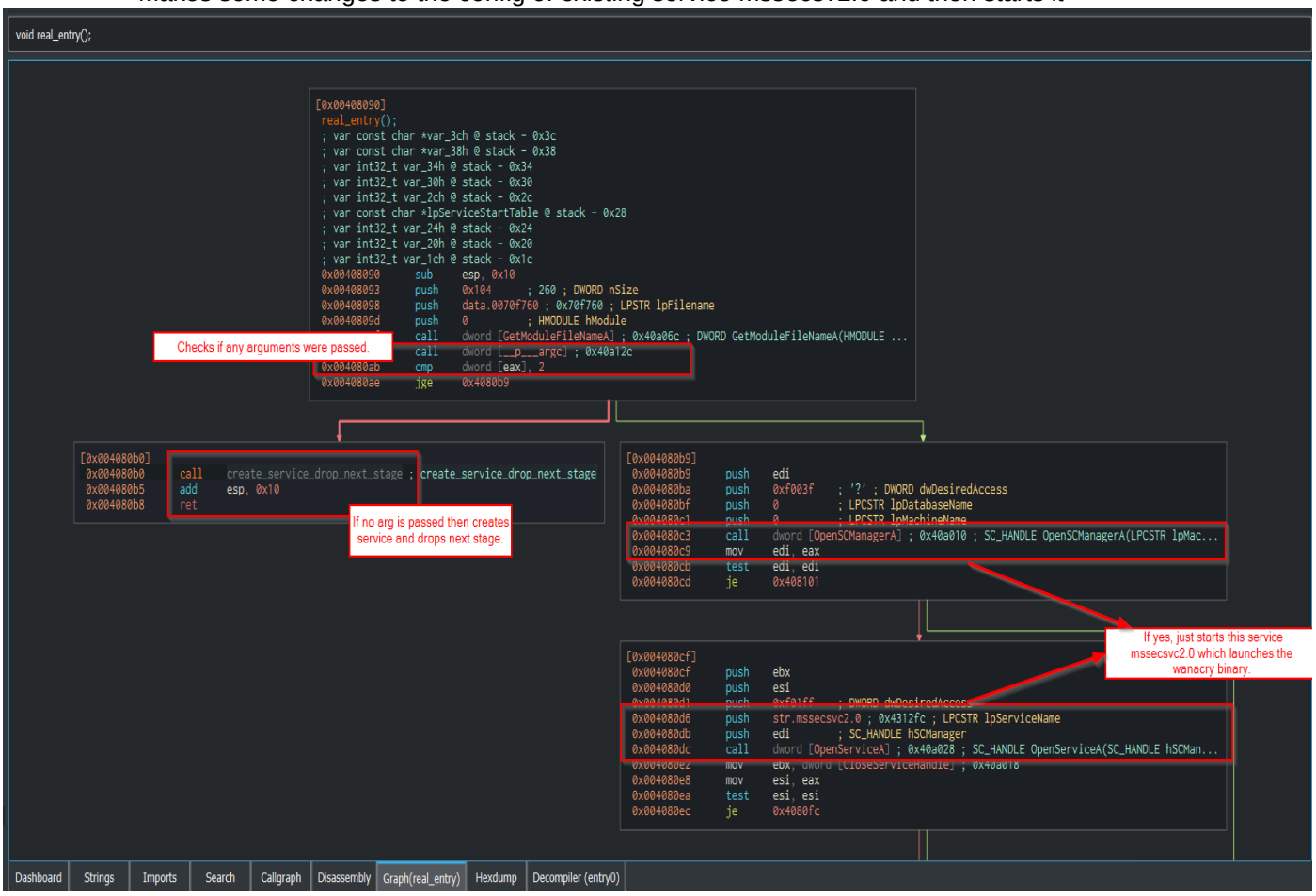
Initially checks if there are any arguments :

If not then :

⇨ Creates service and drops the next stage

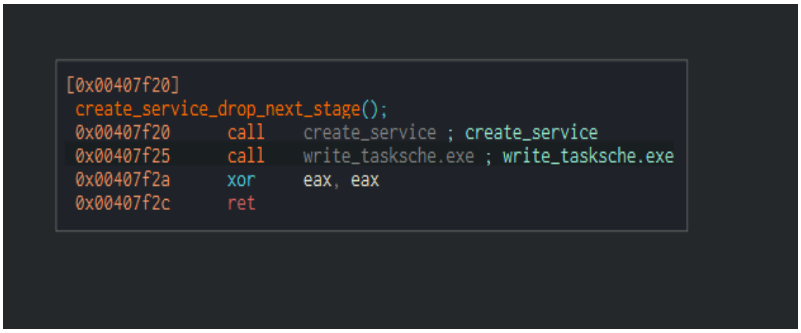
If yes then :

⇨ Makes some changes to the config of existing service mssecsv2.0 and then starts it



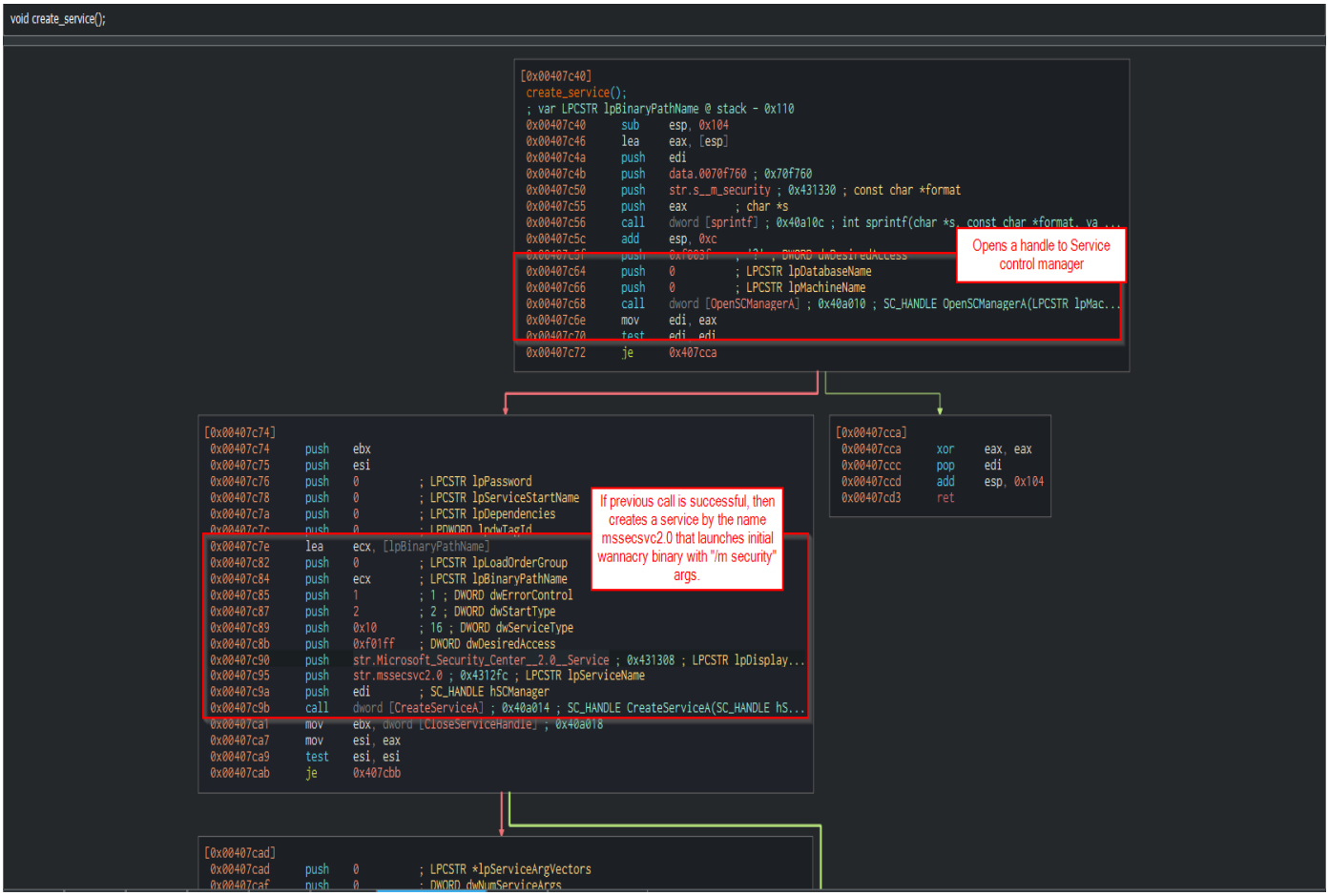
Create_service_drop_next_stage_function call

As the name suggests, makes a function call to create a service for persistence and then writes the next stage payload to tasksche.exe

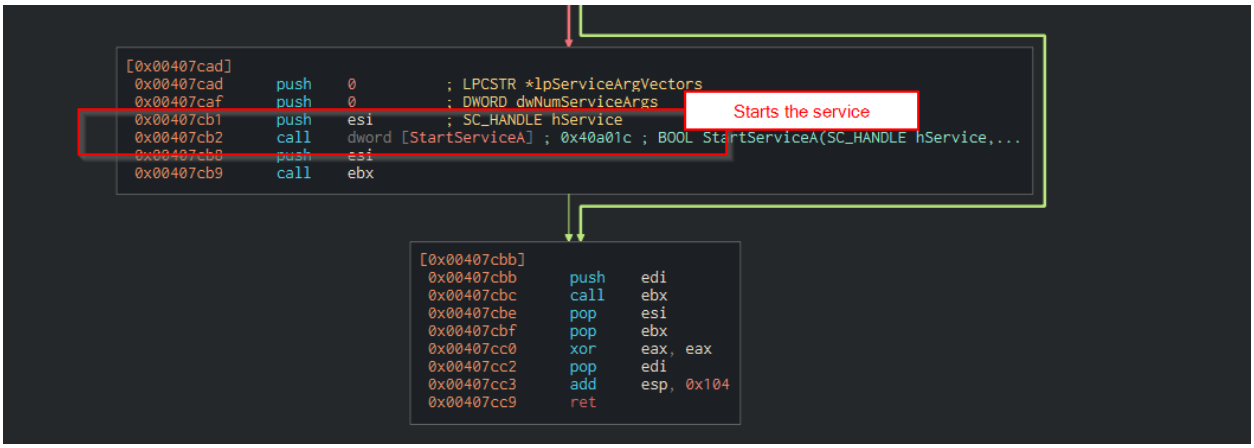


Creation of service

Name of the service – mssecsvc2 ➔based on tcpview logs seen in dynamic analysis this is responsible for scanning/spreading over local network by exploiting smb vulnerabilty

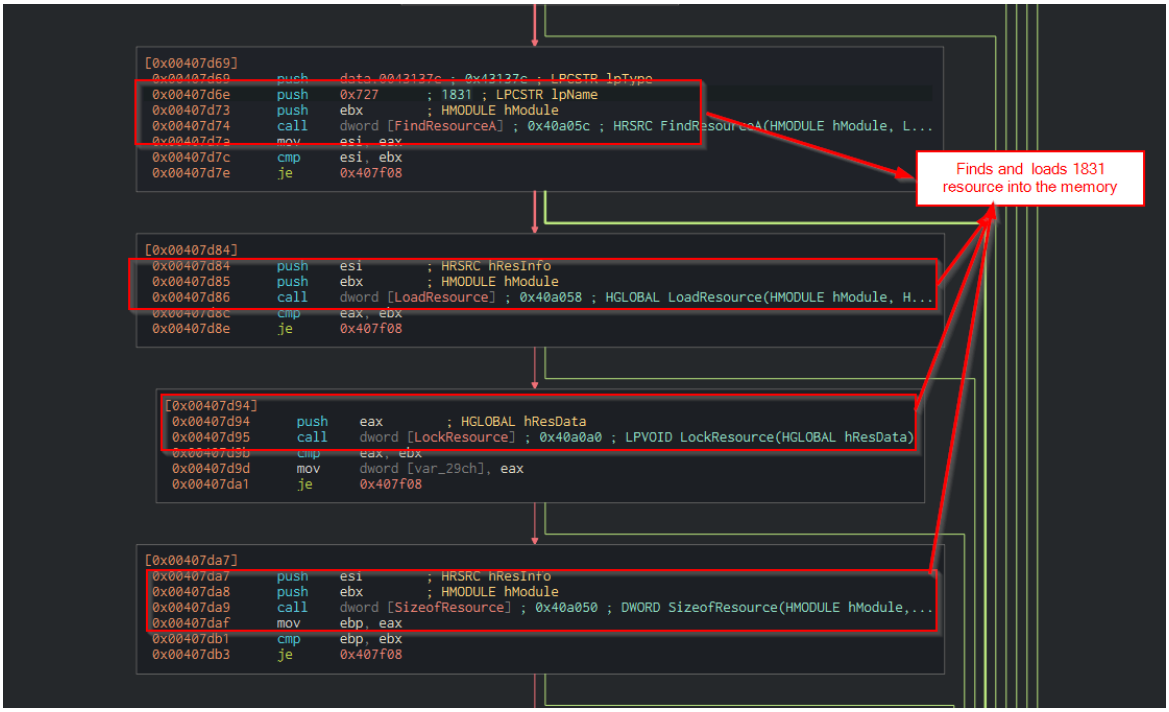


Then just starts this newly created service.

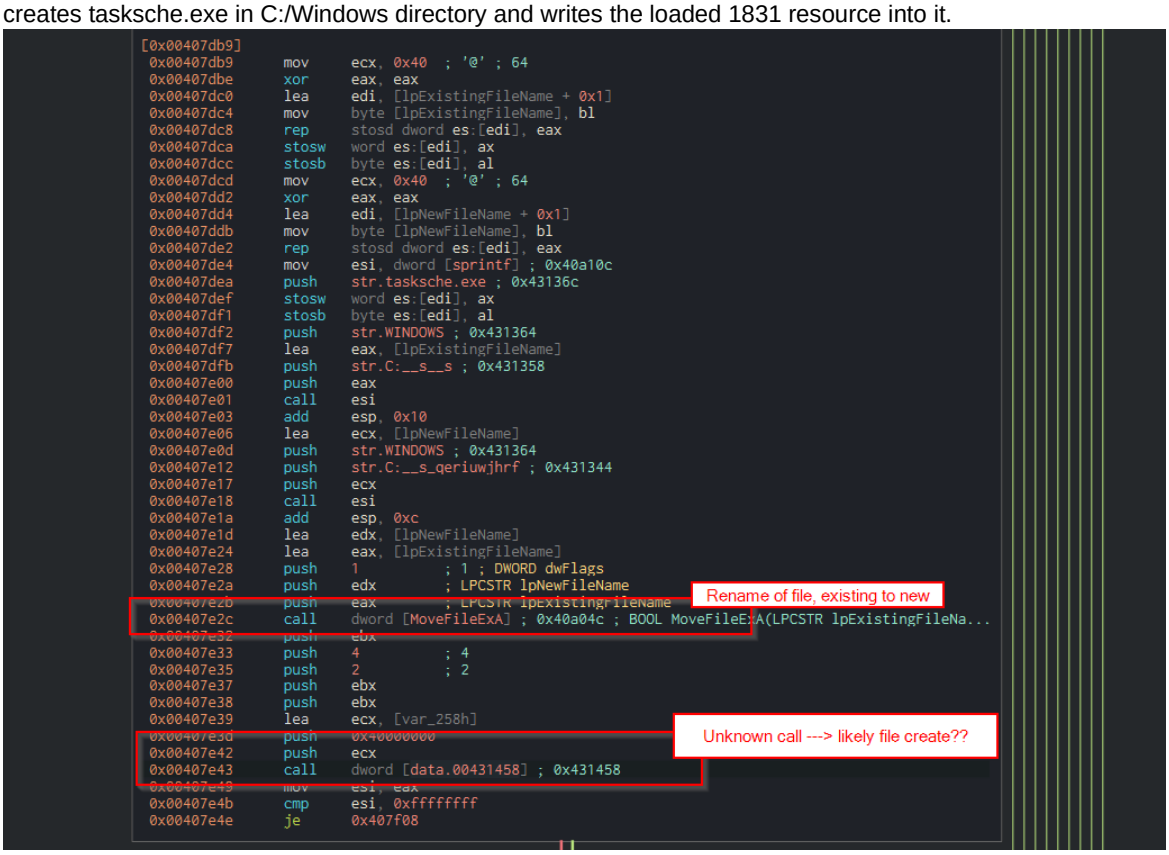


Creation of next stage payload

Finds and loads embedded 1831 resource.



Next part is is not clear in disassembled output, but we can see the translated function calls in debugger. Basically it creates tasksche.exe in C:/Windows directory and writes the loaded 1831 resource into it.

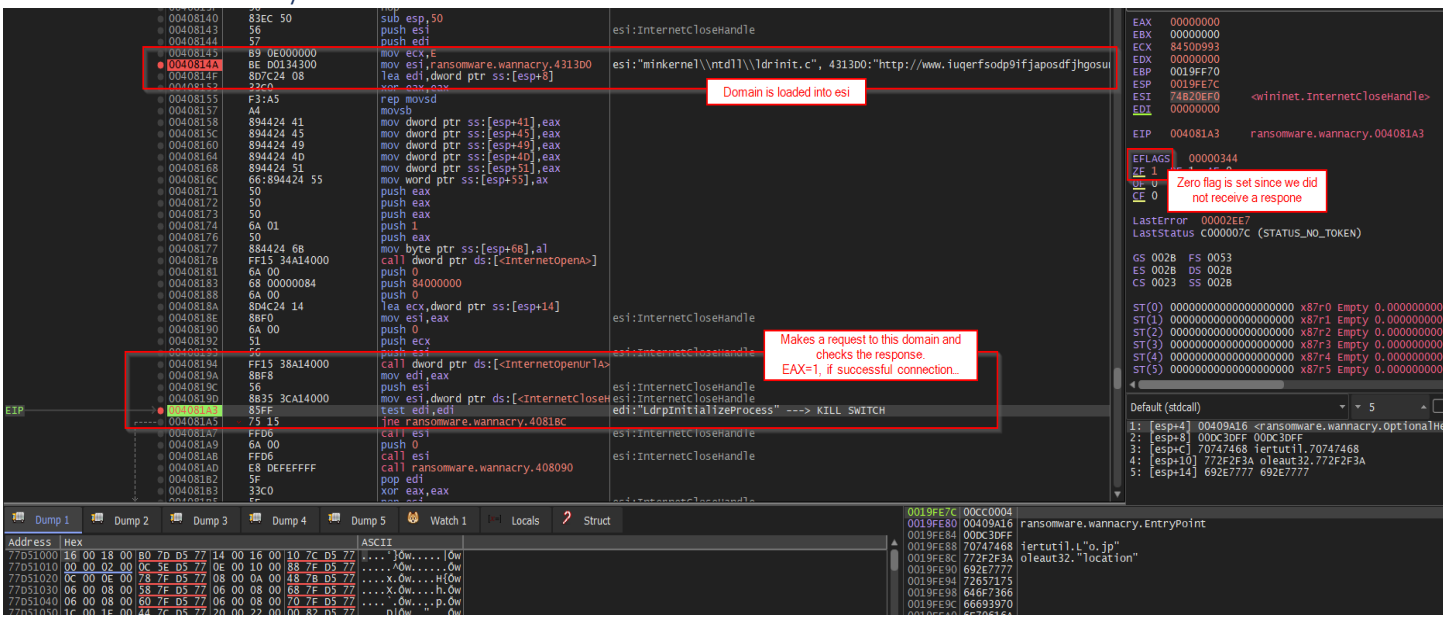


Advanced Dynamic Analysis

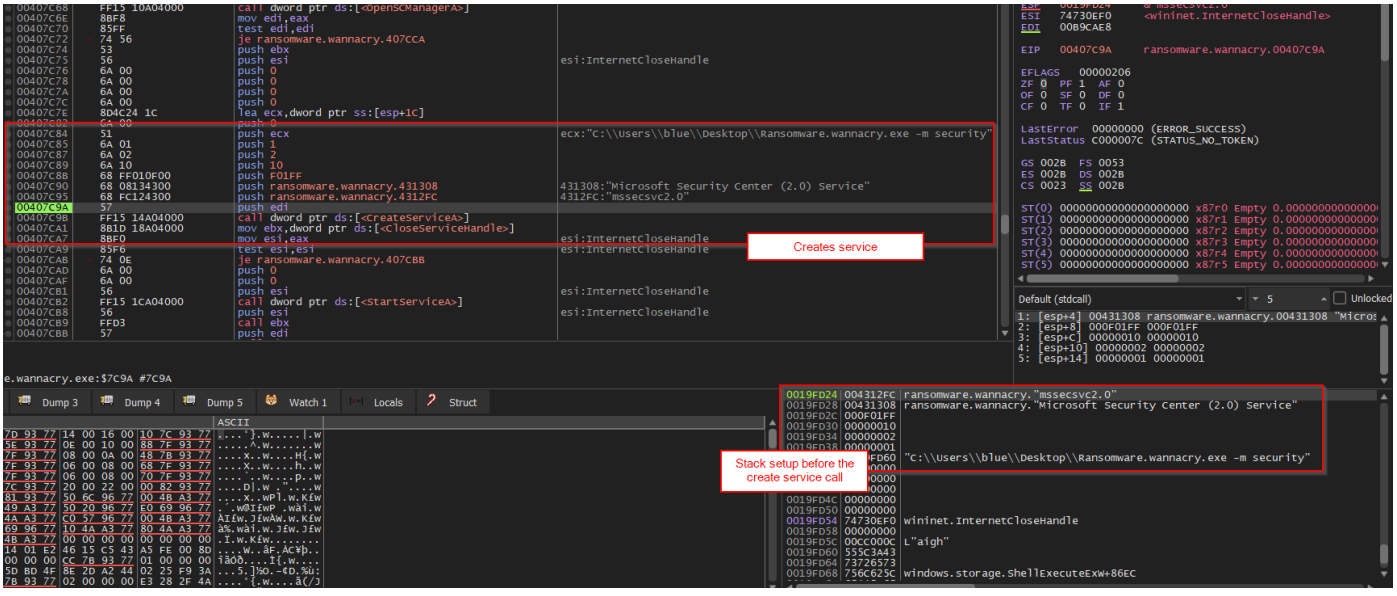
{Screenshots and description about advanced dynamic artifacts and methods}

Debugging Wannacry.exe – Initial payload

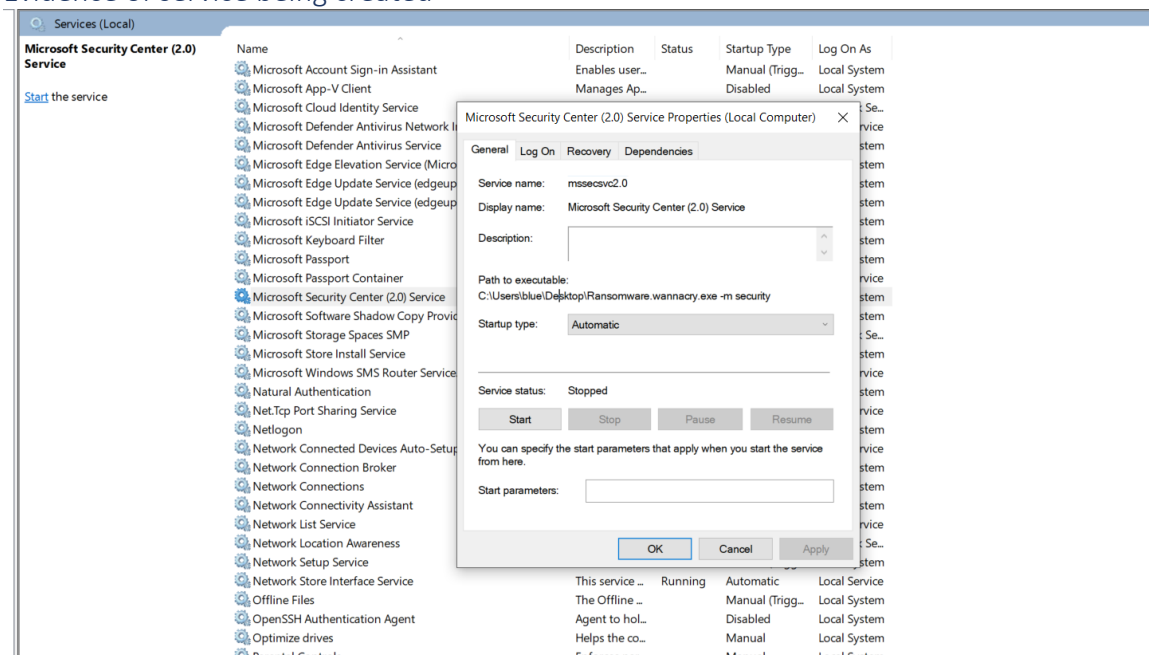
Kill Switch functionality



Service Creation:

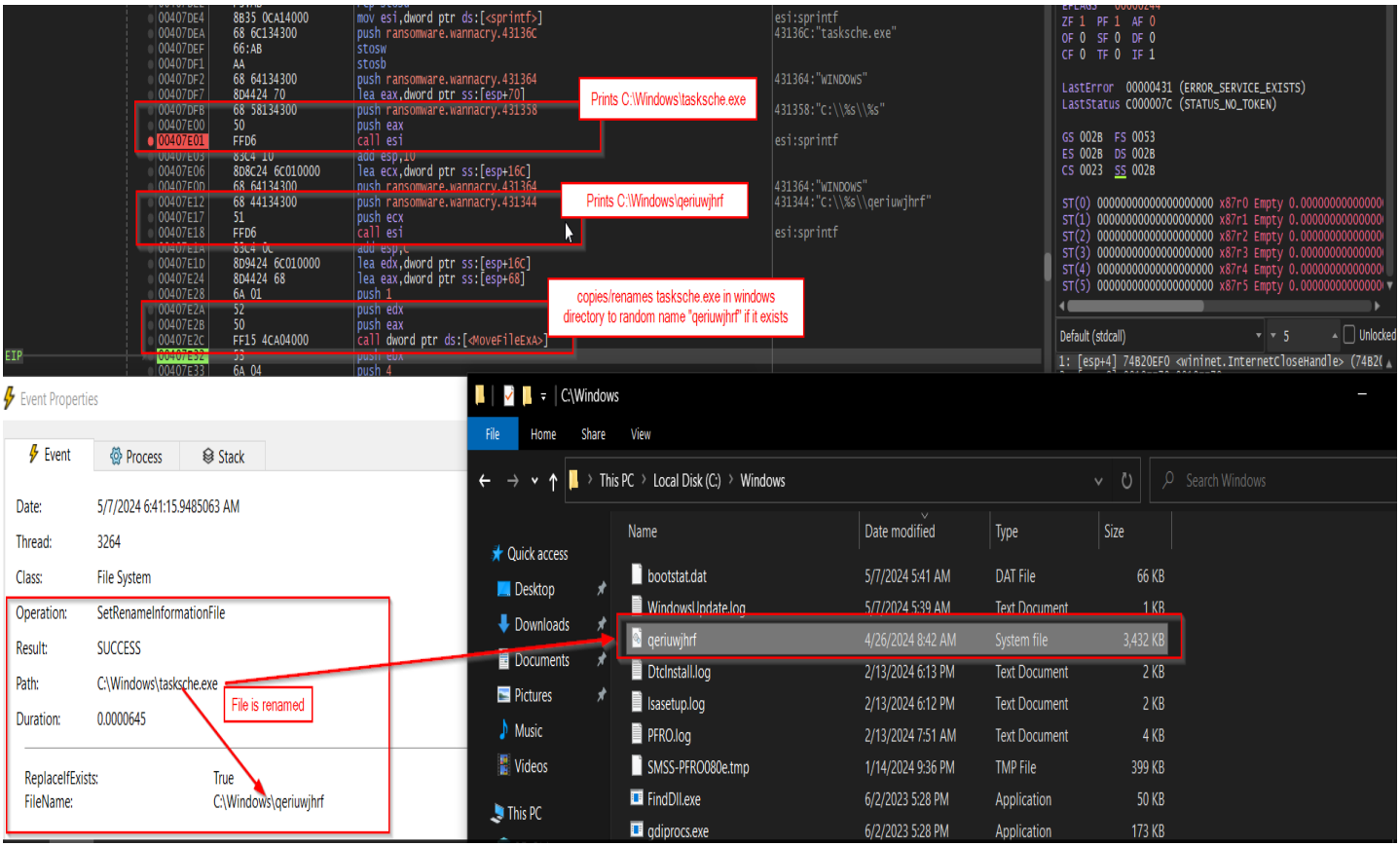


Evidence of service being created

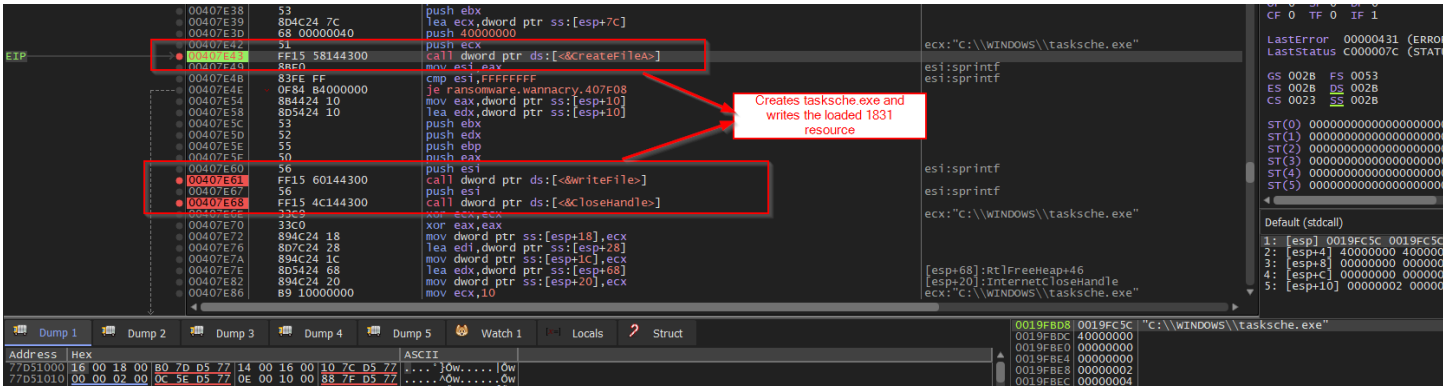


Write/Rename Next stage payload

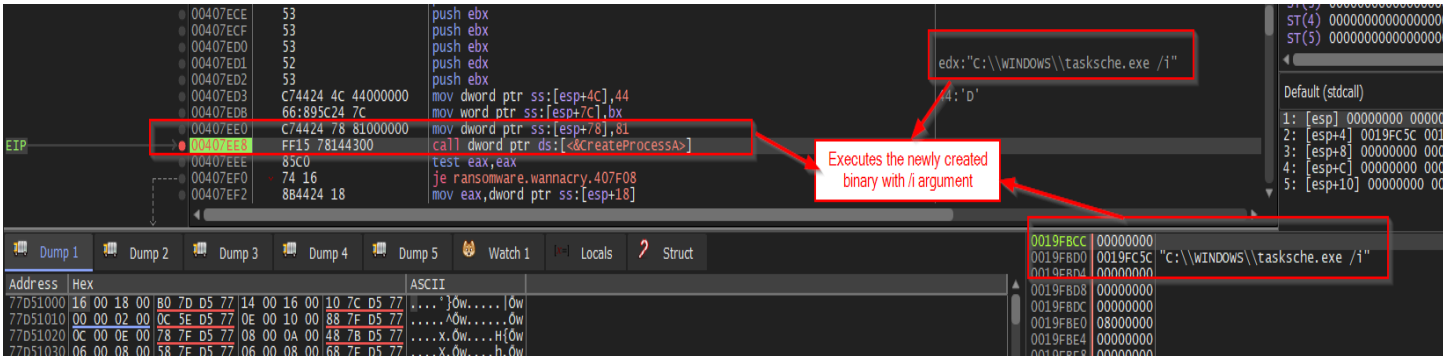
Checks if tasksche.exe exists, if yes then renames the file into a weird random name: qeruwjhrf.



Else, creates tasksche.exe in windows directory and writes the 1831 resource.



Executes the newly created file tasksche.exe with /i argument



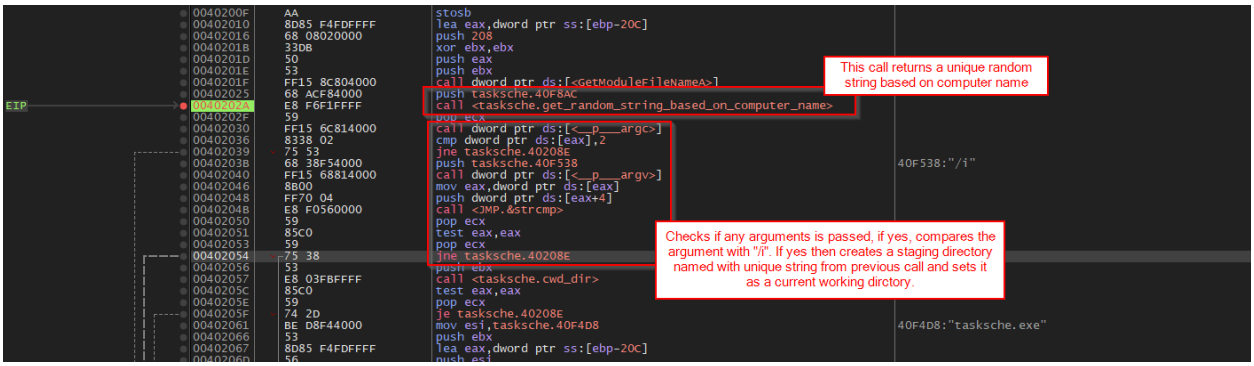
Once this is executed, all important files in our computer gets encrypted and a decryptor program is launched stating the instruction on how to recover the files.

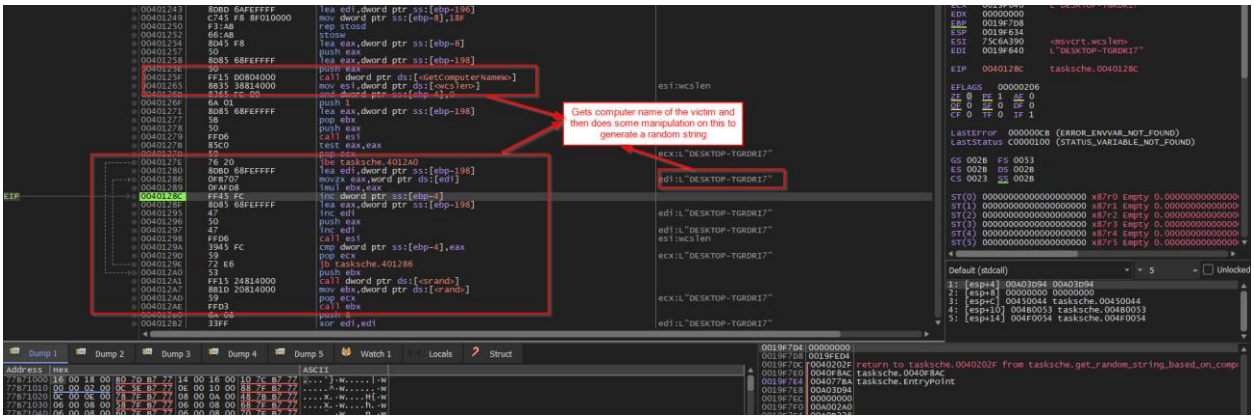
We can place a breakpoint at the create process call, and stop the execution of the binary at this point and then look at the second stage – tasksche.exe and dissect it further.

Debugging tasksche.exe – next stage payload

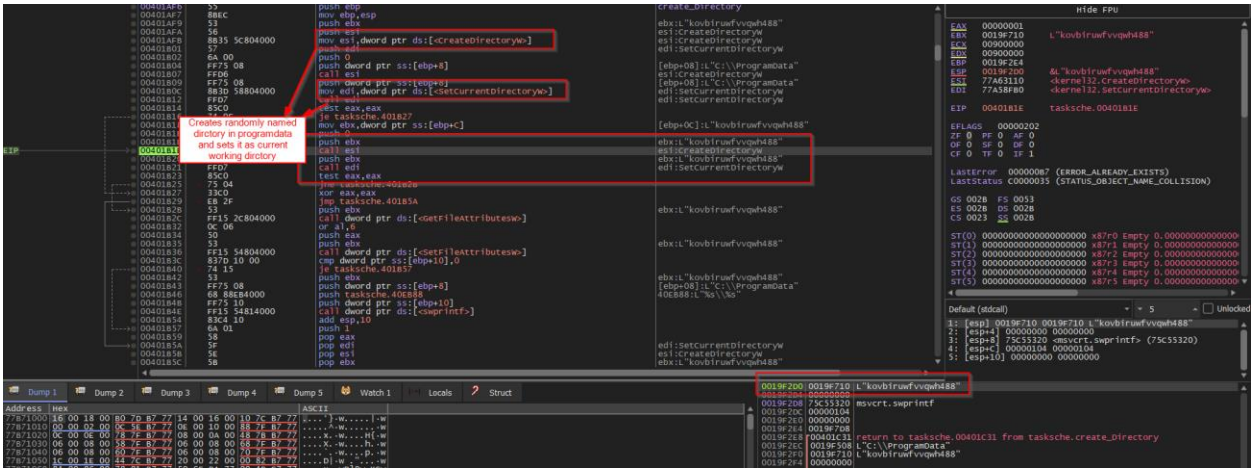
Stage Folder Creation

Initially calls a function which uses the computer name to create a unique randomized string. This string is later used to name the staging folder and service created by the program.



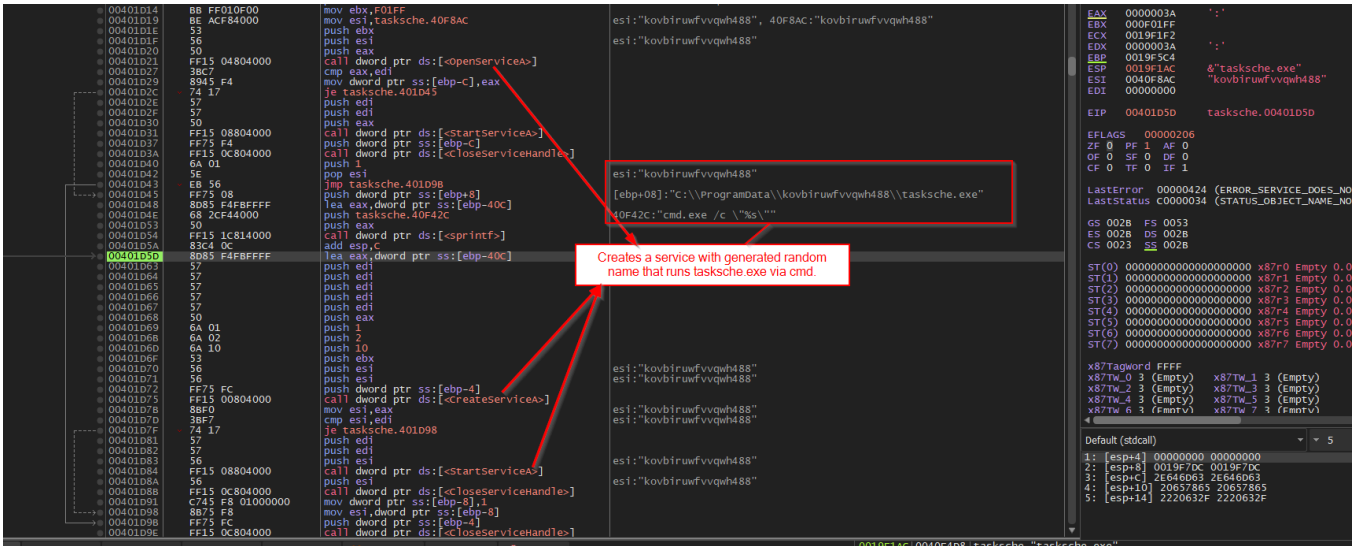


Creation of staged folder using the randomly generated string in previous function:

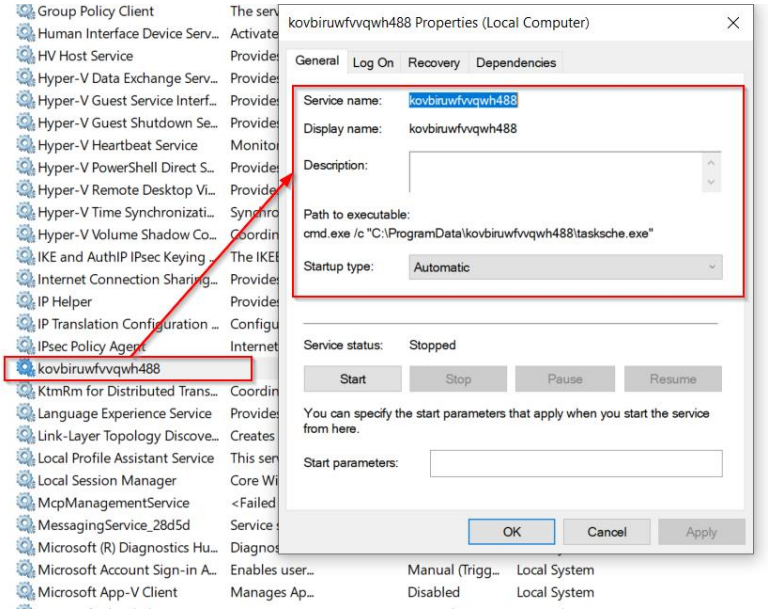


Persistence – Creation of service

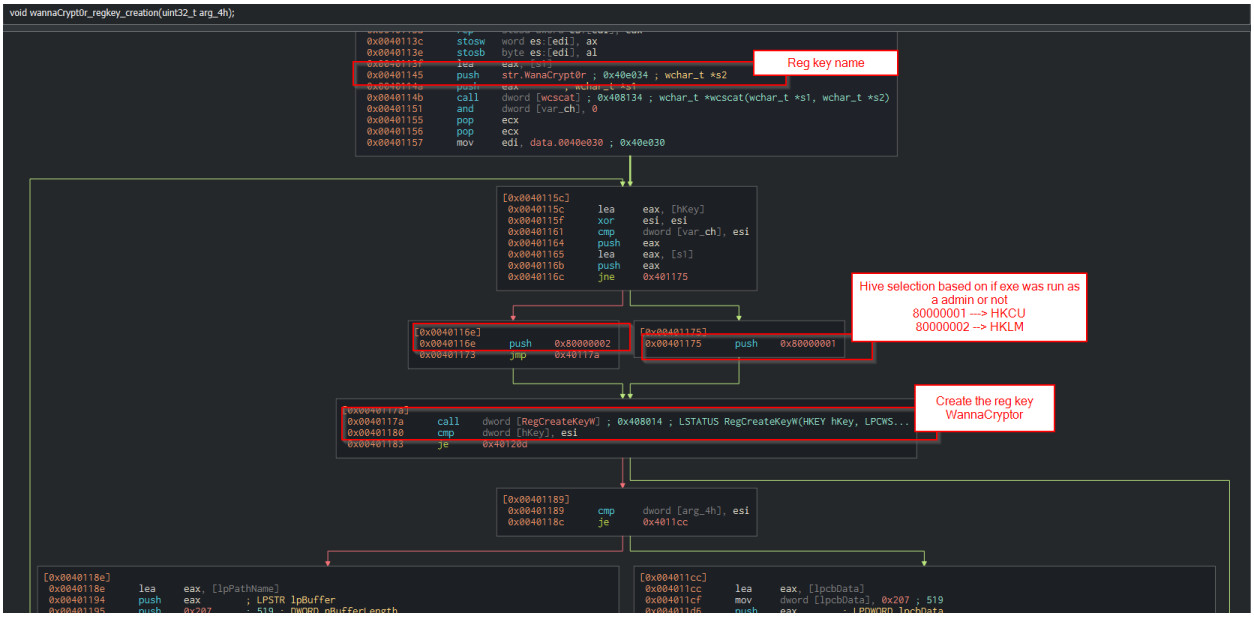
Creation of tasksche.exe service for persistence



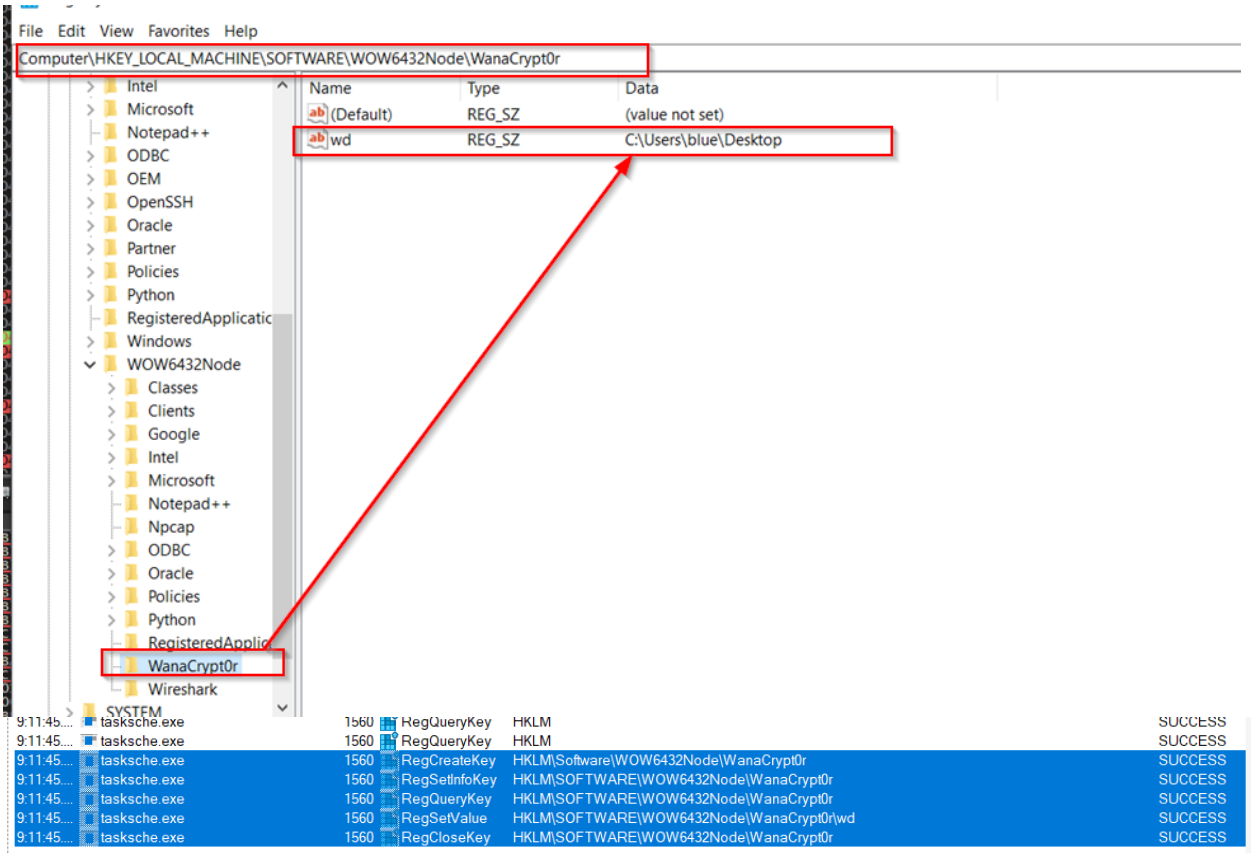
Evidence of service being created



Creates a reg key for wanacrypt0r in (HKLM|HKCU)/SOFTWARE/ hive



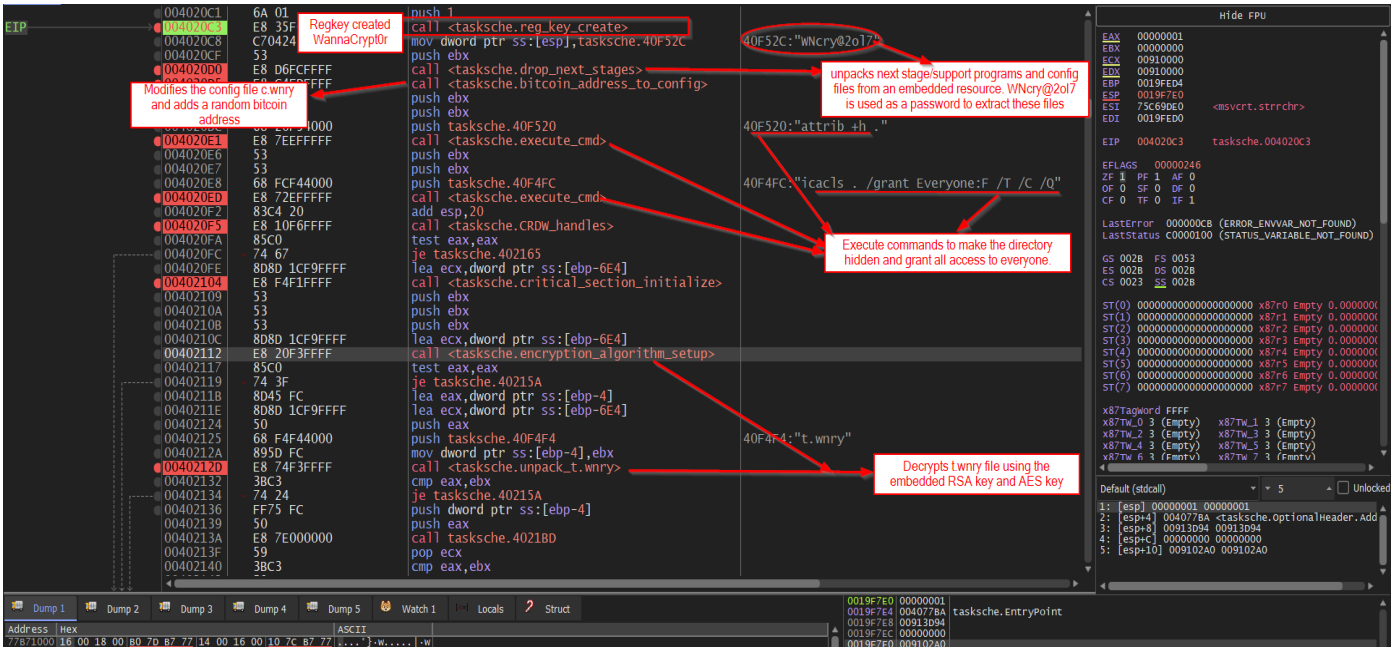
And the registry key value is set to current working directory



High level overview of the main functions of tasksche.exe

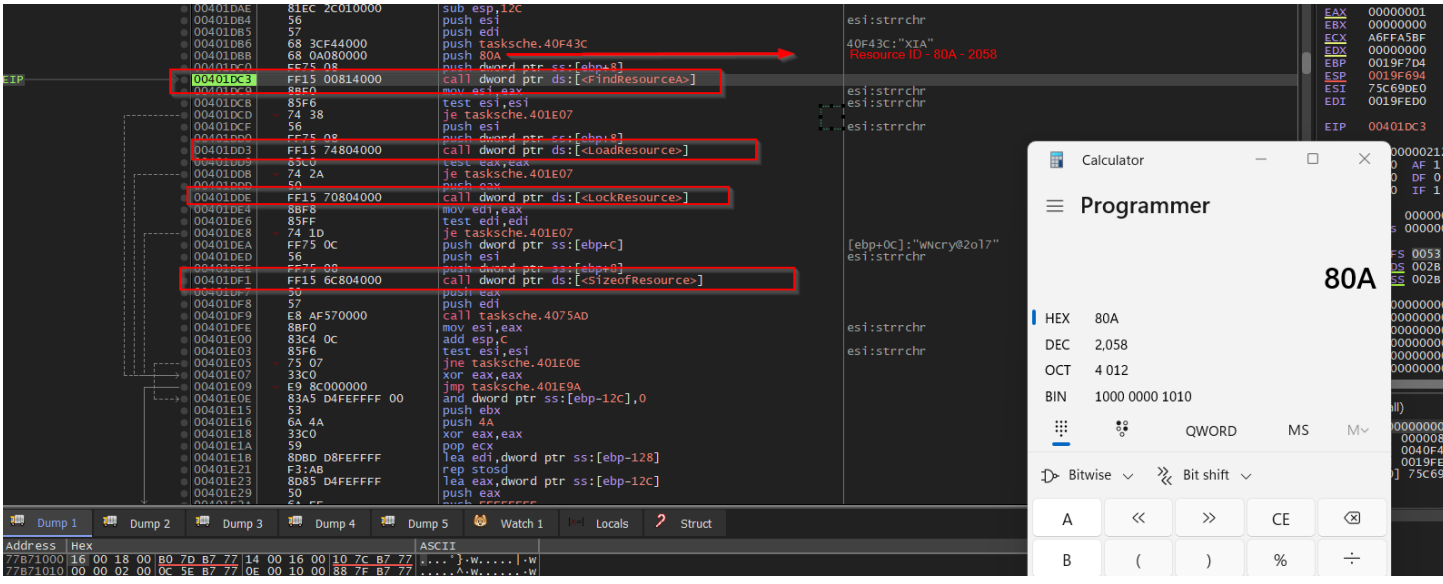
After creating/setting the registry key, it unpacks an embedded resource and creates new files. Once the files have been extracted, it modifies the config file (c.wnry) and adds a random bitcoin address. Executes commands to make the directory hidden and then grants all access to everyone to all the subdirectories.

In the end, decrypts the t.wnry sample using an embedded rsa and aes key and then finally executes this program.

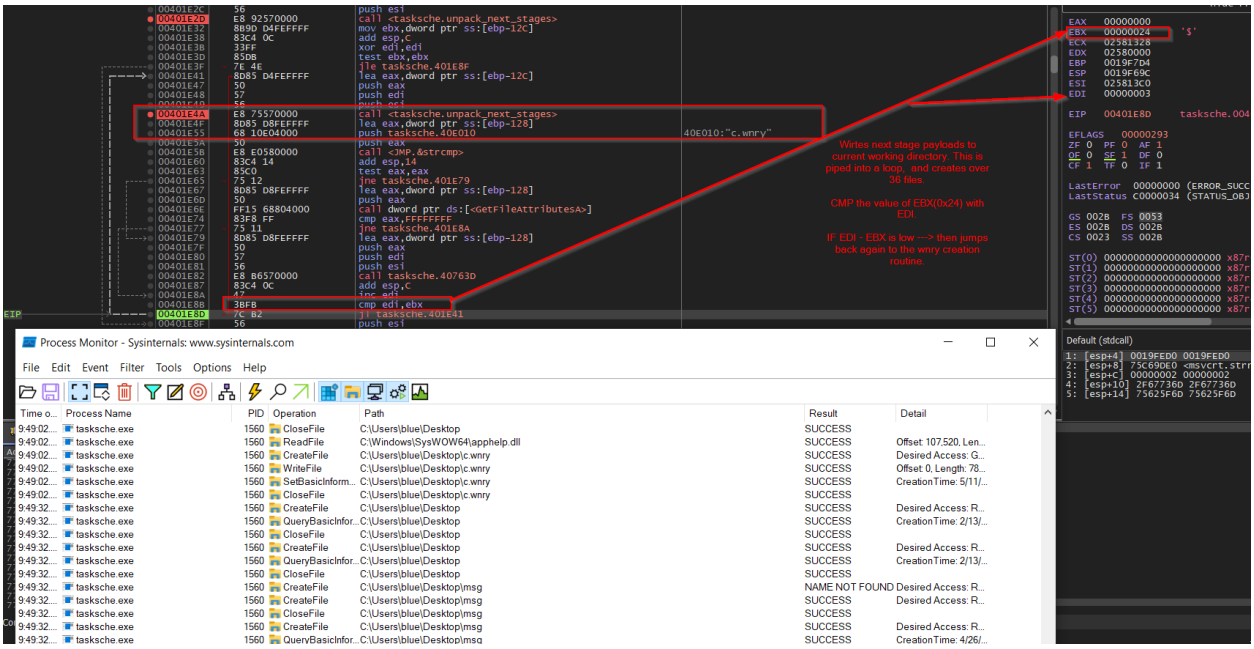


Detailed view of extracted files from 2058 resource in tasksche.exe

Initially loads the 2058 resource :

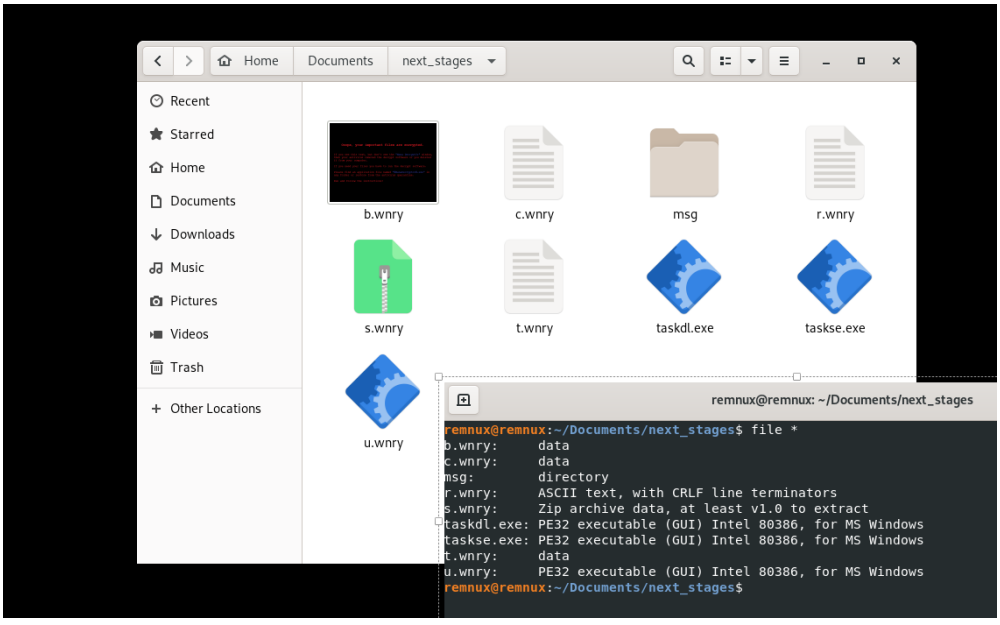


And then extracts all the files from this loaded resource. Over 36 files are created in this process.

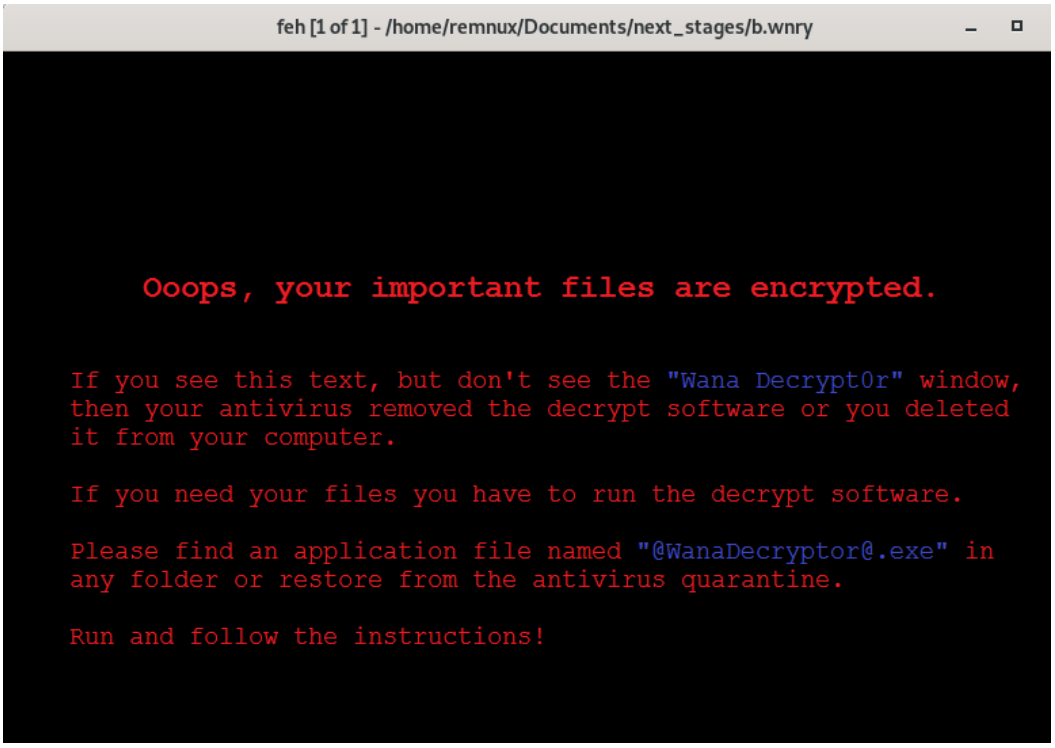


List of extracted files

Below is the list of files and directories that gets created. These files have been already detailed in the malware composition section.

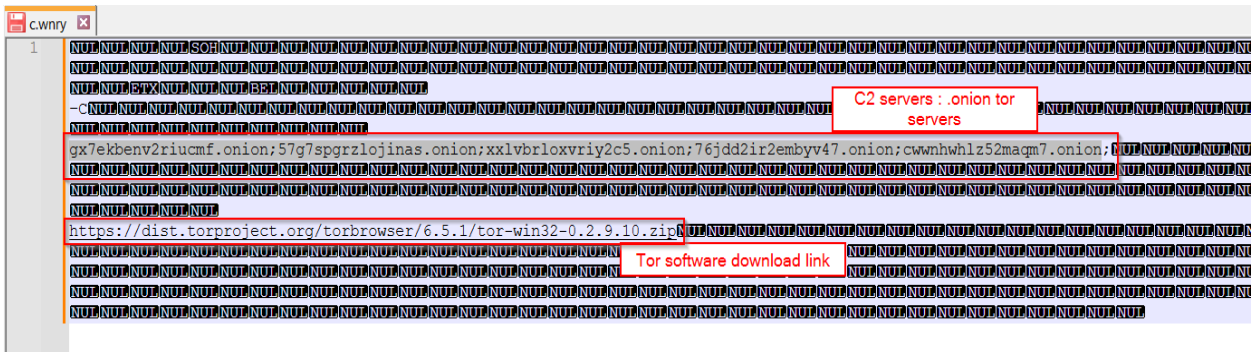


1. b.wnry is a bitmap image that appears as a desktop wallpaper after infection :



2. c.wnry → is a config file containing the tor server(.onion domains) which are likely the command and control servers used by the threat actor.
IOC domains :

gx7ekbenv2riucmf[.]onion
57g7spgrzlojinas[.]onion
xxlvbrloxvriy2c5[.]onion
76jdd2ir2embyv47[.]onion
cwwnhwhlz52maqm7[.]onion



3. msg -> directory containing ransom note in different languages

```
remnux@remnux:~/Documents/next_stages$ ls msg
 m_bulgarian.wnry                m_english.wnry      m_italian.wnry      m_romanian.wnry
'm_chinese (simplified).wnry'    m_filipino.wnry     m_japanese.wnry     m_russian.wnry
'm_chinese (traditional).wnry'   m_finnish.wnry      m_korean.wnry       m_slovak.wnry
 m_croatian.wnry                 m_french.wnry       m_latvian.wnry      m_spanish.wnry
 m_czech.wnry                    m_german.wnry       m_norwegian.wnry    m_swedish.wnry
 m_danish.wnry                   m_greek.wnry        m_polish.wnry       m_turkish.wnry
 m_dutch.wnry                    m_indonesian.wnry   m_portuguese.wnry   m_vietnamese.wnry
remnux@remnux:~/Documents/next_stages$
```

4. r.wnry => text file containing the instructions on how to recover the files

```
remnux@remnux:~/Documents/next_stages$ cat r.wnry
Q:  What's wrong with my files?

A:  Ooops, your important files are encrypted. It means you will not be able to access them anymore until they are decrypted.
    If you follow our instructions, we guarantee that you can decrypt all your files quickly and safely!
    Let's start decrypting!

Q:  What do I do?

A:  First, you need to pay service fees for the decryption.
    Please send %s to this bitcoin address: %s

    Next, please find an application file named "%s". It is the decrypt software.
    Run and follow the instructions! (You may need to disable your antivirus for a while.)

Q:  How can I trust?

A:  Don't worry about decryption.
    We will decrypt your files surely because nobody will trust us if we cheat users.


*   If you need our assistance, send a message by clicking <Contact Us> on the decryptor window.
remnux@remnux:~/Documents/next_stages$
```

5. s.wnry -> zip file containing the tor software and its dependency files

```
remnux@remnux:~/Documents/next_stages$ file s.wnry
s.wnry: Zip archive data, at least v1.0 to extract
remnux@remnux:~/Documents/next_stages$
remnux@remnux:~/Documents/next_stages$
remnux@remnux:~/Documents/next_stages$ unzip s.wnry
Archive:  s.wnry
replace Tor/libeay32.dll? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
  inflating: Tor/libeay32.dll
  inflating: Tor/libevent-2-0-5.dll
  inflating: Tor/libevent_core-2-0-5.dll
  inflating: Tor/libevent_extra-2-0-5.dll
  inflating: Tor/libgcc_s_sjlj-1.dll
  inflating: Tor/libssp-0.dll
  inflating: Tor/ssleay32.dll
  inflating: Tor/tor.exe
  inflating: Tor/zlib1.dll
remnux@remnux:~/Documents/next_stages$
```

6. taskdl.exe => support program to delete files from system

```
remnux@remnux:~/Documents/next_stages$ file taskdl.exe
taskdl.exe: PE32 executable (GUI) Intel 80386, for MS Windows
remnux@remnux:~/Documents/next_stages$
```

7. t.wnry → encrypted dll file which consists the encryption routine for the malware

```
remnux@remnux:~/Documents/next_stages$ file t.wnry
t.wnry: data
remnux@remnux:~/Documents/next_stages$
remnux@remnux:~/Documents/next_stages$ xxd t.wnry | head -n 5
00000000: 5741 4e41 4352 5921 0001 0000 1e38 2227  WANACRY!.....8"'
00000010: fde6 7f0c 5de7 7e3e 28a7 affd 2a50 6449  ....].~>(...*PdI
00000020: 66c6 b627 176d 3ed2 ff1c 32cb 8c30 8860  f..'.m>...2..0.`
00000030: 70f6 eae9 9981 5e15 fe03 2349 7cbb ce3c  p.....^...#I|..<
00000040: ee57 e042 dc3d afa8 82b8 4d01 057a 7846  .W.B.=....M..zxF
remnux@remnux:~/Documents/next_stages$
```

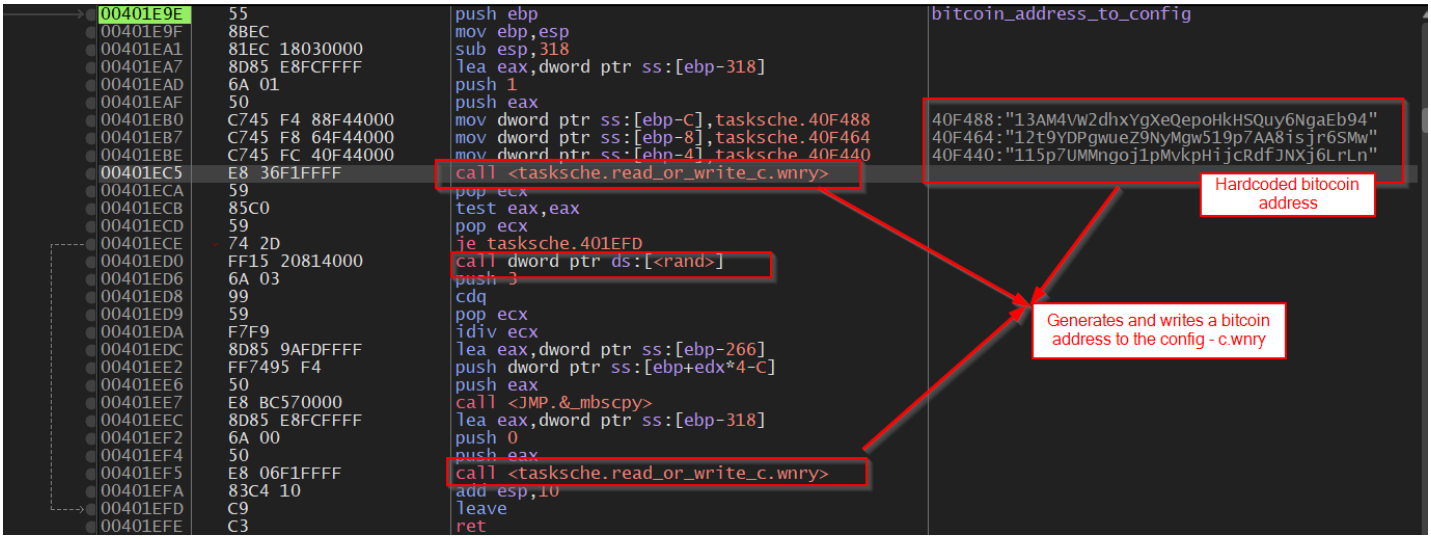
8. taskse.exe → PE file – unknown functionality

```
remnux@remnux:~/Documents/next_stages$ file taskse.exe
taskse.exe: PE32 executable (GUI) Intel 80386, for MS Windows
remnux@remnux:~/Documents/next_stages$
```

9. u.wnry → PE file --- unknown functionality

```
remnux@remnux:~/Documents/next_stages$ file u.wnry
u.wnry: PE32 executable (GUI) Intel 80386, for MS Windows
remnux@remnux:~/Documents/next_stages$
```

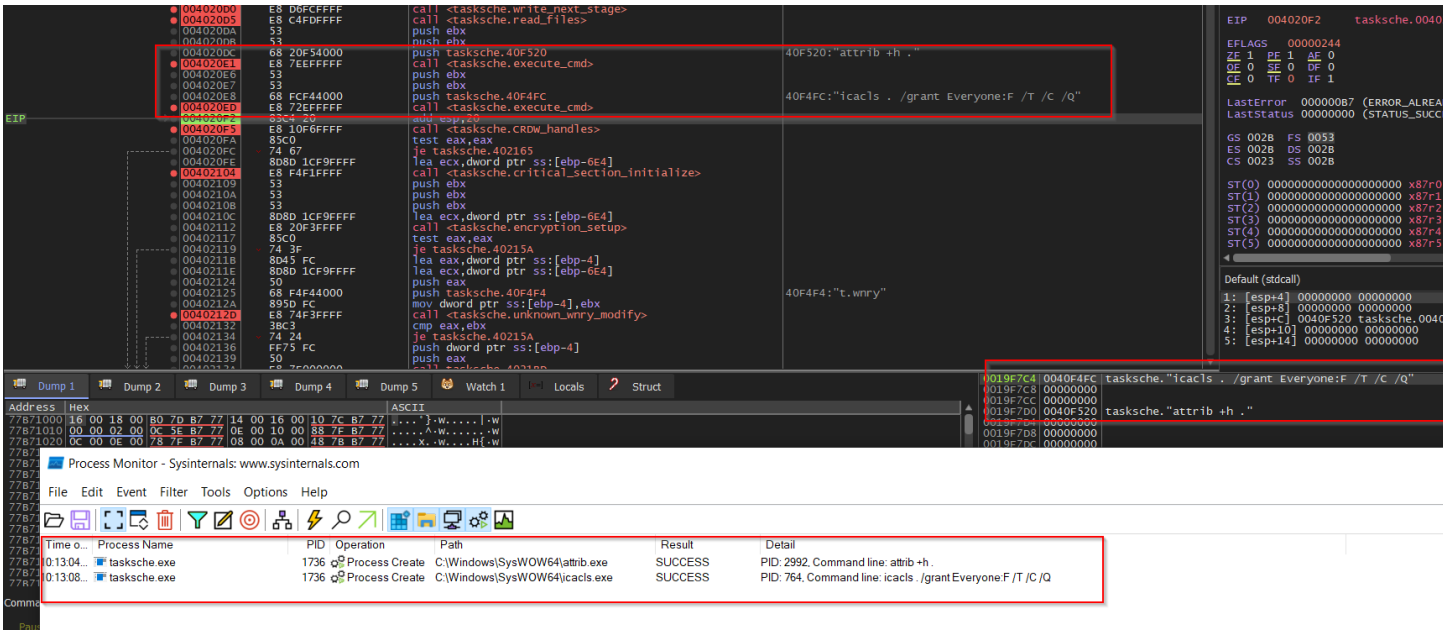
Next modifies c.wnry file and adds a random bitcoin address



After modification, we can see that c.wnry has a bitcoin address appended at the beginning

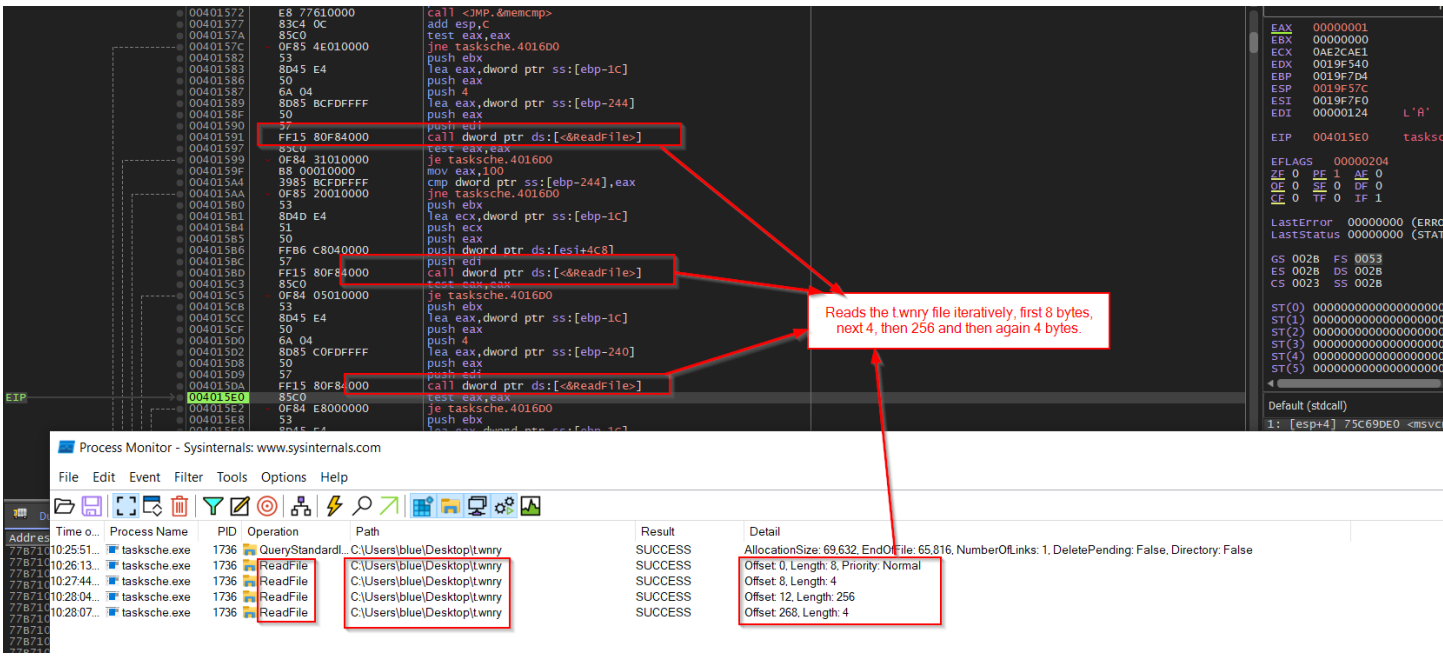
Command Execution

The program executes commands like “attrib +h” & “icalcs . /grant Everyone:F /T /C /Q” to make the directory hidden and grant all access respectively.

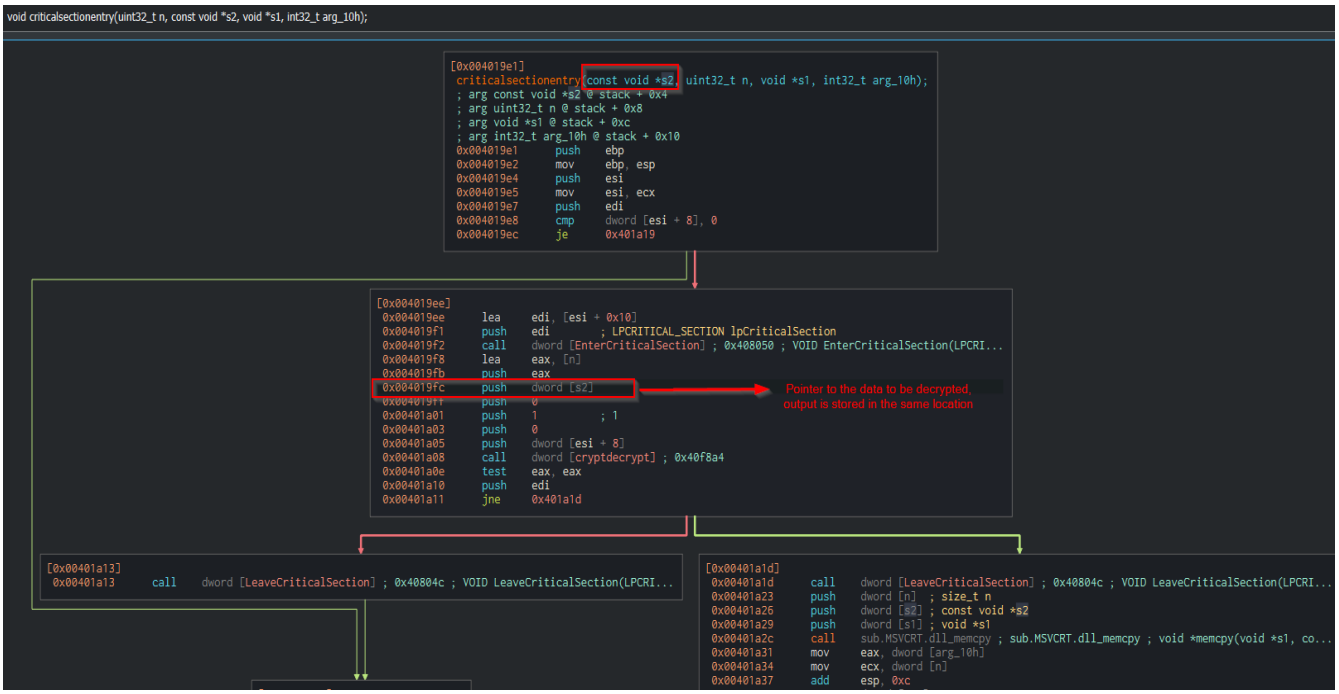


Unpack the t.wnry file

Initially reads the t.wnry sample, iteratively reading initial 8 bytes, next 4 and then 256 and then again 4 bytes.



Then makes a call to a function that decrypts the data that was read from t.wnry using an embedded RSA key in the program.



Decryption routine/ unpacking call

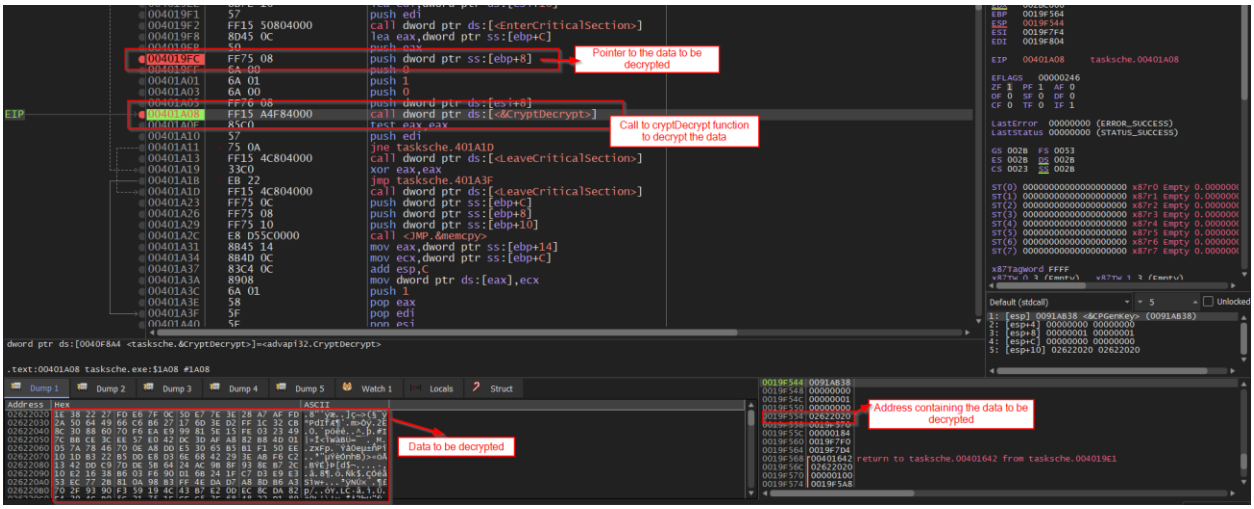


Figure 16 Before Decryption call

Data after decryption

Doing some OSINT on the t.wnry file, I found that the first 16 bytes this decrypted data contains an AES key which can be used to decrypt the rest of the encrypted data in t.wnry sample.

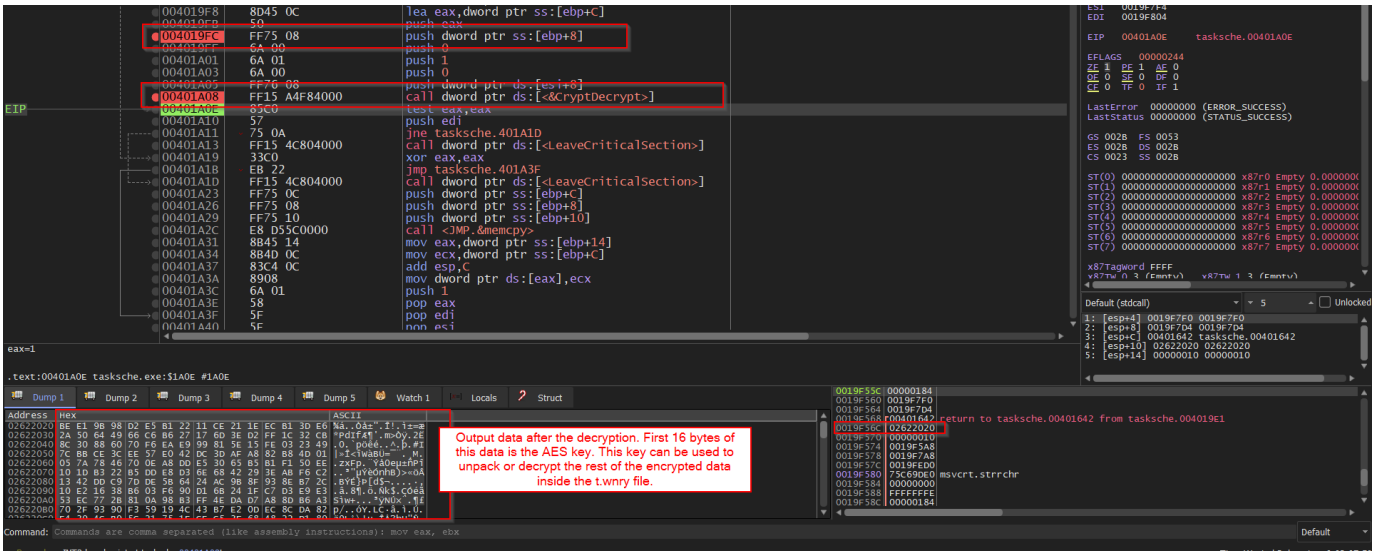
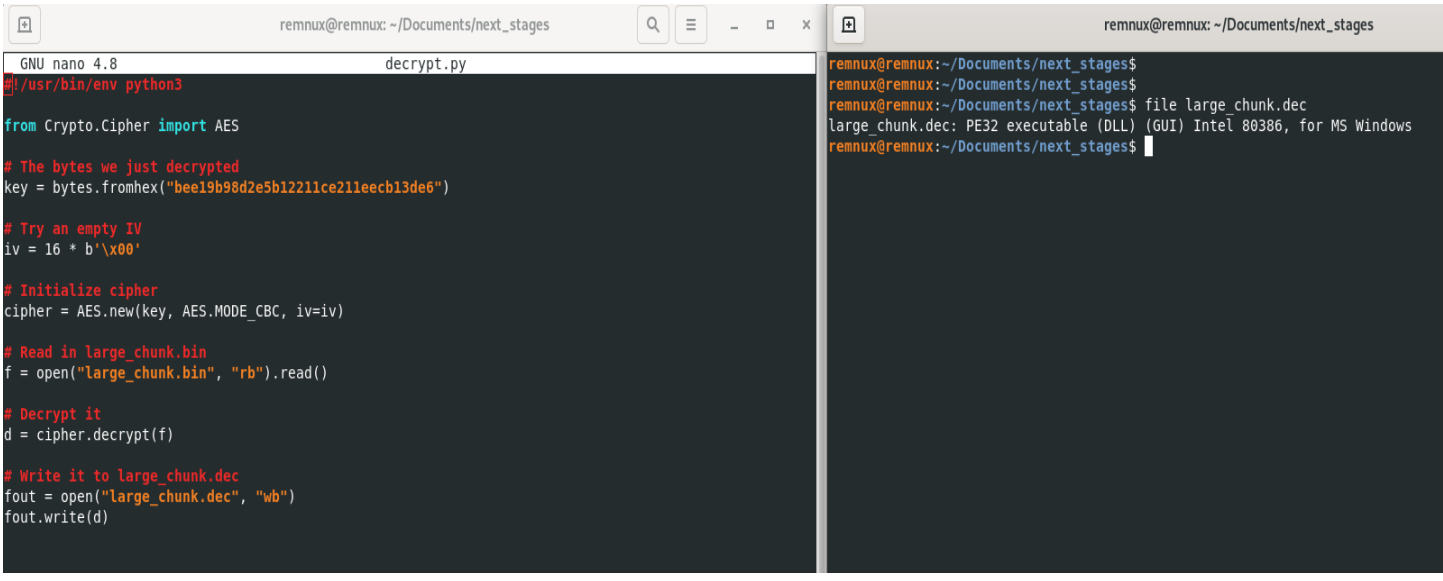


Figure 17 After the decryption call

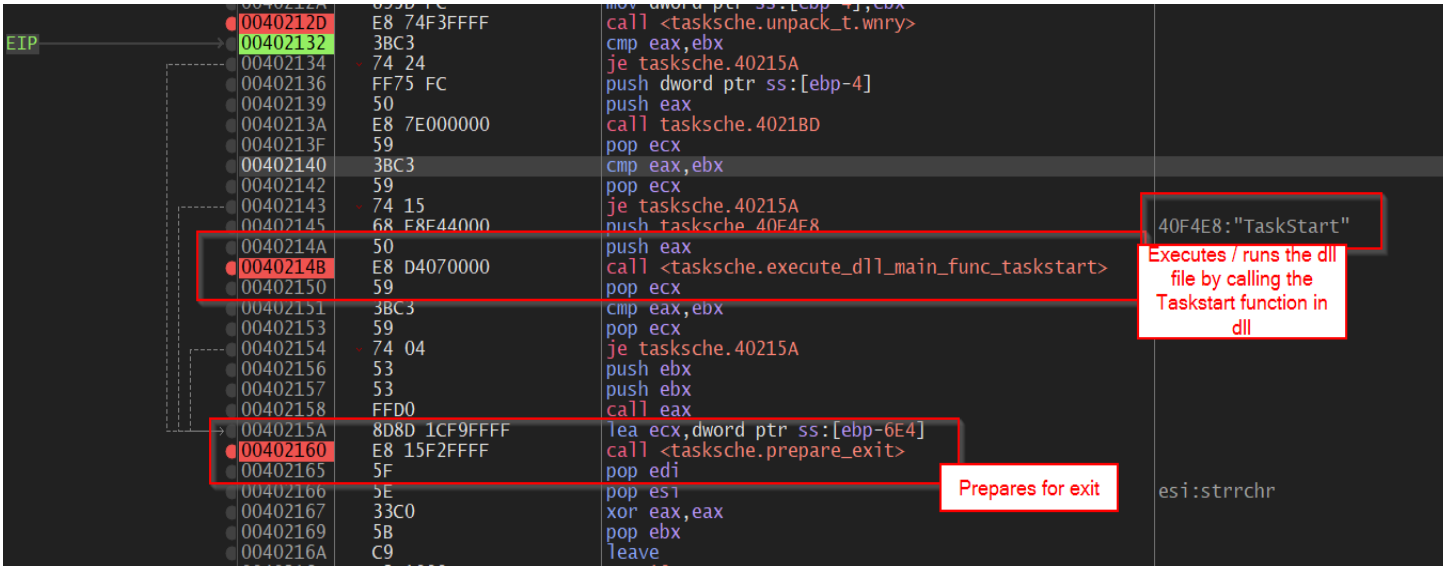
A simple python script can be used to decrypt or unpack this sample.

Used this script from ghidraninja to decrypt the rest of the data in t.wnry file
https://github.com/ghidraninja/ReversingWannacry/blob/master/decrypt_large_chunk.py

And found this to be a windows dll file.



After this, it executes this decrypted dll file by calling the taskstart function and then prepares the program for exit by clearing all the pointers and memory.



I am going to stop the advanced analysis at this point. The execution of dll leads to the encryption of all our files on computer and the popup for decryptor appears on the desktop.

It looks like the dll likely has the encryption routine functionality.

Indicators of Compromise

The full list of IOCs can be found in the Appendices.

Network Indicators

Initial killswitch domain:

www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]com

Command and control servers from the config file (c.wnry):

gx7ekbenv2riucmf[.]onion
57g7spgrzlojinas[.]onion
xxlvbrloxvriy2c5[.]onion
76jdd2ir2embyv47[.]onion
cwwnhwhlz52maqm7[.]onion

Host-based Indicators

See the malware composition & list of extracted files section for host-based indicators

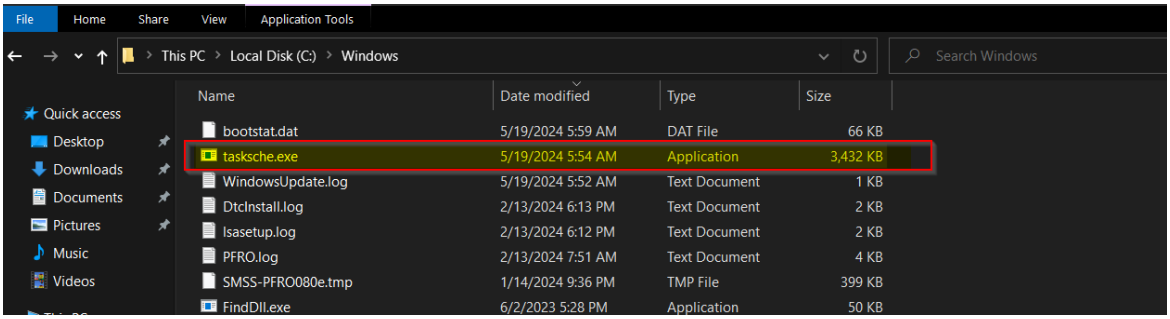


Figure 18 Main payload of wannacry dropped by initial detonation

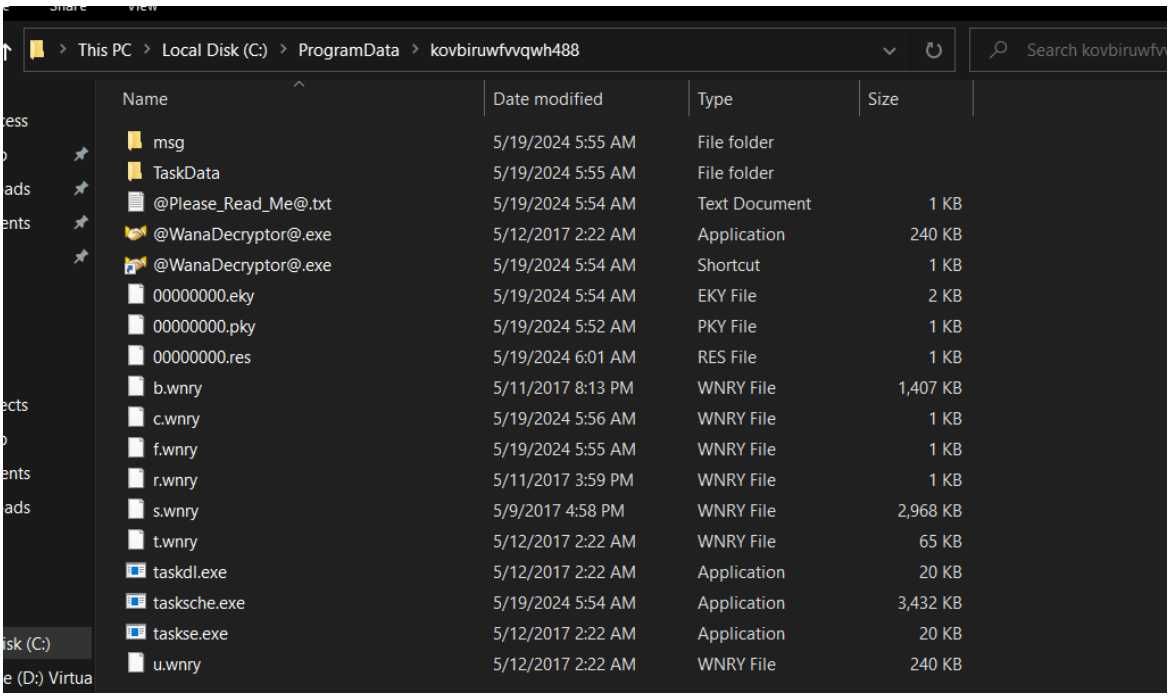


Figure 19 Staging folder and support programs for wannacry

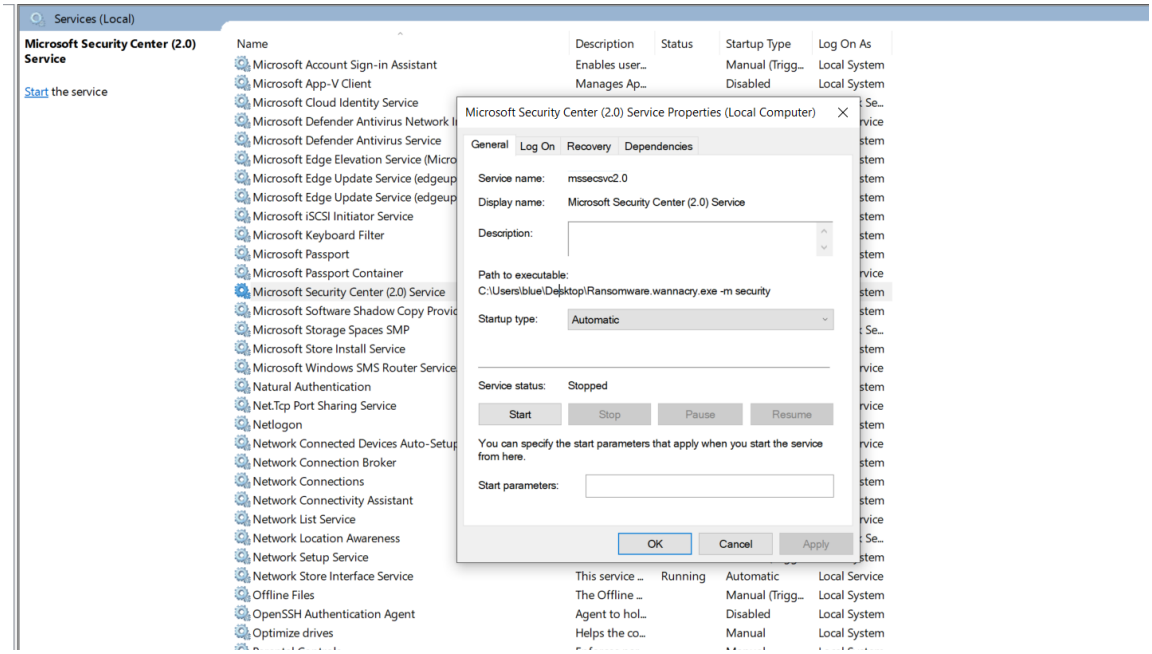


Figure 20 mscsvc service that launches wannacry payload to spread through smb

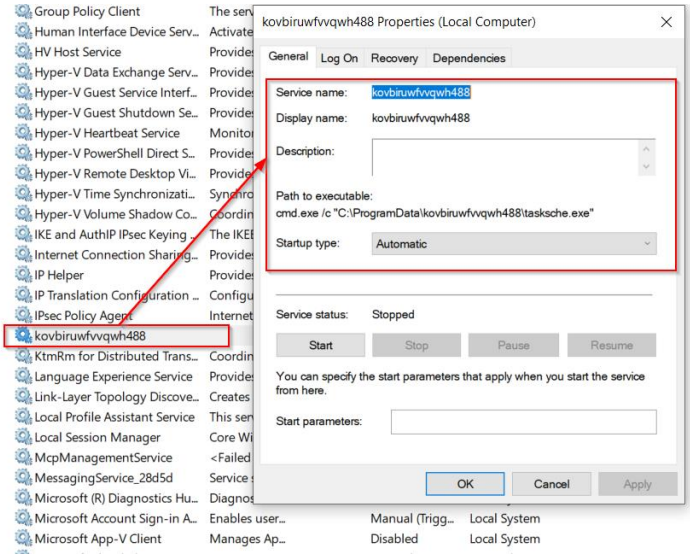


Figure 21 Tasksche service

Rules & Signatures

A. Yara Rules

```
rule ransomware_wannacry {
    meta:
        author = "Aniksha Shetty"
        description = "Detect wanancryransomware file"
        created = "04-15-2024"


    strings:
        $weird_file_name = "C:\\%s\\qeriuwjhrf" ascii
        $kill_switch_url = "iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com" ascii
        $bitcoin_addr1 = "115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn" ascii
        $bitcoin_addr2 = "12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw" ascii
        $bitcoin_addr3 = "13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94" ascii
        $passkey = "WNcry@2ol7" ascii
        $ipc_share = "\\%s\\IPC$" ascii
        $file_extension = ".wnry"
        $pe_magic_byte = "MZ"

    condition:
        $pe_magic_byte at 0 and
        3 of them
}
```

B. Callback URLs

Domain	Type
www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]com	Kill Switch domain
76jdd2ir2embyv47[.]onion	C2
cwwnhwhlz52maqm7[.]onion	C2
xxlvbrloxvriy2c5[.]onion	C2
gx7ekbenv2riucmf[.]onion	C2
57g7spgrzlojinas[.]onion	C2