MALWARE ANALYSIS

Michael Awoyemi

CMP320: ADVANCED ETHICAL HACKING

2023/24

Note that the Information contained in this document is for educational purposes.

.

Abstract

The sole objective of this report was to conduct an exhaustive and comprehensive analysis of a corrupted file. The chosen methodology for the analysis was inspired by the Systematic Approach to Malware Analysis (SAMA). SAMA is a well-known technique for malicious software behavioural assessments. The aim was to gather a deeper insight about the intended functionality of the file and how it was programmed by its developers to act.

The analysis was carried out in 2 stages. Firstly, examining the code and structure of the corrupted file using static technique tools and a disassembler. This allowed for a detailed understanding of the file's inner workings, without executing it. The second part was dynamic analysis, during the analysis, the malware was executed within a secure environment to visualise how it behaved. Thanks to specialised tools, the interactions with the virtual machine were visualised in real-time. The combinations of both types of analysis provided a deeper understanding of the corrupted file.

By conducting thorough examinations and continuous monitorization of the file behaviour, it became abundantly clear that the file in question was, in fact, a highly malicious software. The type of malware in question was the popular and destructive ransomware WannaCry. WannaCry exploited a Windows vulnerability called Eternal Blue. What made this ransomware particularly dangerous was its ability to hide behind the corporation name Microsoft. Once the malware was executed it immediately began creating threads to conduct its malicious activities. It created, encrypted, and deleted files, among other malicious functions, were discovered.

Finally, the report emphasises what companies and individuals can do to stay protected from malicious software in general. Concluding with the suggestions of policies and action plans like incident response, business continuity, and disaster recovery.

.

Contents

.

1 INTRODUCTION

1.1 BACKGROUND

Ransomware is a highly destructive type of malware (malicious software) with the purpose of extorting its victims by demanding payments to undo the changes and restore the damage caused by the malware. The impact of ransomware can be severe and long-lasting, ranging from the deletion of important information to the complete encryption and erasure of data, or even the total blockage of access to a victim's computer. Usually, once the malware is executed, a message is displayed where the perpetrators will leave information about payment and how to regain access or recover data.

Over the past few years, there has been a significant rise in the number of ransomware attacks and viruses. Shockingly, the number of victims of ransomware has increased by 128.18% between 2022 and 2023. These types of attacks can have a devastating impact on everyone, from large corporations to individual computers.

Malware analysis can be described as the art of meticulously studying the composition of the software in order to fully understand its intention. There are two types of analysis: static and dynamic. Static analysis consists of specific tools to trace the instructions of the malware code to see how it acts. Dynamic analysis, on the other hand, is the practice of running the malware and seeing it in real-time to confirm the details that were found during the static analysis. The dynamic part must always be run from a controlled environment, either a virtual machine or a sandbox, to not damage the original machine. This way, we can effectively analyse the damage caused by the malware on the VM and gain valuable insights into how to prevent future attacks.

1.2 AIM

The main objective of this project is to conduct an in-depth analysis of a corrupt file using both static and dynamic techniques. The purpose of this analysis is to gather information about the intended functionality of the file, as well as how it has been programmed to execute its intended tasks. This comprehensive analysis will help in identifying any potential security risks or vulnerabilities associated with the corrupt file.

1.3 METHODOLOGY

The methodology used to perform a malware analysis test on the suspicious file was a combination of static, dynamic, and disassembly analysis. The tools used for the malware functionality through source code inspection were, VirusTotal, String, Floss, DIE, Dependency Walker, Pestudio, and IDA Pro, among others. The dynamic analysis was performed in a virtual environment on a Windows 10 FlareVM machine. From the virtual machine, the corrupted file was executed and tools such as Procmon, Regshot, Wireshark, FakeNet, Procexp, and among others were used.

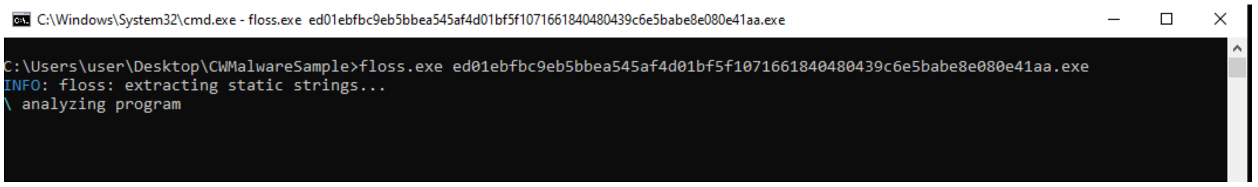
2 PROCEDURE

2.1 STATIC ANALYSIS

VirusTotal is a very powerful tool for discovering information about malware. If a file is suspected of being malicious, the user can search for the hashes of the suspected file in the extensive database of VirusTotal. Once scanned, Virus Total will compare the file with the other hashes in its database. If the search is successful, it will provide a report on the type of malware, its function, its behaviour, and how dangerous or harmful it is, among other information available on the internet about the file in question. If the search was not successful, it does not automatically mean that the file is not dangerous. It means that it is not yet registered in the database and users must proceed with caution.

In this case, the search was successful, and the file type was WannaCry. WannaCry is a type of ransomware that was released in 2017 and has caused major losses to large companies and individuals worldwide. In Figure 3, under the static analysis discussion, it can be seen Virustotal's report in which it found the hash of the file that was introduced in 66 out of the 71 sites it compared the hash with. The tool also provided information such as the security vendor that has the file flagged, details about the file in general, relationship, and behaviour among others.

The next tools used to parse the file are strings.exe and floss.exe. Both tools are run from the CLI (command line interface). The function of these tools is to search for "strings" in the file that can later benefit the user in understanding the malware's operation.



```
C:\Windows\System32\cmd.exe - floss.exe  ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe

C:\Users\user\Desktop\CWMalwareSample>floss.exe ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe
INFO: floss: extracting static strings...
\ analyzing program
```

Figure 1: Running Floss.exe

PEid is another statics analysing tool that determines the compilation of the file and finds out if the file is packed. When an executable file is packed, the executable code is compressed. This means that the code can be modified without changing the underlying function of the file. The analyst decided to use this tool first because if the file was packed some tools would not be able to read the headings.

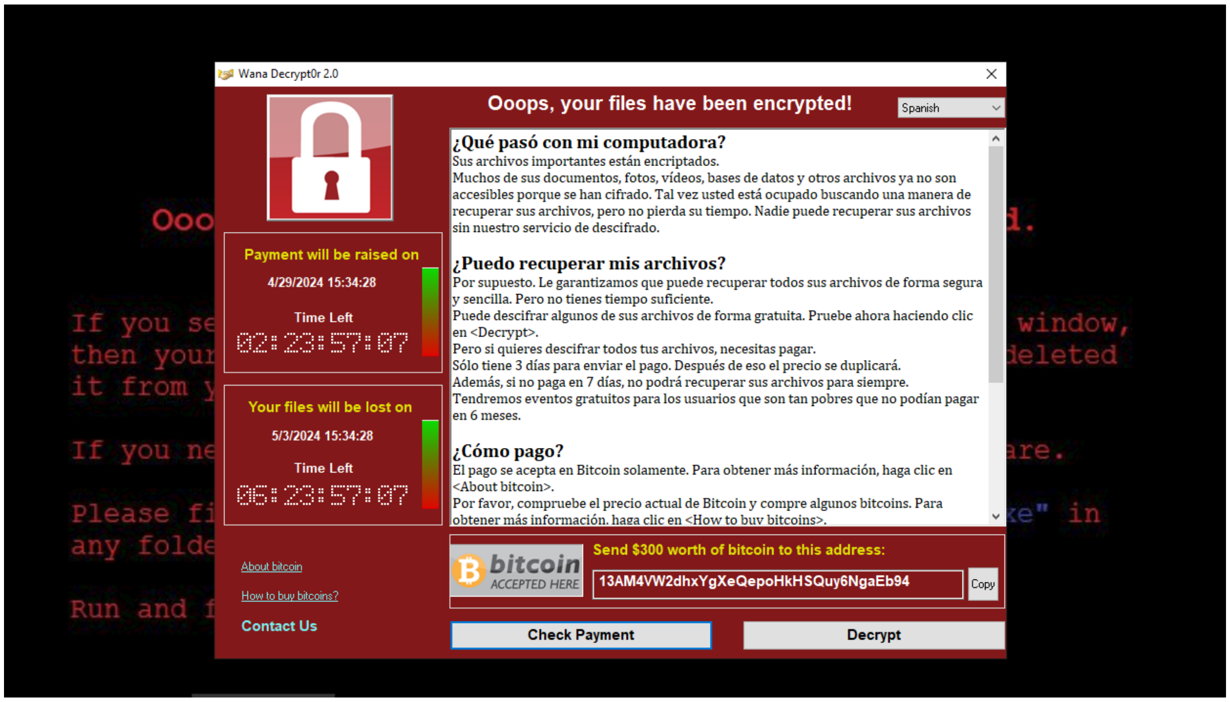
DIE (Detect it easy) was also used to scan the file. The Entropy sub-tool discovered more information about the taste of the file. It was useful when looking at PE headers within an executable. PE headers are usually broken up into .text, .data, .rdata, .idata, .edata, .pdata, .rsrc, and .reloc. This will be reviewed and expanded further when discussing the findings. Evidence can be found in Figure 4.

Finally, the following tools were used to conclude the static analysis of the file; IDA Pro, Pestudio, and Dependency Walker. Dependency walker was used to discover the DLLs in the file, and what the DLLs are executing. The findings of the DLLs and their instructions can be found in Appendix B.

2.2 Dynamic Analysis

As mentioned in the introduction, in this phase of the analysis, the analyst examined the behaviour of the code using a set of tools and also by running the file. This was done using a Windows 10 virtual machine, specifically from FlareVM. The analyst removed the network interfaces from the VM to prevent it from spreading, adding another layer of protection to his personal device.

The analyst's approach was to run the malware in the VM, see what happened, and analyse how it affected the network and registry. Upon executing the file, the Windows User Account Control prompted the analyst to allow changes to the device, as seen in Figure 24.



¿Qué pasó con mi computadora?
Sus archivos importantes están encriptados.
Muchos de sus documentos, fotos, vídeos, bases de datos y otros archivos ya no son accesibles porque se han cifrado. Tal vez usted está ocupado buscando una manera de recuperar sus archivos, pero no pierda su tiempo. Nadie puede recuperar sus archivos sin nuestro servicio de descifrado.

¿Puedo recuperar mis archivos?
Por supuesto. Le garantizamos que puede recuperar todos sus archivos de forma segura y sencilla. Pero no tienes tiempo suficiente.
Puede descifrar algunos de sus archivos de forma gratuita. Pruebe ahora haciendo clic en <Decrypt>.
Pero si quieres descifrar todos tus archivos, necesitas pagar.
Sólo tiene 3 días para enviar el pago. Después de eso el precio se duplicará.
Además, si no paga en 7 días, no podrá recuperar sus archivos para siempre.
Tendremos eventos gratuitos para los usuarios que son tan pobres que no podían pagar en 6 meses.

¿Cómo pago?
El pago se acepta en Bitcoin solamente. Para obtener más información, haga clic en <About bitcoin>.
Por favor, compruebe el precio actual de Bitcoin y compre algunos bitcoins. Para obtener más información, haga clic en <How to buy bitcoins>.

Send $300 worth of bitcoin to this address:
13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94

Figure 2: Malware executed.

Procmon (Process Explorer) and Procexp (Process Explorer) were both used to understand what happened within the system once the malware was executed. They both captured the process running and provided information such as process name, description, image path, lifetime, company, and command, among others.

Registry keys were also found using the tool Regshot. These keys store important information about the system configuration and application settings. Registry keys are organized hierarchically and contain values and subkeys. They are crucial for maintaining system stability and performance. Windows uses these keys to access and modify system settings as needed. The registry keys are essential for the proper functioning of the Windows operating system.

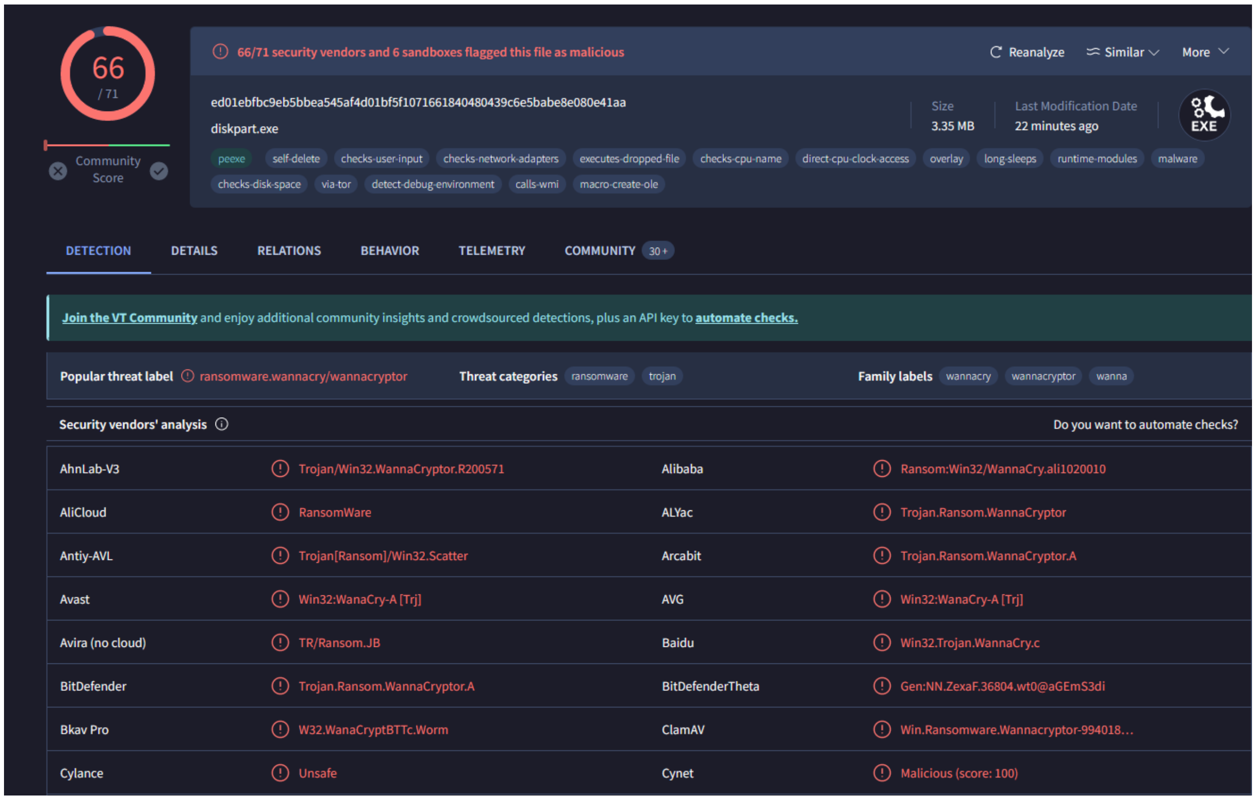
Additional analysis was done, this type in search of any possible communication between the malware and the internet. This was performed with the well-known tool Wireshark, and FakeNet.

3 Discussion

3.1 Static Analysis Discussion

VirusTotal was the first tool used to scan the file the analyst found suspicious. The result came back as WannaCry, an infamous virus that struck the world by surprise back in 2017. Infecting and causing major losses to the NHS in the United Kingdom, Telefonica in Spain, FedEx in the USA, and the German railway among others.

As it was an infamous ransomware attack, VirusTotal was able to detect the hash in its database, cross-referencing it with 71 security vendors, of which 66 had the file flagged as malicious as can be seen in the image below.



66/71 security vendors and 6 sandboxes flagged this file as malicious

ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa
diskpart.exe

Size 3.35 MB
Last Modification Date 22 minutes ago

peexe · self-delete · checks-user-input · checks-network-adapters · executes-dropped-file · checks-cpu-name · direct-cpu-clock-access · overlay · long-sleeps · runtime-modules · malware · checks-disk-space · via-tor · detect-debug-environment · calls-wmi · macro-create-ole

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: ransomware.wannacry/wannacryptor
Threat categories: ransomware, trojan
Family labels: wannacry, wannacryptor, wanna

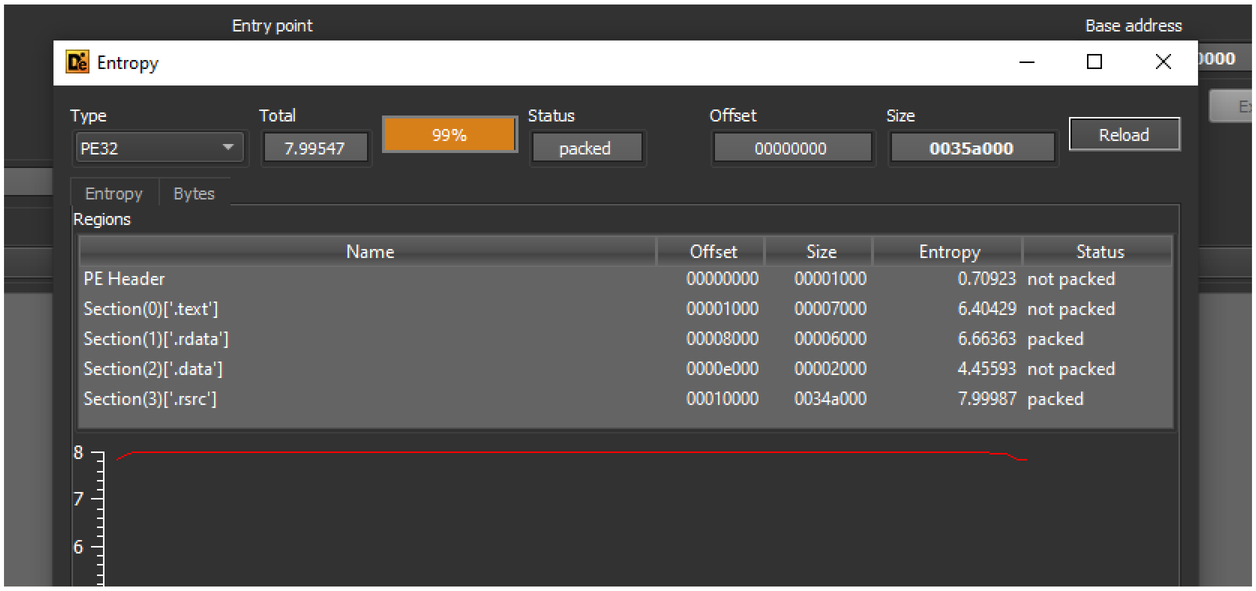
Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	Trojan/Win32.WannaCryptor.R200571	Alibaba	Ransom:Win32/WannaCry.ali1020010
AliCloud	RansomWare	ALYac	Trojan.Ransom.WannaCryptor
Antiy-AVL	Trojan[Ransom]/Win32.Scatter	Arcabit	Trojan.Ransom.WannaCryptor.A
Avast	Win32:WanaCry-A [Trj]	AVG	Win32:WanaCry-A [Trj]
Avira (no cloud)	TR/Ransom.JB	Baidu	Win32.Trojan.WannaCry.c
BitDefender	Trojan.Ransom.WannaCryptor.A	BitDefenderTheta	Gen:NN.ZexaF.36804.wt0@aGEmS3di
Bkav Pro	W32.WanaCryptBTTc.Worm	ClamAV	Win.Ransomware.Wannacryptor-994018...
Cylance	Unsafe	Cynet	Malicious (score: 100)

Figure 3: VirusTotal Findings.

After uploading the malware file through Detect it Easy (DIE), from the Entropy sections of the tool, it was discovered that some sections of the files were packed and compiled with Microsoft Visual C++ 6.0. As the PE header, .text, and .data were not packed, more information about them was found using other tools.



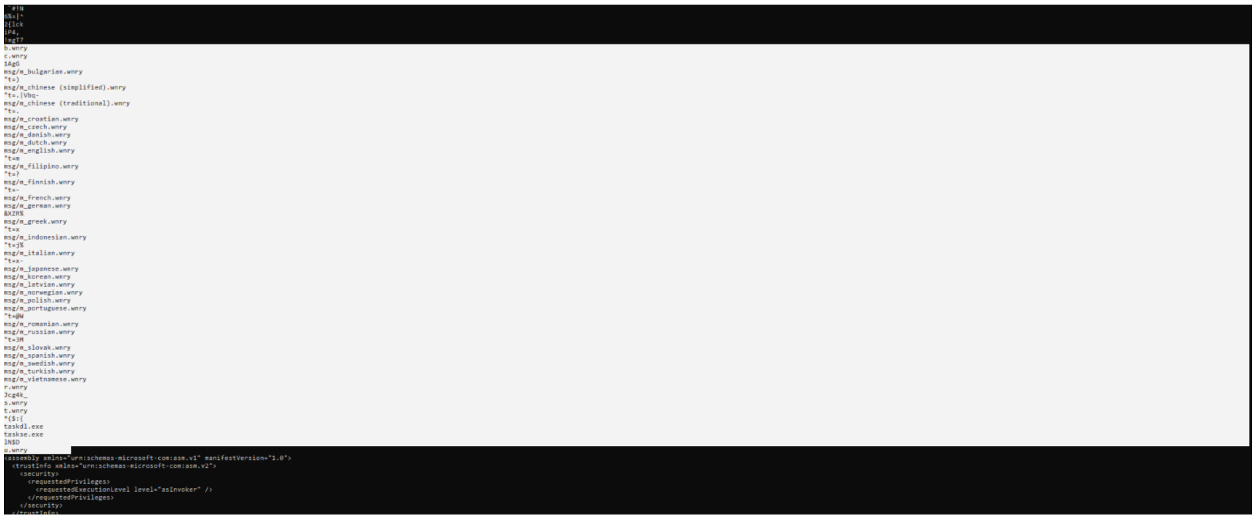
Type	Total		Status	Offset	Size
PE32	7.99547	99%	packed	00000000	0035a000

Regions

Name	Offset	Size	Entropy	Status
PE Header	00000000	00001000	0.70923	not packed
Section(0)['.text']	00001000	00007000	6.40429	not packed
Section(1)['.rdata']	00008000	00006000	6.66363	packed
Section(2)['.data']	0000e000	00002000	4.45593	not packed
Section(3)['.rsrc']	00010000	0034a000	7.99987	packed

Figure 4: Entropy (DIE) result of the file

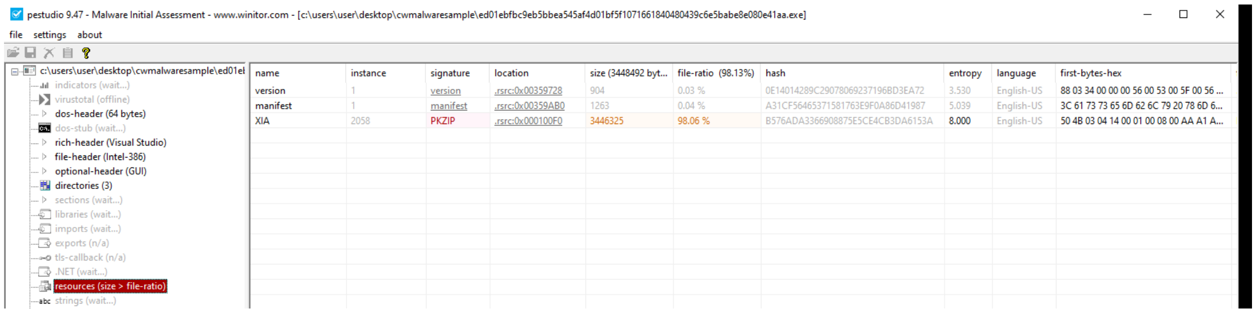
Once the file was deemed malicious and several sections were unpacked, the analyst used several tools to find out more information on how the file was composed by its developer. The tools Strings.exe and Floss.exe as mentioned in the static analysis procedure are both run from CLI. Both tools returned a list of executable files. Furthermore, a list of languages was also retrieved. It gives the impression the malware is created to target multiple countries with multiple languages. Evidence of this can be found in the image below and Appendix B. Additionally, the tool Floss was able to find a stack string named Oftware\.



```
 `#!N
6%=|^
2{lck
iP4,
!mgT7
b.wnry
c.wnry
1AgG
msg/m_bulgarian.wnry
"t=)
msg/m_chinese (simplified).wnry
"t=.|Vbq-
msg/m_chinese (traditional).wnry
"t=.
msg/m_croatian.wnry
msg/m_czech.wnry
msg/m_danish.wnry
msg/m_dutch.wnry
msg/m_english.wnry
"t=m
msg/m_filipino.wnry
"t=?
msg/m_finnish.wnry
"t=-
msg/m_french.wnry
msg/m_german.wnry
&XZR%
msg/m_greek.wnry
"t=x
msg/m_indonesian.wnry
"t=j%
msg/m_italian.wnry
"t=x-
msg/m_japanese.wnry
msg/m_korean.wnry
msg/m_latvian.wnry
msg/m_norwegian.wnry
msg/m_polish.wnry
msg/m_portuguese.wnry
"t=@W
msg/m_romanian.wnry
msg/m_russian.wnry
"t=3M
msg/m_slovak.wnry
msg/m_spanish.wnry
msg/m_swedish.wnry
msg/m_turkish.wnry
msg/m_vietnamese.wnry
r.wnry
Jcg4k_
s.wnry
t.wnry
"($:{
taskdl.exe
taskse.exe
lN$D
u.wnry
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v2">
    <security>
      <requestedPrivileges>
        <requestedExecutionLevel level="asInvoker" />
      </requestedPrivileges>
    </security>
  </trustInfo>
```

Figure 5: String.exe list of languages.

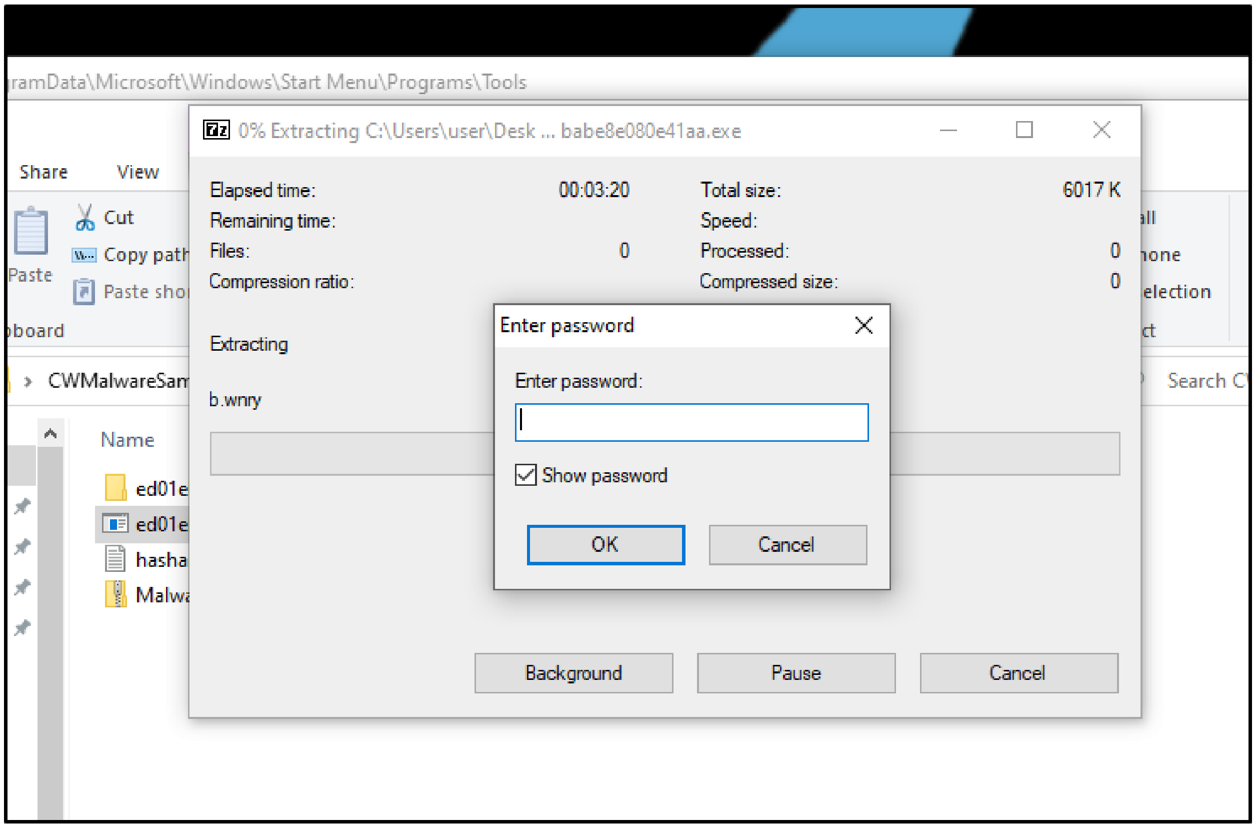
The next tool used was Pestudio. During the analysis of the resources, the tool found 3 resources: vision, manifest and XIA. They can be seen in the image below. Additionally, the file ratio of the resources was 98% XIA. Further investigation discovered that XIA is packed into a PKZIP, verifying ".rsrc" was packed which was already found with the tool DIE.


pestudio 9.47 - Malware Initial Assessment - www.winitor.com - [c:\users\user\desktop\cwmalwaresample\ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe]

name	instance	signature	location	size (3448492 byt...	file-ratio (98.13%)	hash	entropy	language	first-bytes-hex
version	1	version	.rsrc:0x00359728	904	0.03 %	0E14014289C29078069237196BD3EA72	3.530	English-US	88 03 34 00 00 00 56 00 53 00 5F 00 56 ...
manifest	1	manifest	.rsrc:0x00359AB0	1263	0.04 %	A31CF56465371581763E9F0A86D41987	5.039	English-US	3C 61 73 73 65 6D 62 6C 79 20 78 6D 6...
XIA	2058	PKZIP	.rsrc:0x000100F0	3446325	98.06 %	B576ADA3366908875E5CE4CB3DA6153A	8.000	English-US	50 4B 03 04 14 00 01 00 08 00 AA A1 A...

Figure 6: Pestudio, the discovery of the PKZIP

Knowing that 98% of the resources in the file were packed in a PKZIP, the next option was to find out more about that file. Research on PKZIP files revealed they could be unzipped using the tool ZIP-7. When this was tried, it asked for a password. The analyst noticed that the file extracted with sub-directories and other files even with the wrong password, but unfortunately, they were empty. Further research with the disassembler led to the discovery of the password WNcry@20l7. When unzipped with the correct password the folder had the files with content. These files can be seen in Figures 8 and 9.



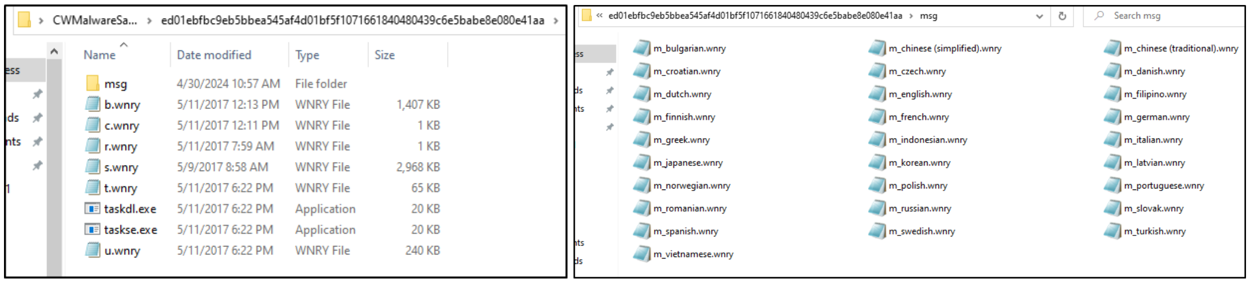
0% Extracting C:\Users\user\Desk ... babe8e080e41aa.exe

Elapsed time: 00:03:20
Remaining time:
Files: 0
Compression ratio:
Total size: 6017 K
Speed:
Processed: 0
Compressed size: 0

Extracting
b.wnry

Enter password:

Figure 7: Zip file required password.

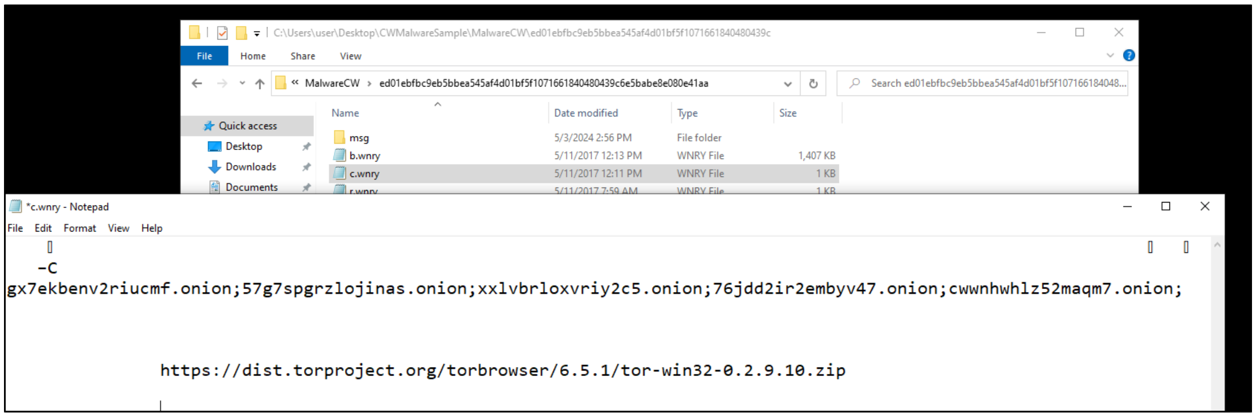

CWMalwareSa... › ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa

Name	Date modified	Type	Size
msg	4/30/2024 10:57 AM	File folder	
b.wnry	5/11/2017 12:13 PM	WNRY File	1,407 KB
c.wnry	5/11/2017 12:11 PM	WNRY File	1 KB
r.wnry	5/11/2017 7:59 AM	WNRY File	1 KB
s.wnry	5/9/2017 8:58 AM	WNRY File	2,968 KB
t.wnry	5/11/2017 6:22 PM	WNRY File	65 KB
taskdl.exe	5/11/2017 6:22 PM	Application	20 KB
taskse.exe	5/11/2017 6:22 PM	Application	20 KB
u.wnry	5/11/2017 6:22 PM	WNRY File	240 KB

« ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa › msg

m_bulgarian.wnry	m_chinese (simplified).wnry	m_chinese (traditional).wnry
m_croatian.wnry	m_czech.wnry	m_danish.wnry
m_dutch.wnry	m_english.wnry	m_filipino.wnry
m_finnish.wnry	m_french.wnry	m_german.wnry
m_greek.wnry	m_indonesian.wnry	m_italian.wnry
m_japanese.wnry	m_korean.wnry	m_latvian.wnry
m_norwegian.wnry	m_polish.wnry	m_portuguese.wnry
m_romanian.wnry	m_russian.wnry	m_slovak.wnry
m_spanish.wnry	m_swedish.wnry	m_turkish.wnry
m_vietnamese.wnry		

Figure 8 & 9: Content of the file extracted using Zip-7 and the password "WNCry2ol7".


C:\Users\user\Desktop\CWMalwareSample\MalwareCW\ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c

« MalwareCW › ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa

Name	Date modified	Type	Size
msg	5/3/2024 2:56 PM	File folder	
b.wnry	5/11/2017 12:13 PM	WNRY File	1,407 KB
c.wnry	5/11/2017 12:11 PM	WNRY File	1 KB
r.wnry	5/11/2017 7:59 AM	WNRY File	1 KB

*c.wnry - Notepad

```
    -C
gx7ekbenv2riucmf.onion;57g7spgrzlojinas.onion;xxlvbrloxvriy2c5.onion;76jdd2ir2embyv47.onion;cwwnhwhlz52maqm7.onion;


            https://dist.torproject.org/torbrowser/6.5.1/tor-win32-0.2.9.10.zip
```

Figure 10: Content of the c.wnry file".

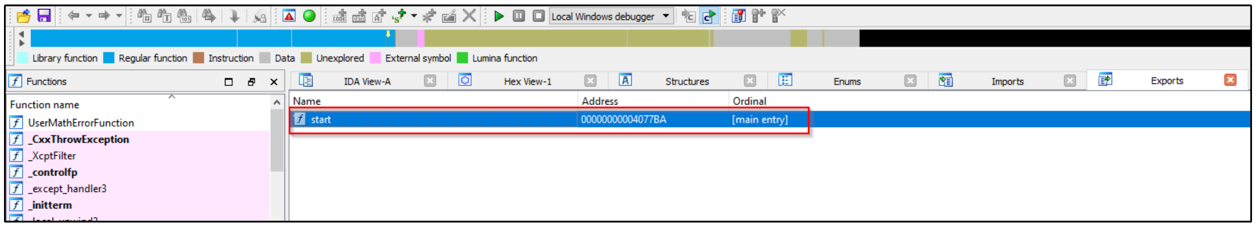
Dependency Walker was used to retrieve all the DLLs (Dynamic Link Library) within the composition of the malware. There were 4 DLLs named KERNERL32.DLL which contains core Windows functionality, USER32.DLL contains related to user interfaces, ADVAPI32.DLL related to advanced Windows services and security functions, and MSVCRT.DLL which has functions for memory management, string manipulation, file I/O operations. Each DLL is executed at runtime and calls all its functions. A list of all the functions within each DLL can be found in Appendix B. Here are a few that were interesting to the analyst.

KERNEL32.LL: CopyFileA, CreateDirectotyA, CreateDirectoryW, CreateProcessA, DeleteCriticalSection, GetProcAddress, GetFileAttributesA, etc.

USER32.DLL: wsprintfA

ADBAPI32.DLL: CreateServiceA, RegCloseKey, RegCreateKeyW, RegQueryValueEXA, StartServiceA

Further static analysis was done using the IDA pro disassembler. This allowed the analyst to examine the code in assembly language. When the malware was loaded into IDA Pro, the main entry function also called start was reviewed. Evidence of this can be found in Figure 11. The analyst entered every function and analysed its actions. Here are some of the relevant calls that the malware does.



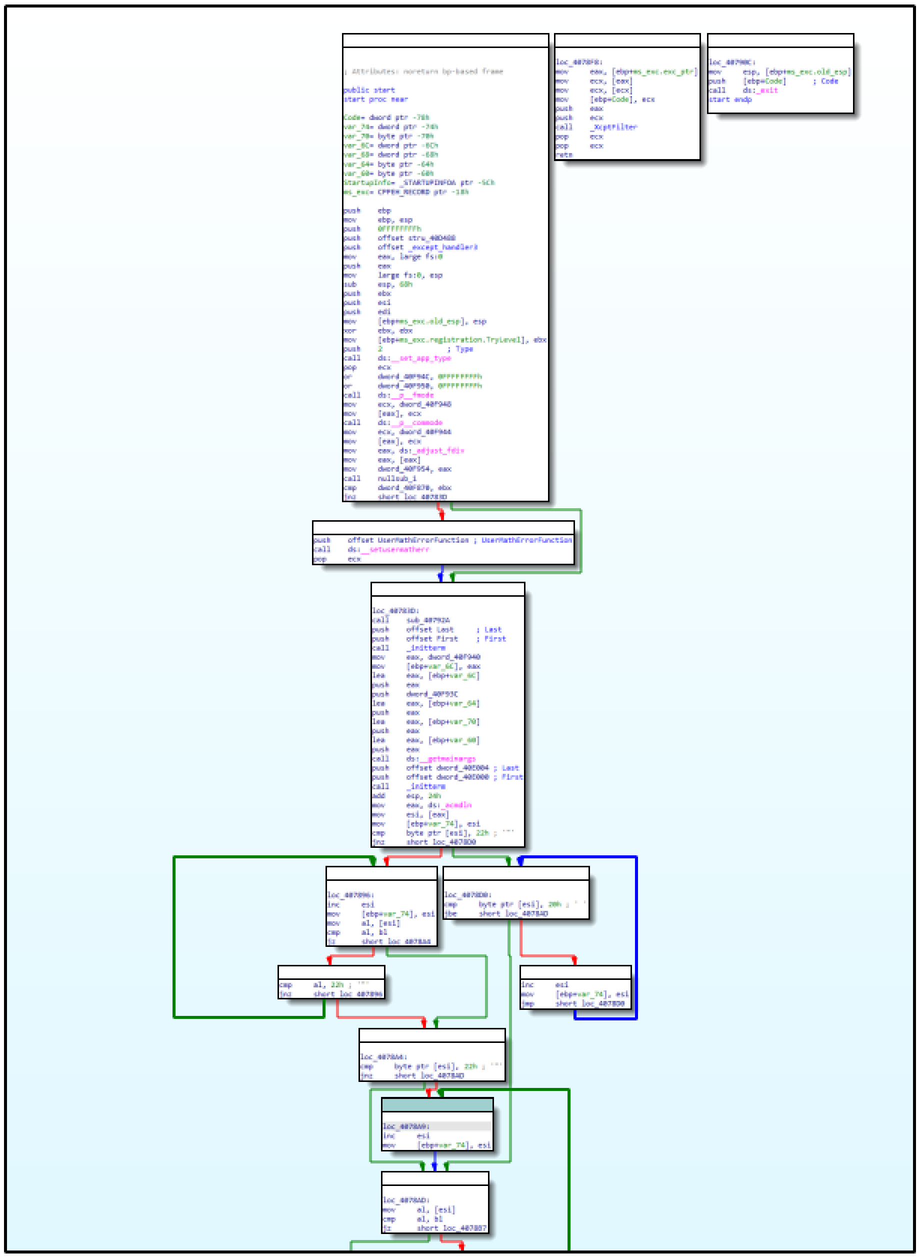
Library function | Regular function | Instruction | Data | Unexplored | External symbol | Lumina function

Functions

Function name
- UserMathErrorFunction
- **_CxxThrowException**
- _XcptFilter
- **_controlfp**
- _except_handler3
- **_initterm**

Name	Address	Ordinal
start	00000000004077BA	[main entry]

Figure 11: main entry function.



```
; Attributes: noreturn bp-based frame

public start
start proc near

Code= dword ptr -78h
var_74= dword ptr -74h
var_70= byte ptr -70h
var_6C= dword ptr -6Ch
var_68= dword ptr -68h
var_64= byte ptr -64h
var_60= byte ptr -60h
StartupInfo= _STARTUPINFOA ptr -5Ch
ms_exc= CPPEH_RECORD ptr -18h

push    ebp
mov     ebp, esp
push    0FFFFFFFFh
push    offset stru_4004D8
push    offset _except_handler3
mov     eax, large fs:0
push    eax
mov     large fs:0, esp
sub     esp, 68h
push    ebx
push    esi
push    edi
mov     [ebp+ms_exc.old_esp], esp
xor     ebx, ebx
mov     [ebp+ms_exc.registration.TryLevel], ebx
push    2               ; Type
call    ds:__set_app_type
pop     ecx
or      dword_40F94C, 0FFFFFFFFh
or      dword_40F950, 0FFFFFFFFh
call    ds:__p__fmode
mov     ecx, dword_40F948
mov     [eax], ecx
call    ds:__p__commode
mov     ecx, dword_40F944
mov     [eax], ecx
mov     eax, ds:_adjust_fdiv
mov     eax, [eax]
mov     dword_40F954, eax
call    nullsub_1
cmp     dword_40F870, ebx
jnz     short loc_40783D
```

```
loc_4077F8:
mov     eax, [ebp+ms_exc.exc_ptr]
mov     ecx, [eax]
mov     ecx, [ecx]
mov     [ebp+Code], ecx
push    eax
push    ecx
call    _XcptFilter
pop     ecx
pop     ecx
retn
```

```
loc_40780C:
mov     esp, [ebp+ms_exc.old_esp]
push    [ebp+Code]      ; Code
call    ds:_exit
start endp
```

```
push    offset UserMathErrorFunction ; UserMathErrorFunction
call    ds:__setusermatherr
pop     ecx
```

```
loc_40783D:
call    sub_40792A
push    offset Last     ; Last
push    offset First    ; First
call    _initterm
mov     eax, dword_40F940
mov     [ebp+var_6C], eax
lea     eax, [ebp+var_6C]
push    eax
push    dword_40F93C
lea     eax, [ebp+var_64]
push    eax
lea     eax, [ebp+var_70]
push    eax
lea     eax, [ebp+var_60]
push    eax
call    ds:__getmainargs
push    offset dword_40E004 ; Last
push    offset dword_40E000 ; First
call    _initterm
add     esp, 24h
mov     eax, ds:_acmdln
mov     esi, [eax]
mov     [ebp+var_74], esi
cmp     byte ptr [esi], 22h ; '"'
jnz     short loc_4078D0
```

```
loc_407896:
inc     esi
mov     [ebp+var_74], esi
mov     al, [esi]
cmp     al, bl
jz      short loc_4078A4
```

```
loc_4078D0:
cmp     byte ptr [esi], 20h ; ' '
jbe     short loc_4078AD
```

```
cmp     al, 22h ; '"'
jnz     short loc_407896
```

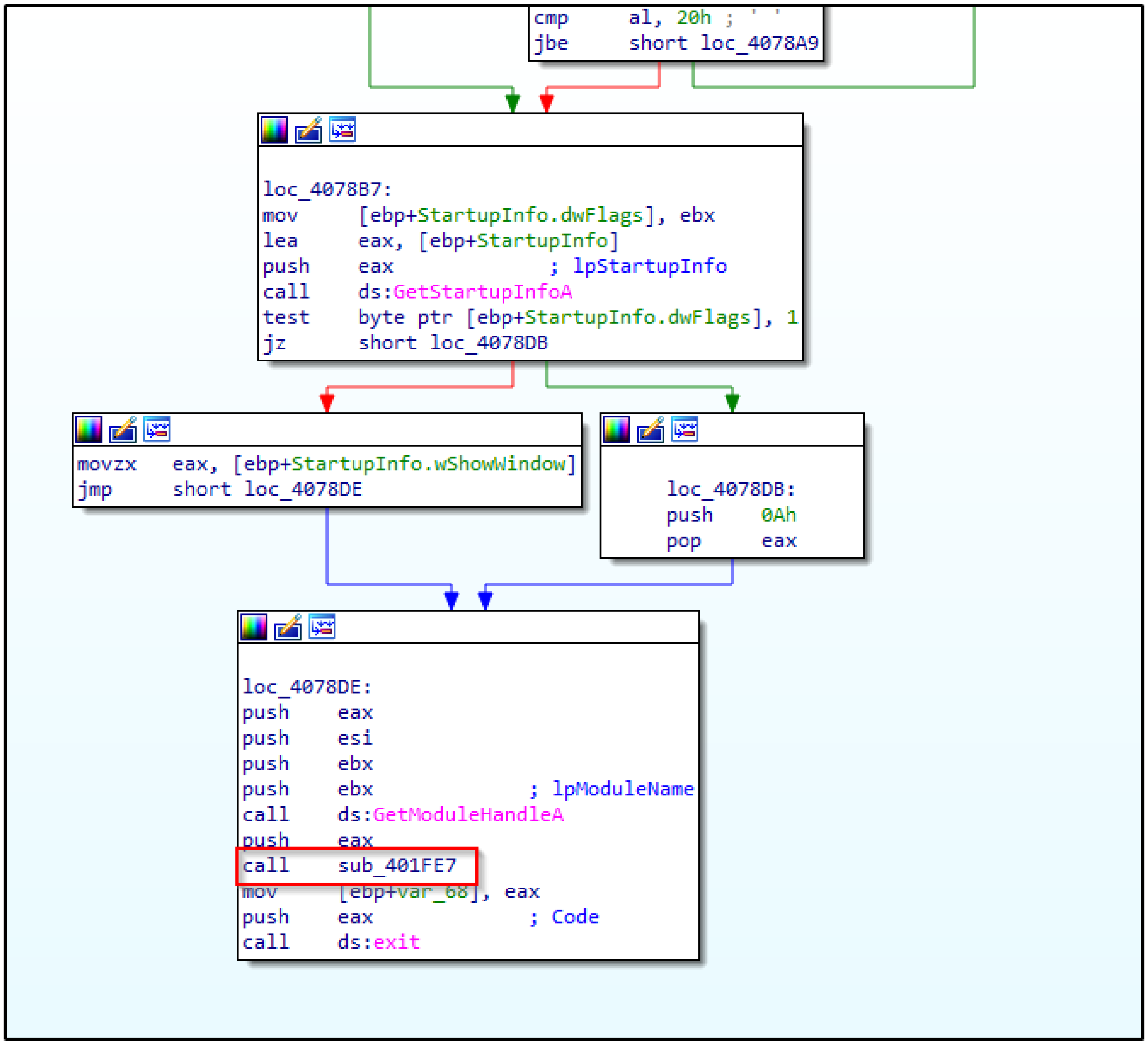
```
inc     esi
mov     [ebp+var_74], esi
jmp     short loc_4078D0
```

```
loc_4078A4:
cmp     byte ptr [esi], 22h ; '"'
jnz     short loc_4078AD
```

```
loc_4078A9:
inc     esi
mov     [ebp+var_74], esi
```

```
loc_4078AD:
mov     al, [esi]
cmp     al, bl
jz      short loc_4078B7
```

Figure 12: main entry function blocks.



```
cmp     al, 20h ; ' '
jbe     short loc_4078A9
```

```
loc_4078B7:
mov     [ebp+StartupInfo.dwFlags], ebx
lea     eax, [ebp+StartupInfo]
push    eax             ; lpStartupInfo
call    ds:GetStartupInfoA
test    byte ptr [ebp+StartupInfo.dwFlags], 1
jz      short loc_4078DB
```

```
movzx   eax, [ebp+StartupInfo.wShowWindow]
jmp     short loc_4078DE
```

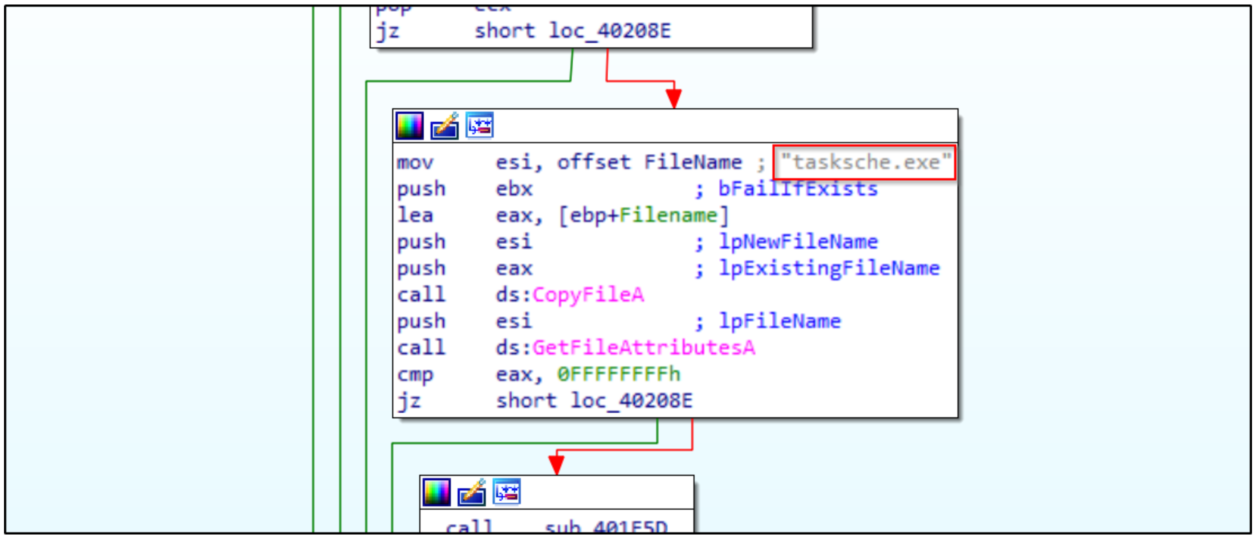
```
loc_4078DB:
push    0Ah
pop     eax
```

```
loc_4078DE:
push    eax
push    esi
push    ebx
push    ebx             ; lpModuleName
call    ds:GetModuleHandleA
push    eax
call    sub_401FE7
mov     [ebp+var_68], eax
push    eax             ; Code
call    ds:exit
```

Figure 13: sub-401FE7 (most relevant subroutine from entry block).

The interesting block that was discovered can be seen in the image above. This block was involved in handling module operations and possibly terminating the process. First getting the handle of a module using GetModuleHandleA, then calling a subroutine at address sub_401FE7, likely to perform some operation with the module handle. Finally, it stored the result in [ebp+var_68].

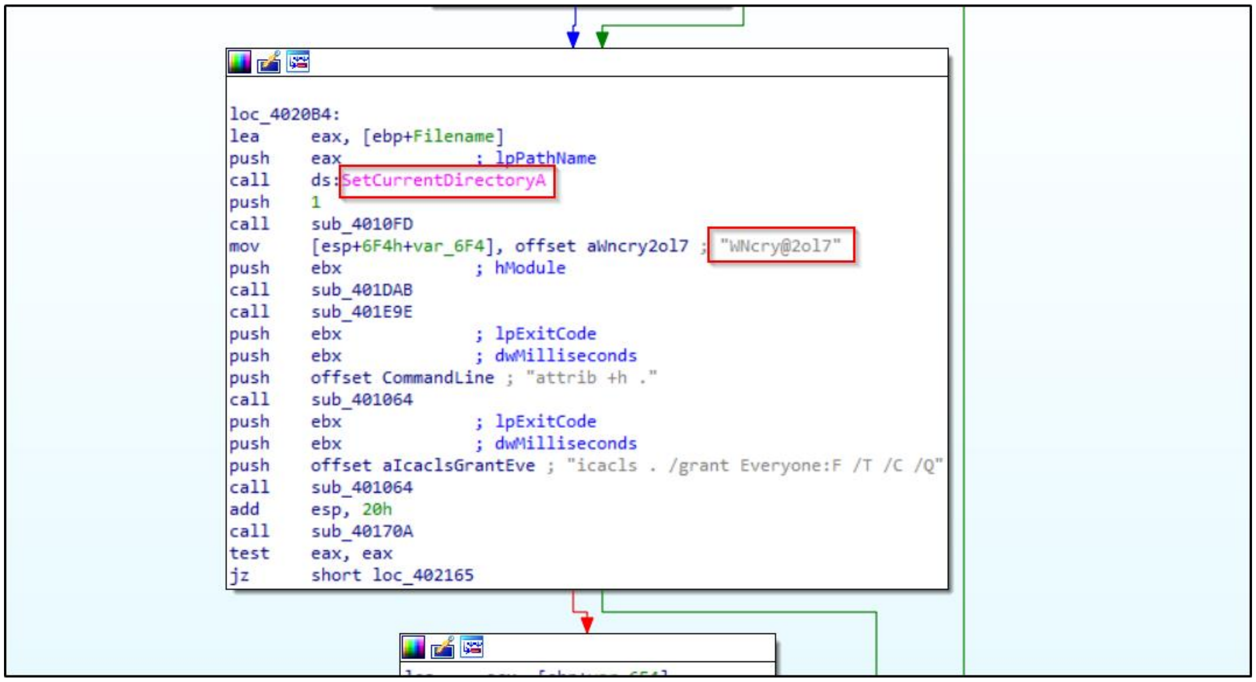
Inside the sub_401FE7 seen in the previous block, there was another block that called tasksche.exe. This block of code copies a file named "tasksche.exe" to a new location and then checks if the file exists using the GetFileAttributesA function. If the file does not exist, it proceeds to an address called loc_40208E.



```
jz      short loc_40208E

mov     esi, offset FileName ; "tasksche.exe"
push    ebx             ; bFailIfExists
lea     eax, [ebp+Filename]
push    esi             ; lpNewFileName
push    eax             ; lpExistingFileName
call    ds:CopyFileA
push    esi             ; lpFileName
call    ds:GetFileAttributesA
cmp     eax, 0FFFFFFFFh
jz      short loc_40208E

call    sub_401F5D
```

Figure 14: Executable tasksche.exe.

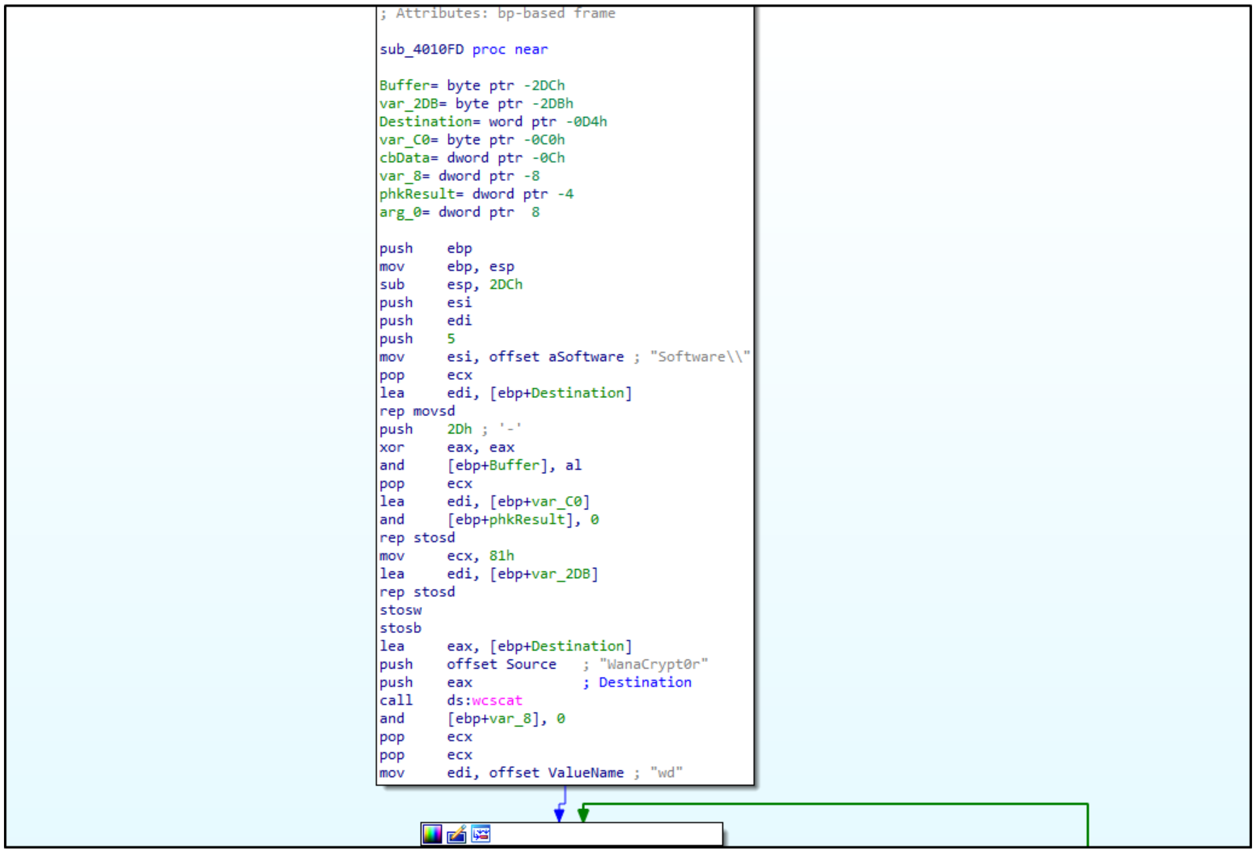
In the same sub_401FE7, there was another block that was set to manipulate files, directories, and permission. It also called several other subroutines as part of the block. The discovery of the "WNcry@2ol7" when analysing other blocks that dealt with directories gave the impression of it being a password to decrypt the PKZIP file.



```
loc_4020B4:
lea     eax, [ebp+Filename]
push    eax             ; lpPathName
call    ds:SetCurrentDirectoryA
push    1
call    sub_4010FD
mov     [esp+6F4h+var_6F4], offset aWncry2ol7 ; "WNcry@2ol7"
push    ebx             ; hModule
call    sub_401DAB
call    sub_401E9E
push    ebx             ; lpExitCode
push    ebx             ; dwMilliseconds
push    offset CommandLine ; "attrib +h ."
call    sub_401064
push    ebx             ; lpExitCode
push    ebx             ; dwMilliseconds
push    offset aIcaclsGrantEve ; "icacls . /grant Everyone:F /T /C /Q"
call    sub_401064
add     esp, 20h
call    sub_40170A
test    eax, eax
jz      short loc_402165
```

Figure 15: Creation of directories and password discovery.

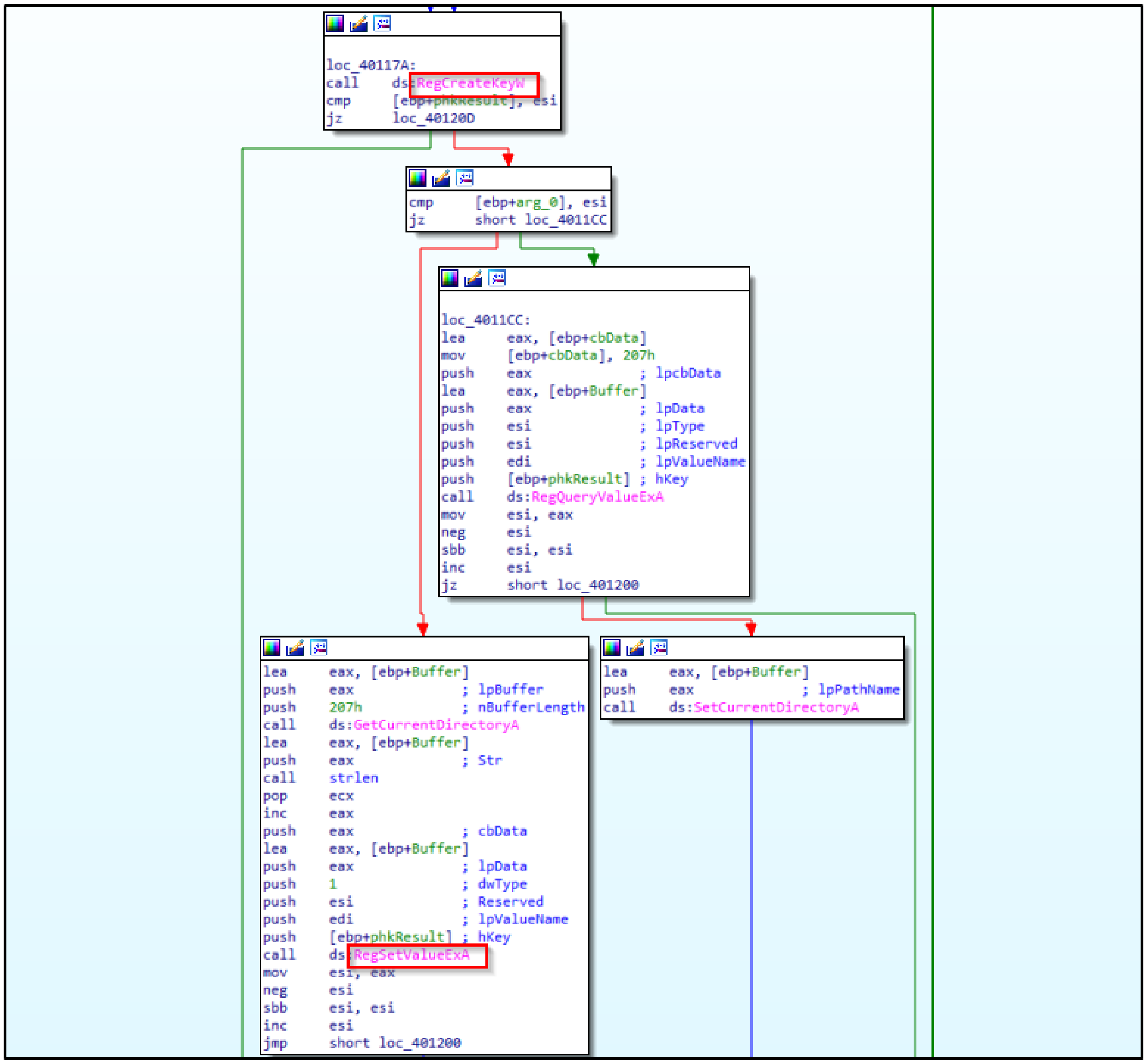
From the subroutine sub_4010FD inside the block above, the first things that caught the analyst's eyes were the 3 comments, "Software\\", "WanaCryt0r", and "wd".



```
; Attributes: bp-based frame

sub_4010FD proc near

Buffer= byte ptr -2DCh
var_2DB= byte ptr -2DBh
Destination= word ptr -0D4h
var_C0= byte ptr -0C0h
cbData= dword ptr -0Ch
var_8= dword ptr -8
phkResult= dword ptr -4
arg_0= dword ptr  8

push    ebp
mov     ebp, esp
sub     esp, 2DCh
push    esi
push    edi
push    5
mov     esi, offset aSoftware ; "Software\\"
pop     ecx
lea     edi, [ebp+Destination]
rep movsd
push    2Dh ; '-'
xor     eax, eax
and     [ebp+Buffer], al
pop     ecx
lea     edi, [ebp+var_C0]
and     [ebp+phkResult], 0
rep stosd
mov     ecx, 81h
lea     edi, [ebp+var_2DB]
rep stosd
stosw
stosb
lea     eax, [ebp+Destination]
push    offset Source   ; "WanaCrypt0r"
push    eax             ; Destination
call    ds:wcscat
and     [ebp+var_8], 0
pop     ecx
pop     ecx
mov     edi, offset ValueName ; "wd"
```

Figure 16: subroutine sub_4010FD.

Other subroutines showed further evidence of attempts to create and edit new registry keys using functions such as 'RegCreateKeyW', 'RegQueryValueExA' and 'RegSetValueExA'. This can be seen in Figure 17.



```
loc_40117A:
call    ds:RegCreateKeyW
cmp     [ebp+phkResult], esi
jz      loc_40120D
```

```
cmp     [ebp+arg_0], esi
jz      short loc_4011CC
```

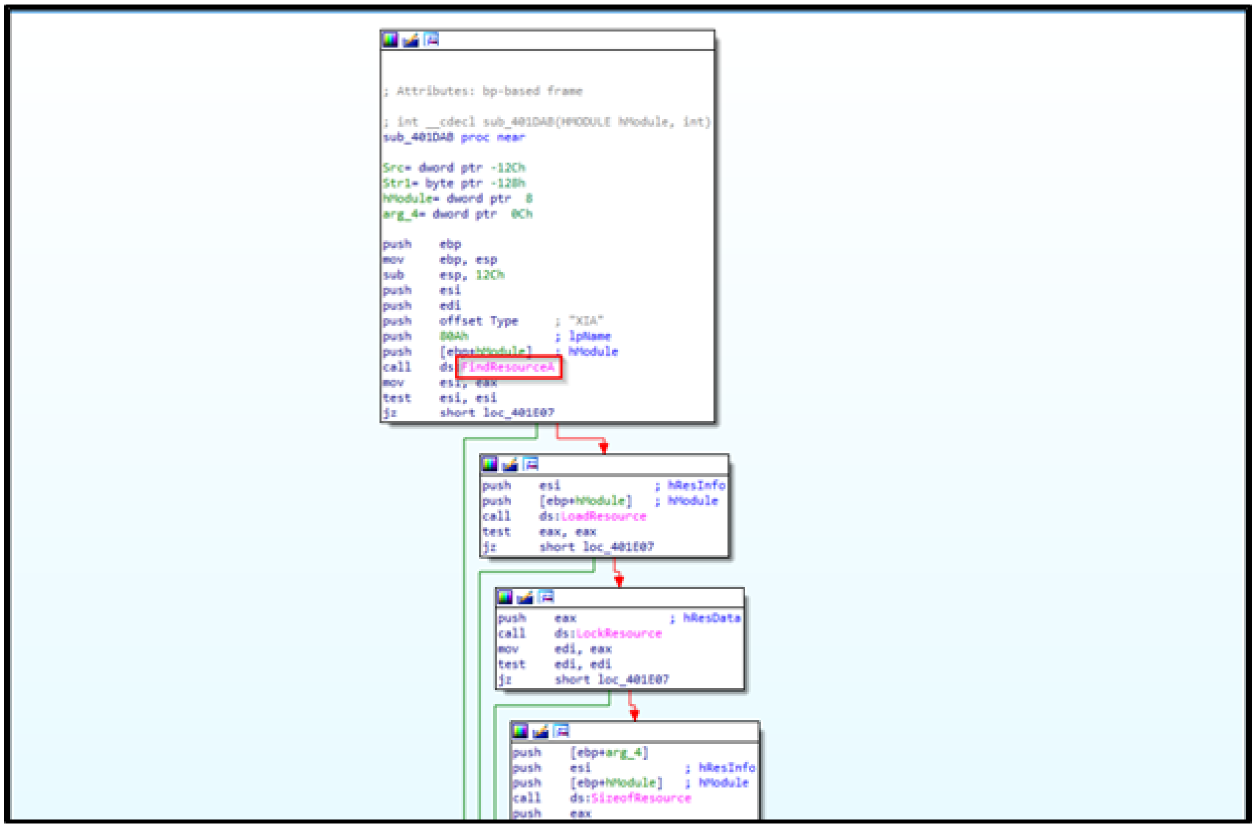
```
loc_4011CC:
lea     eax, [ebp+cbData]
mov     [ebp+cbData], 207h
push    eax                 ; lpcbData
lea     eax, [ebp+Buffer]
push    eax                 ; lpData
push    esi                 ; lpType
push    esi                 ; lpReserved
push    edi                 ; lpValueName
push    [ebp+phkResult]     ; hKey
call    ds:RegQueryValueExA
mov     esi, eax
neg     esi
sbb     esi, esi
inc     esi
jz      short loc_401200
```

```
lea     eax, [ebp+Buffer]
push    eax                 ; lpBuffer
push    207h                ; nBufferLength
call    ds:GetCurrentDirectoryA
lea     eax, [ebp+Buffer]
push    eax                 ; Str
call    strlen
pop     ecx
inc     eax
push    eax                 ; cbData
lea     eax, [ebp+Buffer]
push    eax                 ; lpData
push    1                   ; dwType
push    esi                 ; Reserved
push    edi                 ; lpValueName
push    [ebp+phkResult]     ; hKey
call    ds:RegSetValueExA
mov     esi, eax
neg     esi
sbb     esi, esi
inc     esi
jmp     short loc_401200
```

```
lea     eax, [ebp+Buffer]
push    eax                 ; lpPathName
call    ds:SetCurrentDirectoryA
```

Figure 17: Creation of registry keys, retrieving values, and setting directories.

The sub_401DAB subroutine called the function “FindResourceA” to find the file type “XIA” which was already discovered using DIE. Once it finds the resource the subroutine would load the resource and push the file “c.winry” onto the stack.



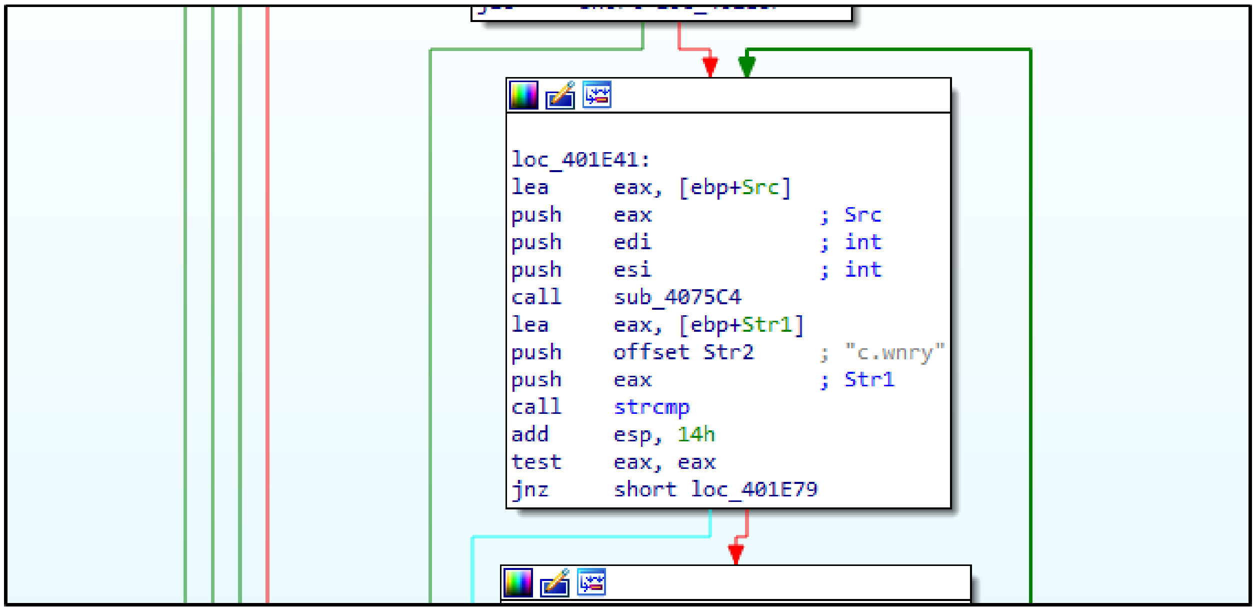
```
; Attributes: bp-based frame

; int __cdecl sub_401DAB(HMODULE hModule, int)
sub_401DAB proc near

Src= dword ptr -12Ch
Str1= byte ptr -128h
hModule= dword ptr  8
arg_4= dword ptr  0Ch

push    ebp
mov     ebp, esp
sub     esp, 12Ch
push    esi
push    edi
push    offset Type       ; "XIA"
push    80Ah              ; lpName
push    [ebp+hModule]     ; hModule
call    ds:FindResourceA
mov     esi, eax
test    esi, esi
jz      short loc_401E07
```

```
push    esi               ; hResInfo
push    [ebp+hModule]     ; hModule
call    ds:LoadResource
test    eax, eax
jz      short loc_401E07
```

```
push    eax               ; hResData
call    ds:LockResource
mov     edi, eax
test    edi, edi
jz      short loc_401E07
```

```
push    [ebp+arg_4]
push    esi               ; hResInfo
push    [ebp+hModule]     ; hModule
call    ds:SizeofResource
push    eax
```

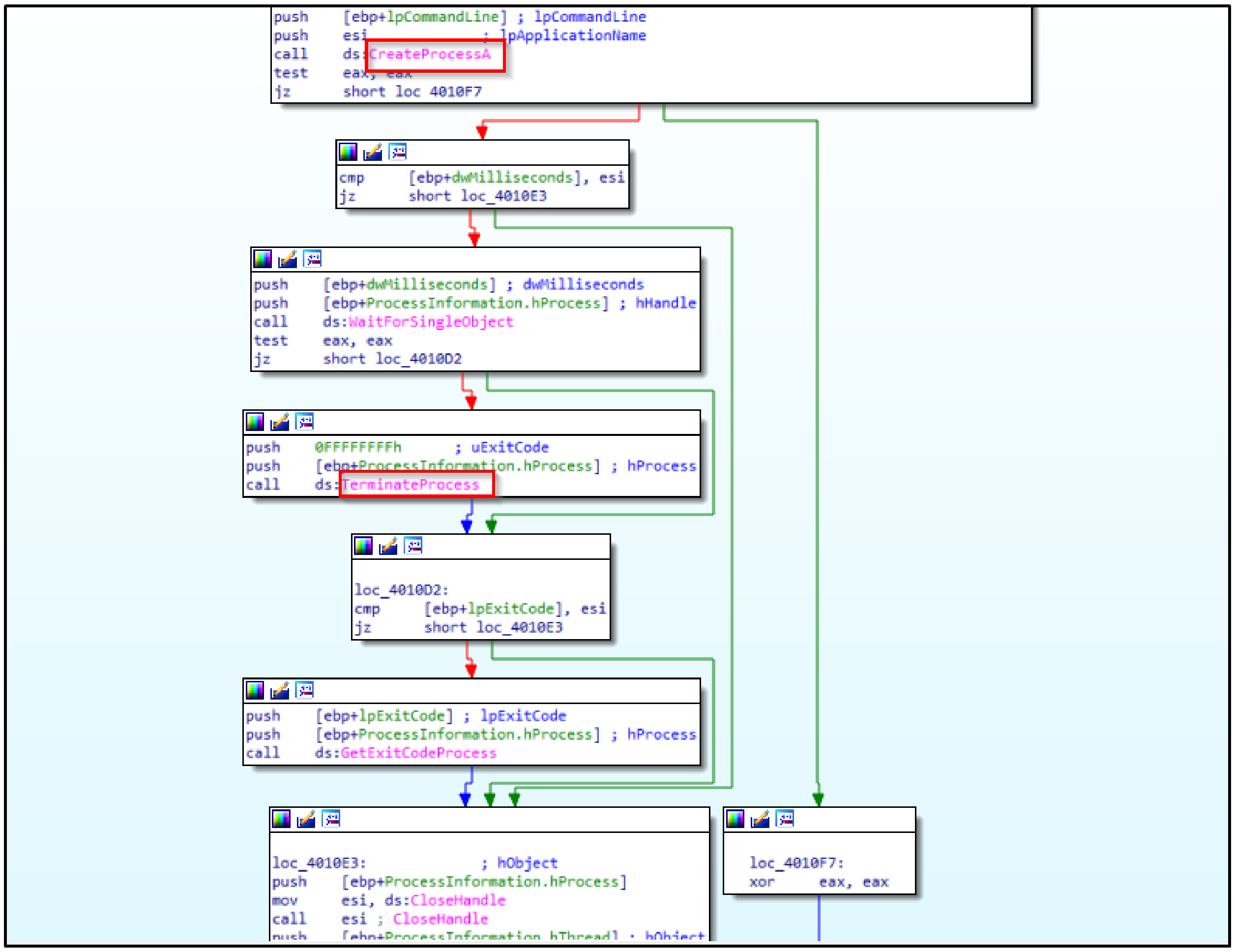
Figure 18: Manipulation of resources.



```
loc_401E41:
lea     eax, [ebp+Src]
push    eax               ; Src
push    edi               ; int
push    esi               ; int
call    sub_4075C4
lea     eax, [ebp+Str1]
push    offset Str2       ; "c.wnry"
push    eax               ; Str1
call    strcmp
add     esp, 14h
test    eax, eax
jnz     short loc_401E79
```

Figure 19: Passing c.wnry into the stack.

Another subroutine called sub_401064 was discovered. Within its blocks, it had instructions that called functions such as createProcessA, and TerminateProcess.



```
push    [ebp+lpCommandLine] ; lpCommandLine
push    esi             ; lpApplicationName
call    ds:CreateProcessA
test    eax, eax
jz      short loc_4010F7
```

```
cmp     [ebp+dwMilliseconds], esi
jz      short loc_4010E3
```

```
push    [ebp+dwMilliseconds] ; dwMilliseconds
push    [ebp+ProcessInformation.hProcess] ; hHandle
call    ds:WaitForSingleObject
test    eax, eax
jz      short loc_4010D2
```

```
push    0FFFFFFFFh      ; uExitCode
push    [ebp+ProcessInformation.hProcess] ; hProcess
call    ds:TerminateProcess
```

```
loc_4010D2:
cmp     [ebp+lpExitCode], esi
jz      short loc_4010E3
```

```
push    [ebp+lpExitCode] ; lpExitCode
push    [ebp+ProcessInformation.hProcess] ; hProcess
call    ds:GetExitCodeProcess
```

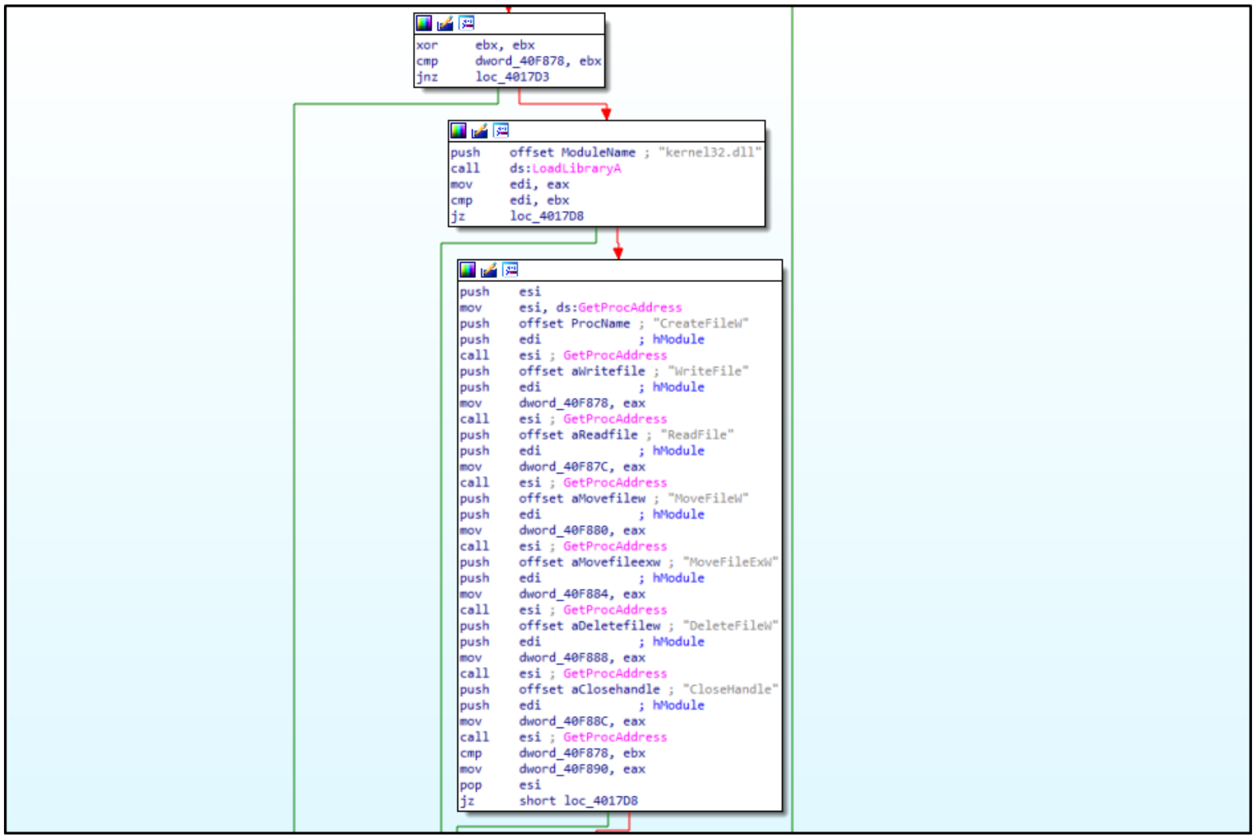
```
loc_4010E3:                 ; hObject
push    [ebp+ProcessInformation.hProcess]
mov     esi, ds:CloseHandle
call    esi ; CloseHandle
push    [ebp+ProcessInformation.hThread] ; hObject
```

```
loc_4010F7:
xor     eax, eax
```

Figure 20: Manipulation of processes.

The subroutine that was in charge of loading the kernel.dll library was creating sub_40170A. There were also several functions:

- GetProcAdress: GetProcAddress is used to retrieve the address of exported functions or procedures from a specified dynamic-link library (DLL).
- Retrieve CreateFileW function address: Pushes the string "CreateFileW" onto the stack, followed by the handle to the module where the function resides (hModule). Then, it calls GetProcAddress to retrieve the address of CreateFileW and stores it at location dword_40F878. It repeats this for other functions WriteFile, ReadFile, MoveFileW, MoveFileExW, DeleteFileW, and CloseHandle. For each function, it pushes the function name, hModule, and then calls GetProcAddress to retrieve and store the function address.

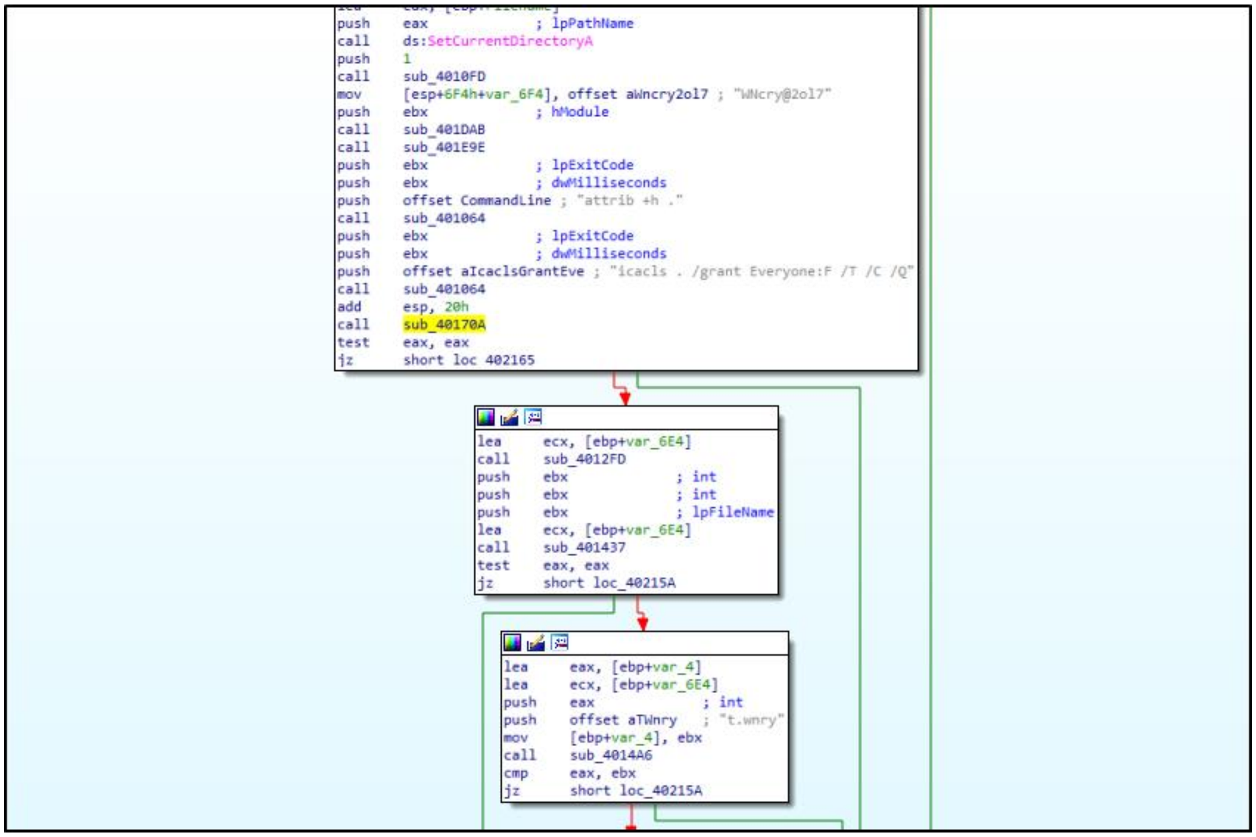


```
xor     ebx, ebx
cmp     dword_40F878, ebx
jnz     loc_4017D3
```

```
push    offset ModuleName ; "kernel32.dll"
call    ds:LoadLibraryA
mov     edi, eax
cmp     edi, ebx
jz      loc_4017D8
```

```
push    esi
mov     esi, ds:GetProcAddress
push    offset ProcName ; "CreateFileW"
push    edi             ; hModule
call    esi ; GetProcAddress
push    offset aWritefile ; "WriteFile"
push    edi             ; hModule
mov     dword_40F878, eax
call    esi ; GetProcAddress
push    offset aReadfile ; "ReadFile"
push    edi             ; hModule
mov     dword_40F87C, eax
call    esi ; GetProcAddress
push    offset aMovefilew ; "MoveFileW"
push    edi             ; hModule
mov     dword_40F880, eax
call    esi ; GetProcAddress
push    offset aMovefileexw ; "MoveFileExW"
push    edi             ; hModule
mov     dword_40F884, eax
call    esi ; GetProcAddress
push    offset aDeletefilew ; "DeleteFileW"
push    edi             ; hModule
mov     dword_40F888, eax
call    esi ; GetProcAddress
push    offset aClosehandle ; "CloseHandle"
push    edi             ; hModule
mov     dword_40F88C, eax
call    esi ; GetProcAddress
cmp     dword_40F878, ebx
mov     dword_40F890, eax
pop     esi
jz      short loc_4017D8
```

Figure 21: File manipulation.



```
push    eax             ; lpPathName
call    ds:SetCurrentDirectoryA
push    1
call    sub_4010FD
mov     [esp+6F4h+var_6F4], offset aWncry2ol7 ; "WNcry@2ol7"
push    ebx             ; hModule
call    sub_401DAB
call    sub_401E9E
push    ebx             ; lpExitCode
push    ebx             ; dwMilliseconds
push    offset CommandLine ; "attrib +h ."
call    sub_401064
push    ebx             ; lpExitCode
push    ebx             ; dwMilliseconds
push    offset aIcaclsGrantEve ; "icacls . /grant Everyone:F /T /C /Q"
call    sub_401064
add     esp, 20h
call    sub_40170A
test    eax, eax
jz      short loc_402165
```

```
lea     ecx, [ebp+var_6E4]
call    sub_4012FD
push    ebx             ; int
push    ebx             ; int
push    ebx             ; lpFileName
lea     ecx, [ebp+var_6E4]
call    sub_401437
test    eax, eax
jz      short loc_40215A
```

```
lea     eax, [ebp+var_4]
lea     ecx, [ebp+var_6E4]
push    eax             ; int
push    offset aTWnry   ; "t.wnry"
mov     [ebp+var_4], ebx
call    sub_4014A6
cmp     eax, ebx
jz      short loc_40215A
```

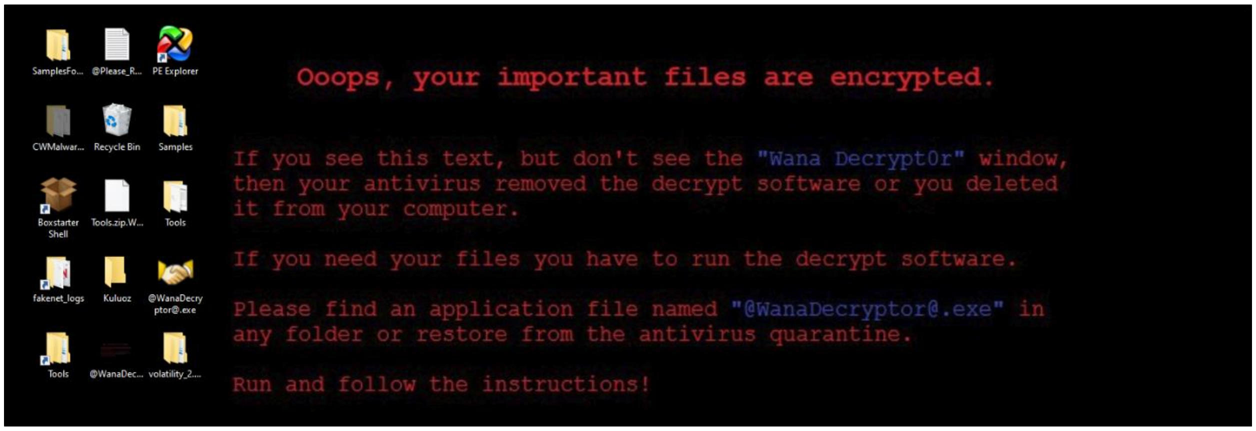
Figure 22: Passing t.wnry into the stack.

The final interesting block is the image above where t.wnry is pushed into the stack. This file was already discovered and examined in Figure 8.

3.2 Dynamic Analysis Discussion

The malware was executed several times on the Virtual machine. A snapshot of the initial state of the VM was created to revert back for the use of different tools.

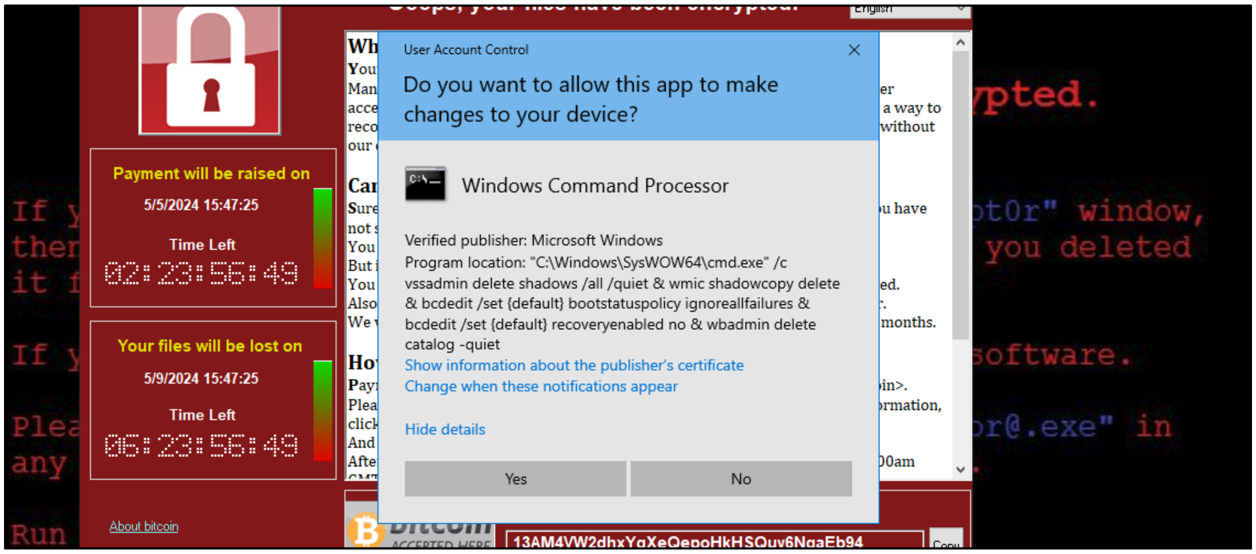
The initial phase of this analysis was to run the malware and see what happens. At first glance, new files appeared to be created on the Desktop. Local files started getting encrypted, and the background changed to an image that displayed a message.



```
Ooops, your important files are encrypted.

If you see this text, but don't see the "Wana Decrypt0r" window,
then your antivirus removed the decrypt software or you deleted
it from your computer.

If you need your files you have to run the decrypt software.

Please find an application file named "@WanaDecryptor@.exe" in
any folder or restore from the antivirus quarantine.

Run and follow the instructions!
```

Figure 23: New desktop view with new files and background

The User Account Control also requested the analyst's permission to make changes. The changes required can be found when clicking "show details" and in the figure below. It was requesting to delete all the Windows backups using "*wmic shadowcopy*" and "*wbadmin delete catalog –quiet*" (the "-quiet" flag is used to operate silently without asking for confirmation). With the command "*Bcdedit /set {default} bootstatuspolicy ignorerealfailures"* and "*Bcdedit /set {default} recoveryenabled no*" it will set the boot status policy to ignore failures and won't automatically launch into recovery mode if it encounters boot problems.



User Account Control

Do you want to allow this app to make changes to your device?

Windows Command Processor

Verified publisher: Microsoft Windows
Program location: "C:\Windows\SysWOW64\cmd.exe" /c vssadmin delete shadows /all /quiet & wmic shadowcopy delete & bcdedit /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default} recoveryenabled no & wbadmin delete catalog -quiet

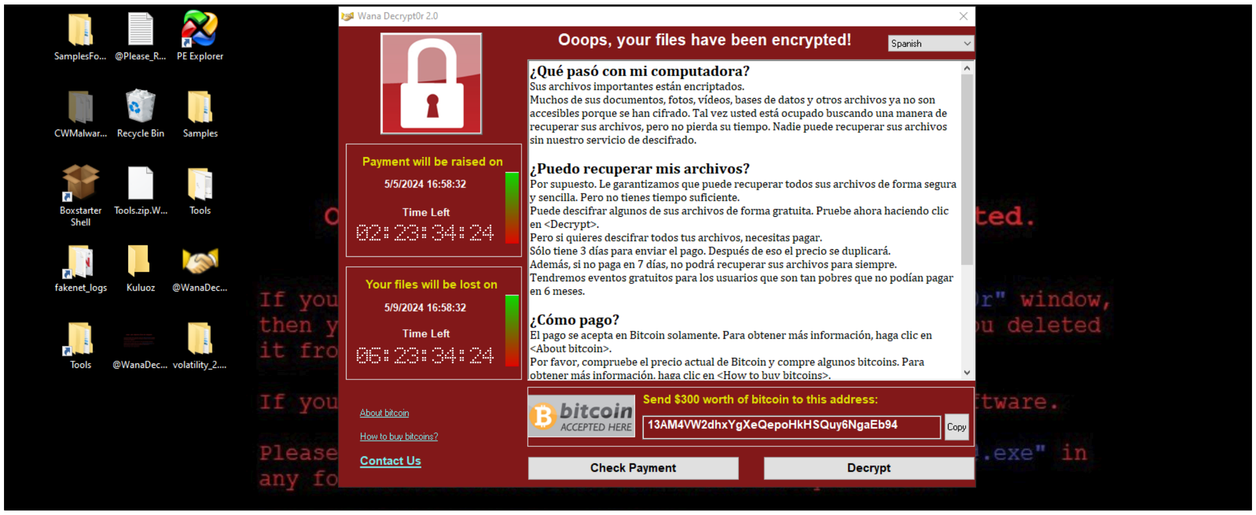
Show information about the publisher's certificate
Change when these notifications appear

Hide details

Yes No

Figure 24: User Account Control

@WanaDecrypt@.exe, one of the new files created on the desktop was an executable. It was executed automatically containing information on the decryption process. A message displayable in multiple languages showed information on what was happening. This was previously discovered using string.exe during the static analysis. The information was about what happened, how to recover the files and how to pay. Furthermore, 2 timers displayed how much is left before the 300$ payment increases, and how much is left for the files to be deleted permanently. Ironically, several help links were added that explained what a bitcoin is, how to buy it, how to contact the developer, and the crypto wallet where the bitcoin should be paid into. Lastly, there were 2 buttons, a check payment, and a decrypt button. The decrypt button will decrypt some random file as shown in Figure 26.



Ooops, your files have been encrypted!

Payment will be raised on
5/5/2024 16:58:32
Time Left
02:23:34:24

Your files will be lost on
5/9/2024 16:58:32
Time Left
06:23:34:24

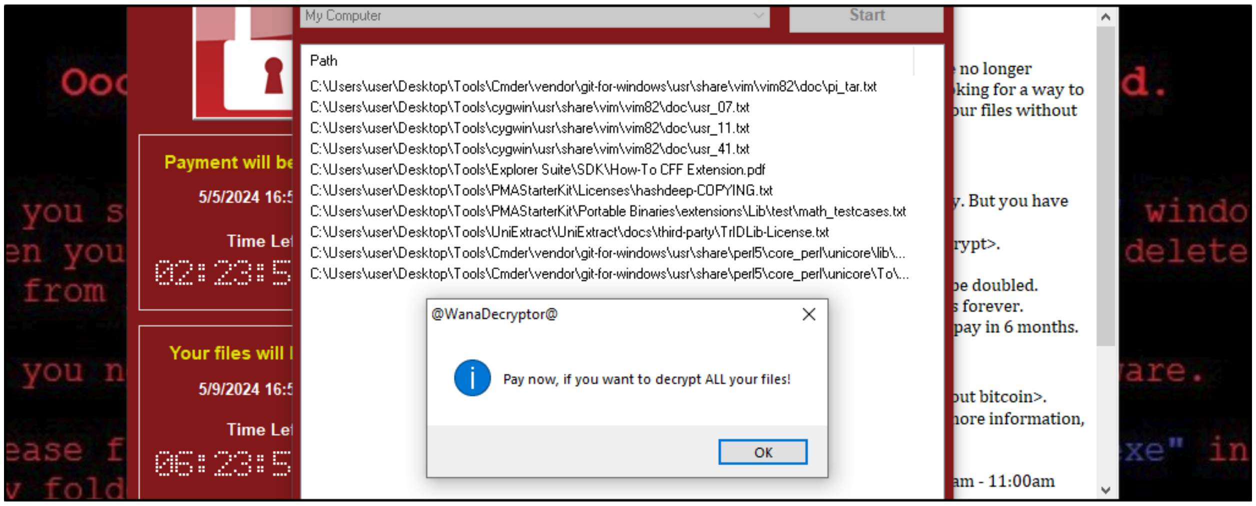
¿Qué pasó con mi computadora?
Sus archivos importantes están encriptados.
Muchos de sus documentos, fotos, vídeos, bases de datos y otros archivos ya no son accesibles porque se han cifrado. Tal vez usted está ocupado buscando una manera de recuperar sus archivos, pero no pierda su tiempo. Nadie puede recuperar sus archivos sin nuestro servicio de descifrado.

¿Puedo recuperar mis archivos?
Por supuesto. Le garantizamos que puede recuperar todos sus archivos de forma segura y sencilla. Pero no tienes tiempo suficiente.
Puede descifrar algunos de sus archivos de forma gratuita. Pruebe ahora haciendo clic en <Decrypt>.
Pero si quieres descifrar todos tus archivos, necesitas pagar.
Sólo tiene 3 días para enviar el pago. Después de eso el precio se duplicará.
Además, si no paga en 7 días, no podrá recuperar sus archivos para siempre.
Tendremos eventos gratuitos para los usuarios que son tan pobres que no podían pagar en 6 meses.

¿Cómo pago?
El pago se acepta en Bitcoin solamente. Para obtener más información, haga clic en <About bitcoin>.
Por favor, compruebe el precio actual de Bitcoin y compre algunos bitcoins. Para obtener más información, haga clic en <How to buy bitcoins>.

bitcoin ACCEPTED HERE
Send $300 worth of bitcoin to this address:
13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94

Figure 25: @WanaDecrypt@.exe

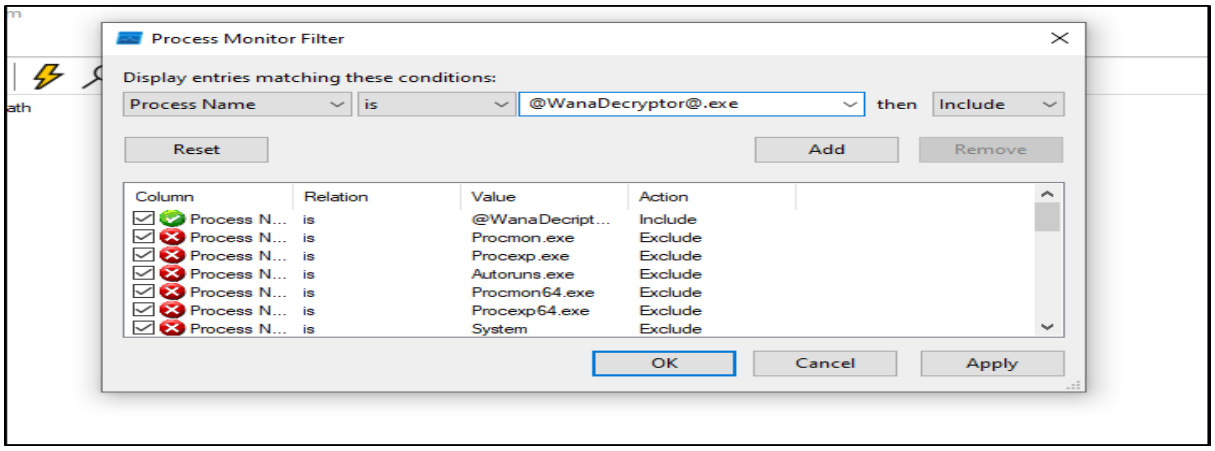


```
Path
C:\Users\user\Desktop\Tools\Cmder\vendor\git-for-windows\usr\share\vim\vim82\doc\pi_tar.txt
C:\Users\user\Desktop\Tools\cygwin\usr\share\vim\vim82\doc\usr_07.txt
C:\Users\user\Desktop\Tools\cygwin\usr\share\vim\vim82\doc\usr_11.txt
C:\Users\user\Desktop\Tools\cygwin\usr\share\vim\vim82\doc\usr_41.txt
C:\Users\user\Desktop\Tools\Explorer Suite\SDK\How-To CFF Extension.pdf
C:\Users\user\Desktop\Tools\PMAStarterKit\Licenses\hashdeep-COPYING.txt
C:\Users\user\Desktop\Tools\PMAStarterKit\Portable Binaries\extensions\Lib\test\math_testcases.txt
C:\Users\user\Desktop\Tools\UniExtract\UniExtract\docs\third-party\TrIDLib-License.txt
C:\Users\user\Desktop\Tools\Cmder\vendor\git-for-windows\usr\share\perl5\core_perl\unicore\lib\...
C:\Users\user\Desktop\Tools\Cmder\vendor\git-for-windows\usr\share\perl5\core_perl\unicore\To\...
```

@WanaDecryptor@
Pay now, if you want to decrypt ALL your files!

Figure 26: Files decrypted.

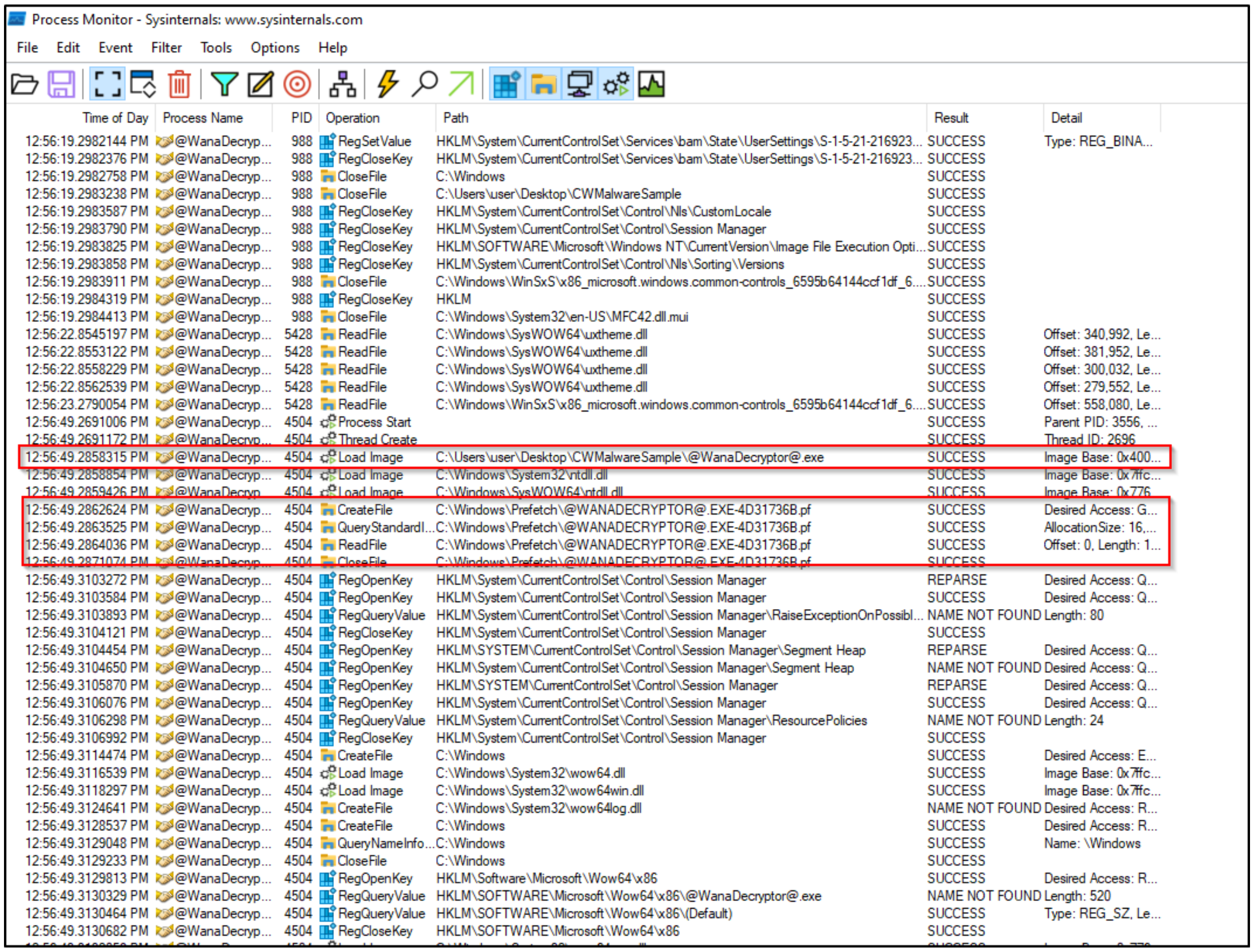
Reverting to the initial state using the snapshot, the tool Procmon (Process Monitor) was used before executing the file to trace the process of the @WanaDecriptor@.exe file using a filter that searches for the process name.



```
Process Monitor Filter
Display entries matching these conditions:
Process Name   is   @WanaDecryptor@.exe   then   Include
Reset                                    Add   Remove

Column         Relation   Value             Action
Process N...   is         @WanaDecript...   Include
Process N...   is         Procmon.exe       Exclude
Process N...   is         Procexp.exe       Exclude
Process N...   is         Autoruns.exe      Exclude
Process N...   is         Procmon64.exe     Exclude
Process N...   is         Procexp64.exe     Exclude
Process N...   is         System            Exclude

OK   Cancel   Apply
```

Figure 27: Procmon filter

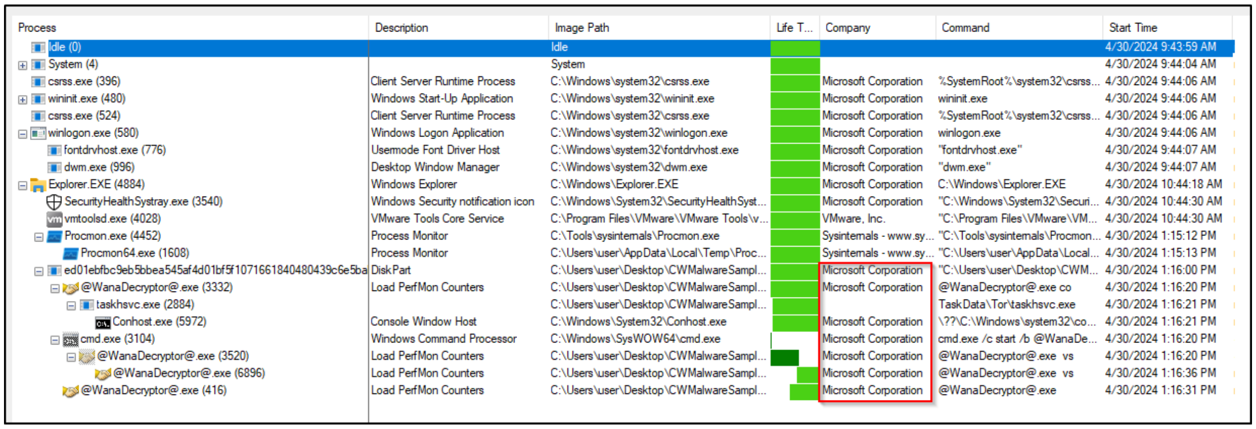
There were a lot of processes that were running under the name of @wanaDecryptor.exe, from loading the background image to, creating, reading, and closing files as shown in the images below.



```
Process Monitor - Sysinternals: www.sysinternals.com
File  Edit  Event  Filter  Tools  Options  Help

Time of Day        Process Name       PID   Operation        Path                                                                         Result           Detail
12:56:19.2982144 PM @WanaDecryp...   988   RegSetValue      HKLM\System\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-21-216923... SUCCESS   Type: REG_BINA...
12:56:19.2982376 PM @WanaDecryp...   988   RegCloseKey      HKLM\System\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-21-216923... SUCCESS
12:56:19.2982758 PM @WanaDecryp...   988   CloseFile        C:\Windows                                                                   SUCCESS
12:56:19.2983238 PM @WanaDecryp...   988   CloseFile        C:\Users\user\Desktop\CWMalwareSample                                        SUCCESS
12:56:19.2983587 PM @WanaDecryp...   988   RegCloseKey      HKLM\System\CurrentControlSet\Control\Nls\CustomLocale                       SUCCESS
12:56:19.2983790 PM @WanaDecryp...   988   RegCloseKey      HKLM\System\CurrentControlSet\Control\Session Manager                        SUCCESS
12:56:19.2983825 PM @WanaDecryp...   988   RegCloseKey      HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Opti...SUCCESS
12:56:19.2983858 PM @WanaDecryp...   988   RegCloseKey      HKLM\System\CurrentControlSet\Control\Nls\Sorting\Versions                   SUCCESS
12:56:19.2983911 PM @WanaDecryp...   988   CloseFile        C:\Windows\WinSxS\x86_microsoft.windows.common-controls_6595b64144ccf1df_6....SUCCESS
12:56:19.2984319 PM @WanaDecryp...   988   RegCloseKey      HKLM                                                                         SUCCESS
12:56:19.2984413 PM @WanaDecryp...   988   CloseFile        C:\Windows\System32\en-US\MFC42.dll.mui                                      SUCCESS
12:56:22.8545197 PM @WanaDecryp...   5428  ReadFile         C:\Windows\SysWOW64\uxtheme.dll                                              SUCCESS          Offset: 340,992, Le...
12:56:22.8553122 PM @WanaDecryp...   5428  ReadFile         C:\Windows\SysWOW64\uxtheme.dll                                              SUCCESS          Offset: 381,952, Le...
12:56:22.8558229 PM @WanaDecryp...   5428  ReadFile         C:\Windows\SysWOW64\uxtheme.dll                                              SUCCESS          Offset: 300,032, Le...
12:56:22.8562539 PM @WanaDecryp...   5428  ReadFile         C:\Windows\SysWOW64\uxtheme.dll                                              SUCCESS          Offset: 279,552, Le...
12:56:23.2790054 PM @WanaDecryp...   5428  ReadFile         C:\Windows\WinSxS\x86_microsoft.windows.common-controls_6595b64144ccf1df_6....SUCCESS          Offset: 558,080, Le...
12:56:49.2691006 PM @WanaDecryp...   4504  Process Start                                                                                 SUCCESS          Parent PID: 3556, ...
12:56:49.2691172 PM @WanaDecryp...   4504  Thread Create                                                                                 SUCCESS          Thread ID: 2696
12:56:49.2858315 PM @WanaDecryp...   4504  Load Image       C:\Users\user\Desktop\CWMalwareSample\@WanaDecryptor@.exe                    SUCCESS          Image Base: 0x400...
12:56:49.2858854 PM @WanaDecryp...   4504  Load Image       C:\Windows\System32\ntdll.dll                                                SUCCESS          Image Base: 0x7ffc...
12:56:49.2859426 PM @WanaDecryp...   4504  Load Image       C:\Windows\SysWOW64\ntdll.dll                                                SUCCESS          Image Base: 0x776...
12:56:49.2862624 PM @WanaDecryp...   4504  CreateFile       C:\Windows\Prefetch\@WANADECRYPTOR@.EXE-4D31736B.pf                          SUCCESS          Desired Access: G...
12:56:49.2863525 PM @WanaDecryp...   4504  QueryStandardI...C:\Windows\Prefetch\@WANADECRYPTOR@.EXE-4D31736B.pf                          SUCCESS          AllocationSize: 16,...
12:56:49.2864036 PM @WanaDecryp...   4504  ReadFile         C:\Windows\Prefetch\@WANADECRYPTOR@.EXE-4D31736B.pf                          SUCCESS          Offset: 0, Length: 1...
12:56:49.2871074 PM @WanaDecryp...   4504  CloseFile        C:\Windows\Prefetch\@WANADECRYPTOR@.EXE-4D31736B.pf                          SUCCESS
12:56:49.3103272 PM @WanaDecryp...   4504  RegOpenKey       HKLM\System\CurrentControlSet\Control\Session Manager                        REPARSE          Desired Access: Q...
12:56:49.3103584 PM @WanaDecryp...   4504  RegOpenKey       HKLM\System\CurrentControlSet\Control\Session Manager                        SUCCESS          Desired Access: Q...
12:56:49.3103893 PM @WanaDecryp...   4504  RegQueryValue    HKLM\System\CurrentControlSet\Control\Session Manager\RaiseExceptionOnPossibl... NAME NOT FOUND Length: 80
12:56:49.3104121 PM @WanaDecryp...   4504  RegCloseKey      HKLM\System\CurrentControlSet\Control\Session Manager                        SUCCESS
12:56:49.3104454 PM @WanaDecryp...   4504  RegOpenKey       HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Segment Heap           REPARSE          Desired Access: Q...
12:56:49.3104650 PM @WanaDecryp...   4504  RegOpenKey       HKLM\System\CurrentControlSet\Control\Session Manager\Segment Heap           NAME NOT FOUND   Desired Access: Q...
12:56:49.3105870 PM @WanaDecryp...   4504  RegOpenKey       HKLM\SYSTEM\CurrentControlSet\Control\Session Manager                        REPARSE          Desired Access: Q...
12:56:49.3106076 PM @WanaDecryp...   4504  RegOpenKey       HKLM\System\CurrentControlSet\Control\Session Manager                        SUCCESS          Desired Access: Q...
12:56:49.3106298 PM @WanaDecryp...   4504  RegQueryValue    HKLM\System\CurrentControlSet\Control\Session Manager\ResourcePolicies       NAME NOT FOUND   Length: 24
12:56:49.3106992 PM @WanaDecryp...   4504  RegCloseKey      HKLM\System\CurrentControlSet\Control\Session Manager                        SUCCESS
12:56:49.3114474 PM @WanaDecryp...   4504  CreateFile       C:\Windows                                                                   SUCCESS          Desired Access: E...
12:56:49.3116539 PM @WanaDecryp...   4504  Load Image       C:\Windows\System32\wow64.dll                                                SUCCESS          Image Base: 0x7ffc...
12:56:49.3118297 PM @WanaDecryp...   4504  Load Image       C:\Windows\System32\wow64win.dll                                             SUCCESS          Image Base: 0x7ffc...
12:56:49.3124641 PM @WanaDecryp...   4504  CreateFile       C:\Windows\System32\wow64log.dll                                             NAME NOT FOUND   Desired Access: R...
12:56:49.3128537 PM @WanaDecryp...   4504  CreateFile       C:\Windows                                                                   SUCCESS          Desired Access: R...
12:56:49.3129048 PM @WanaDecryp...   4504  QueryNameInfo... C:\Windows                                                                   SUCCESS          Name: \Windows
12:56:49.3129233 PM @WanaDecryp...   4504  CloseFile        C:\Windows                                                                   SUCCESS
12:56:49.3129813 PM @WanaDecryp...   4504  RegOpenKey       HKLM\Software\Microsoft\Wow64\x86                                            SUCCESS          Desired Access: R...
12:56:49.3130329 PM @WanaDecryp...   4504  RegQueryValue    HKLM\SOFTWARE\Microsoft\Wow64\x86\@WanaDecryptor@.exe                        NAME NOT FOUND   Length: 520
12:56:49.3130464 PM @WanaDecryp...   4504  RegQueryValue    HKLM\SOFTWARE\Microsoft\Wow64\x86\(Default)                                  SUCCESS          Type: REG_SZ, Le...
12:56:49.3130682 PM @WanaDecryp...   4504  RegCloseKey      HKLM\SOFTWARE\Microsoft\Wow64\x86                                            SUCCESS
```

Figure 28: malware process from Procmon.

Looking at the process tree it was also discovered that the malware was hiding behind the company name “Microsoft Corporation” to evade being detected. It was also noticed 2 new executables called taskhsvc.exe and Conhost.exe.



Process	Description	Image Path	Life T...	Company	Command	Start Time
Idle (0)		Idle				4/30/2024 9:43:59 AM
System (4)		System				4/30/2024 9:44:04 AM
csrss.exe (396)	Client Server Runtime Process	C:\Windows\system32\csrss.exe		Microsoft Corporation	%SystemRoot%\system32\csrss...	4/30/2024 9:44:06 AM
wininit.exe (480)	Windows Start-Up Application	C:\Windows\system32\wininit.exe		Microsoft Corporation	wininit.exe	4/30/2024 9:44:06 AM
csrss.exe (524)	Client Server Runtime Process	C:\Windows\system32\csrss.exe		Microsoft Corporation	%SystemRoot%\system32\csrss...	4/30/2024 9:44:06 AM
winlogon.exe (580)	Windows Logon Application	C:\Windows\system32\winlogon.exe		Microsoft Corporation	winlogon.exe	4/30/2024 9:44:06 AM
fontdrvhost.exe (776)	Usermode Font Driver Host	C:\Windows\system32\fontdrvhost.exe		Microsoft Corporation	"fontdrvhost.exe"	4/30/2024 9:44:07 AM
dwm.exe (996)	Desktop Window Manager	C:\Windows\system32\dwm.exe		Microsoft Corporation	"dwm.exe"	4/30/2024 9:44:07 AM
Explorer.EXE (4884)	Windows Explorer	C:\Windows\Explorer.EXE		Microsoft Corporation	C:\Windows\Explorer.EXE	4/30/2024 10:44:18 AM
SecurityHealthSystray.exe (3540)	Windows Security notification icon	C:\Windows\System32\SecurityHealthSyst...		Microsoft Corporation	"C:\Windows\System32\Securi...	4/30/2024 10:44:30 AM
vmtoolsd.exe (4028)	VMware Tools Core Service	C:\Program Files\VMware\VMware Tools\v...		VMware, Inc.	"C:\Program Files\VMware\VM...	4/30/2024 10:44:30 AM
Procmon.exe (4452)	Process Monitor	C:\Tools\sysinternals\Procmon.exe		Sysinternals - www.sy...	"C:\Tools\sysinternals\Procmon...	4/30/2024 1:15:12 PM
Procmon64.exe (1608)	Process Monitor	C:\Users\user\AppData\Local\Temp\Proc...		Sysinternals - www.sy...	"C:\Users\user\AppData\Local...	4/30/2024 1:15:13 PM
ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5ba	DiskPart	C:\Users\user\Desktop\CWMalwareSampl...		Microsoft Corporation	"C:\Users\user\Desktop\CWM...	4/30/2024 1:16:00 PM
@WanaDecryptor@.exe (3332)	Load PerfMon Counters	C:\Users\user\Desktop\CWMalwareSampl...		Microsoft Corporation	@WanaDecryptor@.exe co	4/30/2024 1:16:20 PM
taskhsvc.exe (2884)		C:\Users\user\Desktop\CWMalwareSampl...			TaskData\Tor\taskhsvc.exe	4/30/2024 1:16:21 PM
Conhost.exe (5972)	Console Window Host	C:\Windows\System32\Conhost.exe		Microsoft Corporation	\??\C:\Windows\system32\co...	4/30/2024 1:16:21 PM
cmd.exe (3104)	Windows Command Processor	C:\Windows\SysWOW64\cmd.exe		Microsoft Corporation	cmd.exe /c start /b @WanaDe...	4/30/2024 1:16:20 PM
@WanaDecryptor@.exe (3520)	Load PerfMon Counters	C:\Users\user\Desktop\CWMalwareSampl...		Microsoft Corporation	@WanaDecryptor@.exe vs	4/30/2024 1:16:20 PM
@WanaDecryptor@.exe (6896)	Load PerfMon Counters	C:\Users\user\Desktop\CWMalwareSampl...		Microsoft Corporation	@WanaDecryptor@.exe vs	4/30/2024 1:16:36 PM
@WanaDecryptor@.exe (416)	Load PerfMon Counters	C:\Users\user\Desktop\CWMalwareSampl...		Microsoft Corporation	@WanaDecryptor@.exe	4/30/2024 1:16:31 PM

Figure 29: Process tree, malware company name.

Using Process Explorer, the tester was able to see that as soon as the malware was executed the CPU usage skyrocketed as the initialisation of threads, folders and files were being created.

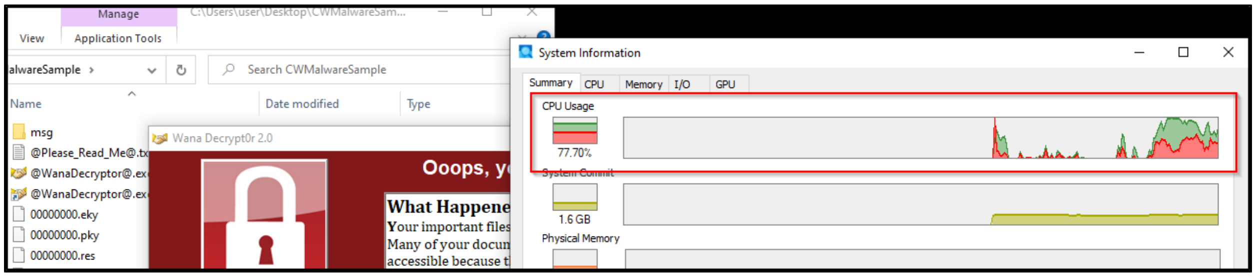
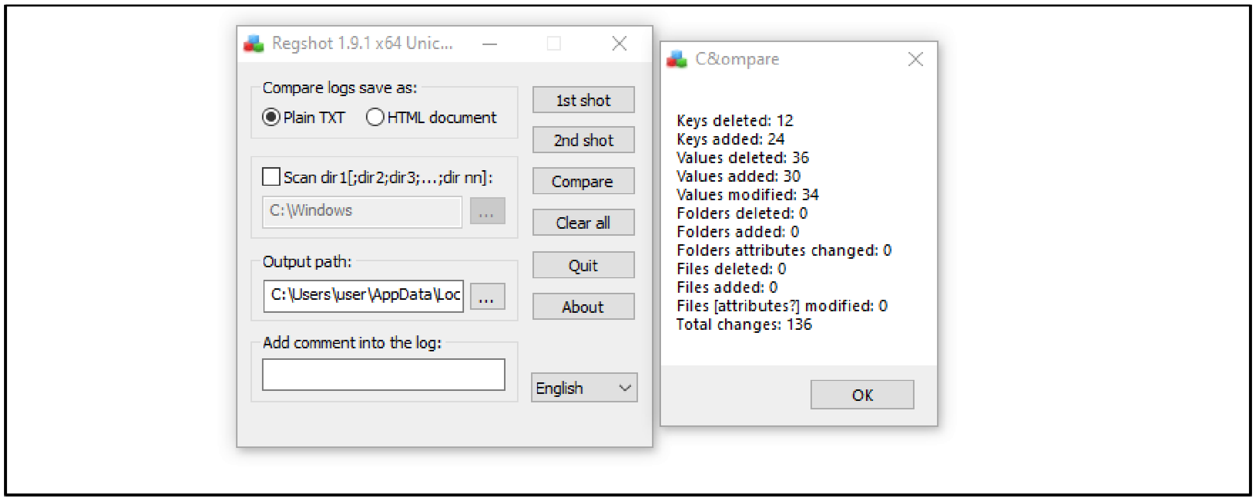


Figure 30: CPU Usage peaked.

Moreover, a registry analysis was performed with the tool Regshot. Comparing 2 shots of the system's registry keys taken before and after running the malware helped the analyst understand further what was going on. The malware made a total of 136 changes to the VM. 12 keys were deleted, 24 new keys were added, 36 values were deleted, 30 values were added, and 34 values were modified. This can be seen in the image below and more in detail in Appendix C.

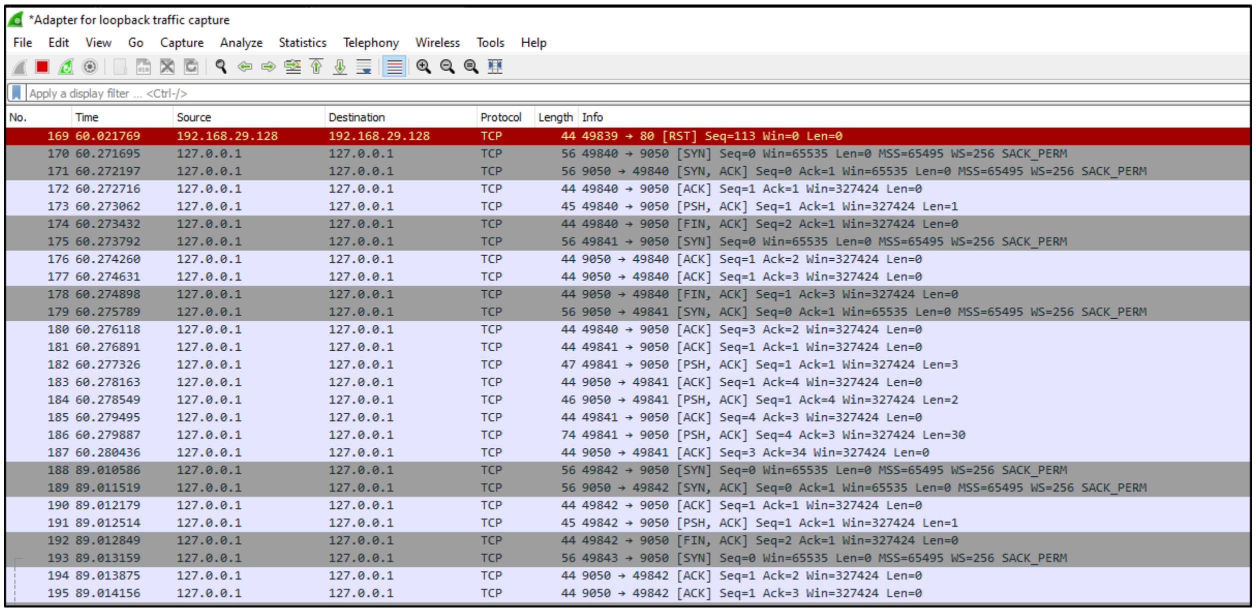


Plain TXT [✓] HTML document
Scan dir1[;dir2;dir3;...;dir nn]:
C:\Windows
Output path:
C:\Users\user\AppData\Loc
Add comment into the log:

```
Keys deleted: 12
Keys added: 24
Values deleted: 36
Values added: 30
Values modified: 34
Folders deleted: 0
Folders added: 0
Folders attributes changed: 0
Files deleted: 0
Files added: 0
Files [attributes?] modified: 0
Total changes: 136
```

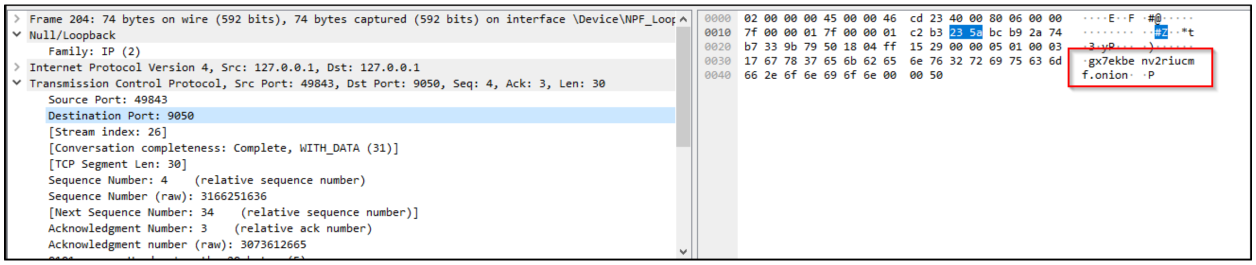
Figure 31: Regshot comparison 1st and 2nd shot.

During the network analysis, two tools were used - FakeNet and Wireshark. FakeNet is a network simulator that is effective in analysing malware activity during the execution process. Since there was no other traffic on the network, Wireshark was able to provide a comprehensive analysis of the traffic without any interference. Upon reviewing the traffic, it was observed that a lot of traffic was being directed to port 9050. By using the filter "tcp.port== 9050" and analysing the traffic, several URLs of web pages with the domain onion were attempted to access were discovered. This domain is used on the dark web. Therefore, it can be deduced that the malware was attempting to connect to a domain on the dark web. This information is presented in Figures 32 and 33.


*Adapter for loopback traffic capture
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.  Time          Source          Destination     Protocol Length Info
 169 60.021769     192.168.29.128  192.168.29.128  TCP      44 49839 → 80 [RST] Seq=113 Win=0 Len=0
 170 60.271695     127.0.0.1       127.0.0.1       TCP      56 49840 → 9050 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
 171 60.272197     127.0.0.1       127.0.0.1       TCP      56 9050 → 49840 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
 172 60.272716     127.0.0.1       127.0.0.1       TCP      44 49840 → 9050 [ACK] Seq=1 Ack=1 Win=327424 Len=0
 173 60.273062     127.0.0.1       127.0.0.1       TCP      45 49840 → 9050 [PSH, ACK] Seq=1 Ack=1 Win=327424 Len=1
 174 60.273432     127.0.0.1       127.0.0.1       TCP      44 49840 → 9050 [FIN, ACK] Seq=2 Ack=1 Win=327424 Len=0
 175 60.273792     127.0.0.1       127.0.0.1       TCP      56 49841 → 9050 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
 176 60.274260     127.0.0.1       127.0.0.1       TCP      44 9050 → 49840 [ACK] Seq=1 Ack=2 Win=327424 Len=0
 177 60.274631     127.0.0.1       127.0.0.1       TCP      44 9050 → 49840 [ACK] Seq=1 Ack=3 Win=327424 Len=0
 178 60.274898     127.0.0.1       127.0.0.1       TCP      44 9050 → 49840 [FIN, ACK] Seq=1 Ack=3 Win=327424 Len=0
 179 60.275789     127.0.0.1       127.0.0.1       TCP      56 9050 → 49841 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
 180 60.276118     127.0.0.1       127.0.0.1       TCP      44 49840 → 9050 [ACK] Seq=3 Ack=2 Win=327424 Len=0
 181 60.276891     127.0.0.1       127.0.0.1       TCP      44 49841 → 9050 [ACK] Seq=1 Ack=1 Win=327424 Len=0
 182 60.277326     127.0.0.1       127.0.0.1       TCP      47 49841 → 9050 [PSH, ACK] Seq=1 Ack=1 Win=327424 Len=3
 183 60.278163     127.0.0.1       127.0.0.1       TCP      44 9050 → 49841 [ACK] Seq=1 Ack=4 Win=327424 Len=0
 184 60.278549     127.0.0.1       127.0.0.1       TCP      46 9050 → 49841 [PSH, ACK] Seq=1 Ack=4 Win=327424 Len=2
 185 60.279495     127.0.0.1       127.0.0.1       TCP      44 49841 → 9050 [ACK] Seq=4 Ack=3 Win=327424 Len=0
 186 60.279887     127.0.0.1       127.0.0.1       TCP      74 49841 → 9050 [PSH, ACK] Seq=4 Ack=3 Win=327424 Len=30
 187 60.280436     127.0.0.1       127.0.0.1       TCP      44 9050 → 49841 [ACK] Seq=3 Ack=34 Win=327424 Len=0
 188 89.010586     127.0.0.1       127.0.0.1       TCP      56 49842 → 9050 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
 189 89.011519     127.0.0.1       127.0.0.1       TCP      56 9050 → 49842 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
 190 89.012179     127.0.0.1       127.0.0.1       TCP      44 49842 → 9050 [ACK] Seq=1 Ack=1 Win=327424 Len=0
 191 89.012514     127.0.0.1       127.0.0.1       TCP      45 49842 → 9050 [PSH, ACK] Seq=1 Ack=1 Win=327424 Len=1
 192 89.012849     127.0.0.1       127.0.0.1       TCP      44 49842 → 9050 [FIN, ACK] Seq=2 Ack=1 Win=327424 Len=0
 193 89.013159     127.0.0.1       127.0.0.1       TCP      56 49843 → 9050 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
 194 89.013875     127.0.0.1       127.0.0.1       TCP      44 9050 → 49842 [ACK] Seq=1 Ack=2 Win=327424 Len=0
 195 89.014156     127.0.0.1       127.0.0.1       TCP      44 9050 → 49842 [ACK] Seq=1 Ack=3 Win=327424 Len=0
```

Figure 32: Wireshark traffic.



```
> Frame 204: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_Loop
v Null/Loopback
    Family: IP (2)
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
v Transmission Control Protocol, Src Port: 49843, Dst Port: 9050, Seq: 4, Ack: 3, Len: 30
    Source Port: 49843
    Destination Port: 9050
    [Stream index: 26]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 30]
    Sequence Number: 4    (relative sequence number)
    Sequence Number (raw): 3166251636
    [Next Sequence Number: 34    (relative sequence number)]
    Acknowledgment Number: 3    (relative ack number)
    Acknowledgment number (raw): 3073612665
```

```
0000  02 00 00 00 45 00 00 46  cd 23 40 00 80 06 00 00   ····E··F ·#@·····
0010  7f 00 00 01 7f 00 00 01  c2 b3 23 5a bc b9 2a 74   ········ ··#Z··*t
0020  b7 33 9b 79 50 18 04 ff  15 29 00 00 05 01 00 03   ·3·yP··· ·)······
0030  17 67 78 37 65 6b 62 65  6e 76 32 72 69 75 63 6d   ·gx7ekbe nv2riucm
0040  66 2e 6f 6e 69 6f 6e 00  00 50                     f.onion· ·P
```

Figure 33: Discovery of .onion URL.

3.3 Conclusion

After examining the file in question, the analysis concluded that it was a type of malware known as WannaCry ransomware. To further verify the findings, the hash was entered into the VirusTotal database and found that other security vendors had also registered the file as malware. More analysis showed that the malware is designed to deceive the user by posing as a trusted Windows provider. It requests permission to delete all backups and ignore boot failures via the User Account control prompt. Furthermore, static, and dynamic analysis confirmed that the malware manipulated registries, files and resources as seen from the IDA pro disassembler, among others, and confirmed in the dynamic test. Once the files were encrypted the victim had 3 days before the ransom increases, and 7 days before the files were permanently deleted. In addition, network analysis was also conducted and discovered that the malware attempted to access several web pages belonging to the "onion" domain. This suggests that the malware developers use these pages to keep track of the infected systems and remotely control the decryption of files once the ransom is paid.

3.4 Countermeasures

For a company to protect from this type of specific malware, it is essential to have several updated backup copies. Moreover, companies should always have the most secure updated version of the system to minimise the number of susceptible vulnerabilities. Also, having an Incident Response (IR), Business Continuity (BC) and Disaster Recovery (DR) plan in place is essential to prepare for the worst.

Staff training should also be required, as humans can cause several malware exposures if they download or run executables that are not allowed on the network. It can also be considered the prohibition of BYOD policies (bring your own device), as devices may have been infected with malware which can expand through the network. Another idea for companies can be segmenting the network. Dividing the network into several zones with different permissions in place can help prevent and limit the spread of malware across the network.

In terms of individuals, simple actions like not connecting your devices to an untrusted network, refraining from downloading files from untrusted sources, and having an antivirus or some type of malicious software detector can be effective countermeasures.

3.5 Future Work

If the analyst had more time and resources, the malware would have undergone a more in-depth analysis with the disassembly. Tools like Ghidra, which are more complex, would have been used to achieve a better analysis result. Additionally, the analyst would have analysed the virus from a sandbox environment to gather more information about how it affects memory and other system processes. Furthermore, the analyst would have investigated whether the malware has any propagation capabilities, i.e., the ability to spread to other computers within the network. This would have been done using multiple virtual machines connected to each other to simulate a network environment and observe the virus's behaviour.

References

Group, I.D.M. (2024) *Los Ataques de ransomware casi se duplicaron en 2023*, *Vulnerabilidades IT Digital Security*. Available at: https://www.itdigitalsecurity.es/vulnerabilidades/2024/03/los-ataques-de-ransomware-casi-se-duplicaron-en-2023 (Accessed: 28 April 2024).

What is systematic approach to malware analysis (Sama)? (2023) YouTube. Available at: https://www.youtube.com/watch?v=r2qSO25wJBo (Accessed: 28 April 2024).

(No date) *Virustotal*. Available at: https://www.virustotal.com/gui/file/ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa/detection (Accessed: 28 April 2024).

Assembly - introduction (no date) Tutorialspoint. Available at: https://www.tutorialspoint.com/assembly_programming/assembly_introduction.htm (Accessed: 29 April 2024).

Hex rays (no date) Hex Rays - State-of-the-art binary code analysis solutions. Available at: https://hex-rays.com/products/ida/support/idadoc/index.shtml (Accessed: 29 April 2024).

Bermejo Higuera, J. et al. (2020) Systematic approach to malware analysis (SAMA), MDPI. Available at: https://www.mdpi.com/2076-3417/10/4/1360 (Accessed: 29 April 2024).

Gewarren (no date) *Identifying functions in dlls - .NET framework*, *.NET Framework | Microsoft Learn*. Available at: https://learn.microsoft.com/en-us/dotnet/framework/interop/identifying-functions-in-dlls (Accessed: 30 April 2024).

Fern&aacute;ndez, G. (2019) Reversing wannacry, LinkedIn. Available at: https://www.linkedin.com/pulse/reversing-wannacry-gerardo-fern%C3%A1ndez-navarrete/ (Accessed: 03 May 2024).

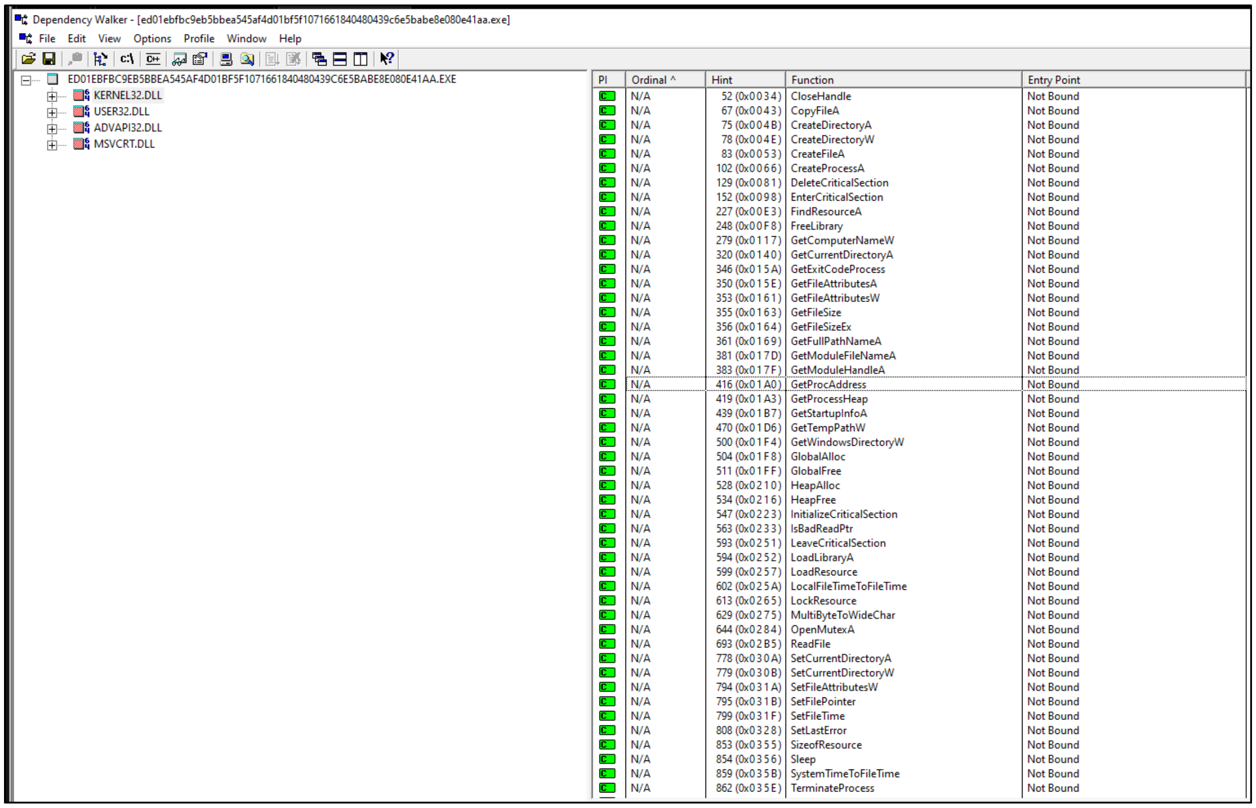
Ghidra: Primer Análisis Con La Herramienta de Ingeniería Inversa 'cortesía' De La National Security Agency (no date) *Ghidra: Primer análisis con la herramienta de Ingeniería Inversa 'cortesía' de la National Security Agency*. Available at: https://www.elladodelmal.com/2019/03/ghidra-primer-analisis-con-la.html (Accessed: 02 May 2024).

APPENDICES

APPENDIX A – FLOSS.EXE RESULT

```
--------------------------------
| FLOSS UTF-16LE STRINGS (194) |
----------------------
WanaCrypt0r
Software\
.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
.snt
.onetoc2
.dwg
.pdf
.wk1
.wks
.123
.rtf
.csv
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
.docb
.docx
.doc
%s\%s
%s\Intel
%s\ProgramData
VS_VERSION_INFO
StringFileInfo
040904B0
.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
CompanyName
Microsoft Corporation
FileDescription
DiskPart
FileVersion
6.1.7601.17514 (win7sp1_rtm.101119-1850)
InternalName
diskpart.exe
LegalCopyright
Microsoft Corporation. All rights reserved.
OriginalFilename
diskpart.exe
ProductName
Microsoft
Windows
Operating System
ProductVersion
6.1.7601.17514
VarFileInfo
Translation
----------------------
| FLOSS STACK STRINGS (1) |
----------------------
oftware\
----------------------
| FLOSS TIGHT STRINGS (0) |
----------------------
----------------------
| FLOSS DECODED STRINGS (0) |
```

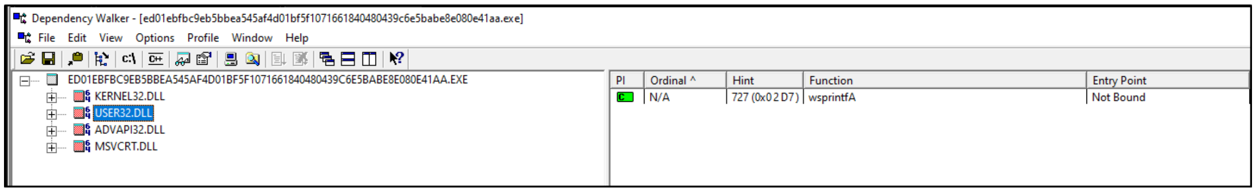
Appendix B – Dependency Walker (DLL functions)


Dependency Walker - [ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe]

```
ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA.EXE
    KERNEL32.DLL
    USER32.DLL
    ADVAPI32.DLL
    MSVCRT.DLL
```

PI	Ordinal ^	Hint	Function	Entry Point
🟩	N/A	52 (0x0034)	CloseHandle	Not Bound
🟩	N/A	67 (0x0043)	CopyFileA	Not Bound
🟩	N/A	75 (0x004B)	CreateDirectoryA	Not Bound
🟩	N/A	78 (0x004E)	CreateDirectoryW	Not Bound
🟩	N/A	83 (0x0053)	CreateFileA	Not Bound
🟩	N/A	102 (0x0066)	CreateProcessA	Not Bound
🟩	N/A	129 (0x0081)	DeleteCriticalSection	Not Bound
🟩	N/A	152 (0x0098)	EnterCriticalSection	Not Bound
🟩	N/A	227 (0x00E3)	FindResourceA	Not Bound
🟩	N/A	248 (0x00F8)	FreeLibrary	Not Bound
🟩	N/A	279 (0x0117)	GetComputerNameW	Not Bound
🟩	N/A	320 (0x0140)	GetCurrentDirectoryA	Not Bound
🟩	N/A	346 (0x015A)	GetExitCodeProcess	Not Bound
🟩	N/A	350 (0x015E)	GetFileAttributesA	Not Bound
🟩	N/A	353 (0x0161)	GetFileAttributesW	Not Bound
🟩	N/A	355 (0x0163)	GetFileSize	Not Bound
🟩	N/A	356 (0x0164)	GetFileSizeEx	Not Bound
🟩	N/A	361 (0x0169)	GetFullPathNameA	Not Bound
🟩	N/A	381 (0x017D)	GetModuleFileNameA	Not Bound
🟩	N/A	383 (0x017F)	GetModuleHandleA	Not Bound
🟩	N/A	416 (0x01A0)	GetProcAddress	Not Bound
🟩	N/A	419 (0x01A3)	GetProcessHeap	Not Bound
🟩	N/A	439 (0x01B7)	GetStartupInfoA	Not Bound
🟩	N/A	470 (0x01D6)	GetTempPathW	Not Bound
🟩	N/A	500 (0x01F4)	GetWindowsDirectoryW	Not Bound
🟩	N/A	504 (0x01F8)	GlobalAlloc	Not Bound
🟩	N/A	511 (0x01FF)	GlobalFree	Not Bound
🟩	N/A	528 (0x0210)	HeapAlloc	Not Bound
🟩	N/A	534 (0x0216)	HeapFree	Not Bound
🟩	N/A	547 (0x0223)	InitializeCriticalSection	Not Bound
🟩	N/A	563 (0x0233)	IsBadReadPtr	Not Bound
🟩	N/A	593 (0x0251)	LeaveCriticalSection	Not Bound
🟩	N/A	594 (0x0252)	LoadLibraryA	Not Bound
🟩	N/A	599 (0x0257)	LoadResource	Not Bound
🟩	N/A	602 (0x025A)	LocalFileTimeToFileTime	Not Bound
🟩	N/A	613 (0x0265)	LockResource	Not Bound
🟩	N/A	629 (0x0275)	MultiByteToWideChar	Not Bound
🟩	N/A	644 (0x0284)	OpenMutexA	Not Bound
🟩	N/A	693 (0x02B5)	ReadFile	Not Bound
🟩	N/A	778 (0x030A)	SetCurrentDirectoryA	Not Bound
🟩	N/A	779 (0x030B)	SetCurrentDirectoryW	Not Bound
🟩	N/A	794 (0x031A)	SetFileAttributesW	Not Bound
🟩	N/A	795 (0x031B)	SetFilePointer	Not Bound
🟩	N/A	799 (0x031F)	SetFileTime	Not Bound
🟩	N/A	808 (0x0328)	SetLastError	Not Bound
🟩	N/A	853 (0x0355)	SizeofResource	Not Bound
🟩	N/A	854 (0x0356)	Sleep	Not Bound
🟩	N/A	859 (0x035B)	SystemTimeToFileTime	Not Bound
🟩	N/A	862 (0x035E)	TerminateProcess	Not Bound

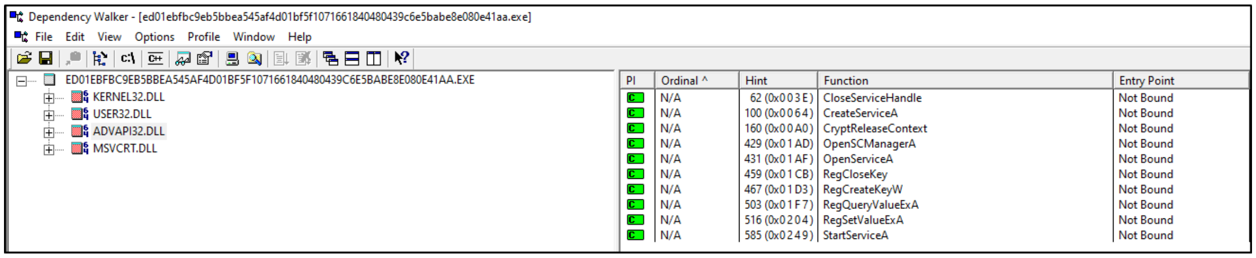
Figure B.1: KERNEL.dll functions


Dependency Walker - [ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe]

```
ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA.EXE
    KERNEL32.DLL
    USER32.DLL
    ADVAPI32.DLL
    MSVCRT.DLL
```

PI	Ordinal ^	Hint	Function	Entry Point
🟩	N/A	727 (0x02D7)	wsprintfA	Not Bound

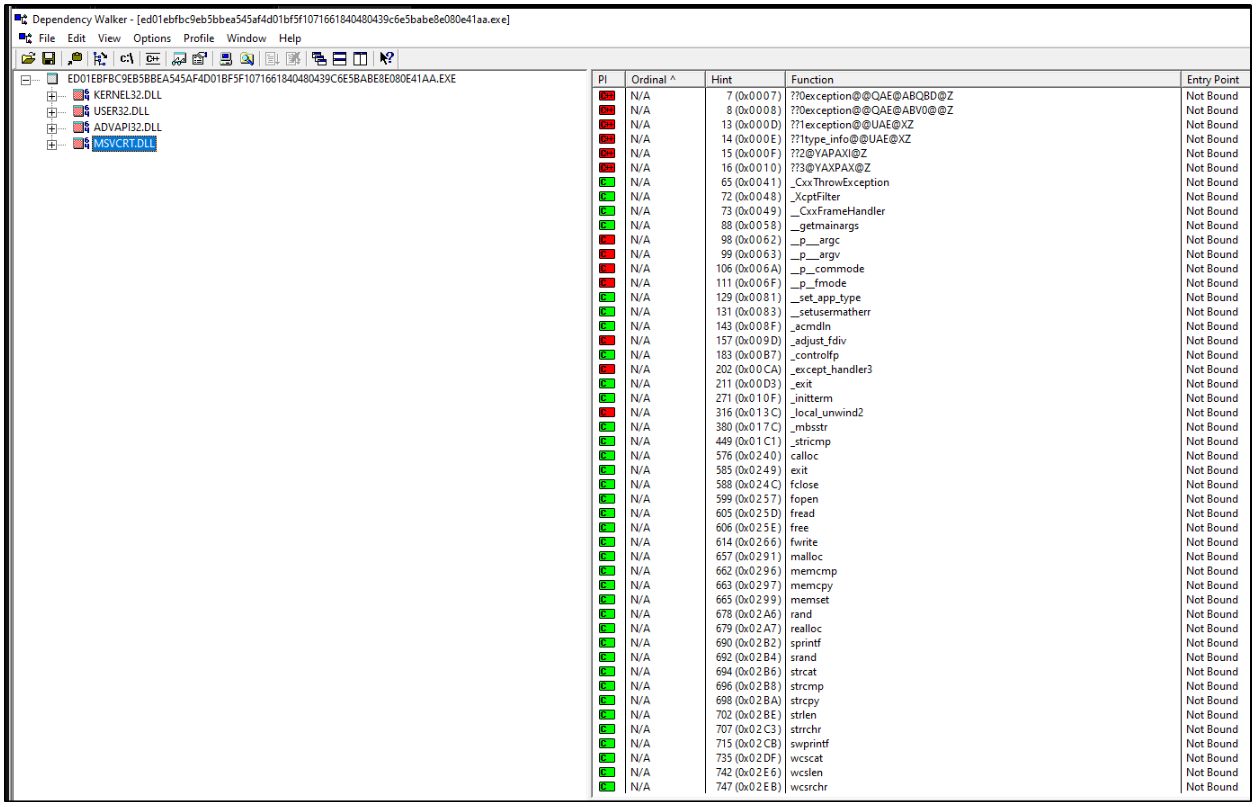
Figure B.2: USER.dll functions


Dependency Walker - [ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe]

```
ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA.EXE
    KERNEL32.DLL
    USER32.DLL
    ADVAPI32.DLL
    MSVCRT.DLL
```

PI	Ordinal ^	Hint	Function	Entry Point
🟩	N/A	62 (0x003E)	CloseServiceHandle	Not Bound
🟩	N/A	100 (0x0064)	CreateServiceA	Not Bound
🟩	N/A	160 (0x00A0)	CryptReleaseContext	Not Bound
🟩	N/A	429 (0x01AD)	OpenSCManagerA	Not Bound
🟩	N/A	431 (0x01AF)	OpenServiceA	Not Bound
🟩	N/A	459 (0x01CB)	RegCloseKey	Not Bound
🟩	N/A	467 (0x01D3)	RegCreateKeyW	Not Bound
🟩	N/A	503 (0x01F7)	RegQueryValueExA	Not Bound
🟩	N/A	516 (0x0204)	RegSetValueExA	Not Bound
🟩	N/A	585 (0x0249)	StartServiceA	Not Bound

Figure B.3: ADVAPI.dll functions.



Dependency Walker - [ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe]

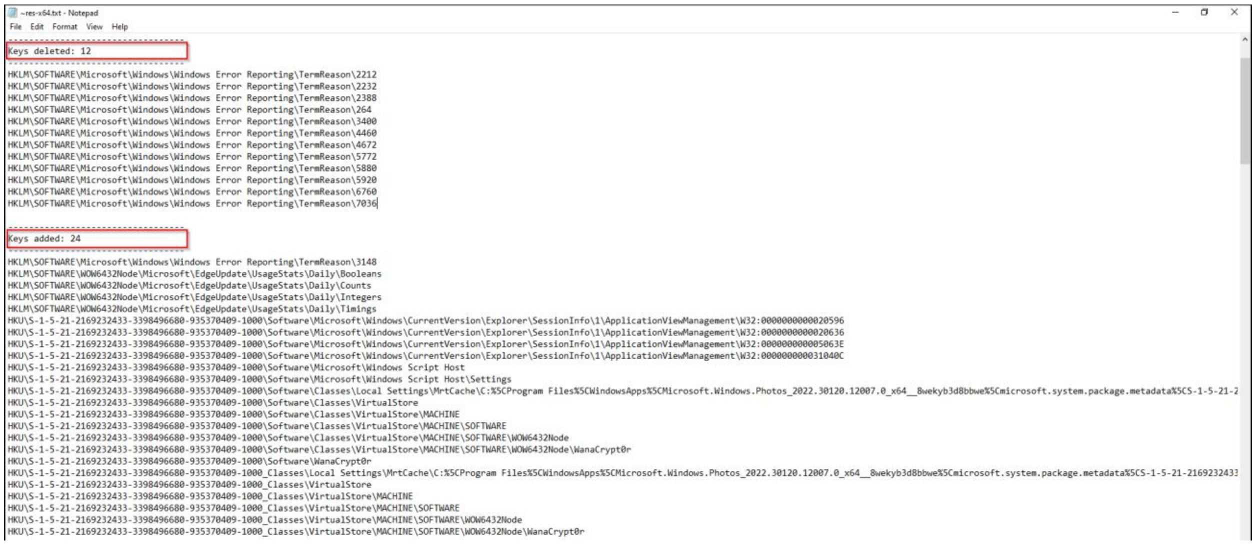
File Edit View Options Profile Window Help

```
ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA.EXE
    KERNEL32.DLL
    USER32.DLL
    ADVAPI32.DLL
    MSVCRT.DLL
```

PI	Ordinal ^	Hint	Function	Entry Point
🟥 C++	N/A	7 (0x0007)	??0exception@@QAE@ABQBD@Z	Not Bound
🟥 C++	N/A	8 (0x0008)	??0exception@@QAE@ABV0@@Z	Not Bound
🟥 C++	N/A	13 (0x000D)	??1exception@@UAE@XZ	Not Bound
🟥 C++	N/A	14 (0x000E)	??1type_info@@UAE@XZ	Not Bound
🟥 C++	N/A	15 (0x000F)	??2@YAPAXI@Z	Not Bound
🟥 C++	N/A	16 (0x0010)	??3@YAXPAX@Z	Not Bound
🟩 C	N/A	65 (0x0041)	_CxxThrowException	Not Bound
🟩 C	N/A	72 (0x0048)	_XcptFilter	Not Bound
🟩 C	N/A	73 (0x0049)	__CxxFrameHandler	Not Bound
🟩 C	N/A	88 (0x0058)	__getmainargs	Not Bound
🟥 C	N/A	98 (0x0062)	__p__argc	Not Bound
🟥 C	N/A	99 (0x0063)	__p__argv	Not Bound
🟥 C	N/A	106 (0x006A)	__p__commode	Not Bound
🟥 C	N/A	111 (0x006F)	__p__fmode	Not Bound
🟩 C	N/A	129 (0x0081)	__set_app_type	Not Bound
🟩 C	N/A	131 (0x0083)	__setusermatherr	Not Bound
🟩 C	N/A	143 (0x008F)	_acmdln	Not Bound
🟥 C	N/A	157 (0x009D)	_adjust_fdiv	Not Bound
🟩 C	N/A	183 (0x00B7)	_controlfp	Not Bound
🟥 C	N/A	202 (0x00CA)	_except_handler3	Not Bound
🟩 C	N/A	211 (0x00D3)	_exit	Not Bound
🟩 C	N/A	271 (0x010F)	_initterm	Not Bound
🟥 C	N/A	316 (0x013C)	_local_unwind2	Not Bound
🟩 C	N/A	380 (0x017C)	_mbsstr	Not Bound
🟩 C	N/A	449 (0x01C1)	_stricmp	Not Bound
🟩 C	N/A	576 (0x0240)	calloc	Not Bound
🟩 C	N/A	585 (0x0249)	exit	Not Bound
🟩 C	N/A	588 (0x024C)	fclose	Not Bound
🟩 C	N/A	599 (0x0257)	fopen	Not Bound
🟩 C	N/A	605 (0x025D)	fread	Not Bound
🟩 C	N/A	606 (0x025E)	free	Not Bound
🟩 C	N/A	614 (0x0266)	fwrite	Not Bound
🟩 C	N/A	657 (0x0291)	malloc	Not Bound
🟩 C	N/A	662 (0x0296)	memcmp	Not Bound
🟩 C	N/A	663 (0x0297)	memcpy	Not Bound
🟩 C	N/A	665 (0x0299)	memset	Not Bound
🟩 C	N/A	678 (0x02A6)	rand	Not Bound
🟩 C	N/A	679 (0x02A7)	realloc	Not Bound
🟩 C	N/A	690 (0x02B2)	sprintf	Not Bound
🟩 C	N/A	692 (0x02B4)	srand	Not Bound
🟩 C	N/A	694 (0x02B6)	strcat	Not Bound
🟩 C	N/A	696 (0x02B8)	strcmp	Not Bound
🟩 C	N/A	698 (0x02BA)	strcpy	Not Bound
🟩 C	N/A	702 (0x02BE)	strlen	Not Bound
🟩 C	N/A	707 (0x02C3)	strrchr	Not Bound
🟩 C	N/A	715 (0x02CB)	swprintf	Not Bound
🟩 C	N/A	735 (0x02DF)	wcscat	Not Bound
🟩 C	N/A	742 (0x02E6)	wcslen	Not Bound
🟩 C	N/A	747 (0x02EB)	wcsrchr	Not Bound

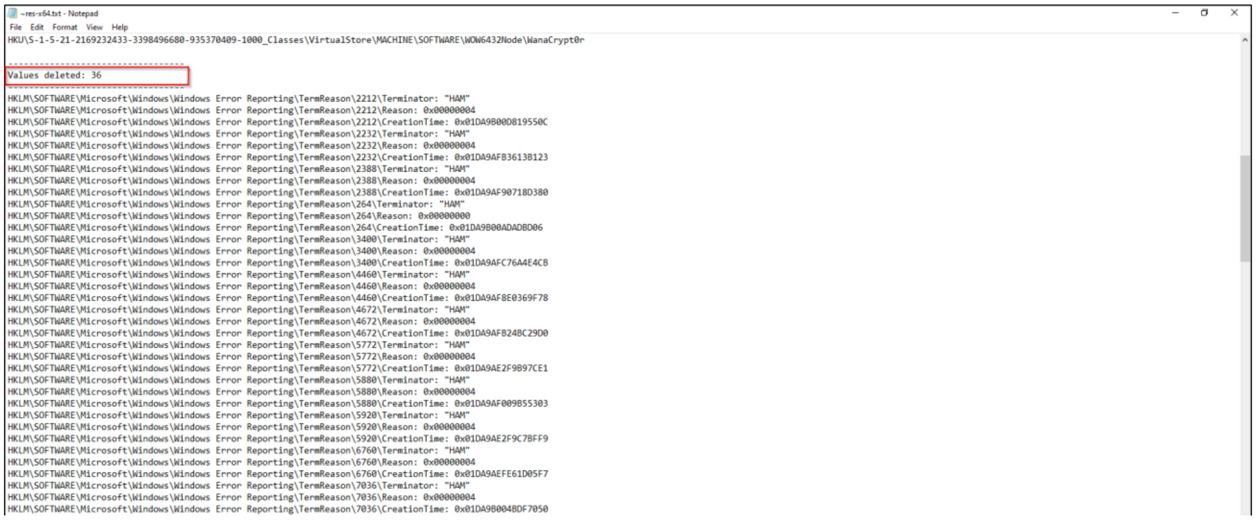
Figure B.4: MSVCRT.dll functions

APPENDIX C – REGSHOT OUTPUT



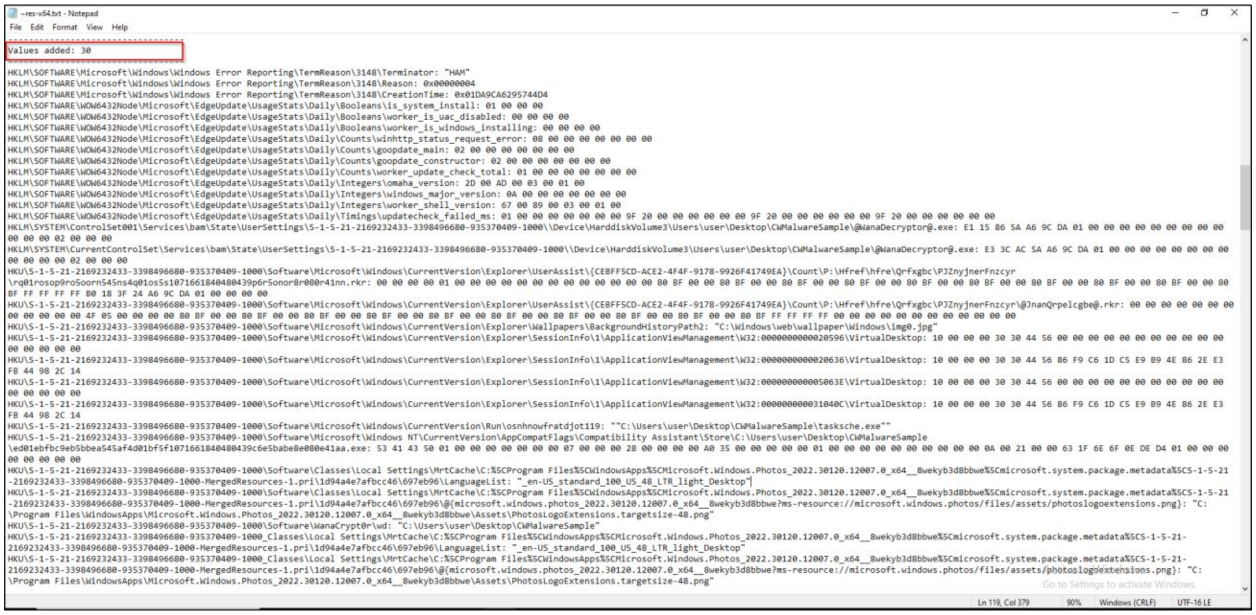
```
----------------------------------
Keys deleted: 12
----------------------------------
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\2212
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\2232
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\2388
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\264
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\3400
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\4460
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\4672
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\5772
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\5880
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\5920
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\6760
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\7036

----------------------------------
Keys added: 24
----------------------------------
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\3148
HKLM\SOFTWARE\WOW6432Node\Microsoft\EdgeUpdate\UsageStats\Daily\Booleans
HKLM\SOFTWARE\WOW6432Node\Microsoft\EdgeUpdate\UsageStats\Daily\Counts
HKLM\SOFTWARE\WOW6432Node\Microsoft\EdgeUpdate\UsageStats\Daily\Integers
HKLM\SOFTWARE\WOW6432Node\Microsoft\EdgeUpdate\UsageStats\Daily\Timings
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:0000000000020596
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:0000000000020636
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:000000000005063E
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:000000000031040C
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Microsoft\Windows Script Host
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Microsoft\Windows Script Host\Settings
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Classes\Local Settings\MrtCache\C:%5CProgram Files%5CWindowsApps%5CMicrosoft.Windows.Photos_2022.30120.12007.0_x64__8wekyb3d8bbwe%5Cmicrosoft.system.package.metadata%5CS-1-5-21-2
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Classes\VirtualStore
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Classes\VirtualStore\MACHINE
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Classes\VirtualStore\MACHINE\SOFTWARE
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Classes\VirtualStore\MACHINE\SOFTWARE\WOW6432Node
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Classes\VirtualStore\MACHINE\SOFTWARE\WOW6432Node\WanaCrypt0r
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\WanaCrypt0r
HKU\S-1-5-21-2169232433-3398496680-935370409-1000_Classes\Local Settings\MrtCache\C:%5CProgram Files%5CWindowsApps%5CMicrosoft.Windows.Photos_2022.30120.12007.0_x64__8wekyb3d8bbwe%5Cmicrosoft.system.package.metadata%5CS-1-5-21-2169232433
HKU\S-1-5-21-2169232433-3398496680-935370409-1000_Classes\VirtualStore
HKU\S-1-5-21-2169232433-3398496680-935370409-1000_Classes\VirtualStore\MACHINE
HKU\S-1-5-21-2169232433-3398496680-935370409-1000_Classes\VirtualStore\MACHINE\SOFTWARE
HKU\S-1-5-21-2169232433-3398496680-935370409-1000_Classes\VirtualStore\MACHINE\SOFTWARE\WOW6432Node
HKU\S-1-5-21-2169232433-3398496680-935370409-1000_Classes\VirtualStore\MACHINE\SOFTWARE\WOW6432Node\WanaCrypt0r
```

Figure C.1: key manipulation



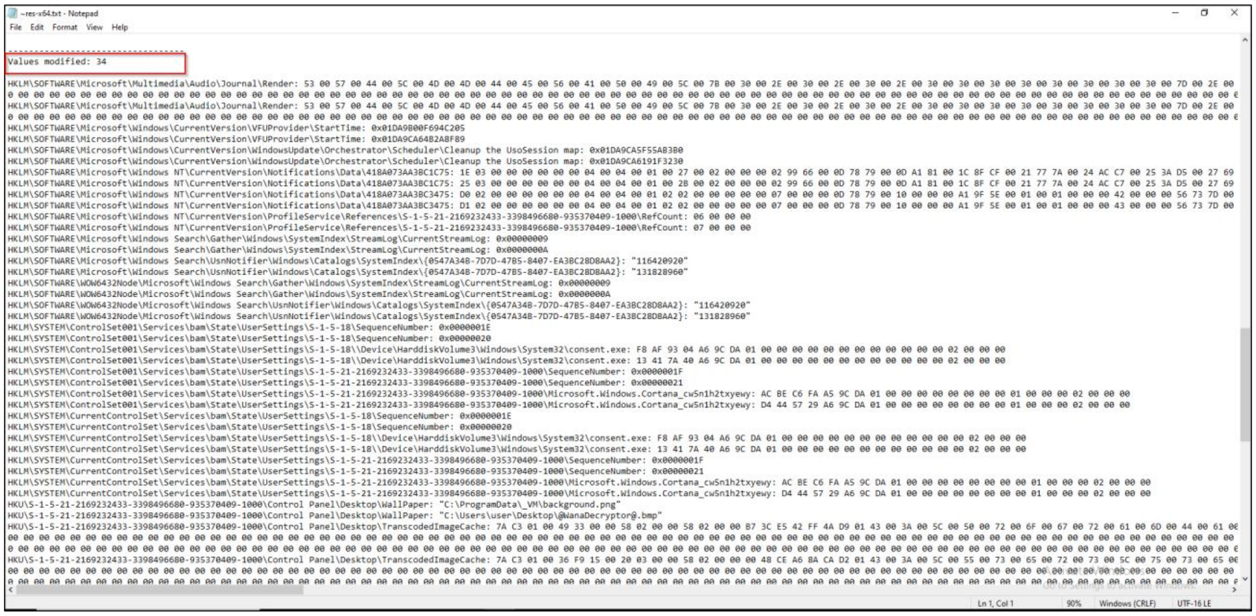
```
HKU\S-1-5-21-2169232433-3398496680-935370409-1000_Classes\VirtualStore\MACHINE\SOFTWARE\WOW6432Node\WanaCrypt0r

----------------------------------
Values deleted: 36
----------------------------------
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\2212\Terminator: "HAM"
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\2212\Reason: 0x00000004
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\2212\CreationTime: 0x01DA9B00D819550C
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\2232\Terminator: "HAM"
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\2232\Reason: 0x00000004
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\2232\CreationTime: 0x01DA9AFB3613B123
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\2388\Terminator: "HAM"
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\2388\Reason: 0x00000004
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\2388\CreationTime: 0x01DA9AF90718D380
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\264\Terminator: "HAM"
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\264\Reason: 0x00000000
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\264\CreationTime: 0x01DA9B00ADADBD06
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\3400\Terminator: "HAM"
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\3400\Reason: 0x00000004
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\3400\CreationTime: 0x01DA9AFC76A4E4CB
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\4460\Terminator: "HAM"
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\4460\Reason: 0x00000004
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\4460\CreationTime: 0x01DA9AF8E0369F78
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\4672\Terminator: "HAM"
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\4672\Reason: 0x00000004
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\4672\CreationTime: 0x01DA9AFB24BC29D0
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\5772\Terminator: "HAM"
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\5772\Reason: 0x00000004
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\5772\CreationTime: 0x01DA9AE2F9B97CE1
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\5880\Terminator: "HAM"
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\5880\Reason: 0x00000004
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\5880\CreationTime: 0x01DA9AF009B55303
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\5920\Terminator: "HAM"
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\5920\Reason: 0x00000004
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\5920\CreationTime: 0x01DA9AE2F9C7BFF9
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\6760\Terminator: "HAM"
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\6760\Reason: 0x00000004
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\6760\CreationTime: 0x01DA9AEFE61D05F7
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\7036\Terminator: "HAM"
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\7036\Reason: 0x00000004
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\7036\CreationTime: 0x01DA9B004BDF7050
```

Figure C.2: Value Manipulation 1



```
~res-x64.txt - Notepad
File Edit Format View Help
----------------------------------
Values added: 30

HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\3148\Terminator: "HAM"
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\3148\Reason: 0x00000004
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\3148\CreationTime: 0x01DA9CA6295744D4
HKLM\SOFTWARE\WOW6432Node\Microsoft\EdgeUpdate\UsageStats\Daily\Booleans\is_system_install: 01 00 00 00
HKLM\SOFTWARE\WOW6432Node\Microsoft\EdgeUpdate\UsageStats\Daily\Booleans\worker_is_uac_disabled: 00 00 00 00
HKLM\SOFTWARE\WOW6432Node\Microsoft\EdgeUpdate\UsageStats\Daily\Booleans\worker_is_windows_installing: 00 00 00 00
HKLM\SOFTWARE\WOW6432Node\Microsoft\EdgeUpdate\UsageStats\Daily\Counts\winhttp_status_request_error: 08 00 00 00 00 00 00 00
HKLM\SOFTWARE\WOW6432Node\Microsoft\EdgeUpdate\UsageStats\Daily\Counts\goopdate_main: 02 00 00 00 00 00 00 00
HKLM\SOFTWARE\WOW6432Node\Microsoft\EdgeUpdate\UsageStats\Daily\Counts\goopdate_constructor: 02 00 00 00 00 00 00 00
HKLM\SOFTWARE\WOW6432Node\Microsoft\EdgeUpdate\UsageStats\Daily\Counts\worker_update_check_total: 01 00 00 00 00 00 00 00
HKLM\SOFTWARE\WOW6432Node\Microsoft\EdgeUpdate\UsageStats\Daily\Integers\omaha_version: 2D 00 AD 00 03 00 01 00
HKLM\SOFTWARE\WOW6432Node\Microsoft\EdgeUpdate\UsageStats\Daily\Integers\windows_major_version: 0A 00 00 00 00 00 00 00
HKLM\SOFTWARE\WOW6432Node\Microsoft\EdgeUpdate\UsageStats\Daily\Integers\worker_shell_version: 67 00 89 00 03 00 01 00
HKLM\SOFTWARE\WOW6432Node\Microsoft\EdgeUpdate\UsageStats\Daily\Timings\updatecheck_failed_ms: 01 00 00 00 00 00 00 00 9F 20 00 00 00 00 00 00 9F 20 00 00 00 00 00 00 9F 20 00 00 00 00 00 00
HKLM\SYSTEM\ControlSet001\Services\bam\State\UserSettings\S-1-5-21-2169232433-3398496680-935370409-1000\\Device\HarddiskVolume3\Users\user\Desktop\CWMalwareSample\@WanaDecryptor@.exe: E1 15 86 5A A6 9C DA 01 00 00 00 00 00 00 00 00
00 00 00 02 00 00 00
HKLM\SYSTEM\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-21-2169232433-3398496680-935370409-1000\\Device\HarddiskVolume3\Users\user\Desktop\CWMalwareSample\@WanaDecryptor@.exe: E3 3C AC 5A A6 9C DA 01 00 00 00 00 00 00 00 00
00 00 00 00 02 00 00 00
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count\P:\Hfref\hfre\Qrfxgbc\PJZnyjnerFnzcyr
\rq01rosop9ro5oorn545ns4q01os5s1071661840480439p6r5onor8r080r41nn.rkr: 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80
BF FF FF FF FF B0 18 3F 24 A6 9C DA 01 00 00 00 00
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count\P:\Hfref\hfre\Qrfxgbc\PJZnyjnerFnzcyr\@JnanQrpelcgbe@.rkr: 00 00 00 00 00 00 00
00 00 00 00 00 4F 05 00 00 00 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\Wallpapers\BackgroundHistoryPath2: "C:\Windows\web\wallpaper\Windows\img0.jpg"
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:0000000000020596\VirtualDesktop: 10 00 00 00 30 30 44 56 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:0000000000020636\VirtualDesktop: 10 00 00 00 30 30 44 56 86 F9 C6 1D C5 E9 B9 4E 86 2E E3
FB 44 98 2C 14
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:000000000005063E\VirtualDesktop: 10 00 00 00 30 30 44 56 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:000000000031040C\VirtualDesktop: 10 00 00 00 30 30 44 56 86 F9 C6 1D C5 E9 B9 4E 86 2E E3
FB 44 98 2C 14
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Microsoft\Windows\CurrentVersion\Run\osnhnowfratdjot119: ""C:\Users\user\Desktop\CWMalwareSample\tasksche.exe""
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Compatibility Assistant\Store\C:\Users\user\Desktop\CWMalwareSample
\ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe: 53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00 00 00 00 A0 35 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 0A 00 21 00 00 63 1F 6E 6F 0E DE D4 01 00 00 00
00 00 00 00 00
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Classes\Local Settings\MrtCache\C:%5CProgram Files%5CWindowsApps%5CMicrosoft.Windows.Photos_2022.30120.12007.0_x64__8wekyb3d8bbwe%5Cmicrosoft.system.package.metadata%5CS-1-5-21
-2169232433-3398496680-935370409-1000-MergedResources-1.pri\1d94a4e7afbcc46\697eb96\LanguageList: "_en-US_standard_100_US_48_LTR_light_Desktop"
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\Classes\Local Settings\MrtCache\C:%5CProgram Files%5CWindowsApps%5CMicrosoft.Windows.Photos_2022.30120.12007.0_x64__8wekyb3d8bbwe%5Cmicrosoft.system.package.metadata%5CS-1-5-21
-2169232433-3398496680-935370409-1000-MergedResources-1.pri\1d94a4e7afbcc46\697eb96\@{microsoft.windows.photos_2022.30120.12007.0_x64__8wekyb3d8bbwe?ms-resource://microsoft.windows.photos/files/assets/photoslogoextensions.png}: "C:
\Program Files\WindowsApps\Microsoft.Windows.Photos_2022.30120.12007.0_x64__8wekyb3d8bbwe\Assets\PhotosLogoExtensions.targetsize-48.png"
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Software\WanaCrypt0r\wd: "C:\Users\user\Desktop\CWMalwareSample"
HKU\S-1-5-21-2169232433-3398496680-935370409-1000_Classes\Local Settings\MrtCache\C:%5CProgram Files%5CWindowsApps%5CMicrosoft.Windows.Photos_2022.30120.12007.0_x64__8wekyb3d8bbwe%5Cmicrosoft.system.package.metadata%5CS-1-5-21-
2169232433-3398496680-935370409-1000-MergedResources-1.pri\1d94a4e7afbcc46\697eb96\LanguageList: "_en-US_standard_100_US_48_LTR_light_Desktop"
HKU\S-1-5-21-2169232433-3398496680-935370409-1000_Classes\Local Settings\MrtCache\C:%5CProgram Files%5CWindowsApps%5CMicrosoft.Windows.Photos_2022.30120.12007.0_x64__8wekyb3d8bbwe%5Cmicrosoft.system.package.metadata%5CS-1-5-21-
2169232433-3398496680-935370409-1000-MergedResources-1.pri\1d94a4e7afbcc46\697eb96\@{microsoft.windows.photos_2022.30120.12007.0_x64__8wekyb3d8bbwe?ms-resource://microsoft.windows.photos/files/assets/photoslogoextensions.png}: "C:
\Program Files\WindowsApps\Microsoft.Windows.Photos_2022.30120.12007.0_x64__8wekyb3d8bbwe\Assets\PhotosLogoExtensions.targetsize-48.png"

Ln 119, Col 379    90%    Windows (CRLF)    UTF-16 LE
```

Figure C.3: Value manipulation 2



```
~res-x64.txt - Notepad
File Edit Format View Help
----------------------------------
Values modified: 34

HKLM\SOFTWARE\Microsoft\Multimedia\Audio\Journal\Render: 53 00 57 00 44 00 5C 00 4D 00 4D 00 44 00 45 00 56 00 41 00 50 00 49 00 5C 00 7B 00 30 00 2E 00 30 00 2E 00 30 00 2E 00 30 00 30 00 30 00 30 00 30 00 30 00 30 00 30 00 7D 00 2E 00
0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
HKLM\SOFTWARE\Microsoft\Multimedia\Audio\Journal\Render: 53 00 57 00 44 00 5C 00 4D 00 4D 00 44 00 45 00 56 00 41 00 50 00 49 00 5C 00 7B 00 30 00 2E 00 30 00 2E 00 30 00 2E 00 30 00 30 00 30 00 30 00 30 00 30 00 30 00 30 00 7D 00 2E 00
0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\VFUProvider\StartTime: 0x01DA9B00F694C205
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\VFUProvider\StartTime: 0x01DA9CA64B2A8F89
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Orchestrator\Scheduler\Cleanup the UsoSession map: 0x01DA9CA5F55AB3B0
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Orchestrator\Scheduler\Cleanup the UsoSession map: 0x01DA9CA6191F3230
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Notifications\Data\418A073AA3BC1C75: 1E 03 00 00 00 00 00 00 04 00 04 00 01 00 27 00 02 00 00 00 02 99 66 00 0D 78 79 00 0D A1 81 00 1C 8F CF 00 21 77 7A 00 24 AC C7 00 25 3A D5 00 27 69
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Notifications\Data\418A073AA3BC1C75: 25 03 00 00 00 00 00 00 04 00 04 00 01 00 2B 00 02 00 00 00 02 99 66 00 0D 78 79 00 0D A1 81 00 1C 8F CF 00 21 77 7A 00 24 AC C7 00 25 3A D5 00 27 69
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Notifications\Data\418A073AA3BC3475: D0 02 00 00 00 00 00 00 04 00 04 00 01 02 02 00 00 00 00 00 07 00 00 00 0D 78 79 00 10 00 00 00 A1 9F 5E 00 01 00 01 00 00 00 42 00 00 00 56 73 7D 00
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Notifications\Data\418A073AA3BC3475: D1 02 00 00 00 00 00 00 04 00 04 00 01 02 02 00 00 00 00 00 07 00 00 00 0D 78 79 00 10 00 00 00 A1 9F 5E 00 01 00 01 00 00 00 43 00 00 00 56 73 7D 00
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileService\References\S-1-5-21-2169232433-3398496680-935370409-1000\RefCount: 06 00 00 00
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileService\References\S-1-5-21-2169232433-3398496680-935370409-1000\RefCount: 07 00 00 00
HKLM\SOFTWARE\Microsoft\Windows Search\Gather\Windows\SystemIndex\StreamLog\CurrentStreamLog: 0x00000009
HKLM\SOFTWARE\Microsoft\Windows Search\Gather\Windows\SystemIndex\StreamLog\CurrentStreamLog: 0x0000000A
HKLM\SOFTWARE\Microsoft\Windows Search\UsnNotifier\Windows\Catalogs\SystemIndex\{0547A34B-7D7D-47B5-8407-EA3BC28D8AA2}: "116420920"
HKLM\SOFTWARE\Microsoft\Windows Search\UsnNotifier\Windows\Catalogs\SystemIndex\{0547A34B-7D7D-47B5-8407-EA3BC28D8AA2}: "131828960"
HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows Search\Gather\Windows\SystemIndex\StreamLog\CurrentStreamLog: 0x00000009
HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows Search\Gather\Windows\SystemIndex\StreamLog\CurrentStreamLog: 0x0000000A
HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows Search\UsnNotifier\Windows\Catalogs\SystemIndex\{0547A34B-7D7D-47B5-8407-EA3BC28D8AA2}: "116420920"
HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows Search\UsnNotifier\Windows\Catalogs\SystemIndex\{0547A34B-7D7D-47B5-8407-EA3BC28D8AA2}: "131828960"
HKLM\SYSTEM\ControlSet001\Services\bam\State\UserSettings\S-1-5-18\SequenceNumber: 0x0000001E
HKLM\SYSTEM\ControlSet001\Services\bam\State\UserSettings\S-1-5-18\SequenceNumber: 0x00000020
HKLM\SYSTEM\ControlSet001\Services\bam\State\UserSettings\S-1-5-18\\Device\HarddiskVolume3\Windows\System32\consent.exe: F8 AF 93 04 A6 9C DA 01 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00
HKLM\SYSTEM\ControlSet001\Services\bam\State\UserSettings\S-1-5-18\\Device\HarddiskVolume3\Windows\System32\consent.exe: 13 41 7A 40 A6 9C DA 01 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00
HKLM\SYSTEM\ControlSet001\Services\bam\State\UserSettings\S-1-5-21-2169232433-3398496680-935370409-1000\SequenceNumber: 0x0000001F
HKLM\SYSTEM\ControlSet001\Services\bam\State\UserSettings\S-1-5-21-2169232433-3398496680-935370409-1000\SequenceNumber: 0x00000021
HKLM\SYSTEM\ControlSet001\Services\bam\State\UserSettings\S-1-5-21-2169232433-3398496680-935370409-1000\Microsoft.Windows.Cortana_cw5n1h2txyewy: AC BE C6 FA A5 9C DA 01 00 00 00 00 00 00 00 00 01 00 00 00 02 00 00 00
HKLM\SYSTEM\ControlSet001\Services\bam\State\UserSettings\S-1-5-21-2169232433-3398496680-935370409-1000\Microsoft.Windows.Cortana_cw5n1h2txyewy: D4 44 57 29 A6 9C DA 01 00 00 00 00 00 00 00 00 01 00 00 00 02 00 00 00
HKLM\SYSTEM\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-18\SequenceNumber: 0x0000001E
HKLM\SYSTEM\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-18\SequenceNumber: 0x00000020
HKLM\SYSTEM\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-18\\Device\HarddiskVolume3\Windows\System32\consent.exe: F8 AF 93 04 A6 9C DA 01 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00
HKLM\SYSTEM\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-18\\Device\HarddiskVolume3\Windows\System32\consent.exe: 13 41 7A 40 A6 9C DA 01 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00
HKLM\SYSTEM\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-21-2169232433-3398496680-935370409-1000\SequenceNumber: 0x0000001F
HKLM\SYSTEM\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-21-2169232433-3398496680-935370409-1000\SequenceNumber: 0x00000021
HKLM\SYSTEM\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-21-2169232433-3398496680-935370409-1000\Microsoft.Windows.Cortana_cw5n1h2txyewy: AC BE C6 FA A5 9C DA 01 00 00 00 00 00 00 00 00 01 00 00 00 02 00 00 00
HKLM\SYSTEM\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-21-2169232433-3398496680-935370409-1000\Microsoft.Windows.Cortana_cw5n1h2txyewy: D4 44 57 29 A6 9C DA 01 00 00 00 00 00 00 00 00 01 00 00 00 02 00 00 00
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Control Panel\Desktop\WallPaper: "C:\ProgramData\_VM\background.png"
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Control Panel\Desktop\WallPaper: "C:\Users\user\Desktop\@WanaDecryptor@.bmp"
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Control Panel\Desktop\TranscodedImageCache: 7A C3 01 00 49 33 00 00 58 02 00 00 58 02 00 00 B7 3C E5 42 FF 4A D9 01 43 00 3A 00 5C 00 50 00 72 00 6F 00 67 00 72 00 61 00 6D 00 44 00 61 0
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
HKU\S-1-5-21-2169232433-3398496680-935370409-1000\Control Panel\Desktop\TranscodedImageCache: 7A C3 01 00 36 F9 15 00 20 03 00 00 58 02 00 00 00 48 CE A6 8A CA D2 01 43 00 3A 00 5C 00 55 00 73 00 65 00 72 00 73 00 5C 00 75 00 73 00 65 0
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0

Ln 1, Col 1    90%    Windows (CRLF)    UTF-16 LE
```

Figure C.4: Value manipulation 3