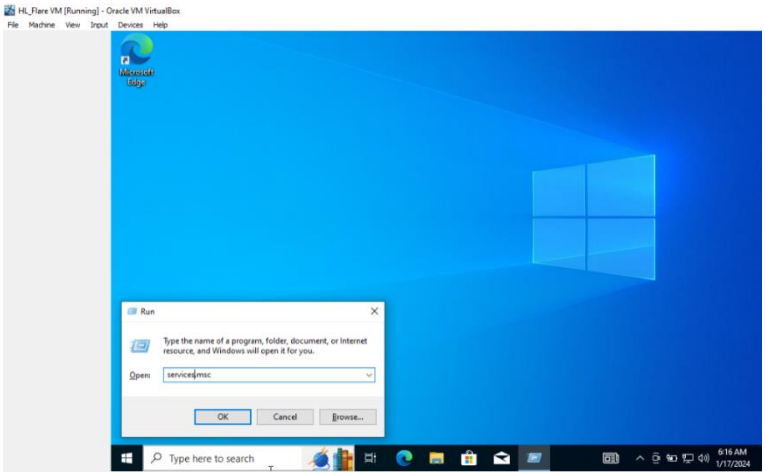
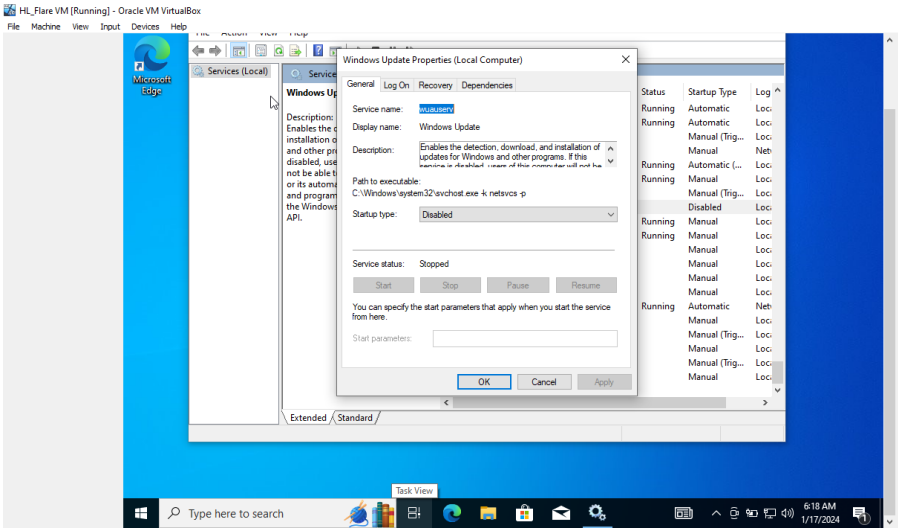
Static Malware Analysis and Disassembly with WannaCry Ransomware

Preparation for creating a sandbox virtual environment

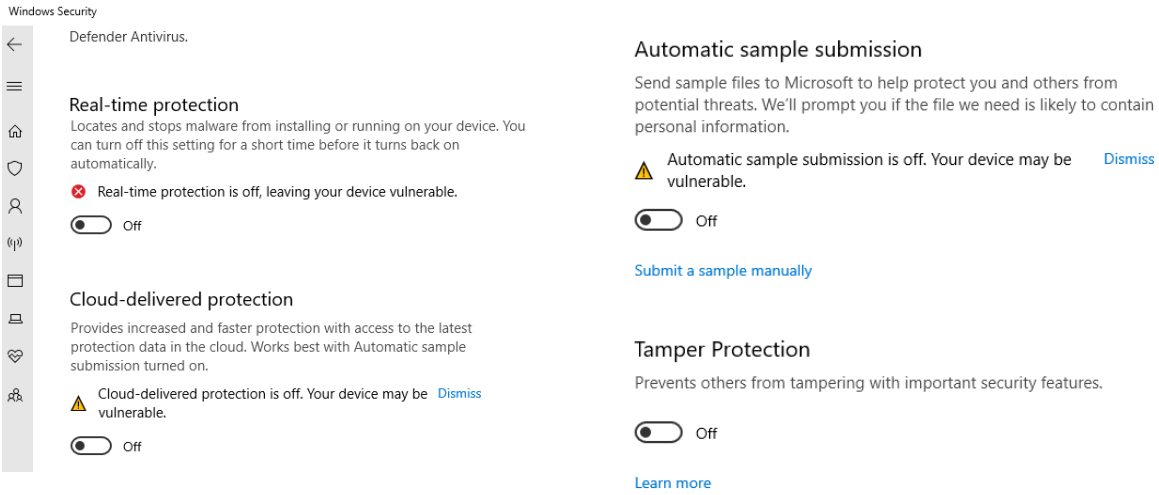
1. The first stage after installing the windows operating system on the virtualbox, some settings will be made by opening services.msc in the Run feature.



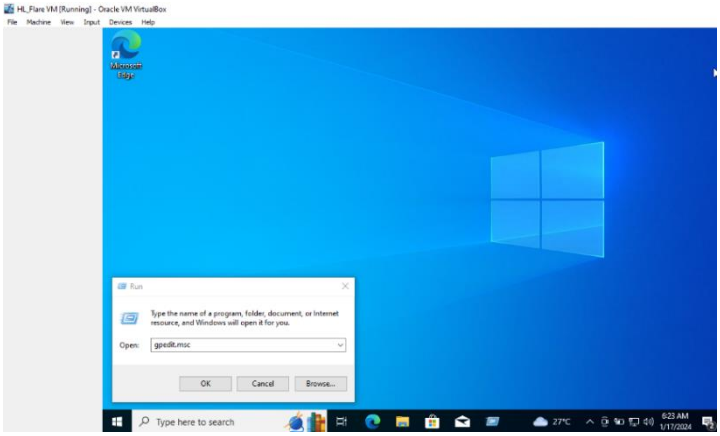
2. Then set the status type to Disabled and the services status to stopped on windows update. Then click ok.

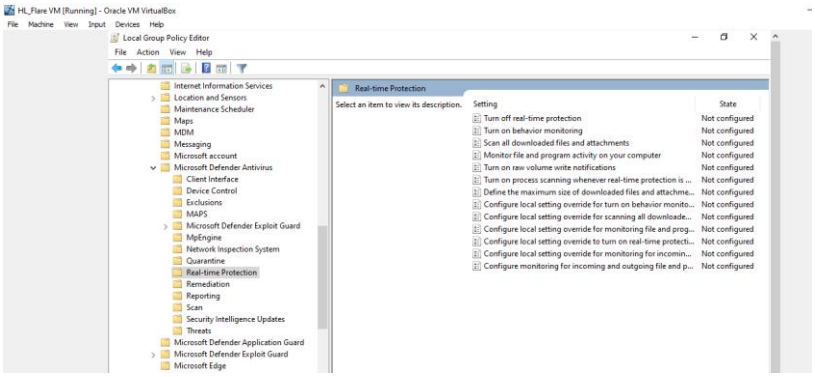


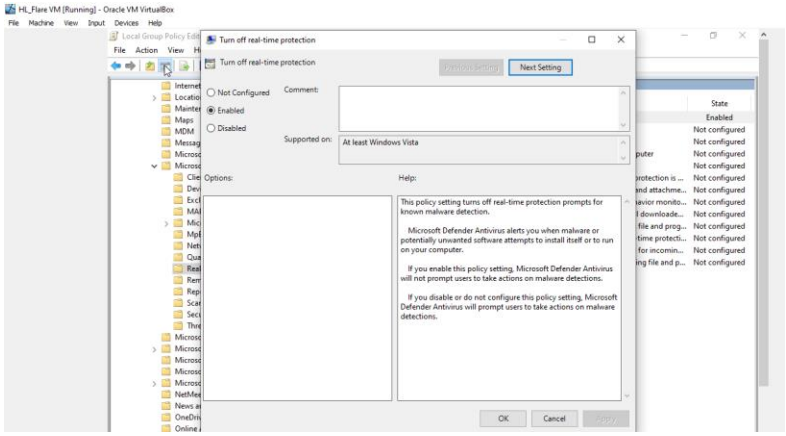
3. Next, do not activate the antivirus defender on windows security as shown below.



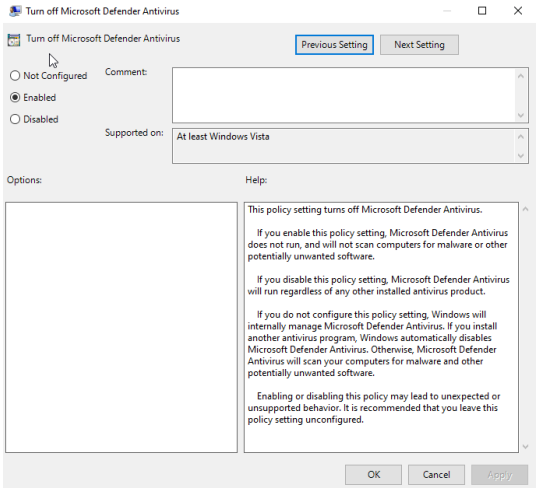
4. In the Run feature open gpedit.msc which refers to Real time Protection and select enabled on Turn off real time protection.



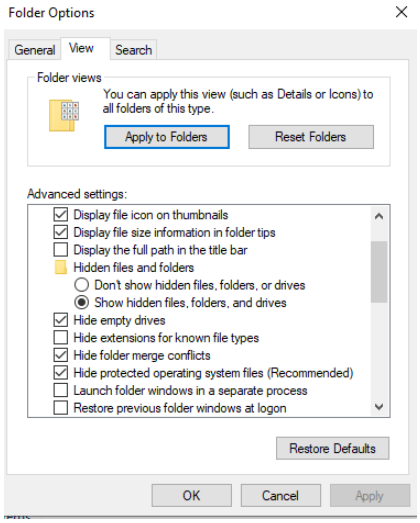




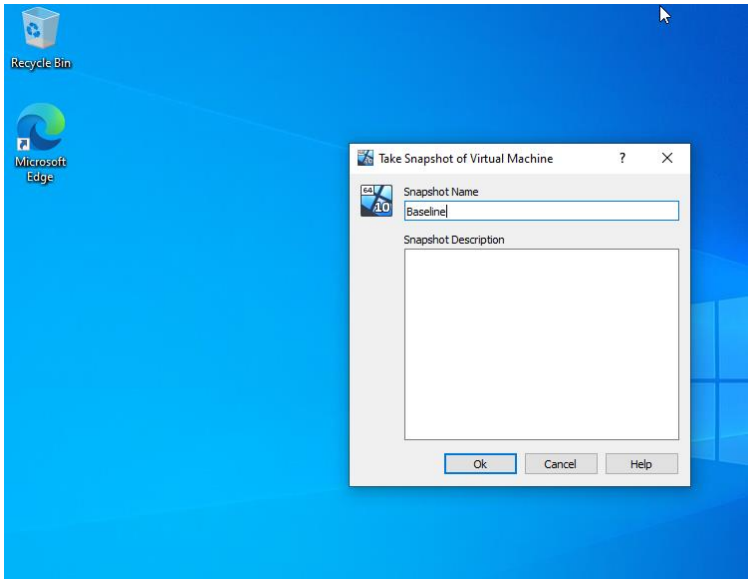
5. In the Turn off Microsoft Defender Antivirus section, select Enabled.



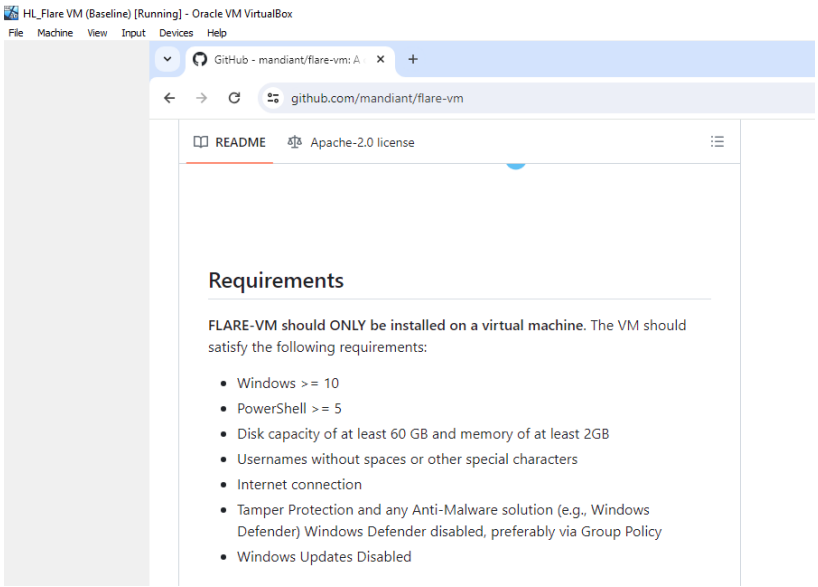
6. In the Turn off Microsoft Defender Antivirus section, select Enabled.



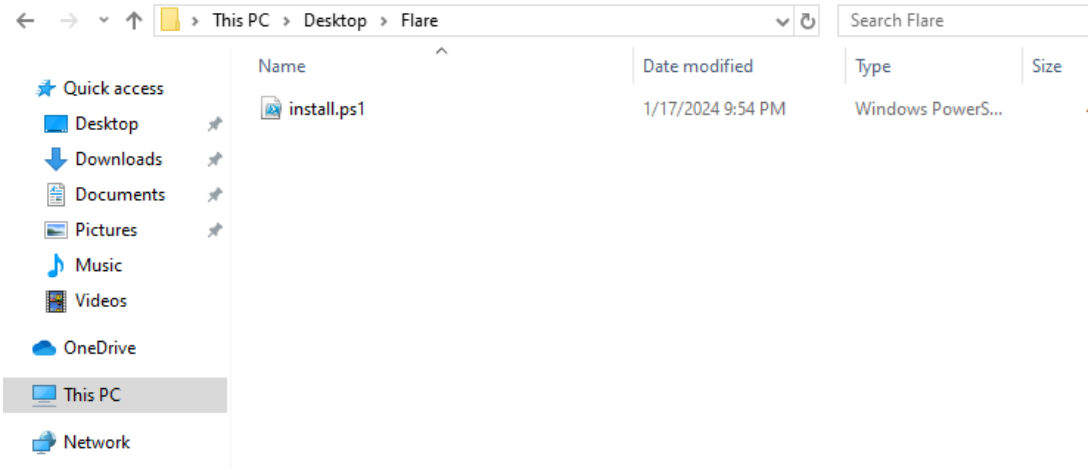
7. After making these settings. We take a snapshot so that it is stored in the virtual machine.



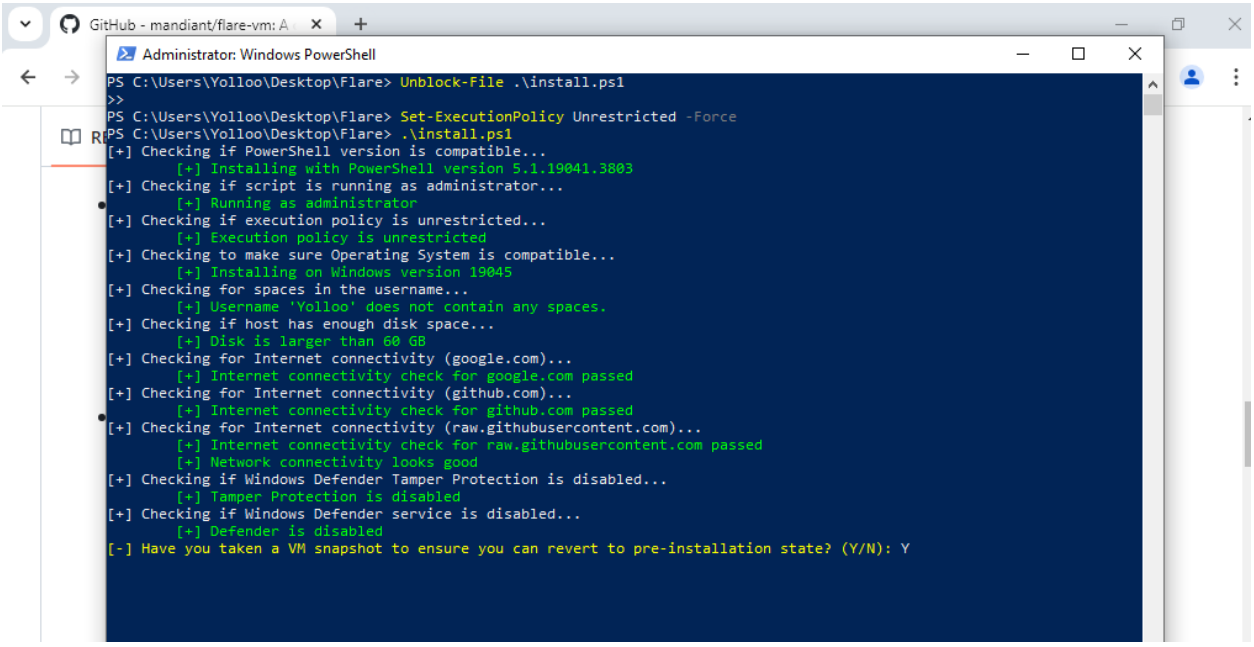
8. On google we will open the flare VM GitHub to install on the windows operating system by following some requirements as follows.



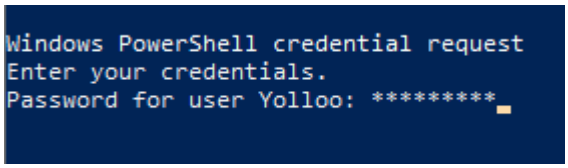
9. Download install.ps1 from flare VM GitHub



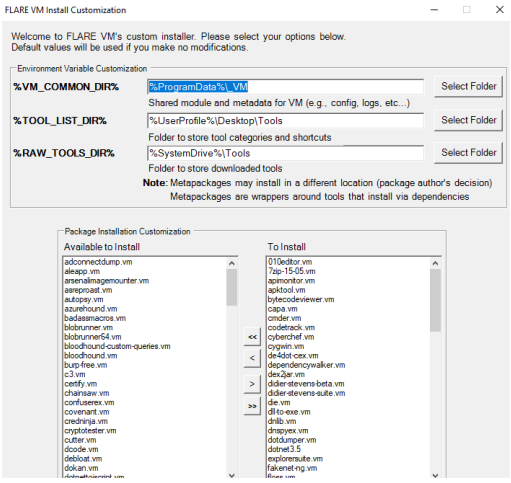
10. In PowerShell windows enter the following command to install the flare VM.



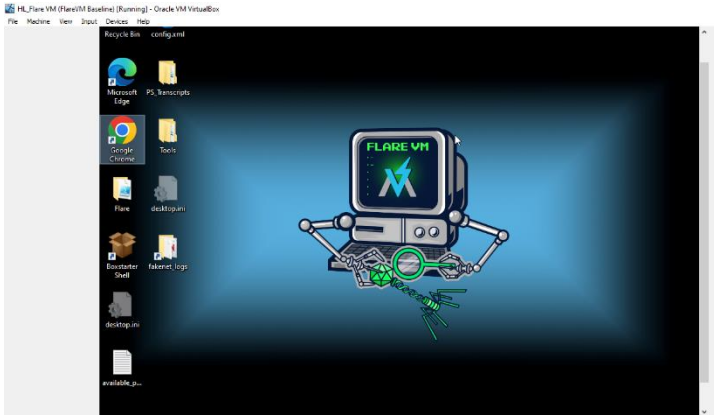
11. Enter the user password to login to the flare VM.



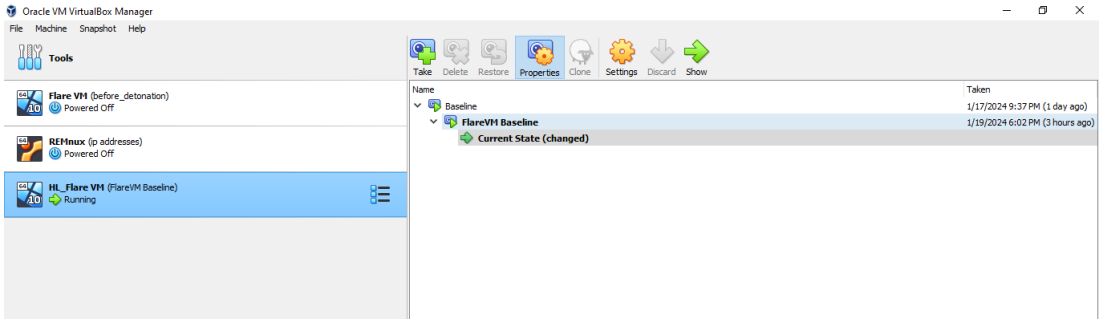
12. During the install process, Flare VM custom will appear and click ok.



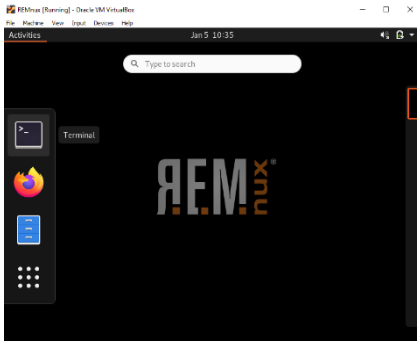
13. After the installation process is complete, the desktop will appear as follows.



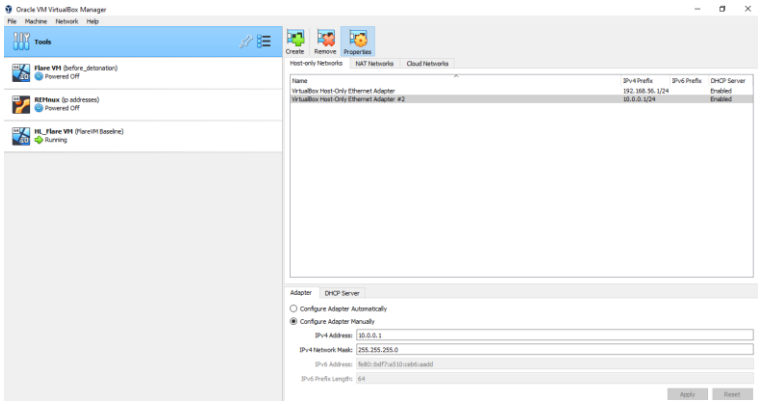
14. Take a snapshot to save the VM flare.



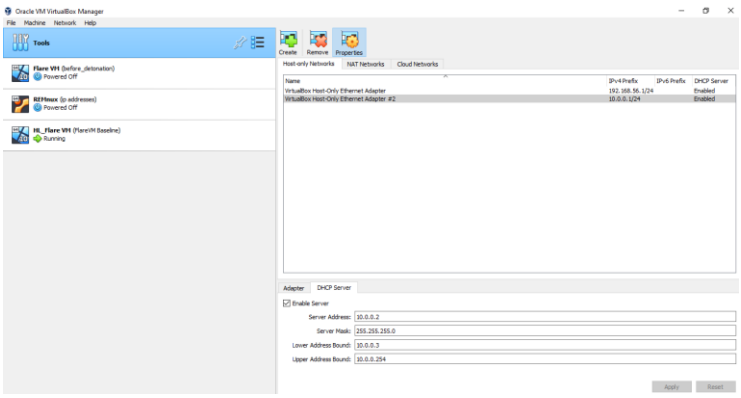
15. After downloading Remnux.ova Install Remnux on the VirtualBox.



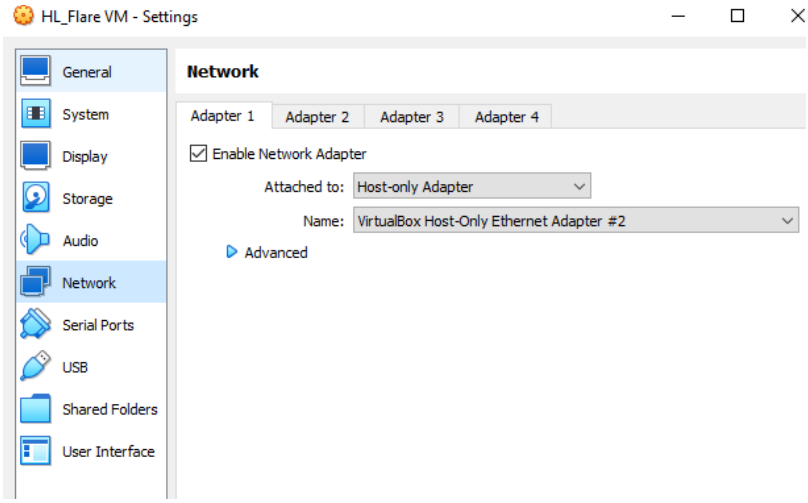
16. In Network VirtualBox add VirtualBox host only ethernet adapter 2. Then in the adapter section select configure adapter manually and enter the IP address.



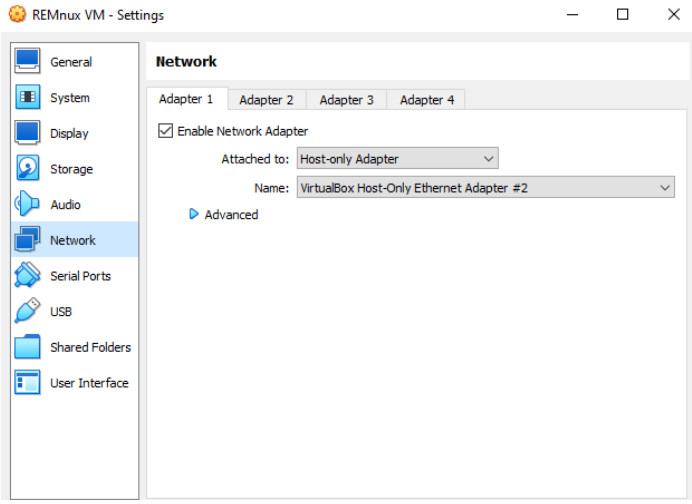
17. On the DHCP server enter the IP Address



18. On the flare VM network adapter select host only adapter with name referring to ethernet adapter 2.



19. On the Remnux VM network adapter select host only adapter and name.



20. In Remnux, check the IP address

```
remnux@remnux:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s17: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 100
0
    link/ether 08:00:27:11:97:67 brd ff:ff:ff:ff:ff:ff
```

21. Enter the following commands for network management.

```
remnux@remnux:~$ sudo vim /etc/netplan/01-netcfg.yaml
```

22. Set dhcp4 to false and addresses to 10.0.0.3/24.

```
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s17:
      dhcp4: false
      addresses: [10.0.0.3/24]
~
~
~
~
~
~
~
~
~
```

23. After making the settings. Enter the sudo netplan apply command and check the IP address again.

```
remnux@remnux:~$ sudo netplan apply
remnux@remnux:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s17: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group de
fault qlen 1000
    link/ether 08:00:27:11:97:67 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.3/24 brd 10.0.0.255 scope global enp0s17
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe11:9767/64 scope link
       valid_lft forever preferred_lft forever
```

24. Use inetsim to create a controlled environment for malware analysis.

```
remnux@remnux:~$ sudo vim /etc/inetsim/inetsim.conf
```

25. Enable start service as follows.

REMnux VM [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
# dns, http, smtp, pop3, tftp, ftp, ntp, time_tcp,
# time_udp, daytime_tcp, daytime_udp, echo_tcp,
# echo_udp, discard_tcp, discard_udp, quotd_tcp,
# quotd_udp, chargen_tcp, chargen_udp, finger,
# ident, syslog, dummy_tcp, dummy_udp, smtps, pop3s,
# ftps, irc, https
#
start_service dns
start_service http
start_service https
start_service smtp
start_service smtps
start_service pop3
start_service pop3s
start_service ftp
start_service ftps
#start_service tftp
#start_service irc
-- INSERT --                                                    19,
```

26. Set the service bind address to 0.0.0.0.

REMnux VM [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
#start_service chargen_udp
#start_service dummy_tcp
#start_service dummy_udp


##########################################
# service_bind_address
#
# IP address to bind services to
#
# Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
service_bind_address    0.0.0.0
```

27. Set the default dns IP to 10.0.0.3.

REMnux VM [Running] - Oracle VM VirtualBox

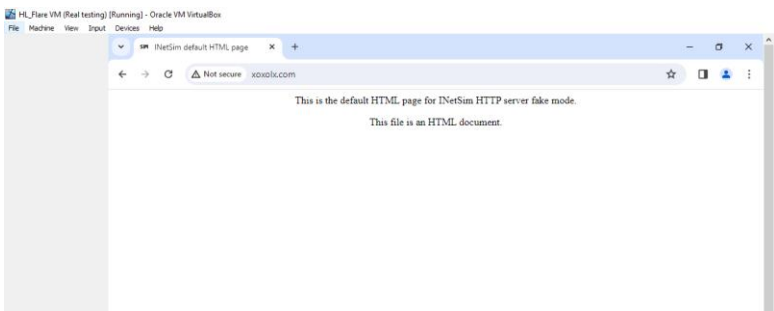
File Machine View Input Devices Help

```
##########################################
# dns_default_ip
#
# Default IP address to return with DNS replies
#
# Syntax: dns_default_ip <IP address>
#
# Default: 127.0.0.1
#
dns_default_ip          10.0.0.3
```

28. Run inetsim on the Remnux VM.

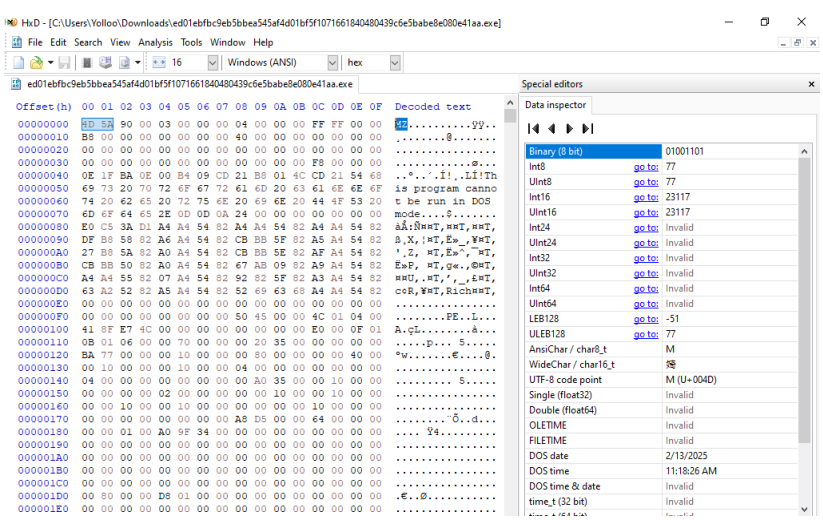
```
remnux@remnux:~$ inetsim
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory:      /var/log/inetsim/
Using data directory:     /var/lib/inetsim/
Using report directory:   /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
=== INetSim main process started (PID 1516) ===
Session ID:     1516
Listening on:   10.0.0.3
Real Date/Time: 2024-01-22 05:28:33
Fake Date/Time: 2024-01-22 05:28:33 (Delta: 0 seconds)
 Forking services...
  * dns_53_tcp_udp - started (PID 1520)
  * smtp_25_tcp - started (PID 1523)
  * smtps_465_tcp - started (PID 1524)
  * ftp_21_tcp - started (PID 1527)
  * ftps_990_tcp - started (PID 1528)
  * https_443_tcp - started (PID 1522)
  * pop3_110_tcp - started (PID 1525)
  * pop3s_995_tcp - started (PID 1526)
  * http_80_tcp - started (PID 1521)
 done.
Simulation running.
```

29. Then open google on flare VM and enter the domain, it will display the default inetsim which indicates that it is connected to the Remnux VM.

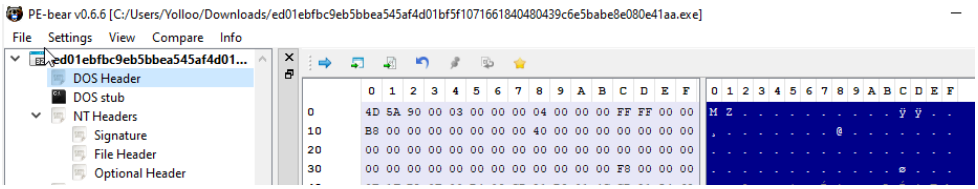


Malware static analysis

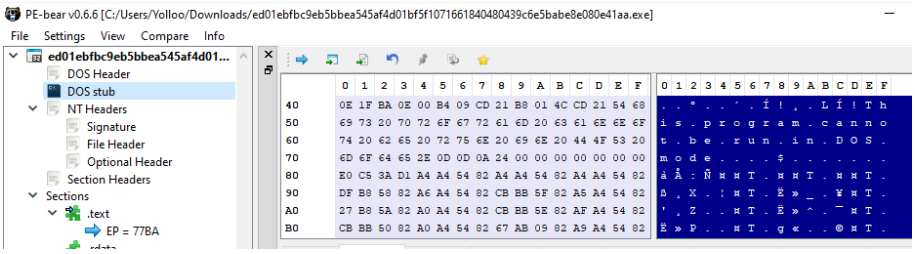
1. Determined PE file type based on hex signatures of 4D 5A with decoded text MZ. This indicates a DOS MZ executable.



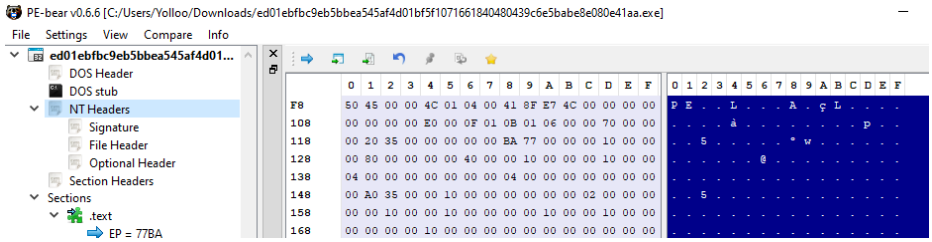
2. Using the PE bear tool which can also be used to find out the DOS Header.



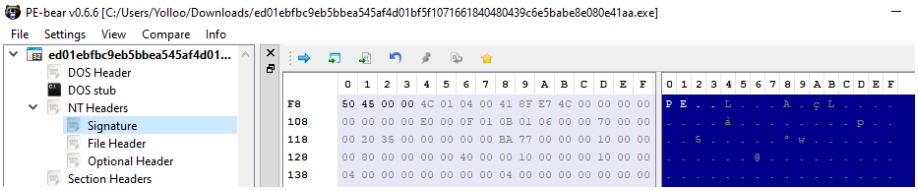
3. DOS stub is an MS-DOS program that prints an error message saying that the executable is not compatible with DOS then exits.



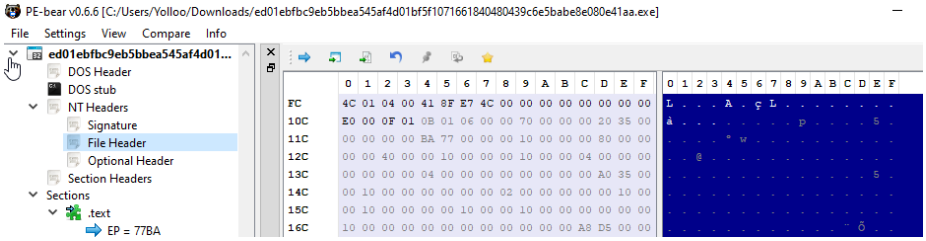
4. The NT header consists of a signature, a file header, and an optional header.



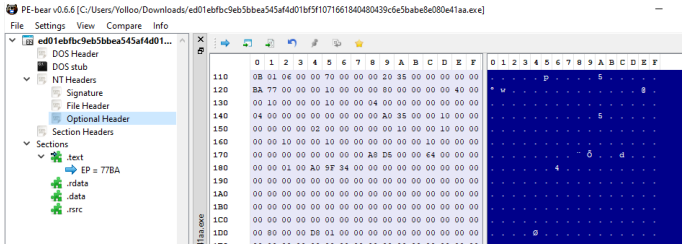
5. PE signature is a DWORD (4-bytes) that identifies the file as a PE image.



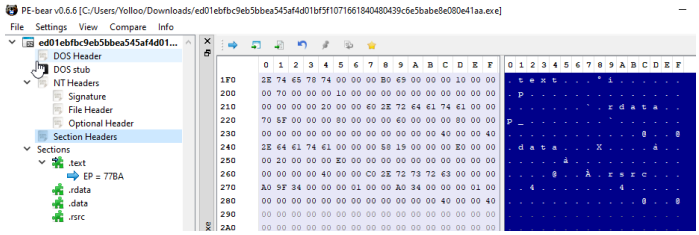
6. File Header contains important information about the PE file.



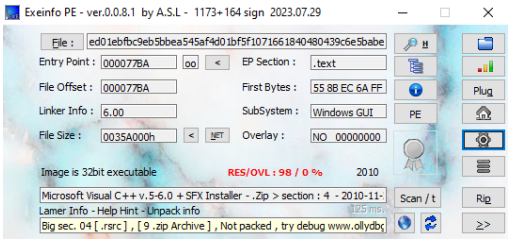
7. Optional Header is the most important part of the NT Headers. It contains a lot of information critical to the execution of the PE file, including the data directories.



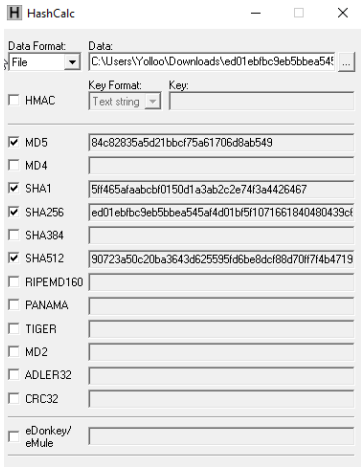
8. The Section Table contains one Section Header per row. Each Section Header contains important information about the PE file sections.



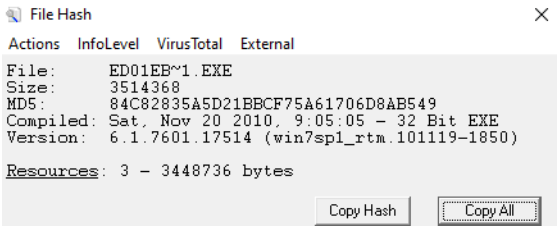
9. Detect packaged or unpacked malware samples.



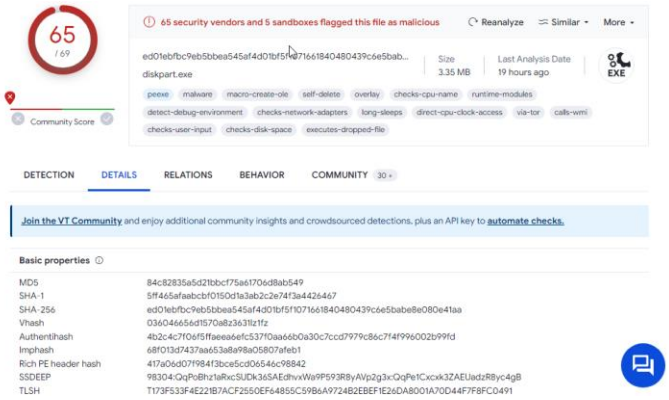
10. Finds the hash of the ransomware consisting of MD5, SHA1, SHA256, SHA512.



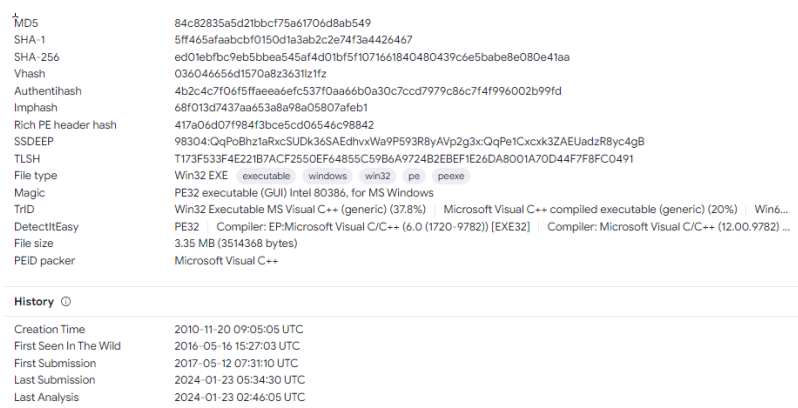
11. Copy the MD5 hash of the WannaCry ransomware exe file.



12. Input the copied hash into Virus Total for scanning and it will obtain a score of 65 security vendors and display other detailed information.



13. In Virus Total, we can look at the history section regarding the creation time to first submission of the ransomware.



14. Showing various filenames of the WannaCry ransomware.

Names

ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe
diskpart.exe
SuperKeyPass.exe
tai lieu can sua.docx.EXE
WannaCry.EXE
Ransomware.WannaCry.exe
Volcado1.bin
WannaCry_Original_MD5.EXE
Muestra1_Volcado1.bin
recursoExtraido
Ransomware_wannacry_exe
WannaCrypt0r.exe
qq.exe
wanna-cry-sample-ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.pdf
Proforma Invoice and Bank swift-REG.PI-0086547654.exe
Muestra1.exe._R__1831_1033.bin.bin
wannaCry
ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.pdf
notch_mod.exe
Endermanch@WannaCrypt0r.exe
ultra.exe
R-muestra1.bin
wannacry.exe

15. Extract the WannaCry ransomware string with the Floss tool into a txt file.

FLOSS

```
examples:
  extract all strings from an executable
    floss suspicious.exe

  do not extract static strings
    floss --no static -- suspicious.exe

  only extract stack and tight strings
    floss --only stack tight -- suspicious.exe

FLARE-VM Thu 01/25/2024 14:36:14.04
C:\Users\Yolloo\Desktop>floss C:\Users\Yolloo\Downloads\ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa
.exe > flossOut.txt
INFO: floss: extracting static strings
finding decoding function features: 100%|| 137/137 [00:00<00:00, 292.40 functions/s, skipped 6 library functions (4%)]
INFO: floss.stackstrings: extracting stackstrings from 102 functions
INFO: floss.results: oftware\
extracting stackstrings: 100%|| 102/102 [00:03<00:00, 33.31 functions/s]
INFO: floss.tightstrings: extracting tightstrings from 13 functions...
extracting tightstrings from function 0x406880: 100%|| 13/13 [00:08<00:00,  1.48 functions/s]
INFO: floss.string_decoder: decoding strings
emulating function 0x403a28 (call 2/2): 100%|| 22/22 [00:07<00:00,  3.13 functions/s]
INFO: floss: finished execution after 80.45 seconds
INFO: floss: rendering results
```

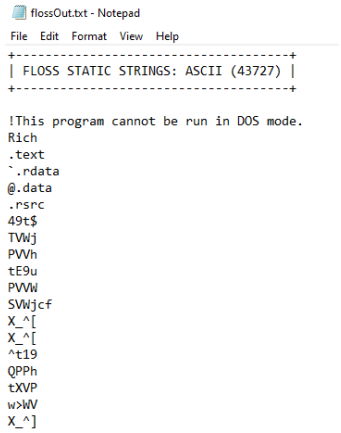
16. The txt file displays the number of static strings and stack strings.

flossOut.txt - Notepad

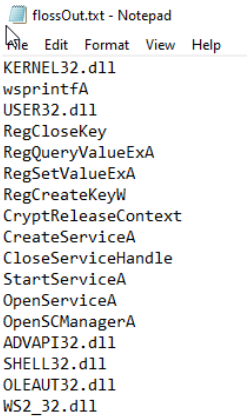
File Edit Format View Help

```
+-----------------------+-------------------------------------------------+
| file path             | C:\Users\Yolloo\Downloads\ed01ebfbc9eb5bbea545af4… |
| identified language   | unknown                                         |
| extracted strings     |                                                 |
|  static strings       | 43921 (206215 characters)                       |
|   language strings    |     0 (     0 characters)                       |
|  stack strings        | 1                                               |
|  tight strings        | 0                                               |
|  decoded strings      | 0                                               |
+-----------------------+-------------------------------------------------+
```

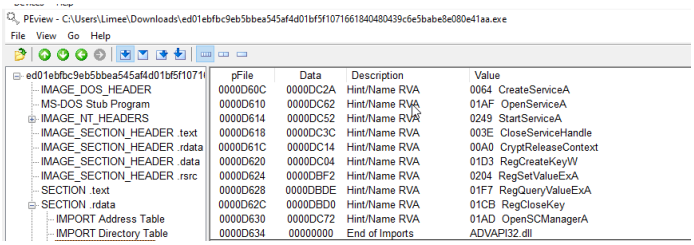
17. The following displays the strings from the WannaCry ransomware file. One of them is This program cannot be run in DOS mode.



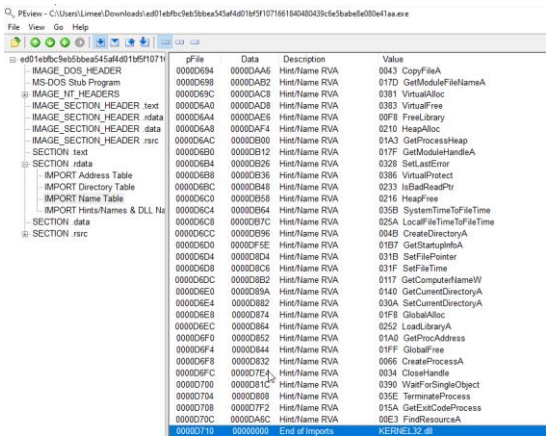
18. In that string there are several important DLLs namely Kernel32.dll, User32.dll, Advapi32.dll, Shell32.dll, Ws2_32.dll.



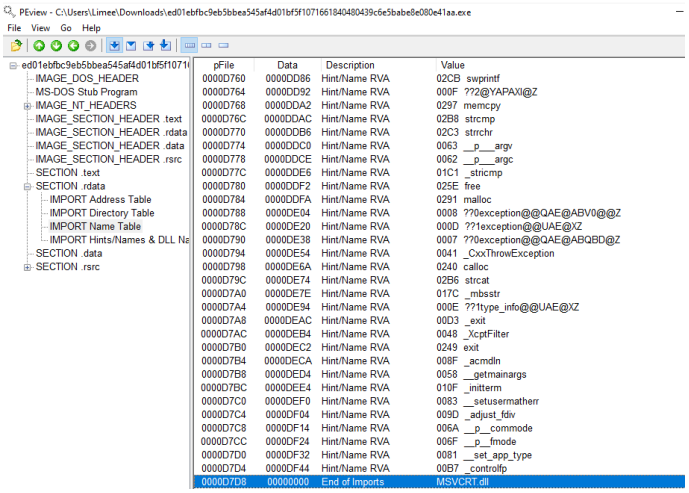
19. In Advapi32.dll there are several functions such as CreateServiceA, OpenServiceA, StartServiceA, CloseServiceHandle, CryptReleaseContext and others.



20. In kernel32.dll there are several functions such as CopyFileA, GetModuleFileNameA, VirtualAlloc, etc.



21. In MSVCRT.dll there are several functions contained in the value column.



22. In User32.dll there is a function in the form of wsprintfA

0000D7DC	0000DBB8	Hint/Name RVA	02D7 wsprintfA
0000D7E0	00000000	End of Imports	USER32.dll

Disassembly Analysis using Ghidra

1. Finds the list of strings in the Ghidra.

Defined Strings - 382 items

Location	String Value	String Representation	Data Type
00400000	MZ	"MZ"	char[2]
004000f8	PE	"PE"	char[4]
004001f0	.text	".text"	char[8]
00400218	.rdata	".rdata"	char[8]
00400240	.data	".data"	char[8]
00400268	.rsrc	".rsrc"	char[8]
0040ce3c	inflate 1.1.3 Copyri...	" inflate 1.1.3 Copy...	ds
0040d453	- unzip 0.15 Copyri...	" - unzip 0.15 Copyri...	ds
0040d7e6	CloseHandle	"CloseHandle"	ds
0040d7f4	GetExitCodeProcess	"GetExitCodeProcess"	ds
0040d80a	TerminateProcess	"TerminateProcess"	ds
0040d81e	WaitForSingleObject	"WaitForSingleObject"	ds
0040d834	CreateProcessA	"CreateProcessA"	ds
0040d846	GlobalFree	"GlobalFree"	ds
0040d854	GetProcAddress	"GetProcAddress"	ds
0040d866	LoadLibraryA	"LoadLibraryA"	ds
0040d876	GlobalAlloc	"GlobalAlloc"	ds
0040d884	SetCurrentDirectoryA	"SetCurrentDirector...	ds
0040d89c	GetCurrentDirectoryA	"GetCurrentDirector...	ds
0040d8b4	GetComputerNameW	"GetComputerNameW"	ds
0040d8c8	SetFileTime	"SetFileTime"	ds
0040d8d6	SetFilePointer	"SetFilePointer"	ds

Defined Strings - 382 items

Location	String Value	String Representation	Data Type
0040db98	CreateDirectoryA	"CreateDirectoryA"	ds
0040dbaa	KERNEL32.dll	"KERNEL32.dll"	ds
0040dbba	wsprintfA	"wsprintfA"	ds
0040dbc4	USER32.dll	"USER32.dll"	ds
0040dbd2	RegCloseKey	"RegCloseKey"	ds
0040dbe0	RegQueryValueExA	"RegQueryValueExA"	ds
0040dbf4	RegSetValueExA	"RegSetValueExA"	ds
0040dc06	RegCreateKeyW	"RegCreateKeyW"	ds
0040dc16	CryptReleaseContext	"CryptReleaseCont...	ds
0040dc2c	CreateServiceA	"CreateServiceA"	ds
0040dc3e	CloseServiceHandle	"CloseServiceHandle"	ds
0040dc54	StartServiceA	"StartServiceA"	ds
0040dc64	OpenServiceA	"OpenServiceA"	ds
0040dc74	OpenSCManagerA	"OpenSCManagerA"	ds
0040dc84	ADVAPI32.dll	"ADVAPI32.dll"	ds
0040dc92	SHELL32.dll	"SHELL32.dll"	ds
0040dc9e	OLEAUT32.dll	"OLEAUT32.dll"	ds
0040dcac	WS2_32.dll	"WS2_32.dll"	ds
0040dcba	fclose	"fclose"	ds
0040dcc4	fwrite	"fwrite"	ds
0040dcce	fread	"fread"	ds
0040dcd6	fopen	"fopen"	ds

Defined Strings - 382 items

Location	String Value	String Representation
0040eb94	CloseHandle	"CloseHandle"
0040eba0	DeleteFileW	"DeleteFileW"
0040ebac	MoveFileExW	"MoveFileExW"
0040ebb8	MoveFileW	"MoveFileW"
0040ebc4	ReadFile	"ReadFile"
0040ebd0	WriteFile	"WriteFile"
0040ebdc	CreateFileW	"CreateFileW"
0040ebe8	kernel32.dll	"kernel32.dll"
0040f08c	Microsoft Enhanced RSA and AES Cry...	"Microsoft Enhanced RSA and AES Cryptographic Provider"
0040f0c4	CryptGenKey	"CryptGenKey"
0040f0d0	CryptDecrypt	"CryptDecrypt"
0040f0e0	CryptEncrypt	"CryptEncrypt"
0040f0f0	CryptDestroyKey	"CryptDestroyKey"
0040f100	CryptImportKey	"CryptImportKey"
0040f110	CryptAcquireContextA	"CryptAcquireContextA"
0040f42c	cmd.exe /c "%s"	"cmd.exe /c \"%s\""
0040f440	115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn	"115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn"
0040f464	12t9YDPgwueZ9NyMgw519p7AA8isjr...	"12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw"
0040f488	13AM4VW2dhxYgXeQepoHkHSQuy6N...	"13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94"
0040f4b4	Global\MsWinZonesCacheCounterMut...	"Global\\MsWinZonesCacheCounterMutexA"
0040f4d8	tasksche.exe	"tasksche.exe"
0040f4e8	TaskStart	"TaskStart"

2. In the decompile file there is a WinMain function as the entry point of the application. Inside this function has several other functions that can be analyzed.

```
hPrevInstance = (HINSTANCE)0x0;
hInstance = GetModuleHandleA((LPCSTR)0x0);
exit_code = WinMain(hInstance,hPrevInstance,(LPSTR)lpCmdLine,nShowCmd);
                    /* WARNING: Subroutine does not return */
exit(exit_code);
```

3. In the Char_0040f538 function there is a /i command to copy the running binary.

```
if (*args == 2) {
  slash_i = &CHAR_0040f538;
  args = (int *)__p___argv();
  is_argv_slash_i = strcmp(*(char **)(*args + 4),(char *)slash_i);
  if ((is_argv_slash_i == 0) &&
     (bVar1 = FUN_00401b5f((wchar_t *)0x0), CONCAT31(extraout_var,bVar1) != 0)) {
    CopyFileA(&local_210,s_tasksche.exe_0040f4d8,0);
    DVar2 = GetFileAttributesA(s_tasksche.exe_0040f4d8);
    if ((DVar2 != 0xffffffff) && (is_argv_slash_i = FUN_00401f5d(), is_argv_slash_i != 0)) {
      return 0;
    }
  }
}
```

```
                    **************************************************
                    CHAR_0040f538
0040f538 2f             ??          2Fh    /
0040f539 69             ??          69h    i
0040f53a 00             ??          00h
0040f53b 00             ??          00h
```

4. In the FUN_004010fd function there is a registry key program that generates a key name in the form of u_Software\\.

```
undefined4 __cdecl FUN_004010fd(int param_1)

{
  size_t sVar1;
  LSTATUS LVar2;
  int iVar3;
  undefined4 *puVar4;
  undefined4 *puVar5;
  bool bVar6;
  HKEY hKey;
  BYTE local_2e0;
  undefined4 local_2df;
  undefined4 local_d8 [5];
  undefined4 local_c4 [45];
  DWORD local_10;
  int local_c;
  HKEY local_8;

  puVar4 = (undefined4 *)u_Software\_0040e04c;
  puVar5 = local_d8;
  for (iVar3 = 5; iVar3 != 0; iVar3 = iVar3 + -
    *puVar5 = *puVar4;
    puVar4 = puVar4 + 1;
```

```
RegCreateKeyW(hKey,(LPCWSTR)local_d8,&local_8);
if (local_8 != (HKEY)0x0) {
  if (param_1 == 0) {
    local_10 = 0x207;
    LVar2 = RegQueryValueExA(local_8,&DAT_0040e030,(LPDWOR
                                &local_10);
    bVar6 = LVar2 == 0;
    if (bVar6) {
      SetCurrentDirectoryA((LPCSTR)&local_2e0);
    }
  }
  else {
    GetCurrentDirectoryA(0x207,(LPSTR)&local_2e0);
    sVar1 = strlen((char *)&local_2e0);
    LVar2 = RegSetValueExA(local_8,&DAT_0040e030,0,1,&loca
    bVar6 = LVar2 == 0;
  }
  RegCloseKey(local_8);
  if (bVar6) {
    return 1;
  }
```

```
MOV     ESI,u_Software\_0040e04c
POP     ECX
LEA     EDI
MOVSD.REP  ES:
PUSH    0x
XOR     EA
AND     by
POP     EC
LEA     EDI
AND     dw
STOSD.REP  ES:
MOV     EC
```

```
u_Software\_0040e04c
0040e04c 53 00 6f        unicode    u"Software\\"
         00 66 00
         74 00 77 ...
0040e060 70 eb 40 00     addr       u_.doc_0040eb70
0040e064 64 eb 40 00     addr       u_.docx_0040eb64
0040e068 58 eb 40 00     addr       u_.docb_0040eb58
0040e06c 4c eb 40 00     addr       u_.docm_0040eb4c
0040e070 40 eb 40 00     addr       u_.dot_0040eb40
0040e074 34 eb 40 00     addr       u_.dotm_0040eb34
0040e078 28 eb 40 00     addr       u_.dotx_0040eb28
0040e07c 1c eb 40 00     addr       u_.xls_0040eb1c
0040e080 10 eb 40 00     addr       u_.xlsx_0040eb10
0040e084 04 eb 40 00     addr       u_.xlsm_0040eb04
0040e088 f8 ea 40 00     addr       u_.xlsb_0040eaf8
0040e08c ec ea 40 00     addr       u_.xlw_0040eaec
0040e090 e0 ea 40 00     addr       u_.xlt_0040eae0
0040e094 d4 ea 40 00     addr       u_.xlm_0040ead4
```

5. Found the password WNcry@2ol7 to extract additional information in the form of XIA.

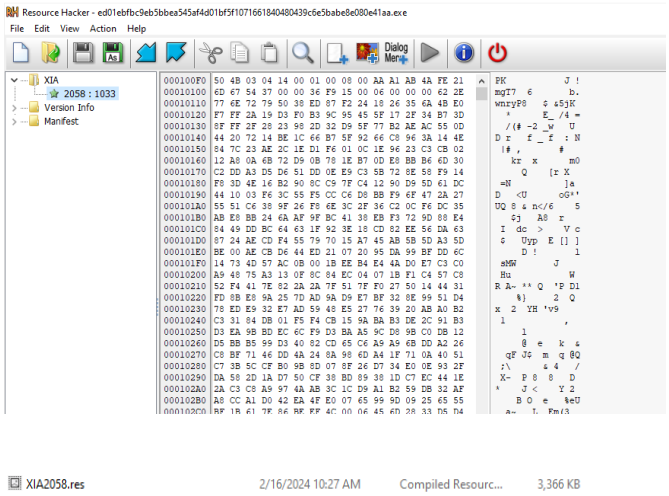
```
undefined4 __cdecl extract_resource(HMODULE const_0)

{
  HRSRC hResInfo;
  HGLOBAL hResData;
  LPVOID pvVar1;
  DWORD DVar2;
  int *piVar3;
  int iVar4;
  int iVar5;
  undefined4 *puVar6;
  char *pcVar7;
  undefined4 in_stack_00000008;
  int local_130;
  undefined4 local_12c [74];

  hResInfo = FindResourceA(const_0,(LPCSTR)2058,&XIA_Res_
  if (((hResInfo != (HRSRC)0x0) &&
      (hResData = LoadResource(const_0,hResInfo), hResDat
     (pvVar1 = LockResource(hResData), pvVar1 != (LPVOID)
    DVar2 = SizeofResource(const_0,hResInfo);
    piVar3 = (int *)FUN_004075ad(pvVar1,DVar2,in_stack_00
```

```
PUSH    0x1
CALL    registry_something
MOV     dword ptr [ESP],s_WNcry@2ol7_0040f52c
PUSH    EBX
CALL    FU
CALL    FU
PUSH    EB
PUSH    EB
PUSH    s_
```

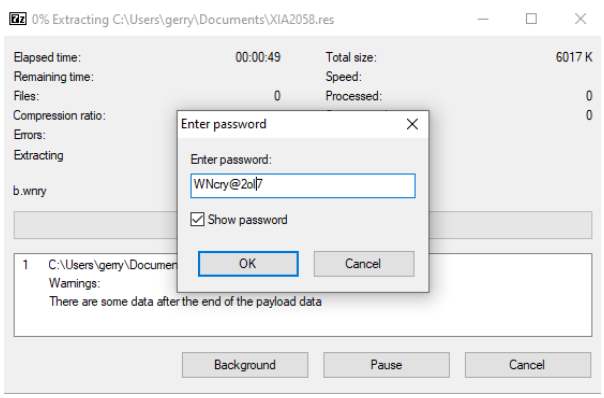
```
s_WNcry@2ol7_0040f52c                    XREF[1
0040f52c 57 4e 63        ds         "WNcry@2ol7"
         72 79 40
         32 6f 6c ...
0040f537 00              ??         00h
```

```
49  SetCurrentDirectoryA(&pfad
50  registry_something(1);
51  FUN_00401dab((HMODULE)0x0)
52  FUN_00401e9e();
53  FUN_00401064(s_attrib_+h_.
54  FUN_00401064(s_icacls_._/g
55  is_argv_slash_i = FUN_0040
```

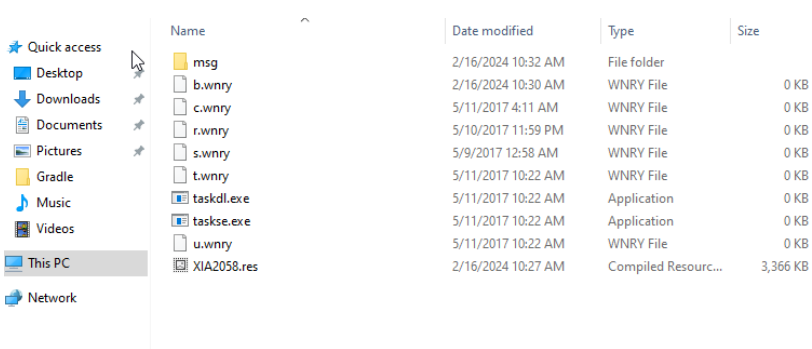
6. Compressed resources using Resource Hacker that contain XIA files.



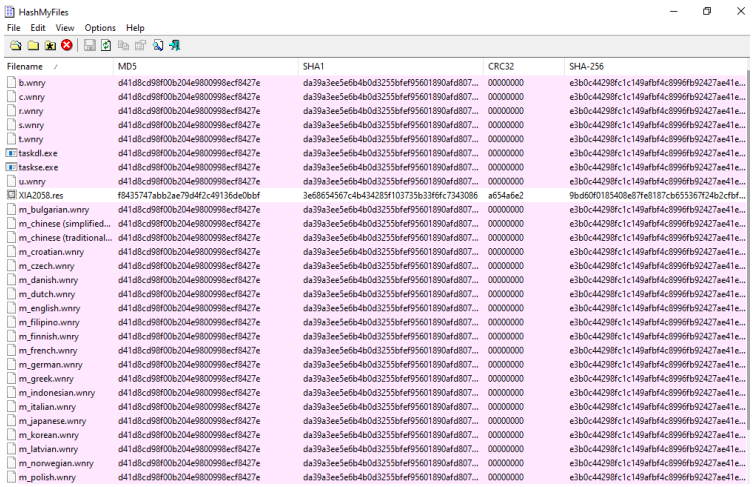
7. Enter the password WNcry@2ol7 to open the XIA file.



8. The XIA file displays several files as follows.



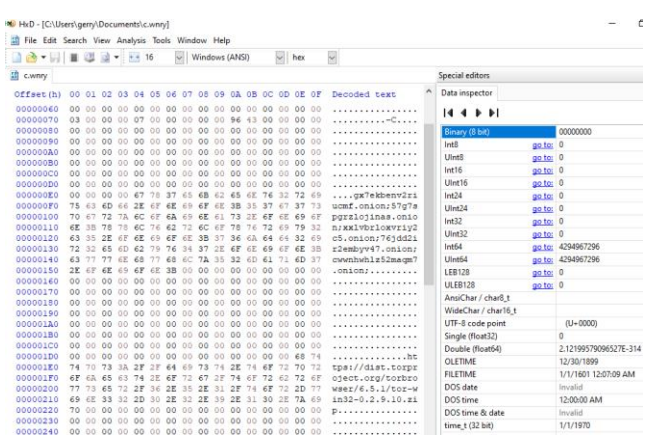
9. Checking the file hash on all data in XIA



Filename	MD5	SHA1	CRC32	SHA-256
b.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
c.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
r.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
s.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
t.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
taskdl.exe	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
taskse.exe	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
u.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
XIA2058.res	f8435747abb2ae79d4f2c49136de0bbf	3e6865457c4b434285f103735b33f6fc7343086	a654a6e2	9bd60f0185408e87fe8187cb655367f24b2cfbf...
m_bulgarian.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_chinese (simplified...	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_chinese (traditional...	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_croatian.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_czech.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_danish.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_dutch.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_english.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_filipino.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_finnish.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_french.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_german.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_greek.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_indonesian.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_italian.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_japanese.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_korean.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_latvian.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_norwegian.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...
m_polish.wnry	d41d8cd98f00b204e9800998ecf8427e	da39a3ee5e6b4b0d3255bfef95601890afd807...	00000000	e3b0c44298fc1c149afbf4c8996fb92427ae41e...

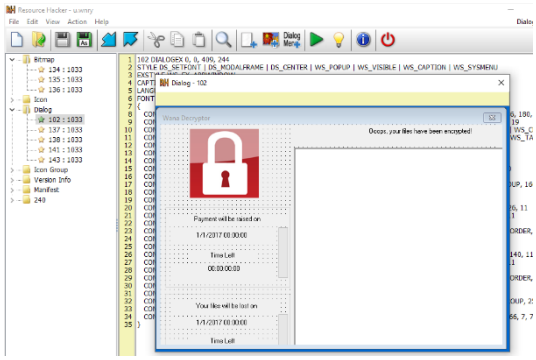
10. Opens the b.wncry file in the form of background from WannaCry ransomware

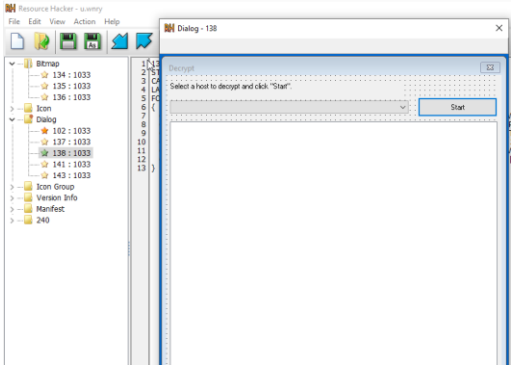
11. The c.wnry file has C2 servers in the form of onion websites and the link for the Tor browser. On ghidra there are also onion addresses.

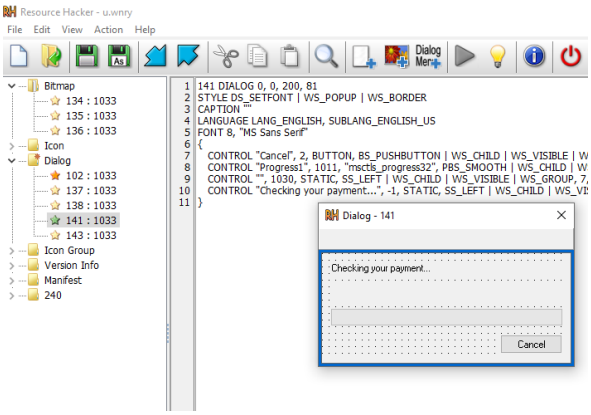


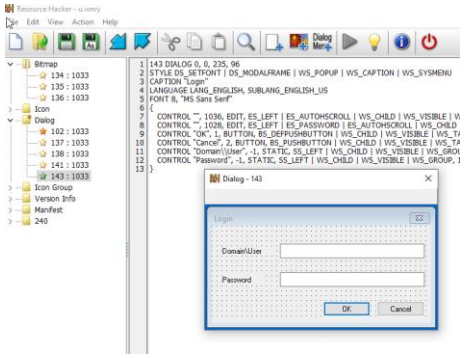

```
void show_onions_and_btc_addrs(void)

{
  bool bVar1;
  undefined3 extraout_var;
  int iVar2;
  undefined local_31c [178];
  char final_btc_addresses [602];
  char *btc_addresses [3];

  btc_addresses[0] = s_13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb_0040f488;
  btc_addresses[1] = s_12t9YDPgwueZ9NyMgw519p7AA8isjr6S_0040f464;
  btc_addresses[2] = s_115p7UMMngoj1pMvkpHijcRdfJNXj6Lr_0040f440;
  bVar1 = show_onion_addresses_etc(local_31c,1);
  if (CONCAT31(extraout_var,bVar1) != 0) {
    iVar2 = rand();
    strcpy(final_btc_addresses,btc_addresses[iVar2 % 3]);
    show_onion_addresses_etc(local_31c,0);
  }
  return;
}
```

12. Open the u.wnry file to get the wannacry ransomware dialog.

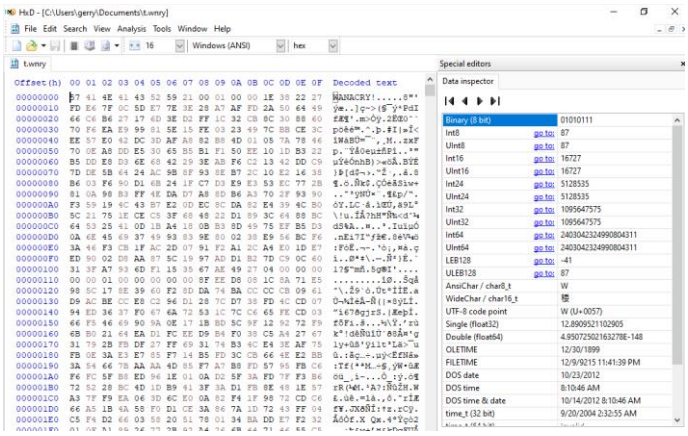




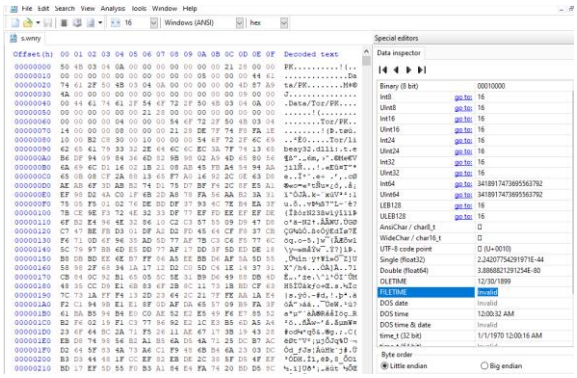




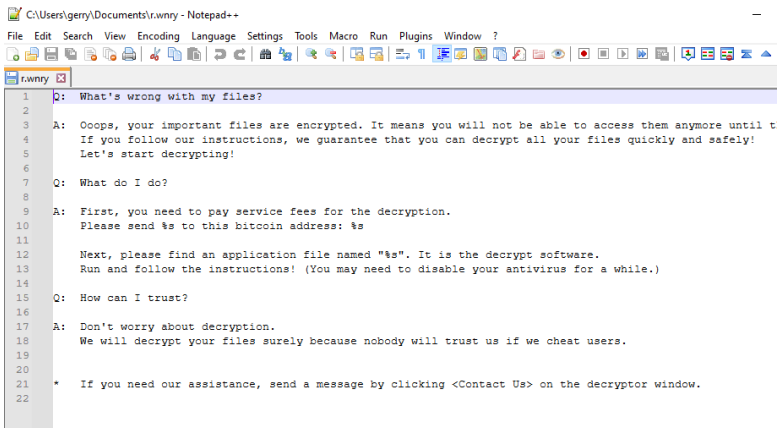
13. In the t.wnry file there is a WANACRY encoded file.



14. The s.wncry file is a compress file with PK format.



15. The r.wnry file contains text on a wannacry ransomware dialog.



16. The malware sets hidden attributes by running the createprocess command consisting of attrib+h and icacls./.

```
execute_command(s_attrib_+h_._0040f520,0,(LPDWORD)0x0);
execute_command(s_icacls_._/grant_Everyone:F_/T_/C_0040f4fc,0,(LPDWORD)0x0);
```

17. The program contains taskstart as a ransomware encryption component.

```
if (is_argv_slash_i != 0) {
  local_8 = 0;
  psVar3 = (short *)something_with_file(local_6e8,s_t.wnry_0040f4f4,&local_8);
  if (((psVar3 != (short *)0x0) &&
      (args = (int *)FUN_004021bd(psVar3,local_8), args != (int *)0x0)) &&
     (pcVar4 = (code *)FUN_00402924(args,s_TaskStart_0040f4e8), pcVar4 != (code
    (*pcVar4)(0,0);
  }
}
FUN_0040137a();
```

18. Display the private RSA key encoded in the malware.

```
undefined4 __thiscall crypto_stuff(int param_1,int param_1_00)

{
  int iVar1;
  int in_ECX;

  iVar1 = Crypto_Context_Setup(in_ECX);
  if (iVar1 != 0) {
    if (param_1 == 0) {
      iVar1 = (*Crypt_Import_Key)(*(undefined4 *)(in_ECX + 4),&public_key_blob,0x494,0,0,in_ECX
      ;
    }
    else {
      iVar1 = FUN_004018f9(*(undefined4 *)(in_ECX + 4),in_ECX + 8,(LPCSTR)param_1);
    }
    if (iVar1 != 0) {
      return 1;
    }
  }
  FUN_004018b9(in_ECX);
  return 0;
}
```

```
                    public_key_blob
0040ebf8 07          ??        07h
0040ebf9 02          ??        02h
0040ebfa 00          ??        00h
0040ebfb 00          ??        00h
0040ebfc 00          ??        00h
0040ebfd a4          ??        A4h
0040ebfe 00          ??        00h
0040ebff 00          ??        00h
0040ec00 52          ??        52h    R
0040ec01 53          ??        53h    S
0040ec02 41          ??        41h    A
0040ec03 32          ??        32h    2
0040ec04 00          ??        00h
0040ec05 08          ??        08h
0040ec06 00          ??        00h
0040ec07 00          ??        00h
```

19. The program contains the AES key for the encrypted file.

```
Decompile: something_with_aes - (ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa)

void __thiscall
something_with_aes(void *this,undefined aes_key,undefined param_2,undefined aes_key_size,
                   undefined const_16)

{
  undefined4 uVar1;
  uint uVar2;
  int iVar3;
  char *pcVar4;
  int iVar5;
  uint *puVar6;
  char *pcVar7;
  undefined4 *puVar8;
  int iVar9;
  int iVar10;
  undefined4 *puVar11;
  undefined3 in_stack_00000005;
  undefined3 in_stack_00000009;
  undefined3 in_stack_0000000d;
  undefined3 in_stack_00000011;
  exception local_18 [20];
```

```
if (_aes_key == (byte *)0x0) {
  _param_2 = (uint *)&DAT_0040f57c;
  exception::exception(local_18,(char **)&param_2);
                /* WARNING: Subroutine does not return */
  _CxxThrowException(local_18,(ThrowInfo *)&pThrowInfo_0040d570);
}
if (((_aes_key_size != 0x10) && (_aes_key_size != 0x18)) && (_aes_key_size != 0x20)) {
  _param_2 = (uint *)&DAT_0040f57c;
  exception::exception(local_18,(char **)&param_2);
                /* WARNING: Subroutine does not return */
  _CxxThrowException(local_18,(ThrowInfo *)&pThrowInfo_0040d570);
}
if (((_const_16 != (byte *)0x10) && (_const_16 != (byte *)0x18)) && (_const_16 != (byt
  _param_2 = (uint *)&DAT_0040f57c;
  exception::exception(local_18,(char **)&param_2);
                /* WARNING: Subroutine does not return */
  _CxxThrowException(local_18,(ThrowInfo *)&pThrowInfo_0040d570);
}
*(byte **)((int)this + 0x3cc) = _const_16;
*(int *)((int)this + 0x3c8) = _aes_key_size;
memcpy((void *)((int)this + 0x3d0),_param_2,(size_t)_const_16);
memcpy((void *)((int)this + 0x3f0), param_2,*(size_t *)((int)this + 0x3cc));
```