

# Raport z poszczególnych etapów projektu

## - analiza malware'u

Grupa: Szymon K, Tomasz N

### I. Przygotowanie środowiska

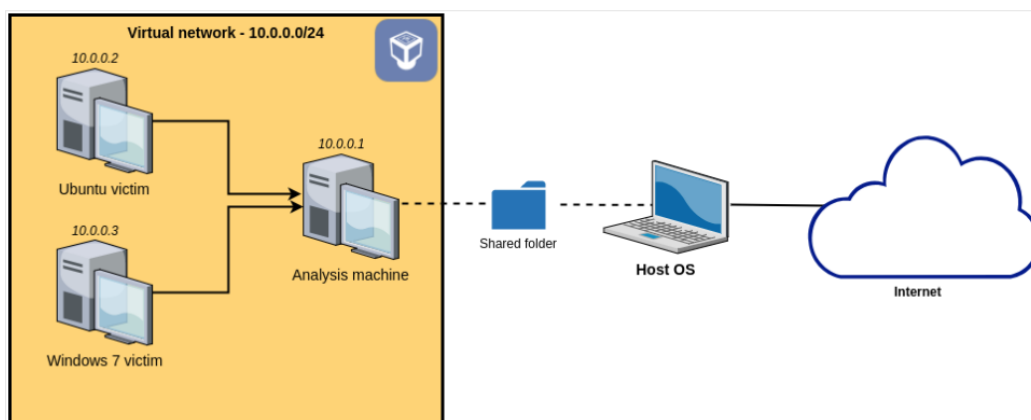
<https://blog.christophetd.fr/malware-analysis-lab-with-virtualbox-inetsim-and-burp/> - ustawienia sieci na przykładzie tego

1. Instalacja oprogramowania INetSim na Kali Linux (powinno być domyślnie)

```
$ sudo su
$ echo "deb http://www.inetsim.org/debian/ binary/" > /etc/apt/sources.list.d/inetsim.list
$ wget -O - http://www.inetsim.org/inetsim-archive-signing-key.asc | apt-key add -
$ apt update
$ apt install inetsim
```

2. Wszystkie interfejsy ustawiamy w VirtualBox na "Internal Network". Windows jest maszyną ofiary (jako że analizowana próbka jest na Win) - w związku z czym musi być całkowicie odizolowana od sieci. Tworzymy zatem sieć wewnętrzną pomiędzy maszyną Win7 i kali (sieć nazywa się malware\_analysis\_network). Chcemy mieć możliwość przechwytywania ruchu sieciowego na maszynie kali i analizy przy użyciu oprogramowania Wireshark. Do przechwytywania ruchu DNS i HTTP użyjemy narzędzia INetSim.

(u nas bez Ubuntu victim)



kali: 10.0.0.1/24 (bez bramy domyślnej)

Win7: 10.0.0.2/24 (brama domyślna 10.0.0.1 - maszyna kali)

3. Interfejs sieciowy na kalim:

```
auto eth0
iface eth0 inet static
address 10.0.0.1
netmask 255.255.255.0
```

Po wpisaniu konfiguracji restartujemy interfejs:

```
$ sudo ifdown eth0
```

```
$ sudo ifup eth0
```

4. Resztę wykonujemy zgodnie z instrukcją. Instalacja InetSim i Burpa. Możliwe problemy - nie wszystkie protokoły na INetSimie działają poprawnie, warto rozważyć wyłączenie tych, z których nie będziemy korzystać; inny problem to brak certyfikatu SSL dla protokołu HTTPS, jeśli chcemy go używać należy przenieść certyfikat i plik klucza z folderu data/inetsim/certs do data/certs.
5. Kolejny punkt - instalujemy ncat i sysinternals suite na Windowsie 7
6. Wyłączamy Windows Firewall oraz Windows Defender
7. Na maszynie kali tworzymy share folder do przekazywania plików binarnych - stamtąd netcatem na Windows 7.

Przygotowane narzędzia

Windows:


PEBear  
Dependency Walker  
Detect It Easy  
Resource Hacker  
Regshot  
Ollydbg  
Pakiet Sysinternals Suite  
ApateDNS

Kali Linux:

Ghidra  
Wireshark  
INetSim  
Burp Suite

## II. Analiza Statyczna

Analizowany program przechowywany jest w systemie Windows 7, w formie skompresowanej jako plik zip. Po pierwsze zatem rozpakowujemy go domyślnym programem. Po tej operacji otrzymujemy pojedynczy folder o nazwie "02" zawierający plik wykonywalny ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe:

Windows 7 (C:) ▾ malware_analysis ▾ bin ▾ projekt ▾ 02_windows ▾ 02 ▾				
Folder				
Name ^	Date modified	Type	Size	
 ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe	5/14/2017 4:29 PM	Application	3,432 KB	

Nazwa wydaje się być skrótem plików programu:

# Hash Analyzer

Tool to identify hash types. Enter a hash to be identified.

Analyze

<b>Hash:</b>	ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa
<b>Salt:</b>	Not Found
<b>Hash type:</b>	SHA2-256
<b>Bit length:</b>	256
<b>Character length:</b>	64
<b>Character type:</b>	hexadecimal

Wyszukujemy również dany hash, aby dowiedzieć się czy plik ten widnieje w sygnaturach.

VirusTotal:

63

169

63 security vendors flagged this file as malicious

ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa

diskpart.exe

direct-cpu-clock-access

executes-dropped-file

overlay

peexe

runtime-modules

self-delete

via-tor

3.35 MB

Size

2021-04-27 12:31:22 UTC

43 minutes ago

EXE

Community Score

DETECTION	DETAILS	RELATIONS	BEHAVIOR	COMMUNITY
Acronis		① Suspicious	Ad-Aware	① Trojan.Ransom.WannaCryptor.A
AegisLab		① Trojan.Win32.Wanna.toNn	AhnLab-V3	① Trojan.Win32.WannaCryptor.R200571
Alibaba		① Ransom.Win32/WannaCry.all1020010	ALYac	① Trojan.Ransom.WannaCryptor
SecureAge APEX		① Malicious	Arcabit	① Trojan.Ransom.WannaCryptor.A
Avast		① Win32:WanaCry-A [Trj]	AVG	① Win32:WanaCry-A [Trj]
Avira (no cloud)		① TR/Ransom.JB	Baidu	① Win32.Trojan.WannaCry.c
BitDefender		① Trojan.Ransom.WannaCryptor.A	BitDefenderTheta	① Gen:NN.Zexaf.34684.wt0@aGEmS3dl
Bkav Pro		① W32.Common.B4665B53	CAT-QuickHeal	① Ransom.WannaCrypt.A4
ClamAV		① Win.Ransomware.WannaCry-6313787-0	Comodo	① TrojWare.Win32.Ransom.WannaCrypt.B@...
CrowdStrike Falcon		① Win/malicious_confidence_100% (W)	Cybereason	① Malicious.5a5d21
Cylance		① Unsafe	Cynet	① Malicious (score: 100)
Cyren		① W32/Trojan.ZTSA-8671	DrWeb	① Trojan.Encoder.11432
eGambit		① Trojan.Generic	Elastic	① Malicious (high Confidence)

any.run:

## General Info

File name

ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe

Full analysis

<https://app.any.run/tasks/33746034-ea7e-4581-bee7-0fd7d5ed7c83>

Verdict

Malicious activity

Threats:

WannaCry

WannaCry is a famous Ransomware that utilizes the EternalBlue exploit. This malware is known for infecting at least 200,000 computers worldwide and it continues to be an active and dangerous threat.

Malware Trends Tracker

More details

Analysis date

6/20/2018, 01:05:22

OS:

Windows 7 Professional Service Pack 1 (build: 7601, 32 bit)

Tags:

ransomware wannacry wannacryptor

Indicators:

MIME:

application/x-dosexec

File info:

PE32 executable (GUI) Intel 80386, for MS Windows

MD5

84C82835A5D21BBCF75A61706D8AB549

SHA1

5FF465AFAACBF0150D1A3AB2C2E74F3A4426467

SHA256

ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA

SSDEEP

98304:QQPOBHZ1ARXCSUDK36SAEDHVXWA9P593R8YAVP2G3X:QQPE1CXCKXK3ZAEUADZR8YC4GB

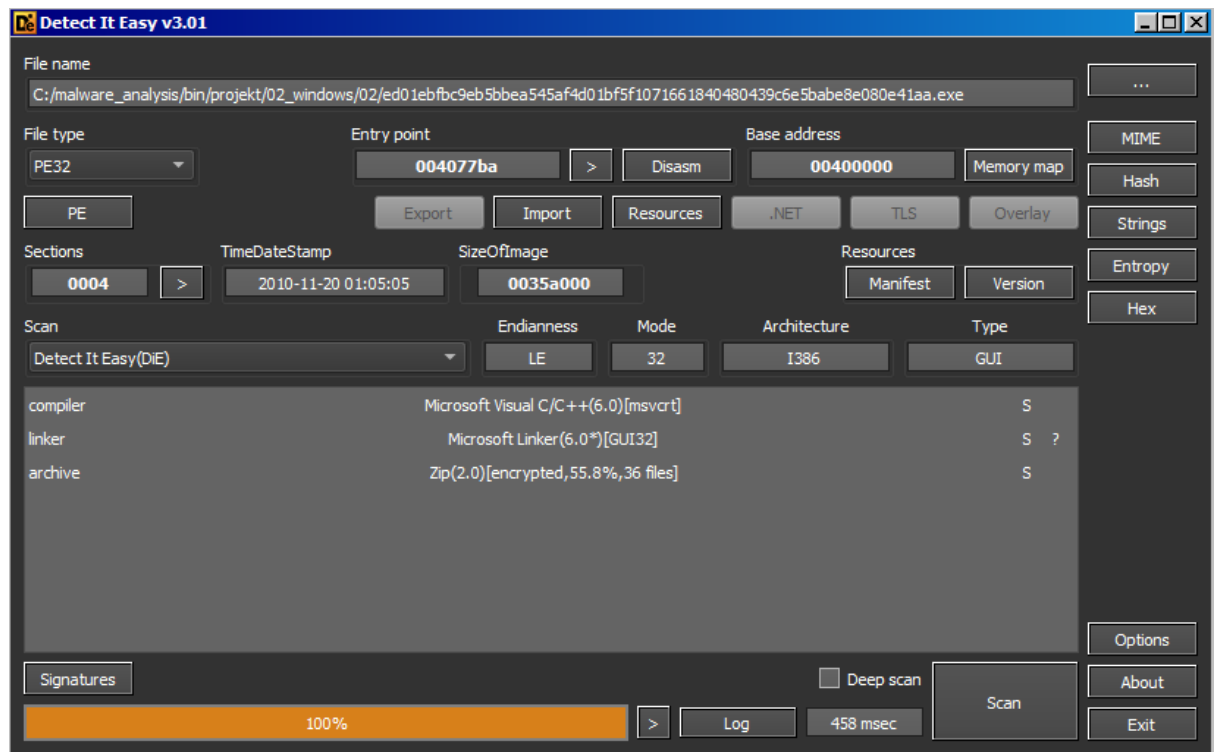
Z analizy sygnaturowej udało nam się zatem dowiedzieć, że z sygnaturą będącą nazwą tego pliku skojarzona jest złośliwa działalność oraz zidentyfikować ją jako pochodzącą ze skrótu SHA-256 ze znanego programu

WannaCry. Jest to malware typu ransomware, który wykorzystywał podatność o nazwie EternalBlue w produktach firmy Microsoft.

Jako podaje avast.com uważa się, że ten wypuszczony w maju 2017 roku malware zainfekował przeszło 230 tysięcy urządzeń Windowsa 150 krajach świata w przeciągu jednego dnia <sup>1</sup>.

Podstawowa analiza statyczna:

Detect it easy



Widzimy, że DiE zidentyfikował badany program jako spakowany  
Data kompilacji wskazuje na 2010-11-20 01:05:05

Z dokładniejszej analizy dowiadujemy się, że plik posiada sekcje .text, .rdata, .data oraz .rsrc

Sekcja zasobów (resources) zawiera podsekcje "Version" oraz "Manifest"

```
VS_VERSION_INFO.StringFileInfo.040904B0.CompanyName:Microsoft Corporation
VS_VERSION_INFO.StringFileInfo.040904B0.FileDescription:DiskPart
VS_VERSION_INFO.StringFileInfo.040904B0.FileVersion:6.1.7601.17514 (win7sp1_rtm.101119-1850)
VS_VERSION_INFO.StringFileInfo.040904B0.InternalName:diskpart.exe
VS_VERSION_INFO.StringFileInfo.040904B0.LegalCopyright:© Microsoft Corporation. All rights reserved.
VS_VERSION_INFO.StringFileInfo.040904B0.OriginalFilename:diskpart.exe
VS_VERSION_INFO.StringFileInfo.040904B0.ProductName:Microsoft® Windows® Operating System
VS_VERSION_INFO.StringFileInfo.040904B0.ProductVersion:6.1.7601.17514
VS_VERSION_INFO.VarFileInfo.Translation:04b00409
```

Informacje o wersji wskazują na plik diskpart.exe

<sup>1</sup> <https://www.avast.com/c-eternalblue>

W pliku Manifest pojawiają się nazwy wspieranych systemów operacyjnych

```
<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>
  <dependency>
    <dependentAssembly>
      <assemblyIdentity
        type="win32"
        name="Microsoft.Windows.Common-Controls"
        version="6.0.0.0"
        processorArchitecture="*"
        publicKeyToken="6595b64144ccf1df"
        language="*"
      />
    </dependentAssembly>
  </dependency>
  <compatibility xmlns="urn:schemas-microsoft-com:compatibility.v1">
    <application>
      <!-- Windows 10 -->
      <supportedOS Id="{8e0f7a12-bfb3-4fe8-b9a5-48fd50a15a9a}"/>
      <!-- Windows 8.1 -->
      <supportedOS Id="{1f676c76-80e1-4239-95bb-83d0f6d0da78}"/>
      <!-- Windows Vista -->
      <supportedOS Id="{e2011457-1546-43c5-a5fe-008deee3d3f0}"/>
      <!-- Windows 7 -->
      <supportedOS Id="{35138b9a-5d96-4fbd-8e2d-a2440225f93a}"/>
      <!-- Windows 8 -->
      <supportedOS Id="{4a2f28e3-53b9-4441-ba9c-d69d4a4a6e38}"/>
    </application>
  </compatibility>
</assembly>
```

Wykonujemy wbudowane polecenie strings:

- znajdujemy dwa stringi potwierdzające fakt zapakowania pliku - prawdopodobnie są to programy, których malware używa aby się rozpakować

193	0000ce3c	0000002e	A	inflate 1.1.3 Copyright 1995-1998 Mark Adler
194	0000d186	00000005	A	j0-=m
195	0000d195	00000007	A	QkkbaL
196	0000d3f1	00000005	A	wn>Jj
197	0000d453	0000002b	A	- unzip 0.15 Copyright 1998 Gilles Vollant

- fragment zawierający nazwy używanych funkcji kryptograficznych (WannaCry to ransomware więc z pewnością będzie zawierał kod szyfrujący pliki)

371	0000f08c	00000035	A	Microsoft Enhanced RSA and AES Cryptographic Provider
372	0000f0c4	0000000b	A	CryptGenKey
373	0000f0d0	0000000c	A	CryptDecrypt
374	0000f0e0	0000000c	A	CryptEncrypt
375	0000f0f0	0000000f	A	CryptDestroyKey
376	0000f100	0000000e	A	CryptImportKey
377	0000f110	00000014	A	CryptAcquireContextA

- stringi wyglądające jak komunikaty błędów:

392	0000f588	0000000f	A	..?AVexception@@
393	0000f598	00000014	A	incompatible version
394	0000f5b0	0000000c	A	buffer error
395	0000f5c0	00000013	A	insufficient memory
396	0000f5d4	0000000a	A	data error
397	0000f5e0	0000000c	A	stream error
398	0000f5f0	0000000a	A	file error
399	0000f5fc	0000000a	A	stream end
400	0000f608	0000000f	A	need dictionary
401	0000f618	00000015	A	invalid distance code
402	0000f630	0000001b	A	invalid literal/length code
403	0000f64c	00000019	A	invalid bit length repeat
404	0000f668	00000023	A	too many length or distance symbols
405	0000f68c	0000001c	A	invalid stored block lengths
406	0000f6ac	00000012	A	invalid block type
407	0000f6c0	00000023	A	incomplete dynamic bit lengths tree
408	0000f6e4	00000027	A	oversubscribed dynamic bit lengths tree

- polecenia i ścieżki, do których być może program wstrzykiwać będzie własne stringi

378	0000f3f8	00000008	U	%s\Intel
379	0000f40c	0000000e	U	%s\ProgramData
380	0000f42c	0000000f	A	cmd.exe /c "%s"

- lista rozszerzeń plików - być może chodzi o rozszerzenia plików, które zostaną zaszyfrowane

319	0000e034	0000000b	U	WanaCrypt0r
320	0000e04c	00000009	U	Software\
321	0000e4b0	00000005	U	.lay6
322	0000e4c8	00000008	U	.sqlite3
323	0000e4dc	00000009	U	.sqlitedb
324	0000e4fc	00000006	U	.accdb
325	0000e678	00000005	U	.java
326	0000e690	00000006	U	.class
327	0000e6fc	00000005	U	.mpeg
328	0000e798	00000005	U	.djvu
329	0000e7d0	00000005	U	.tiff
330	0000e830	00000005	U	.jpeg
331	0000e854	00000007	U	.backup
332	0000e904	00000005	U	.vmdk
333	0000e91c	00000005	U	.sldm
334	0000e928	00000005	U	.sldx
335	0000e970	00000008	U	.onetoc2
336	0000e9e4	00000005	U	.vsdx
337	0000ea38	00000005	U	.potm
338	0000ea44	00000005	U	.potx
339	0000ea50	00000005	U	.ppam
340	0000ea5c	00000005	U	.ppsx
341	0000ea68	00000005	U	.ppsm
342	0000ea8c	00000005	U	.pptm
343	0000ea98	00000005	U	.pptx
344	0000eab0	00000005	U	.xltn
345	0000eabc	00000005	U	.xltx
346	0000eaf8	00000005	U	.xlsb
347	0000eb04	00000005	U	.xlsm
348	0000eb10	00000005	U	.xlsx
349	0000eb28	00000005	U	.dotx
350	0000eb34	00000005	U	.dotm
351	0000eb4c	00000005	U	.docm
352	0000eb58	00000005	U	.docb
353	0000eb64	00000005	U	.docx

- inne charakterystyczne stringi:

354	0000eb7c	00000008	A	WANACRY!
355	0000eb88	00000005	U	%s\%s

Dependency Walker

Program korzysta z następujących bibliotek i funkcji:

**KERNEL32.DLL**



- CreateDirectory, CreateFile, ReadFile, WriteFile - tworzenie i otwieranie katalogów i plików oraz manipulacja nimi
- SetFileAttributes, SetFilePointer
- CreateProcess, TerminateProcess - tworzenie i zamykanie procesów
- EnterCriticalSection, DeleteCriticalSection, OpenMutex - synchronizacja procesów

## USER32.DLL

- wsprintf - zapis danych do podanej struktury/pliku

## ADVAPI32.DLL

- RegCreateKey, RegCloseKey, RegQueryValue, RegSetValue - odczytywanie, tworzenie i modyfikowanie kluczy rejestrów
- OpenSCMService, OpenService, CreateService, StartService - tworzenie i otwieranie serwisów
- CryptReleaseContext

## MSVCRT.DLL

PE-BEAR - informacje z nagłówka

Offset	Name	Value	Meaning
FC	Machine	14c	Intel 386
FE	Sections Count	4	4
100	Time Date Stamp	4ce78f41	Saturday, 20.11.2010 09:05:05 UTC
104	Ptr to Symbol Table	0	0
108	Num. of Symbols	0	0
10C	Size of OptionalHeader	e0	224
10E	Characteristics	10F	
		1	Relocation info stripped from file.
		2	File is executable (i.e. no unresolved external references).
		4	Line numbers stripped from file.
		8	Local symbols stripped from file.
		100	32 bit word machine.

Zaawansowana analiza statyczna

Sygnatury

- WANACRY!

```

*****
s_WANACRY!_0040eb7c          XREF[1]:  FUN_004014a6:00401566(*)
0040eb7c 57 41 4e      ds      "WANACRY!"
          41 43 52
          59 21 00
-----

```

- WanaCrypt0r
- WNcry@2o17 - hasło do rozpakowania

Są to charakterystyczne elementy pozwalające zidentyfikować ten program.

```

0040f52c  57 4e 63      s_WNcry@2o17_0040f52c      XREF[1]:  FUN_00401fe7:004020c8(*)
          72 79 40      ds      "WNcry@2o17"
          32 6f 6c ...
0040f537  00             ??      00h      ? -> 00692f00

```

- tasksche.exe - nazwa tego pliku pojawia się w kodzie jednak nie jest on widoczny po rozpakowaniu, zatem można przypuszczać, że plik tworzony jest dynamicznie po uruchomieniu programu

```

0040f4d8  74 61 73      ds      "tasksche.exe"      XREF[1]:  FUN_00401fe7:00402075(*)
          6b 73 63
          68 65 2e ...
0040f4e5  00             ??      00h
0040f4e6  00             ??      00h
0040f4e7  00             ??      00h



















0040f4e8  54 61 73      s_TaskStart_0040f4e8      XREF[1]:  FUN_00401fe7:00402145(*)
          6b 53 74      ds      "TaskStart"
          61 72 74 00
0040f4f2  00             ??      00h
0040f4f3  00             ??      00h

0040f4f4  74 2e 77      s_t.wnry_0040f4f4      XREF[1]:  FUN_00401fe7:00402125(*)
          6e 72 79 00      ds      "t.wnry"
0040f4fb  00             ??      00h

```

Zawartość próbki po rozpakowaniu

Aby rozpakować próbkę potrzebujemy hasło **WNcry@2o17**

	msg	28.04.2021 22:42	Folder plików	
	b.wnry	11.05.2017 13:13	Plik WNRY	1 407 KB
	c.wnry	11.05.2017 13:11	Plik WNRY	1 KB
	ed01ebfbc9eb5bbea545af4d01bf5f107166...	14.05.2017 16:29	Aplikacja	3 432 KB
	r.wnry	11.05.2017 08:59	Plik WNRY	1 KB
	s.wnry	09.05.2017 09:58	Plik WNRY	2 968 KB
	t.wnry	11.05.2017 19:22	Plik WNRY	65 KB
	taskdl	11.05.2017 19:22	Aplikacja	20 KB
	taskdl.exe.id0	28.04.2021 23:46	Plik ID0	136 KB
	taskdl.exe.id1	28.04.2021 23:46	Plik ID1	56 KB
	taskdl.exe.nam	28.04.2021 23:46	Plik NAM	16 KB
	taskdl.exe.til	28.04.2021 23:46	Plik TIL	2 KB
	taskse	11.05.2017 19:22	Aplikacja	20 KB
	taskse.exe.id0	28.04.2021 23:46	Plik ID0	16 KB
	taskse.exe.id1	28.04.2021 23:46	Plik ID1	0 KB
	taskse.exe.nam	28.04.2021 23:46	Plik NAM	0 KB
	taskse.exe.til	28.04.2021 23:46	Plik TIL	1 KB
	u.wnry	11.05.2017 19:22	Plik WNRY	240 KB

Co robi plik

ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe:

- ładowanie bibliotek i funkcji potrzebnych w dalszym działaniu malware'u"

```
*****
*                                     *
*                                     *
*****
undefined FUN_0040170a()
      AL:1      <RETURN>
      FUN_0040170a      XREF[1]:      FUN_00401fe7:004020f5(c)
0040170a 53      PUSH      EBX
0040170b 57      PUSH      EDI
0040170c e8 34 03      CALL      FUN_00401a45      undefined FUN_00401a45()
      00 00
00401711 85 c0      TEST      EAX,EAX
00401713 0f 84 bf      JZ        LAB_004017d8
      00 00 00
00401719 33 db      XOR      EBX,EBX
0040171b 39 1d 78      CMP      dword ptr [DAT_0040f878],EBX
      f8 40 00
00401721 0f 85 ac      JNZ      LAB_004017d3
      00 00 00
00401727 68 e8 eb      PUSH     s_kernel32.dll_0040ebe8      = "kernel32.dll"
      40 00
0040172c ff 15 e0      CALL     dword ptr [->KERNEL32.DLL::LoadLibraryA]
      80 40 00
00401732 8b f8      MOV      EDI,EAX
00401734 3b fb      CMP      EDI,EBX
00401736 0f 84 9c      JZ        LAB_004017d8
      00 00 00
0040173c 56      PUSH     ESI
0040173d 8b 35 e4      MOV      ESI,dword ptr [->KERNEL32.DLL::GetProcAddress] = 0000d852
      80 40 00
00401743 68 dc eb      PUSH     s_CreateFileW_0040ebdc      = "CreateFileW"
      40 00
00401748 57      PUSH     EDI
00401749 ff d6      CALL     ESI=>KERNEL32.DLL::GetProcAddress
0040174b 68 d0 eb      PUSH     s_WriteFile_0040ebd0      = "WriteFile"
      40 00
00401750 57      PUSH     EDI
00401751 a3 78 f8      MOV      [DAT_0040f878],EAX
      40 00
```

\*na screenie: fragment kodu gdzie ładowana jest biblioteka kernel32.dll (LoadLibraryA), a następnie poszczególne jej funkcje (GetProcAddress), np CreateFile, WriteFile, ReadFile, MoveFile, DeleteFile, CloseHandle

- program tworzy nowe pliki

- ładowanie funkcji kryptograficznych:

```

      18 40 00
00401a4e 57      PUSH     EDI
00401a4f 0f 85 97    JNZ      LAB_00401aec
      00 00 00
00401a55 68 20 e0    PUSH     s_advapi32.dll_0040e020      = "advapi32.dll"
      40 00
00401a5a ff 15 e0    CALL     dword ptr [->KERNEL32.DLL::LoadLibraryA]
      80 40 00
00401a60 8b f8      MOV      EDI,EAX
00401a62 3b fb      CMP      EDI,EBX
00401a64 0f 84 87    JZ       LAB_00401af1
      00 00 00
00401a6a 56      PUSH     ESI
00401a6b 8b 35 e4    MOV      ESI,dword ptr [->KERNEL32.DLL::GetProcAddress] = 0000d852
      80 40 00
00401a71 68 10 f1    PUSH     s_CryptAcquireContextA_0040f110 = "CryptAcquireContextA"
      40 00
00401a76 57      PUSH     EDI
00401a77 ff d6      CALL     ESI=>KERNEL32.DLL::GetProcAddress
00401a79 68 00 f1    PUSH     s_CryptImportKey_0040f100      = "CryptImportKey"
      40 00
00401a7e 57      PUSH     EDI
00401a7f a3 94 f8    MOV      [DAT_0040f894],EAX
      40 00
00401a84 ff d6      CALL     ESI=>KERNEL32.DLL::GetProcAddress
00401a86 68 f0 f0    PUSH     s_CryptDestroyKey_0040f0f0      = "CryptDestroyKey"
      40 00
00401a8b 57      PUSH     EDI

```

\*z biblioteki advapi32.dll funkcji CryptAcquireContext, CryptImportKey, CryptDestroyKey, CryptEncrypt, CryptDecrypt, CryptGenKey

- charakterystyczne ciągi znaków (można wykorzystać jako sygnatury) - są to numery portfeli bitcoinowych, na które przestępcy żądają wpłaty, rzeczywiście gdy uruchomimy malware w komunikacie pojawia się jeden z tych numerów

```

9
10 local_10[0] = s_13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb_0040f488;
11 local_10[1] = s_12t9YDPgwueZ9NyMgw519p7AA8isjr6S_0040f464;
12 local_10[2] = s_115p7UMMngojlPmVkpHijcRdfJNXj6Lr_0040f440;
13 iVar1 = FUN_00401000(local_31c,1);
14 if (iVar1 != 0) {

```



- komunikaty błędów

```

LAB_004050c7                                XREF[1]: 004050b5(j)
04050c7 83 f8 fd    CMP     EAX,-0x3
04050ca 75 09          JNZ     LAB_004050d5
04050cc c7 46 18      MOV     dword ptr [ESI + 0x18],s_oversubscribed_distan... = "oversubscribed distance tree"
          90 f7 40 00
04050d3 eb 39          JMP     LAB_0040510e

LAB_004050d5                                XREF[1]: 004050ca(j)
04050d5 83 f8 fb    CMP     EAX,-0x5
04050d8 75 09          JNZ     LAB_004050e3
04050da c7 46 18      MOV     dword ptr [ESI + 0x18],s_incomplete_distance_t... = "incomplete distance tree"
          74 f7 40 00
04050e1 eb 28          JMP     LAB_0040510b

LAB_004050e3                                XREF[1]: 004050d8(j)
04050e3 83 f8 fc    CMP     EAX,-0x4
04050e6 74 26          JZ      LAB_0040510e

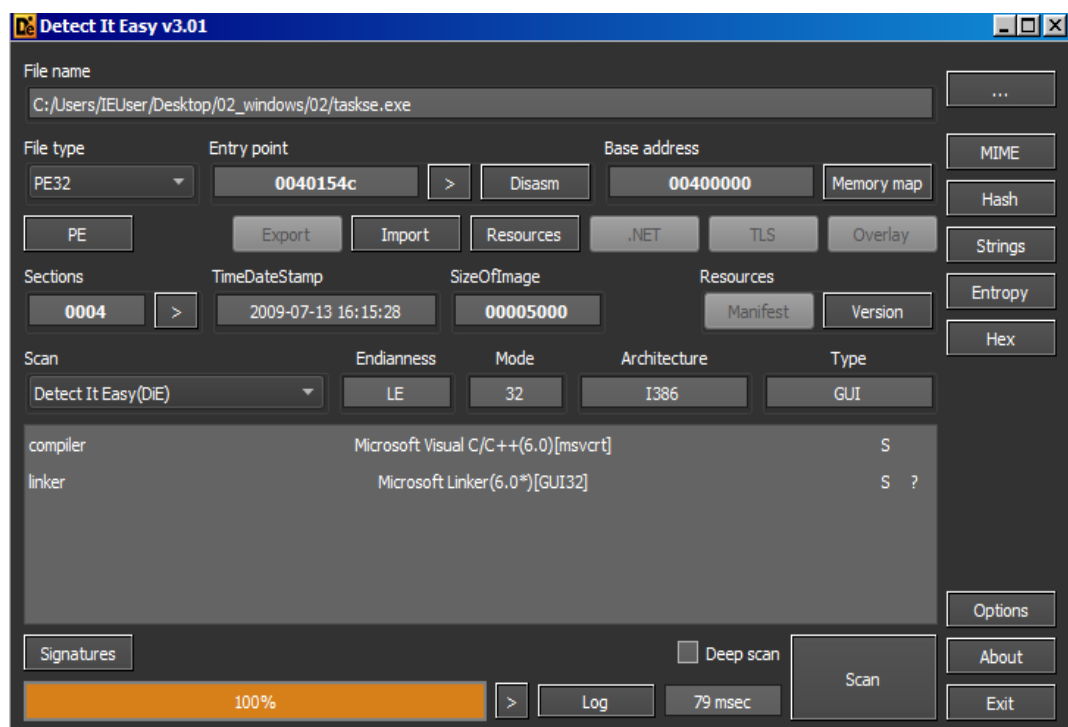
LAB_004050e8                                XREF[1]: 004050c1(j)
04050e8 c7 46 18      MOV     dword ptr [ESI + 0x18],s_empty_distance_tree_w... = "empty distance tree with leng...
          50 f7 40 00
04050ef eb 1a          JMP     LAB_0040510b

LAB_004050f1                                XREF[1]: 0040507b(j)
04050f1 83 f8 fd    CMP     EAX,-0x3
04050f4 75 09          JNZ     LAB_004050ff
04050f6 c7 46 18      MOV     dword ptr [ESI + 0x18],s_oversubscribed_litera... = "oversubscribed literal/length..."
          2a f7 40 00

```

Analiza plików wyodrębnionych przez główny plik:

- taskse.exe



DiE nie zidentyfikował badanego programu jako spakowany  
 Data kompilacji wskazuje na 2009-07-12 17:12:07 w porównaniu do spakowanego pliku głównego z datą kompilacji 2010-11-20 01:05:05 (daty prawdopodobnie są przekłamanie - WannaCry pochodzi z 2017 roku i w tym samym roku ujawniono exploit EternalBlue, który jest przez ten program używany)

Offset	Name	Func. Count	Bound?	OriginalFirstThunk	TimeDateStamp	Forwarder	NameRVA	FirstThunk
2110	KERNEL32.dll	11	FALSE	2160	0	0	2294	2000
2124	MSVCP60.dll	9	FALSE	2190	0	0	2540	2030
2138	MSVCRT.dll	18	FALSE	21B8	0	0	258E	2058

KERNEL32.dll [ 11 entries ]						
Call via	Name	Ordinal	Original Thunk	Thunk	Forwarder	Hint
2000	GetTempPathW	-	2204	2204	-	1D6
2004	GetWindowsDirec...	-	2214	2214	-	1F4
2008	DeleteFileW	-	222C	222C	-	84
200C	FindClose	-	223A	223A	-	CE
2010	FindNextFileW	-	2246	2246	-	DD
2014	FindFirstFileW	-	2256	2256	-	D5
2018	Sleep	-	2268	2268	-	356
201C	GetDriveTypeW	-	2270	2270	-	154
2020	GetLogicalDrives	-	2280	2280	-	178
2024	GetModuleHandleA	-	2654	2654	-	17F
2028	GetStartupInfoA	-	2668	2668	-	1B7

Offset	Name	Func. Count	Bound?	OriginalFirstThunk	TimeDateStamp	Forwarder	NameRVA	FirstThunk
2110	KERNEL32.dll	11	FALSE	2160	0	0	2294	2000
2124	MSVCP60.dll	9	FALSE	2190	0	0	2540	2030
2138	MSVCRT.dll	18	FALSE	21B8	0	0	258E	2058

#### MSVCP60.dll [ 9 entries ]

Call via	Name	Ordinal	Original Thunk	Thunk	Forwarder	Hint
2030	?_Eos@?\$basic_s...	-	2348	2348	-	348
2034	?_Grow@?\$basic...	-	2396	2396	-	393
2038	?_Tidy@?\$basic...	-	23E8	23E8	-	3F9
203C	?assign@?\$basic...	-	2438	2438	-	422
2040	?npos@?\$basic_s...	-	2494	2494	-	662
2044	?_Split@?\$basic...	-	24DE	24DE	-	3F3
2048	?_Xran@std@Y...	-	252C	252C	-	406
204C	??1?\$basic_string...	-	22A2	22A2	-	EA
2050	?_C@?1??_Nullst...	-	22EC	22EC	-	32E

Offset	Name	Func. Count	Bound?	OriginalFirstThunk	TimeDateStamp	Forwarder	NameRVA	FirstThunk
2110	KERNEL32.dll	11	FALSE	2160	0	0	2294	2000
2124	MSVCP60.dll	9	FALSE	2190	0	0	2540	2030
2138	MSVCRT.dll	18	FALSE	21B8	0	0	258E	2058

#### MSVCRT.dll [ 18 entries ]

Call via	Name	Ordinal	Original Thunk	Thunk	Forwarder	Hint
2058	__CxxFrameHan...	-	2562	2562	-	49
205C	??2@YAPAXI@Z	-	2576	2576	-	F
2060	free	-	2586	2586	-	25E
2064	_exit	-	259A	259A	-	D3
2068	_XcptFilter	-	25A2	25A2	-	48
206C	swprintf	-	254C	254C	-	2CB
2070	_acmdln	-	25B8	25B8	-	8F
2074	__getmainargs	-	25C2	25C2	-	58
2078	_initterm	-	25D2	25D2	-	10F
207C	__setusermatherr	-	25DE	25DE	-	83
2080	__adjust_fdiv	-	25F2	25F2	-	9D
2084	__p__commode	-	2602	2602	-	6A
2088	__p__fmode	-	2612	2612	-	6F
208C	__set_app_type	-	2620	2620	-	81
2090	__except_handler3	-	2632	2632	-	CA
2094	__controlfp	-	2646	2646	-	B7
2098	exit	-	25B0	25B0	-	249
209C	wcslen	-	2558	2558	-	2E6

```

1
2 1 VERSIONINFO
3 FILEVERSION 6,1,7600,16385
4 PRODUCTVERSION 6,1,7600,16385
5 FILEOS 0x40004
6 FILETYPE 0x1
7 {
8 BLOCK "StringFileInfo"
9 {
10     BLOCK "040904B0"
11     {
12         VALUE "CompanyName", "Microsoft Corporation"
13         VALUE "FileDescription", "SQL Client Configuration Utility EXE"
14         VALUE "FileVersion", "6.1.7600.16385 (win7_rtm.090713-1255)"
15         VALUE "InternalName", "cliconfg.exe"
16         VALUE "LegalCopyright", "\xA9 Microsoft Corporation. All rights reserved."
17         VALUE "OriginalFilename", "cliconfg.exe"
18         VALUE "ProductName", "Microsoft\xAE Windows\xAE Operating System"
19         VALUE "ProductVersion", "6.1.7600.16385"
20     }
21 }
22
23 BLOCK "VarFileInfo"
24 {
25     VALUE "Translation", 0x0409 0x04B0
26 }
27 }

```

ResourceHacker wskazuje na plik cliconfg.exe (wiele rodzajów malware'u identyfikuje się jako ten plik Windowsa aby ukryć swoją obecność w systemie)

## funkcja WinMain

sprawdza czy przekazano argumenty do programu

```

.text:00401510          ; int __stdcall WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nShowCmd)
.text:00401510          _WinMain@16 proc near
.text:00401510
.text:00401510          hInstance= dword ptr 4
.text:00401510          hPrevInstance= dword ptr 8
.text:00401510          lpCmdLine= dword ptr 0Ch
.text:00401510          nShowCmd= dword ptr 10h
.text:00401510
.text:00401510 000 FF 15 24 20 40 00 call    ds:__p__argc ; sprawdzi ilość przekazanych argumentów
.text:00401516 000 83 38 02          cmp     dword ptr [eax], 2 ; Compare Two Operands
.text:00401519 000 7D 05          jge     short loc_401520 ; powinny być 2 lub więcej argumentów

```

WinMain wywołuje główną funkcję - to na jej analizie się skupimy

## funkcja sub\_401420

ładuje bibliotekę Wtsapi32.dll

```

.text:00401427 020 33 ED          xor     ebp, ebp ; zerowanie rejestru ebp
.text:00401429 020 68 88 31 40 00 push    offset aWtsapi32Dll_0 ; "Wtsapi32.dll"
.text:0040142E 024 89 6C 24 1C    mov     [esp+24h+var_8], ebp
.text:00401432 024 FF 15 08 20 40 00 call    ds:LoadLibraryA ; Indirect Call Near Procedure
.text:00401438 020 8B F0          mov     esi, eax
.text:0040143A 020 3B F5          cmp     esi, ebp ; cmp esi, 0 (bo ebp jest wyzerowany), czyli sprawdzamy czy funkcja zwróciła NULL
.text:0040143C 020 75 08          jnz     short loc_401449 ; Jump if Not Zero (ZF=0)

```

handle do niej zapisywany jest do rejestru esi

Następnie GetProcAddress zwraca adres funkcji WTSEnumerateSession,



```

.text:00401449          loc_401449:
.text:00401449 020 8B 1D 04 20 40 00 mov     ebx, ds:GetProcAddress
.text:0040144F 020 68 70 31 40 00 push    offset aWtsenumeratase ; "WTSEnumerateSessionsA"
.text:00401454 024 56          push    esi ; hModule
.text:00401455 028 FF D3          call     ebx ; GetProcAddress ; Indirect Call Near Procedure
.text:00401457 020 8B F8          mov     edi, eax
.text:00401459 020 3B FD          cmp     edi, ebp ; Compare Two Operands
.text:0040145B 020 75 0B          jnz     short loc_401468 ; Jump if Not Zero (ZF=0)

```

zwrócony adres jest zapisywany do rejestru edi,

oraz adres funkcji WTSFreeMemory

```

.text:00401468          loc_401468:
.text:00401468 020 68 60 31 40 00 push    offset aWtsfreememory ; "WTSFreeMemory"
.text:0040146D 024 56          push    esi ; hModule
.text:0040146E 028 FF D3          call     ebx ; GetProcAddress ; Indirect Call Near Procedure
.text:00401470 020 8B F0          mov     esi, eax
.text:00401472 020 3B F5          cmp     esi, ebp ; Compare Two Operands
.text:00401474 020 89 74 24 1C mov     [esp+20h+var_4], esi
.text:00401478 020 75 0B          jnz     short loc_401485 ; Jump if Not Zero (ZF=0)

```

zwrócony adres jest zapisywany do rejestru esi, a następnie na stosie (zmienna lokalna var\_4)

W obu przypadkach rejestr esi zawiera w sobie wciąż handle do załadowanej biblioteki Wtsapi32.dll

Poniżej widzimy wywołanie funkcji z rejestru edi, a więc chodzi o funkcję WTSEnumerateSession:

```

.text:00401485          loc_401485:
.text:00401485 020 8D 44 24 14 lea     eax, [esp+20h+var_C] ; Load Effective Address
.text:00401489 020 8D 4C 24 10 lea     ecx, [esp+20h+var_10] ; Load Effective Address
.text:0040148D 020 50          push    eax ; *pCount - pointer to the number of WTS_SESSION_INFO structures
.text:0040148E 024 51          push    ecx ; ppSessionInfo - pointer to an array of WTS_SESSION_INFO structures
.text:0040148E          ; to free the returned buffer, call the WTSFreeMemory
.text:0040148F 028 6A 01          push    1 ; Version - must be 1
.text:00401491 02C 55          push    ebp ; Reserved - must be 0
.text:00401492 030 55          push    ebp ; hServer - handle to the RD Session Host server
.text:00401493 034 89 6C 24 24 mov     [esp+34h+var_10], ebp
.text:00401497 034 89 6C 24 28 mov     [esp+34h+var_C], ebp
.text:0040149B 034 FF D7          call     edi ; call WTSEnumerateSessionA
.text:0040149D 020 3B 6C 24 10 cmp     [esp+20h+var_10], ebp ; Compare Two Operands
.text:004014A1 020 75 0B          jnz     short loc_4014AE ; Jump if Not Zero (ZF=0)

```

Z dokumentacji:

“Retrieves a list of sessions on a Remote Desktop Session Host (RD Session Host) server.” - program zwraca zatem listę sesji RDP

Zatem wiemy, że wynik wykonania tej funkcji zostanie zapisany pod adres z rejestru ecx - jest to zmienna lokalna var\_10. Dlatego po wykonaniu funkcji program sprawdza czy ten adres jest różny od zera i tylko jeśli tak wykonuje skok dalej.

Następna struktura kodu wygląda jak pętla - mamy skok powrotny i inkrementację zmiennej.



```

.text:004014C4          loc_4014C4:
.text:004014C4  020 8B 54 24 10      mov     edx, [esp+20h+var_10]
.text:004014C8  020 6A 00            push    0
.text:004014CA  024 6A 05            push    5
.text:004014CC  028 8B 04 16        mov     eax, [esi+edx]
.text:004014CF  028 50              push    eax
.text:004014D0  02C 53              push    ebx
.text:004014D1  030 E8 2A FB FF FF   call    sub_401000 ; Call Procedure
.text:004014D6  030 83 C4 10        add     esp, 10h ; Add
.text:004014D9  020 85 C0            test    eax, eax ; Logical Compare
.text:004014DB  020 75 04            jnz     short loc_4014E1 ; Jump if Not Zero (ZF=0)

```

Te parametry to:

- ebx, czyli wspomniany parametr arg\_0
- eax, jest to suma rejestrów esi (zero na początku z każdą iteracją zwiększany o 0Ch) oraz edx, prawdopodobnie ten sam parametr co ppSessionInfo w poprzedniej funkcji -> "pointer to an array of WTS\_SESSION\_INFO structures"
- 5
- 0

Na koniec pętli widzimy że program wywołuje funkcję Sleep z wartością 100, następnie inkrementuje rejestr edi oraz dodaje 0Ch do esi, po czym sprawdza czy edi jest mniejszy od eax, jeśli tak wykonuje następną iterację pętli

```

.text:004014E1          loc_4014E1:                ; dwMilliseconds
.text:004014E1  020 6A 64            push    64h ; 'd'
.text:004014E3  024 FF D5            call    ebp ; Sleep ; Indirect Call Near Procedure
.text:004014E5  020 8B 44 24 14      mov     eax, [esp+20h+var_C] ; ilość iteracji pętli
.text:004014E9  020 47              inc     edi ; Increment by 1
.text:004014EA  020 83 C6 0C        add     esi, 0Ch ; Add
.text:004014ED  020 3B F8            cmp     edi, eax ; Compare Two Operands
.text:004014EF  020 72 D3            jb      short loc_4014C4 ; Jump if Below (CF=1)

```

Uwaga: bardzo dużo zmiennych lokalnych jest wrzucanych i pobieranych ze stosu, przez co trudno jest śledzić ich wartość. Najlepiej przyjrzyć się temu wykorzystując metody analizy dynamicznej (debugger).

Po zakończeniu pętli wywoływana jest jeszcze funkcja WTSFreeMemory (wyczyszczenie zaalokowanej pamięci dla struktur danych potrzebnych do przechowywania informacji o sesjach RDP)

### funkcja sub\_401000

Na początku procedura korzystając z GetProcAddress ładuje wiele funkcji, które wykorzystywane są w tym programie:

```

if ((pHVar1 != (HMODULE)0x0) ||
    (pHVar1 = LoadLibraryA(s_advapi32.dll_00403150), pHVar1 != (HMODULE)0x0)) {
    /* Tu są ładowane wszystkie funkcje */
    OpenProcessToken_addr = GetProcAddress(pHVar1, s_OpenProcessToken_0040313c);
    LookupPrivilegeValueA_addr = GetProcAddress(pHVar1, s_LookupPrivilegeValueA_00403124);
    AdjustTokenPrivileges_addr = GetProcAddress(pHVar1, s_AdjustTokenPrivileges_0040310c);
    DuplicateTokenEx_addr = GetProcAddress(pHVar1, s_DuplicateTokenEx_004030f8);
    CreateProcessAsUserA_addr = GetProcAddress(pHVar1, s_CreateProcessAsUserA_004030e0);
    if (((OpenProcessToken_addr != (FARPROC)0x0) &&
        (((LookupPrivilegeValueA_addr != (FARPROC)0x0 &&
            (AdjustTokenPrivileges_addr != (FARPROC)0x0)) && (DuplicateTokenEx_addr != (FARPROC)0x0))
            && (CreateProcessAsUserA_addr != (FARPROC)0x0)))) &&
        (pHVar1 = GetModuleHandleA(s_kernel32.dll_004030d0), pHVar1 != (HMODULE)0x0 ||
            (pHVar1 = LoadLibraryA(s_kernel32.dll_004030d0), pHVar1 != (HMODULE)0x0)))) {
        WTSGetActiveConsoleSessionId_addr =
            GetProcAddress(pHVar1, s_WTSGetActiveConsoleSessionId_004030b0);
        GetCurrentProcess_addr = GetProcAddress(pHVar1, s_GetCurrentProcess_0040309c);
        CloseHandle_addr = GetProcAddress(pHVar1, s_CloseHandle_00403090);
        if ((WTSGetActiveConsoleSessionId_addr != (FARPROC)0x0) &&
            (((GetCurrentProcess_addr != (FARPROC)0x0 && (CloseHandle_addr != (FARPROC)0x0)) &&
                ((pHVar1 = GetModuleHandleA(s_userenv.dll_00403084), pHVar1 != (HMODULE)0x0 ||
                    (pHVar1 = LoadLibraryA(s_userenv.dll_00403084), pHVar1 != (HMODULE)0x0)))))) {
            CreateEnvironmentBlock_addr = GetProcAddress(pHVar1, s_CreateEnvironmentBlock_0040306c);
            DestroyEnvironmentBlock_addr = GetProcAddress(pHVar1, s_DestroyEnvironmentBlock_00403054);
            if (((CreateEnvironmentBlock_addr != (FARPROC)0x0) &&
                (DestroyEnvironmentBlock_addr != (FARPROC)0x0)) &&
                ((pHVar1 = GetModuleHandleA(s_wtsapi32.dll_00403044), pHVar1 != (HMODULE)0x0 ||
                    (pHVar1 = LoadLibraryA(s_wtsapi32.dll_00403044), pHVar1 != (HMODULE)0x0)))) &&
                (WTSQueryUserToken_addr = GetProcAddress(pHVar1, s_WTSQueryUserToken_00403030),
                    WTSQueryUserToken_addr != (FARPROC)0x0)) {
                /* Tu się zaczynają wywołania funkcji */
                local_8 = 0;
                iVar2 = (*GetCurrentProcess_addr)(0x28, local_8);
            }
        }
    }
}

```

Następnie następuje fragment kodu, w którym program otwiera tokeny procesów i ustawia odpowiednie uprawnienia do ich modyfikacji.

Kolejno korzystając z SessionID (uzyskane w poprzedniej funkcji) program dokonuje duplikacji (personifikacji tokenu).

```

else {
    SessionId = WTSSESSION_ID;
}
iVar4 = (*WTSQueryUserToken_addr)(SessionId, &phToken);
if ((iVar4 != 0) &&
    (bVar1 = (*DuplicateTokenEx_addr)
        (phToken, 0x20000000, (LPSECURITY_ATTRIBUTES)0x0,
        SecurityIdentification, TokenPrimary, &local_78),

```

Z dokumentacji: *"Impersonation is the ability of a process to take on the security attributes of another process."*

Oznacza to, że duplikując tokeny sesji RDP malware będzie mógł otworzyć nową sesję z tymi samymi uprawnieniami co proces, od którego pochodził pierwszy token.

Dokładnie to dzieje się poniżej:

```

local_e0 = 0x44;
local_dc[1] = s_winsta0\default_00403010;
local_b0 = param_3;
iVar4 = (*CreateEnvironmentBlock_addr) (&lpEnvironment, phNewToken, 1);
if ((iVar4 != 0) &&
    (iVar4 = (*CreateProcessAsUserA_addr)
              (phNewToken, ApplicationName, 0, 0, 0, 0, 0x400, lpEnvironment, 0,
               &local_e0, &local_30), iVar4 != 0)) {
    if (param_4 != 0) {
        WaitForSingleObject(local_30, 0xffffffff);
    }
    local_8 = 0xffffffff;
    FUN_00401398();
    *in_FS_OFFSET = local_14;
    return 0;
}

```

Malware tworzy nowy proces o nazwie ApplicationName (parametr przekazywany w wywołaniu funkcji jako arg\_0 czyli też jako parametr dla pliku taskse.exe) korzystając ze zduplikowanego tokenu (phNewToken) i zmiennych środowiskowych skopiowanych od użytkownika danej sesji RDP (lpEnvironment).

#### Podsumowanie

Działanie programu taskse.exe odpowiada temu czego możemy dowiedzieć się z zewnętrznych źródeł. Jego zadaniem jest znalezienie otwartych sesji RDP, a następnie przeiterowanie po ich liście i duplikacja ich tokenów i uprawnień. Dzięki temu malware może stworzyć własne procesy na tych samych prawach co otwarte sesje RDP. Jest to być może fragment malware'u odpowiedzialny za rozpropagowanie złośliwego oprogramowania po sieci wewnętrznej.

## 2. taskdl.exe

### WinMain

```

DiskDrives = GetLogicalDrives();
iter = 25;
do {
    local_4 = DAT_00403064;
    lpRootPathName = DAT_00403060 & 0xffff0000 | (uint)(ushort)((short)iter + 0x41);
    if ((DiskDrives >> ((byte)iter & 0x1f) & 1) != 0) {
        DiskType = GetDriveTypeW((LPCWSTR)&lpRootPathName);
        if (DiskType != 4) {
            /* if DiskType != DRIVE_REMOTE[the drive is a remote/network drive] */
            FUN_00401080(iter);
            Sleep(10);
        }
    }
    iter = iter + -1;
    /* wykonuje się dopóki iter > 1, czyli wykona się 24 razy */
} while (1 < iter);
return 0;

```

Funkcja WinMain odczytuje wszystkie dostępne w urządzeniu dyski oraz ich typ następnie jeśli rodzaj dysku jest inny niż dysk zdalny wykonuje funkcję 401080 (dodać nazwę)

sub\_401080

Funkcja RetrieveTempFolderPath zwraca ścieżkę do folderu Temp

```

local_4 = (HANDLE)0x0;
local_690 = 0;
RetrieveTempFolderPath(iter, PathToTempFolder);
swprintf(PathToFind, (size_t)u_5s\*%s_00403040, (wchar_t *)PathToTempFolder, u_.WNCRYT_00403050);
hFindFile = FindFirstFileW(PathToFind, (LPWIN32_FIND_DATAW)local_25c);
pbVar1 = local_684;
pbVar7 = local_688;

```

Funkcja znajduje ścieżkę do pliku o rozszerzeniu .WNCRYT następnie plik ten jest usuwany (są to pliki tymczasowe z oryginalnych plików, które malware tworzy w folderze głównym w czasie szyfrowania - ich zawartość jest następnie wpiswana na miejsce oryginalnej zawartości)

Wywołania nieznanymi funkcji (ich nazwy są zobfuskowane) - konieczna dalsza analiza dynamiczna

```

call edi ; swprintf
add esp, 10h
lea ecx, [esp+6A4h+var_67C]
mov [esp+6A4h+var_67C], bl
push 0
call ds:?$_Tidy@$basic_string@GU?$char_traits@G@std@@V?$allocator@G@2@@@std@@AAEX_N@Z ; std::wstring::_Tidy(bool)
lea ecx, [esp+6A4h+Buffer]
push ecx ; String
call ds:wcslen
add esp, 4
mov esi, eax
lea ecx, [esp+6A4h+var_67C]
push 1
push esi
call ds:?$_Grow@$basic_string@GU?$char_traits@G@std@@V?$allocator@G@2@@@std@@AAE_NI_N@Z ; std::wstring::_Grow(uint,bool)
test al, al
jz short loc_4011BC

```

**Detect It Easy v3.01**

File name: C:/Users/IEUser/Desktop/02\_windows/02/taskdl.exe

File type: PE32 Entry point: 004018f6 Base address: 00400000

PE Export Import Resources .NET TLS Overlay

Sections: 0004 TimeDateStamp: 2009-07-13 17:12:07 SizeOfImage: 00005000 Resources: Manifest Version

Scan: Detect It Easy (DiE) Endianness: LE Mode: 32 Architecture: I386 Type: GUI

compiler: Microsoft Visual C++(6.0)[msvcr] S ?

linker: Microsoft Linker(6.0\*)[GUI32] S ?

Signatures: 100% Deep scan: [ ] Scan: 116 msec

Options About Exit

Program DiE wskazał że plik taskdl.exe nie został spakowany. Jego data kompilacji  
2009-07-13 16:15:28

Offset	Name	Func. Count	Bound?	OriginalFirstThunk	TimeDateStamp	Forwarder	NameRVA	FirstThunk
207C	KERNEL32.dll	6	FALSE	20B8	0	0	216C	2000
2090	MSVCRT.dll	16	FALSE	20D4	0	0	21BC	201C

KERNEL32.dll [ 6 entries ]

Call via	Name	Ordinal	Original Thunk	Thunk	Forwarder	Hint
2000	WaitForSingleObj...	-	2118	2118	-	390
2004	GetProcAddress	-	212E	212E	-	1A0
2008	LoadLibraryA	-	2140	2140	-	252
200C	GetModuleHandleA	-	2150	2150	-	17F
2010	Sleep	-	2164	2164	-	356
2014	GetStartupInfoA	-	226E	226E	-	1B7

Offset	Name	Func. Count	Bound?	OriginalFirstThunk	TimeDateStamp	Forwarder	NameRVA	FirstThunk
207C	KERNEL32.dll	6	FALSE	20B8	0	0	216C	2000
2090	MSVCRT.dll	16	FALSE	20D4	0	0	21BC	201C

MSVCRT.dll [ 16 entries ]

Call via	Name	Ordinal	Original Thunk	Thunk	Forwarder	Hint
201C	_local_unwind2	-	218E	218E	-	13C
2020	__p__argv	-	21A0	21A0	-	63
2024	__p__argc	-	21AE	21AE	-	62
2028	_exit	-	21C8	21C8	-	D3
202C	_XcptFilter	-	21D0	21D0	-	48
2030	exit	-	21DE	21DE	-	249
2034	_except_handler3	-	217A	217A	-	CA
2038	__getmainargs	-	21F0	21F0	-	58
203C	_initterm	-	2200	2200	-	10F
2040	_setusermatherr	-	220C	220C	-	83
2044	__adjust_fdiv	-	2220	2220	-	9D
2048	__p__commode	-	2230	2230	-	6A
204C	__p__fmode	-	2240	2240	-	6F
2050	__set_app_type	-	224E	224E	-	81
2054	__controlfp	-	2260	2260	-	B7
2058	_acmdln	-	21E6	21E6	-	8F

MSVCRT.dll



```

1
2 1 VERSIONINFO
3 FILEVERSION 6,1,7600,16385
4 PRODUCTVERSION 6,1,7600,16385
5 FILEOS 0x40004
6 FILETYPE 0x1
7 {
8 BLOCK "StringFileInfo"
9 {
10     BLOCK "040904B0"
11     {
12         VALUE "CompanyName", "Microsoft Corporation"
13         VALUE "FileDescription", "waitfor - wait/send a signal over a network"
14         VALUE "FileVersion", "6.1.7600.16385 (win7_rtm.090713-1255)"
15         VALUE "InternalName", "waitfor.exe"
16         VALUE "LegalCopyright", "\xA9 Microsoft Corporation. All rights reserved."
17         VALUE "OriginalFilename", "waitfor.exe"
18         VALUE "ProductName", "Microsoft\xAE Windows\xAE Operating System"
19         VALUE "ProductVersion", "6.1.7600.16385"
20     }
21 }
22
23 BLOCK "VarFileInfo"
24 {
25     VALUE "Translation", 0x0409 0x04B0
26 }
27 }

```

ResourceHacker wskazuje na plik waitfor.exe

### c.wnry

```

PS C:\Users\IEUser\Desktop\02_windows\02> strings .\c.wnry
Strings v2.53 - Search for ANSI and Unicode strings in binary images.
Copyright (C) 1999-2016 Mark Russinovich
Sysinternals - www.sysinternals.com
gx7ekbenv2riucmf.onion;57g7spgrzlojinas.onion;xxlvbrloxvriy2c5.onion;76jdd2ir2embyv47.onion;cwwnhwhlz52maq7.onion;
https://dist.torproject.org/torbrowser/6.5.1/tor-win32-0.2.9.10.zip
PS C:\Users\IEUser\Desktop\02_windows\02>

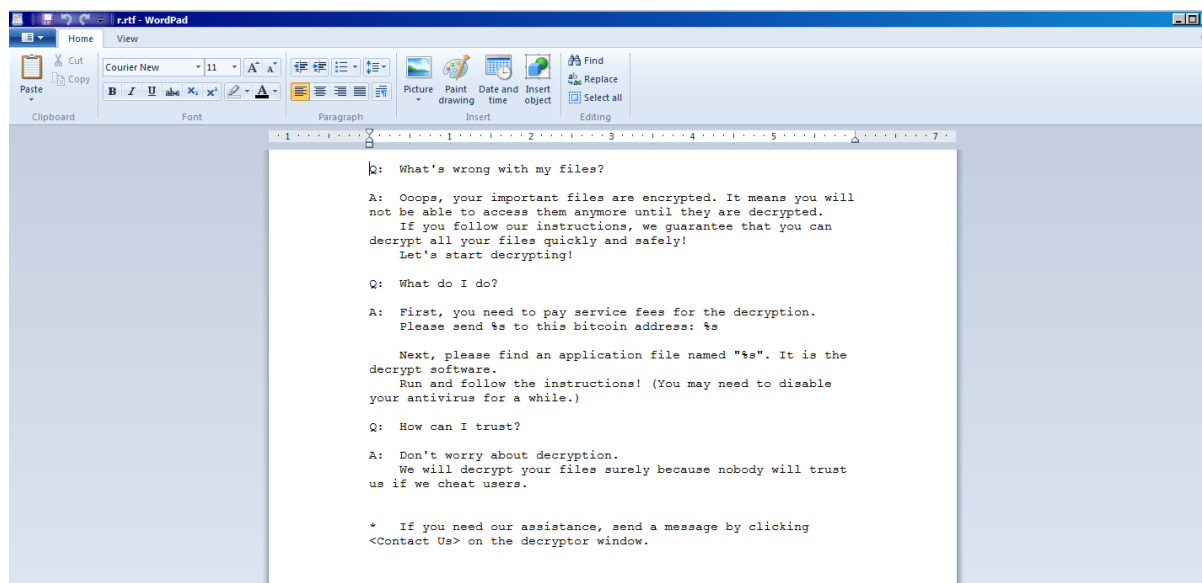
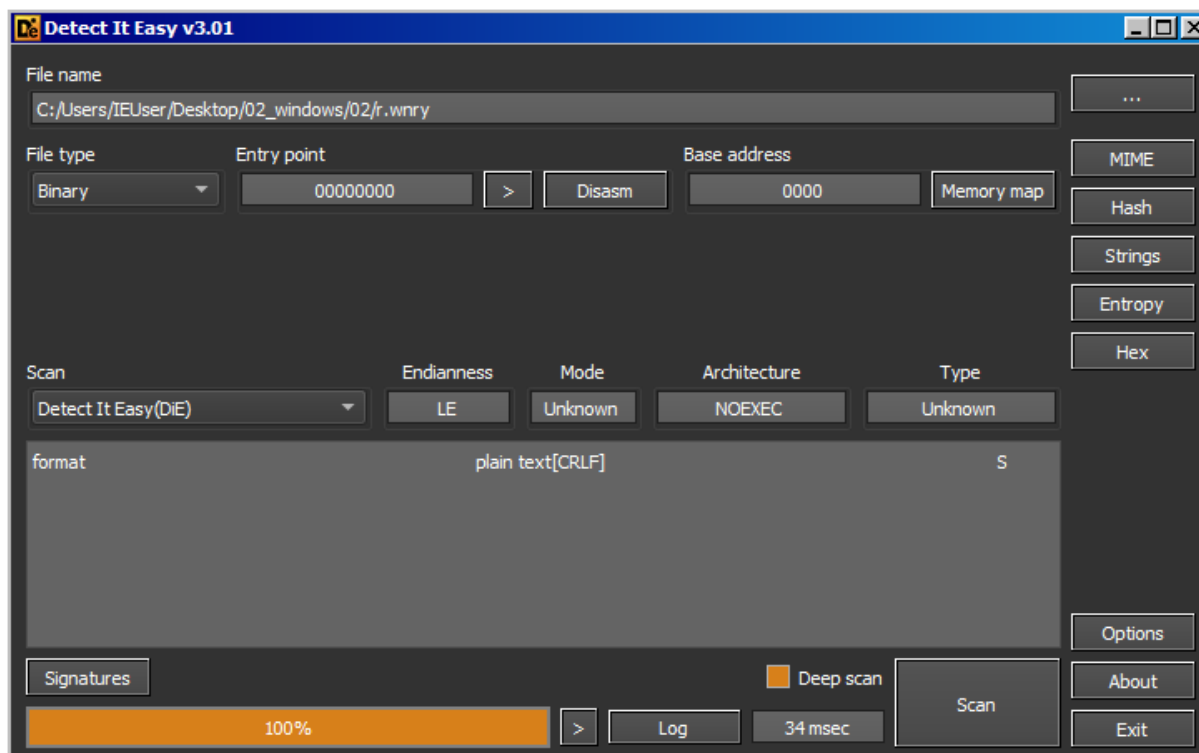
```

W tym pliku możemy zauważyć 5 adresów .onion, oraz adres https z którego zostanie pobrany plik TOR na Windowsa.

Adresy .onion:

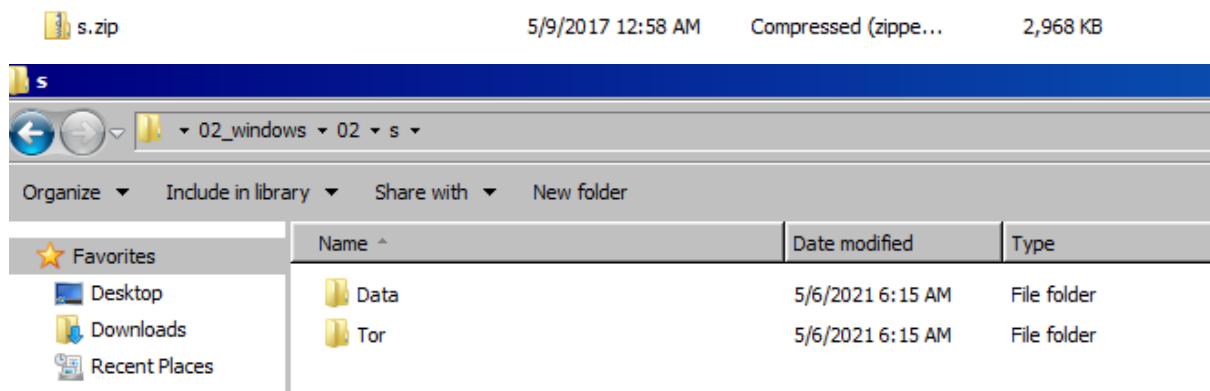
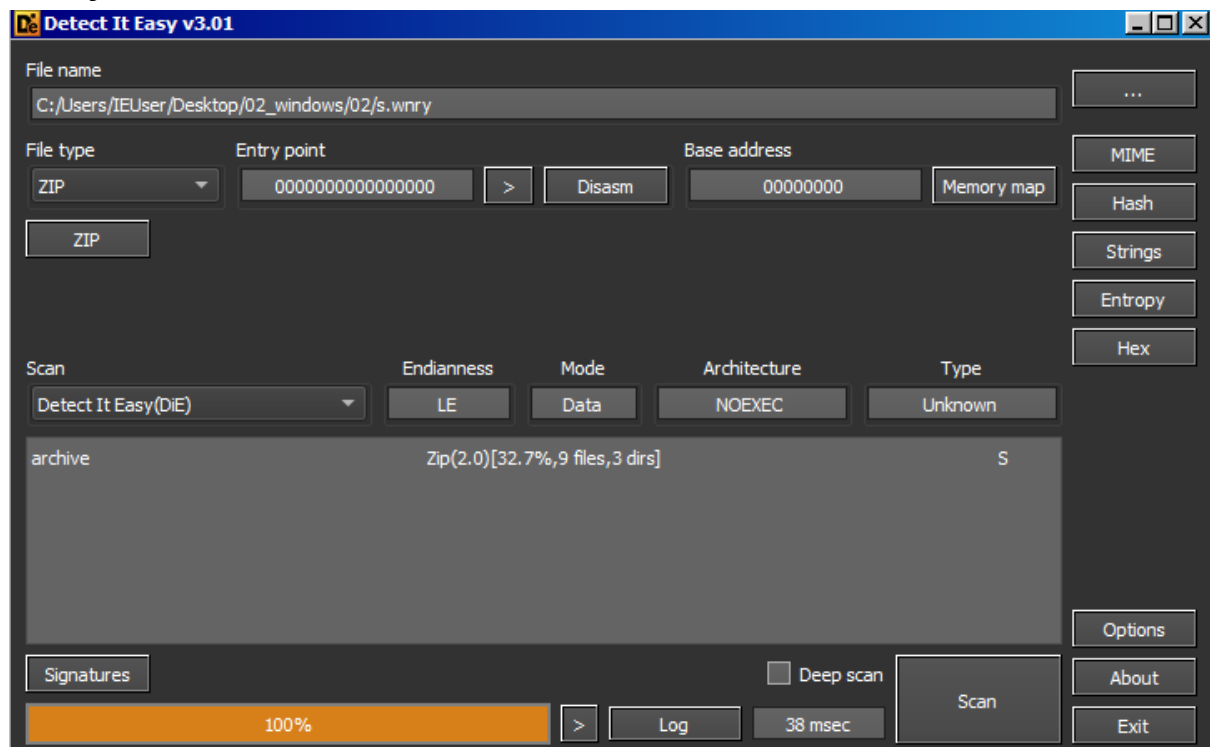
- gx7ekbenv2riucmf.onion
- 57g7spgrzlojinas.onion
- xxlvbrloxvriy2c5.onion
- 76jdd2ir2embyv47.onion
- cwwnhwhlz52maq7.onion

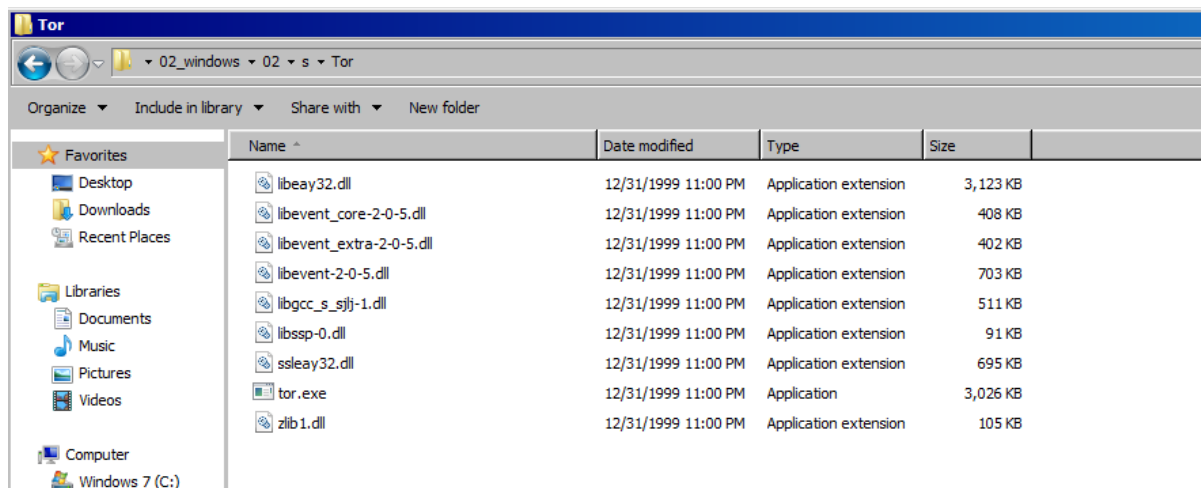
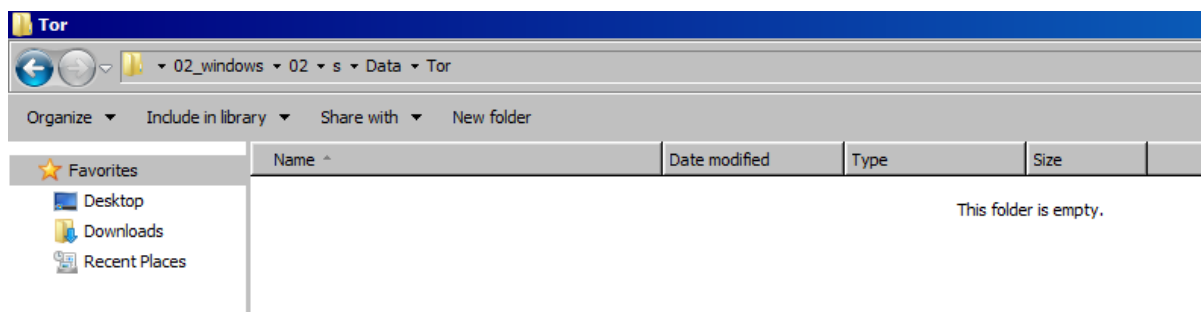
### r.wnry



W tym pliku najprawdopodobniej zawarte są instrukcje dla osoby która została zaatakowana z informacjami co musi zrobić aby odszyfrować dane

s.wnry





Detect-it-Easy rozpoznał plik s.wnry jako archiwum, więc rozpakowałem go i sprawdziłem jego zawartość. W tym archiwum są dwa foldery "Data" i "Tor". Folder Data posiada kolejny folder "Tor" ale narazie są puste więc wracamy do folderu głównego "Tor". W nim znajdują się prawdopodobnie pliki instalacyjne przeglądarki

## t.wnry

```
PS C:\Users\IEUser\Desktop\02_windows\02> strings .\t.wnry

Strings v2.53 - Search for ANSI and Unicode strings in binary images.
Copyright (C) 1999-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

WANACRY!
8'''
~><
*PdIf
#I!
zxPp
nhB>>
[d$
\!u
?hH"
dS%a
nEi?I
`1?
gjr$
,g1y+
iIt
:If<
rR<
```

Wydaje się, że jest to kolejny zaszyfrowany plik. Prawdopodobnie dowiemy się więcej o tym pliku podczas analizy dynamicznej

u.wnry

**Detect It Easy v3.01**

File name  
C:/Users/IEUser/Desktop/02\_windows/02/u.wnry

File type: PE32  
Entry point: 00413102  
Base address: 00400000

PE Export Import Resources .NET TLS Overlay

Sections: 0004  
TimeDateStamp: 2009-07-13 16:19:35  
SizeOfImage: 0003d000

Resources: Manifest Version

Scan: Detect It Easy(DiE)  
Endianness: LE  
Mode: 32  
Architecture: I386  
Type: GUI

library	MFC(4.2)[-]	S ?
compiler	EP:Microsoft Visual C/C++(6.0 (1720-9782))[EXE32]	S
compiler	Microsoft Visual C++(6.0)[msvcrt]	S ?
linker	Microsoft Linker(6.0*)[GUI32]	S ?

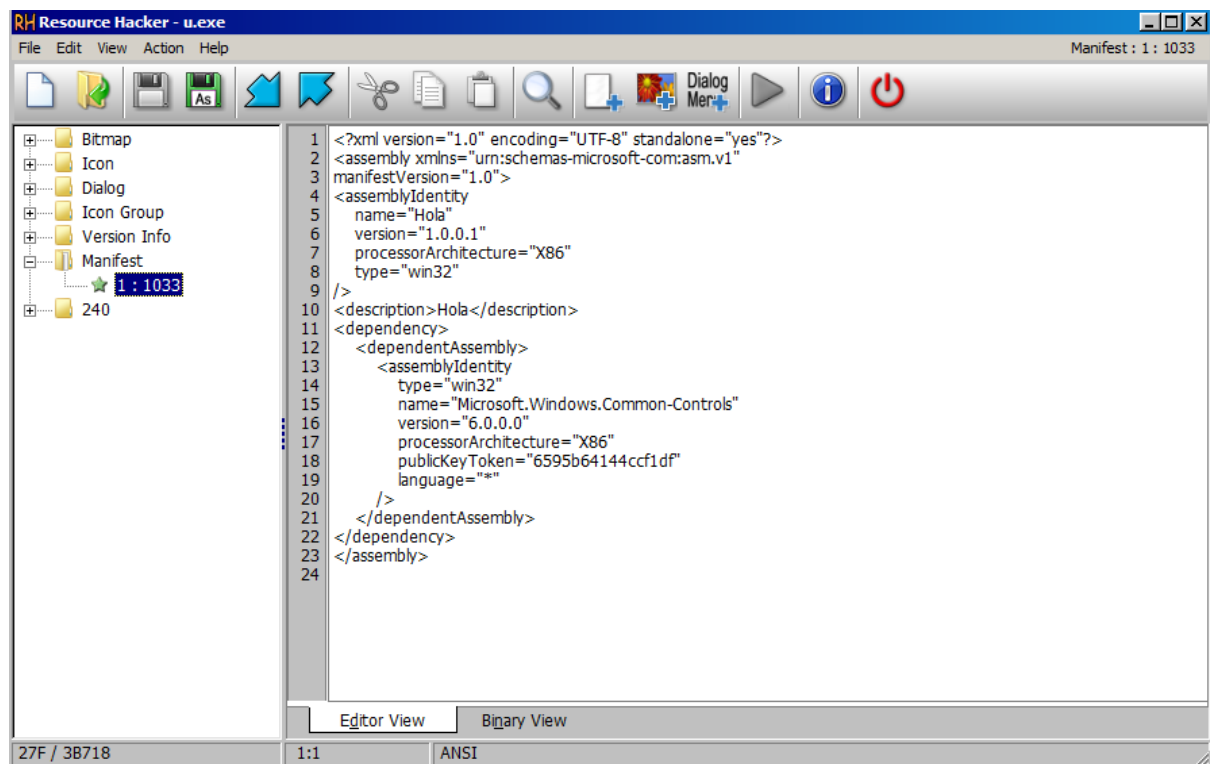
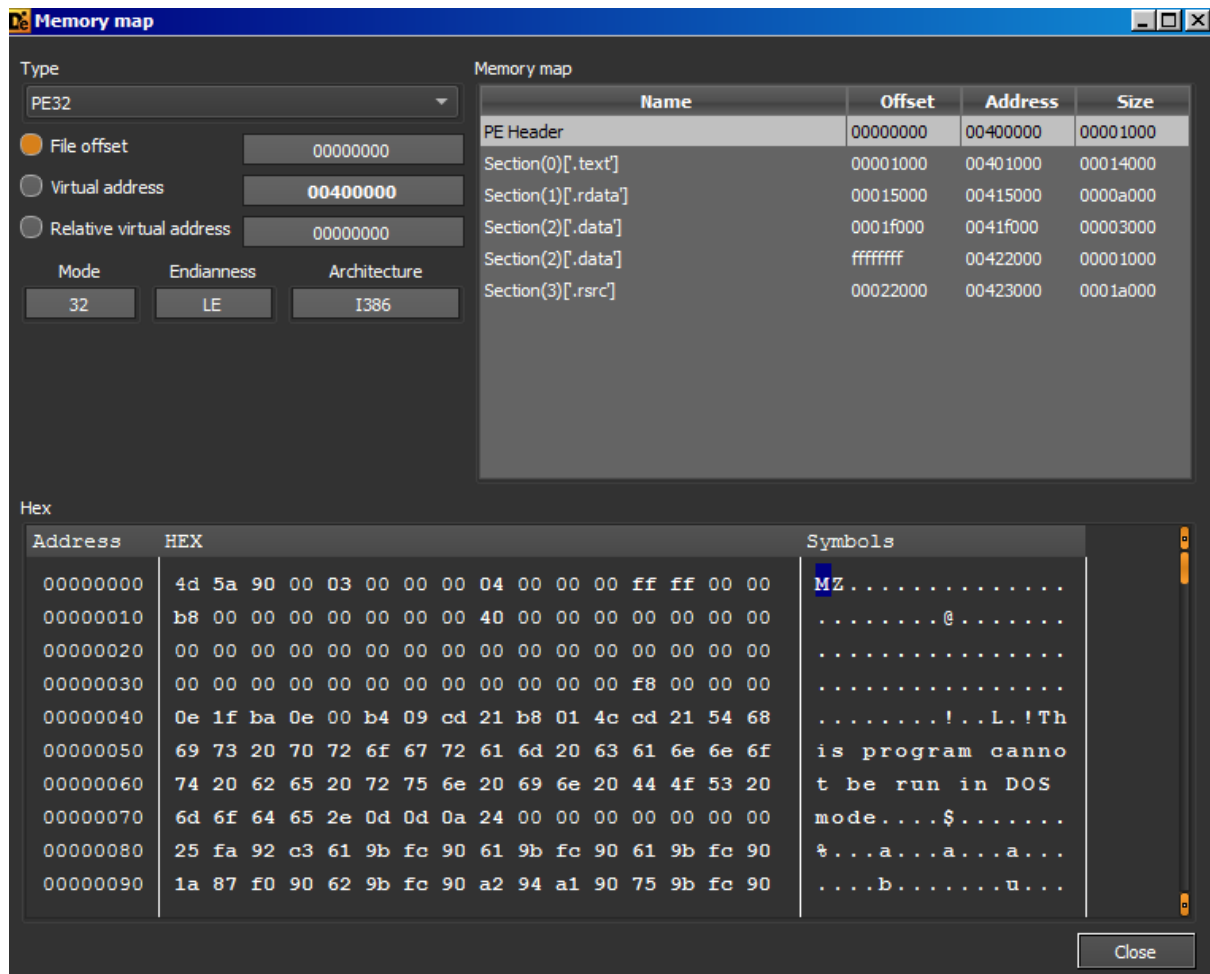
Signatures

100%

Log 128 msec

Scan

Options About Exit



Offset	Name	Func. Count	Bound?	OriginalFirstThunk	TimeDateStamp	Forwarder	NameRVA	FirstThunk
1CB98	MFC42.DLL	205	FALSE	1CE24	0	0	1D374	15174
1CBAC	MSVCRT.dll	58	FALSE	1D188	0	0	1D55A	15408
1CBC0	KERNEL32.dll	59	FALSE	1CD34	0	0	1DA86	15084
1CBD4	USER32.dll	35	FALSE	1D28C	0	0	1DCBA	155DC
1CBE8	GDI32.dll	19	FALSE	1CCE4	0	0	1DDFC	15034
1CBFC	ADVAPI32.dll	9	FALSE	1CCB0	0	0	1DEAE	15000
1CC10	SHELL32.dll	3	FALSE	1D27C	0	0	1DEF2	155CC
1CC24	COMCTL32.dll	2	FALSE	1CCD8	0	0	1DF12	15028
1CC38	OLEAUT32.dll	1	FALSE	1D274	0	0	1DF20	155C4
1CC4C	urlmon.dll	1	FALSE	1D36C	0	0	1DF44	156BC
1CC60	MSVCP60.dll	10	FALSE	1D15C	0	0	1E1FE	154AC
1CC74	WS2_32.dll	17	FALSE	1D324	0	0	1E20A	15674
1CC88	WININET.dll	1	FALSE	1D31C	0	0	1E22C	1566C

Plik wymaga dalszej analizy.

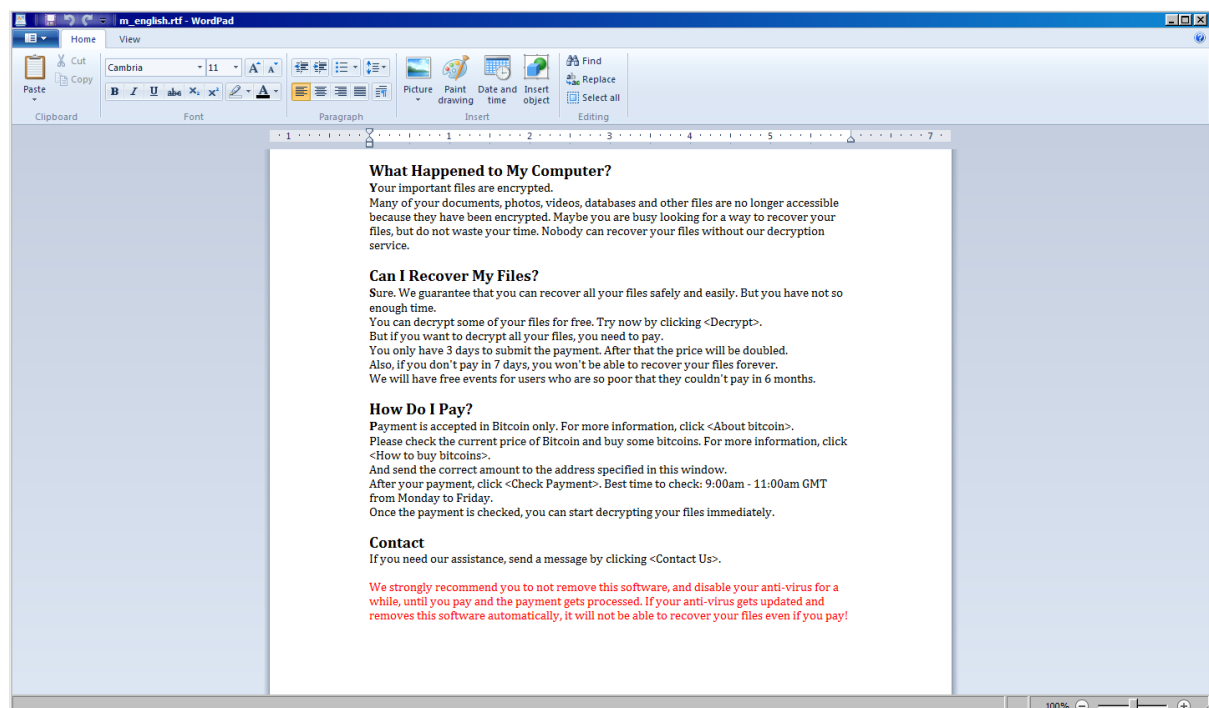
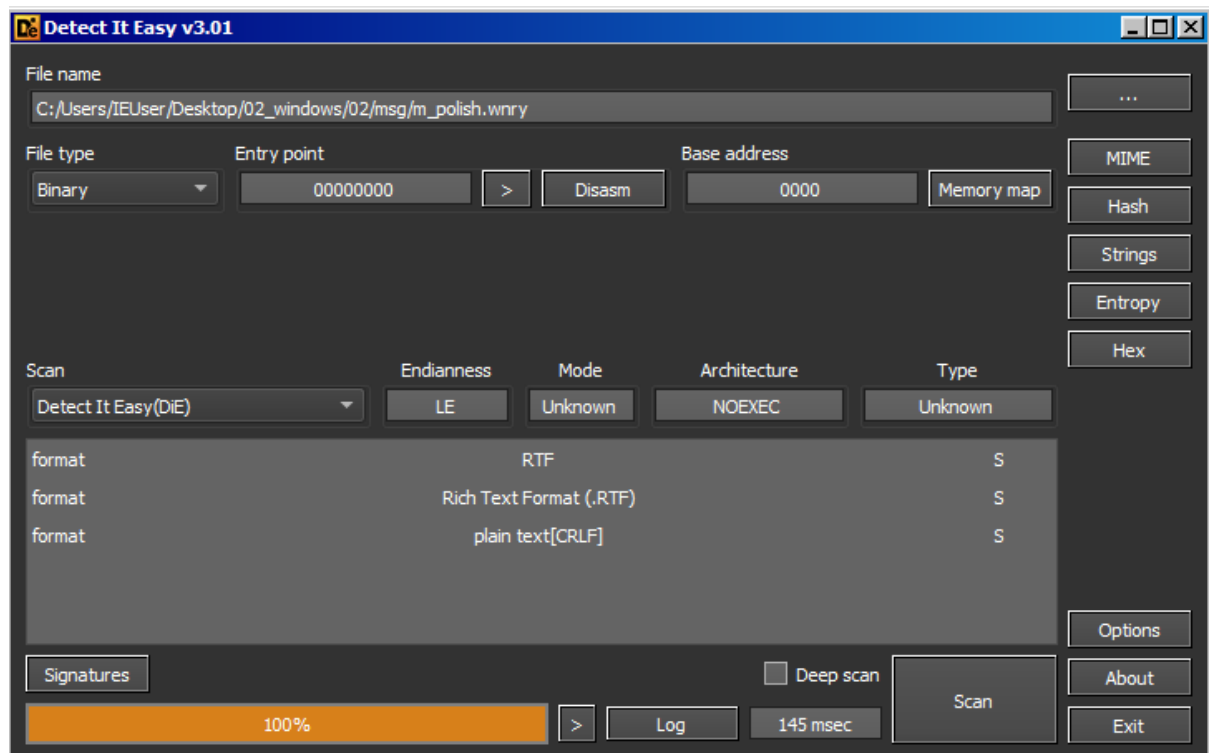
Z wyniku otrzymanego z DiE i ResourceHackera możemy wnioskować, że jest to w oryginale plik LODCTR.exe.

Importy:

- **MFC42.dll** - zawiera w sobie funkcje Microsoft Foundation Classes (MFC), które są używane przez Microsoft Visual Studio, jest ładowany do pamięci RAM. Jeżeli jest umiejscowiony w C:\Windows\System32 to jest bezpieczny, a jeżeli gdzie indziej to może być to Trojan
- **MSVCRT.dll** - standardowa biblioteka C dla Visual C++
- **KERNEL32.dll** - udostępnia aplikacjom podstawowe API Win32, zarządzanie pamięcią, operacje wejścia/wyjścia, tworzenie procesów i wątków oraz funkcje synchronizacji
- **USER32.dll** - Tworzy i obsługuje standardowe elementy interfejsu użytkownika (pulpit, okna, menu). Umożliwia implementację GUI
- **GDI32.dll** - eksportuje funkcje Graphics Device Interface (GDI). Może być użyty do rysowania, linii tekstu, zarządzania czcionkami
- **ADVAPI32.dll** - zapewnia wywołania zabezpieczeń i funkcje służące do manipulowania rejestrem systemu Windows
- **SHELL32.dll** - biblioteka zawierająca funkcje Windows Shell API, które są używane podczas otwierania stron internetowych i plików
- **COMCTL32.dll** - implementuje standardowe kontrolki systemu Windows, takie jak: Otwórz plik, Zapisz, Zapisz jako. Wywołuje funkcję z USER32.dll oraz GDI32.dll dzięki czemu zarządza elementami graficznymi oraz zbiera dane wejściowe
- **OLEAUT32.dll** - współdzielony plik instalowany przez system operacyjny i używany przez program instalacyjny. To bardziej przypomina „kawałek” instalatora, który jest potrzebny do zakończenia instalacji lub instalacji programu.
- **urlmon.dll** - moduł zawierający funkcję używane przez Microsoft OLE (Object Linking and Embedding)
- **MSVCP60.dll** - zawiera standardowe funkcje Microsoft C Runtime Library, takie jak printf, memcpy, cos

- **WS2\_32.dll** - implementuje Winsock API, który zapewnia funkcje sieciowe TCP/IP
- **WININET.dll** - zawiera funkcje internetowe używane przez aplikacje Windows

msg\m\_english.wnry



Program DiE wskazał nam, że jest to plik .rtf zatem zmieniłem końcówkę nazwy aby otworzyć go za pomocą wordpada. Po otwarciu pliku widzimy, że znajdują się tu



informacje dla ofiary co może zrobić aby przywrócić swoje dane z dysku oraz ile ma czasu przed ich ostateczną utratą

### **Podsumowanie**

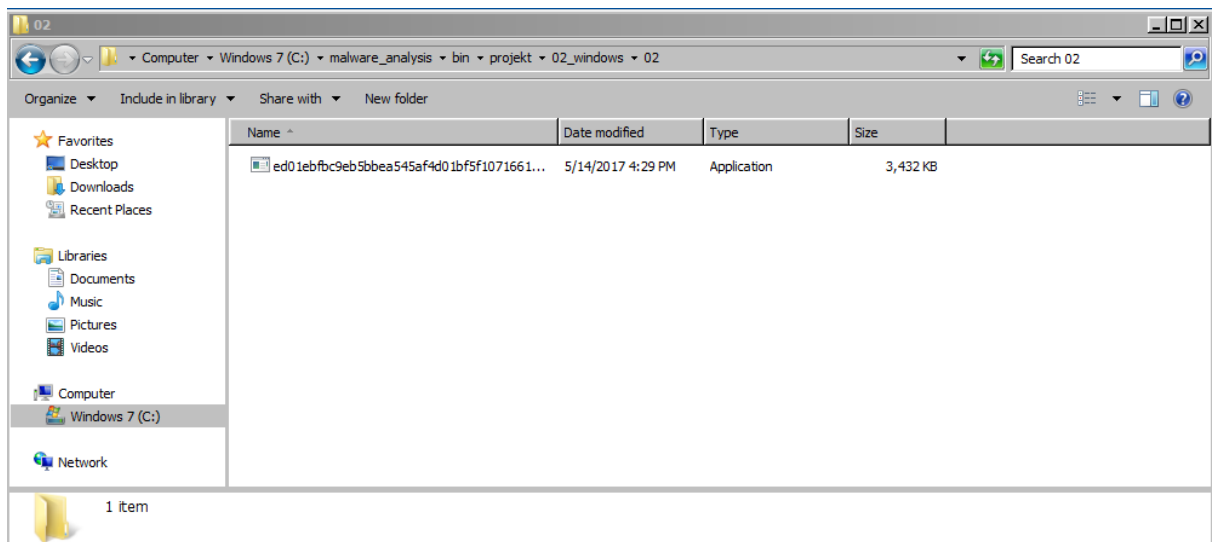
Dostarczony plik wykonywalny zawiera w sobie sygnatury w postaci stringów, rozszerzeń plików oraz sygnatur funkcji skrótu, które wskazują, że mamy do czynienia z programem typu ransomware o nazwie WannaCry. Na ten moment, korzystając z metod analizy statycznej udało się zidentyfikować plik jako plik wykonywalny posiadający zapakowaną wersję tego malware'u. Przy aktywowaniu pliku podwójnym kliknięciem program rozpakowuje się automatycznie stosując zapisane hasło (*WNCry@2017*) oraz natychmiast przechodzi do wykonywania payload'u. W tej fazie analizy badaniu poddaliśmy głównie trzy pliki - spakowany plik dostępny na początku analizy oraz dwa pliki powstałe po jego aktywowaniu. W pierwszym pliku poza odszyfrowywaniem i rozpakowywaniem plików oraz ładowaniem podstawowych funkcji i bibliotek tworzony jest plik *tasksche.exe*, którego nie znaleźliśmy po rozpakowaniu więc zapewne generowany jest dynamicznie.

Następnie plik *taskse.exe* służy do enumeracji otwartych sesji RDP, których tokeny i uprawnienia są duplikowane po czym malware tworzy z nimi nowe procesy - będzie to prawdopodobnie jedna z form rozpropagowywania się malware'u.

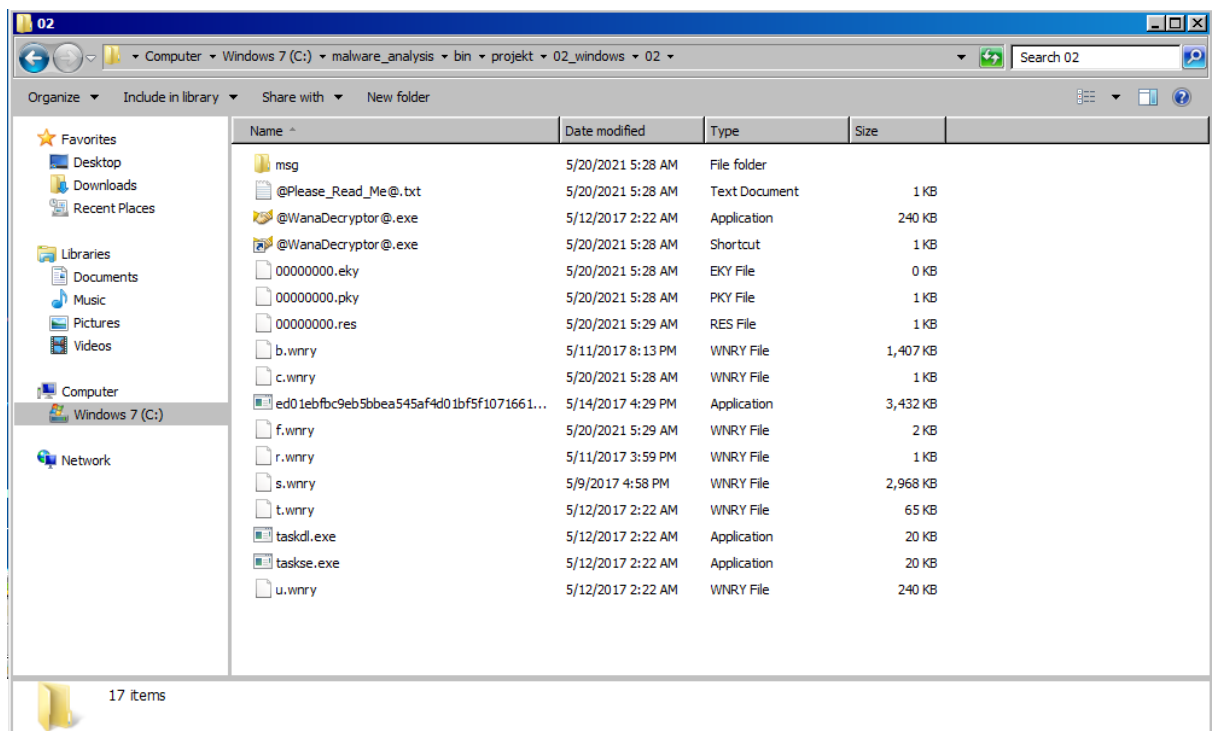
Plik *taskdl.exe* już po wykonaniu głównego programu przeszukuje dyski i usuwa pliki tymczasowe o rozszerzeniu *.WNCRYT*. Są to pliki tymczasowe z oryginalnych plików, które malware tworzy w czasie szyfrowania - ich zawartość jest następnie wpisana na miejsce oryginalnej zawartości

### **III. Analiza dynamiczna**

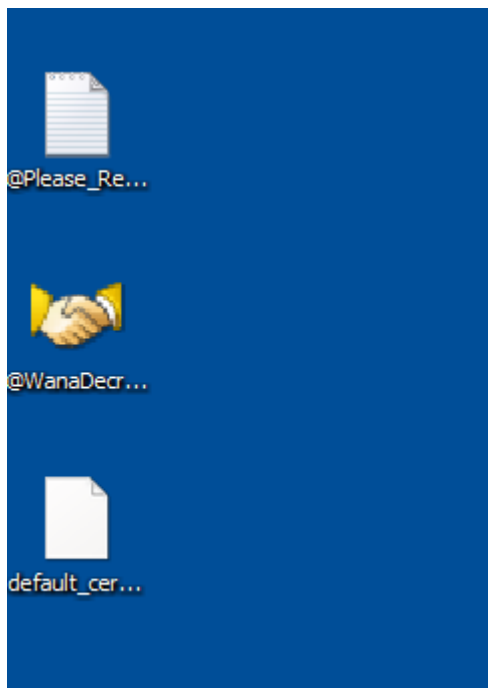
Uruchomienie pliku na maszynie wirtualnej Windows 7 i analiza zachodzących zdarzeń:



Najpierw obserwujemy rozpakowanie się pliku:



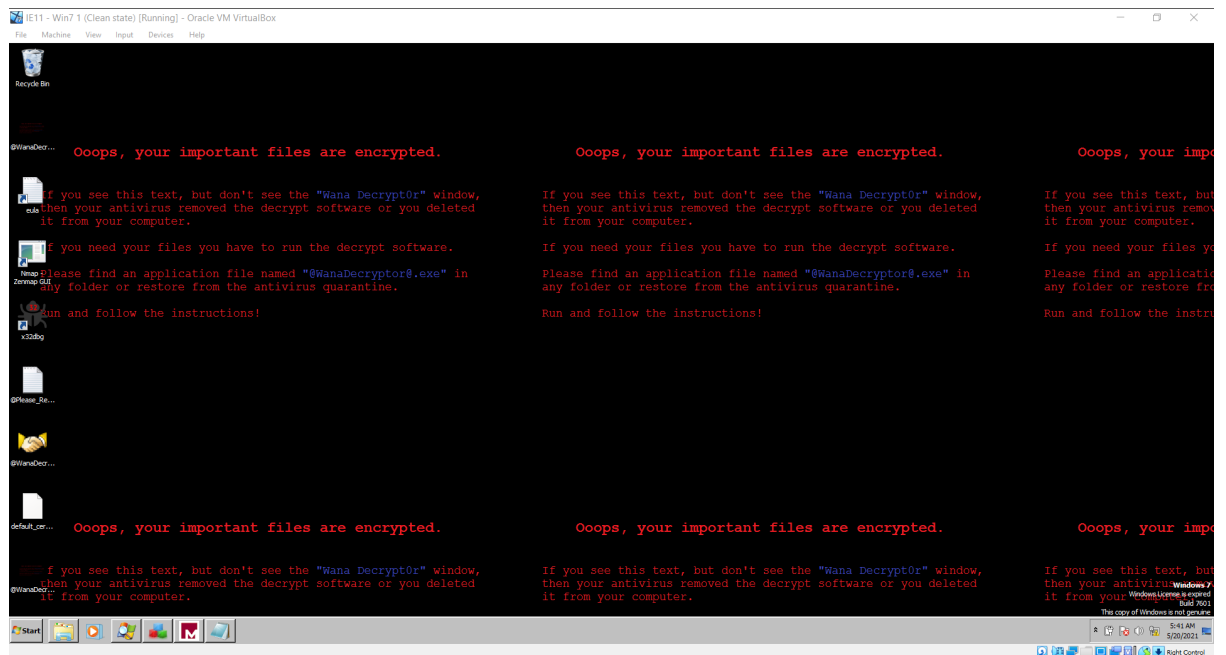
Oraz pojawienie się nowych plików na pulpicie:



Po dłuższej chwili widzimy takie okno:



Oraz zmienia się tło pulpitu na informację o zaszyfrowaniu plików



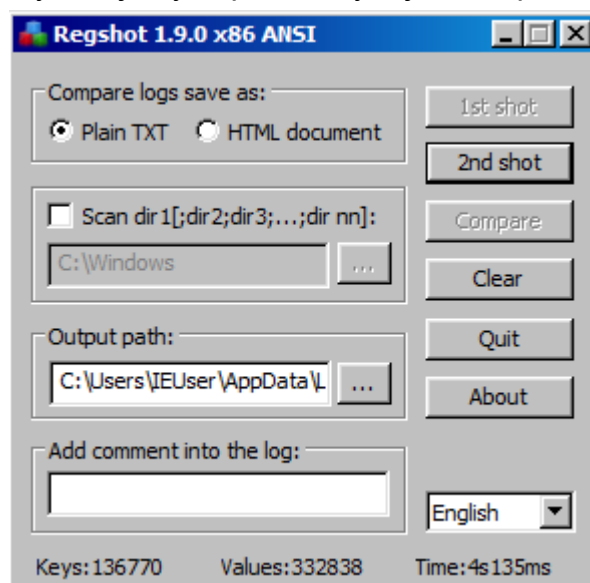
## Wnioski:

Analiza wizualna ostatecznie potwierdza, że mamy do czynienia z przedstawicielem programu typu ransomware pod nazwą WannaCry. Program dokonuje zaszyfrowania plików użytkownika i żąda płatności na portfel bitcoinowy w celu odszyfrowania plików. Zgodnie z podanymi informacjami do odszyfrowania plików potrzebujemy klucza dostarczonego przez atakującego oraz programu WanaDecryptor, który również pojawił się na komputerze wraz ze złośliwym oprogramowaniem.

## Podstawowa analiza dynamiczna:

### Regshot:

Wykonujemy zapis kluczy rejestrów przed uruchomieniem pliku



```
Keys deleted: 1
HKLM\SOFTWARE\Microsoft\Windows Search\Gather\Windows\SystemIndex\Crawls\14

Keys added: 15
HKLM\BCD00000000\Objects\{200d6851-f024-11e7-8a7d-fe2d3f08a936}\Elements\250000e0
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\FveDetect\Cache
HKLM\SOFTWARE\WanaCrypt0r
HKLM\SYSTEM\ControlSet001\services\VSS\Diag\ASR Writer
HKLM\SYSTEM\ControlSet001\services\VSS\Diag\COM+ REGDB Writer
HKLM\SYSTEM\ControlSet001\services\VSS\Diag\Registry Writer
HKLM\SYSTEM\ControlSet001\services\VSS\Diag\Shadow Copy Optimization Writer
HKLM\SYSTEM\ControlSet001\services\VSS\Diag\SwProvider_{b5946137-7b9f-4925-af80-51abd60b20d5}
HKLM\SYSTEM\CurrentControlSet\services\VSS\Diag\ASR Writer
HKLM\SYSTEM\CurrentControlSet\services\VSS\Diag\COM+ REGDB Writer
HKLM\SYSTEM\CurrentControlSet\services\VSS\Diag\Registry Writer
HKLM\SYSTEM\CurrentControlSet\services\VSS\Diag\Shadow Copy Optimization Writer
HKLM\SYSTEM\CurrentControlSet\services\VSS\Diag\SwProvider_{b5946137-7b9f-4925-af80-51abd60b20d5}
HKU\S-1-5-21-3583694148-1414552638-2922671848-1000_Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\107\Shell\{5C4F28B5-F869-4E84-8E60-F11DB97C5C77}
HKU\S-1-5-21-3583694148-1414552638-2922671848-1000_Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\107\Shell\{5C4F28B5-F869-4E84-8E60-F11DB97C5C77}

Values deleted: 5
HKLM\SOFTWARE\Microsoft\Windows Search\Gather\Windows\SystemIndex\Crawls\14\CrawlType: 0x00000002
HKLM\SOFTWARE\Microsoft\Windows Search\Gather\Windows\SystemIndex\Crawls\14\InProgress: 0x00000001
HKLM\SOFTWARE\Microsoft\Windows Search\Gather\Windows\SystemIndex\Crawls\14\DoneAddingCrawlSeeds: 0x00000001
HKLM\SOFTWARE\Microsoft\Windows Search\Gather\Windows\SystemIndex\Crawls\14\IsCatalogLevel: 0x00000000
HKLM\SOFTWARE\Microsoft\Windows Search\Gather\Windows\SystemIndex\Crawls\14\LogStartAddId: 0x00000002

Values added: 37
HKLM\BCD000000000\Objects\{200d6851-f024-11e7-8a7d-fe2d3f08a936}\Elements\250000e0\Element: 01 00 00 00 00 00 00
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\fbqrawoixak113: ""C:\malware_analysis\bin\projekt\02_windows\02\tasksche.exe""
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\SoftwareUpdate\Auto Update\ShowUnableToDetectUI: 0x00000001
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\FveDetect\Cache\DF9C26C60001000000000000: 0x00000000
HKLM\SOFTWARE\WanaCrypt0r\wd: "C:\malware_analysis\bin\projekt\02_windows\02"
HKLM\SYSTEM\ControlSet001\Control\Session Manager\PendingFileRenameOperations: 5C 00 3F 00 3F 00 5C 00 43 00 3A 00 5C 00 55 00 73 00 65 00 72 00 73 00 5C 00 49 00 45 00 55 00 73 00 65 00
00 65 00 64 00 5C 00 68 00 6D 00 70 00 72 00 6E 00 76 00 69 00 6E 00 62 00 44 00 69 00 6A 00 73 00 61 00 73 00 73 00 65 00 60 60 62 00 6C 00 79 00 41 00 6F 00 64 00 44 00 65 00 63 00 6E 00 6D 00
```

HKLM\SOFTWARE\WanaCrypt0r

- program tasksche.exe będzie włączał się przy każdym zalogowaniu użytkownika, zatem jest to sposób na zachowanie trwałości w systemie

- zapisany folder w którym znajdował się malware w celu odnalezienia go do dalszych operacji

"Text Document"

- nie udało się znaleźć związku tych zmian z działaniem malware'u (być może pochodzą od jakiegoś innego programu, który działał w tym samym czasie )

Tu widzimy zmianę tapety na dostarczoną przez malware:

HKU\S-1-5-21-3583694148-1414552638-2922671848-1000\Control

Panel\Desktop\Wallpaper:



"C:\Users\IEUser\AppData\Local\Temp\BGInfo.bmp"

HKU\S-1-5-21-3583694148-1414552638-2922671848-1000\Control

Panel\Desktop\Wallpaper:

"C:\Users\IEUser\Desktop\@WanaDecryptor@.bmp"

Te zmiany i dokładne wartości kluczy rejestrów można również zaobserwować, korzystając z narzędzia Registry Editor:  
HKLM\SOFTWARE\WanaCrypt0r:

Name	Type	Data
 (Default)	REG_SZ	(value not set)
 wd	REG_SZ	C:\malware_analysis\bin\projekt\02_windows\02

Widzimy, że w wartościach ustawiony został adres folderu, w którym znajduje się plik wykonywalny z malware'em.

## Wireshark

W celu analizy sieciowej wykorzystane zostały programy Wireshark, oraz Microsoft Network Monitor. Po uruchomieniu próbki malware możemy zauważyć w wiresharku wiele połączeń po TCP.

tcp						
No.	Time	Source	Destination	Protocol	Length	Info
1532	547.395239895	10.0.0.2	128.31.0.39	TCP	66	49435 → 9101 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
1533	550.306414960	10.0.0.2	128.31.0.39	TCP	66	[TCP Retransmission] 49435 → 9101 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
1536	556.308707200	10.0.0.2	128.31.0.39	TCP	62	[TCP Retransmission] 49435 → 9101 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
608	237.925576021	10.0.0.2	154.35.175.225	TCP	66	49323 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
611	240.944695982	10.0.0.2	154.35.175.225	TCP	66	[TCP Retransmission] 49323 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
615	246.960481503	10.0.0.2	154.35.175.225	TCP	62	[TCP Retransmission] 49323 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
1570	769.556699376	10.0.0.2	163.172.139.104	TCP	66	49443 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
1571	772.557958560	10.0.0.2	163.172.139.104	TCP	66	[TCP Retransmission] 49443 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
1574	778.559866697	10.0.0.2	163.172.139.104	TCP	62	[TCP Retransmission] 49443 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
621	259.952831029	10.0.0.2	163.172.35.249	TCP	66	49327 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
622	262.952423446	10.0.0.2	163.172.35.249	TCP	66	[TCP Retransmission] 49327 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
623	268.971658179	10.0.0.2	163.172.35.249	TCP	62	[TCP Retransmission] 49327 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
627	279.930849099	10.0.0.2	171.25.193.9	TCP	66	49329 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
629	282.931707195	10.0.0.2	171.25.193.9	TCP	66	[TCP Retransmission] 49329 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
631	288.934746080	10.0.0.2	171.25.193.9	TCP	62	[TCP Retransmission] 49329 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
632	291.959259091	10.0.0.2	185.100.84.212	TCP	66	49330 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
633	297.971328531	10.0.0.2	185.100.84.212	TCP	66	[TCP Retransmission] 49330 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
636	303.989803567	10.0.0.2	185.100.84.212	TCP	62	[TCP Retransmission] 49330 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
637	324.299170625	10.0.0.2	185.13.39.197	TCP	66	49332 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
639	327.319713060	10.0.0.2	185.13.39.197	TCP	66	[TCP Retransmission] 49332 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
639	333.316943267	10.0.0.2	185.13.39.197	TCP	62	[TCP Retransmission] 49332 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
644	372.167056710	10.0.0.2	185.97.32.18	TCP	66	49333 → 9001 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
645	375.167568764	10.0.0.2	185.97.32.18	TCP	66	[TCP Retransmission] 49333 → 9001 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
648	381.186439721	10.0.0.2	185.97.32.18	TCP	62	[TCP Retransmission] 49333 → 9001 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
1682	1518.4158627	10.0.0.2	188.166.23.127	TCP	66	49454 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
1683	1521.4304315	10.0.0.2	188.166.23.127	TCP	66	[TCP Retransmission] 49454 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
1684	1527.4335716	10.0.0.2	188.166.23.127	TCP	62	[TCP Retransmission] 49454 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
616	247.883467100	10.0.0.2	193.23.244.244	TCP	66	49324 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
617	250.899974186	10.0.0.2	193.23.244.244	TCP	66	[TCP Retransmission] 49324 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
618	256.964713319	10.0.0.2	193.23.244.244	TCP	62	[TCP Retransmission] 49324 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
337	215.931116009	10.0.0.2	194.109.206.212	TCP	66	49320 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
608	218.929929216	10.0.0.2	194.109.206.212	TCP	66	[TCP Retransmission] 49320 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
605	224.952665636	10.0.0.2	194.109.206.212	TCP	62	[TCP Retransmission] 49320 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
596	215.901114005	10.0.0.2	195.154.122.54	TCP	66	49319 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
599	218.914217781	10.0.0.2	195.154.122.54	TCP	66	[TCP Retransmission] 49319 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
604	224.929682627	10.0.0.2	195.154.122.54	TCP	62	[TCP Retransmission] 49319 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
1527	510.280236499	10.0.0.2	198.96.155.3	TCP	66	49434 → 5001 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
1528	513.302721327	10.0.0.2	198.96.155.3	TCP	66	[TCP Retransmission] 49434 → 5001 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
1529	519.304995397	10.0.0.2	198.96.155.3	TCP	62	[TCP Retransmission] 49434 → 5001 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
673	484.196997325	10.0.0.2	199.254.238.52	TCP	66	49337 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
674	497.4240973180	10.0.0.2	199.254.238.52	TCP	66	[TCP Retransmission] 49337 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
675	413.250270535	10.0.0.2	199.254.238.52	TCP	62	[TCP Retransmission] 49337 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
607	237.925368521	10.0.0.2	51.255.41.65	TCP	66	49322 → 9001 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
612	240.944699966	10.0.0.2	51.255.41.65	TCP	66	[TCP Retransmission] 49322 → 9001 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
614	246.960204798	10.0.0.2	51.255.41.65	TCP	62	[TCP Retransmission] 49322 → 9001 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
1565	1011.3368333	10.0.0.2	86.59.21.38	TCP	66	49445 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
1586	1020.4124877	10.0.0.2	86.59.21.38	TCP	66	[TCP Retransmission] 49445 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
1587	1026.4515727	10.0.0.2	86.59.21.38	TCP	62	[TCP Retransmission] 49445 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
624	276.929597826	10.0.0.2	91.229.20.27	TCP	66	49328 → 9001 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
628	279.948836362	10.0.0.2	91.229.20.27	TCP	66	[TCP Retransmission] 49328 → 9001 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
630	285.964683896	10.0.0.2	91.229.20.27	TCP	62	[TCP Retransmission] 49328 → 9001 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
598	216.980698443	10.0.0.2	92.222.38.67	TCP	66	49321 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
601	219.821434221	10.0.0.2	92.222.38.67	TCP	66	[TCP Retransmission] 49321 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1



W celu uporządkowania tych danych by były bardziej czytelne użyliśmy wbudowanej opcji wiresharka o nazwie Conversations, dzięki której mamy wgląd do statystyk ile połączeń, z jakimi adresami próbowaliśmy się połączyć.

Wireshark - Conversations - eth0											
Ethernet · 6		IPv4 · 21	IPv6 · 3	TCP · 185	UDP · 5						
Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.0.0.1	10.0.0.2	1,536	127k	697	56k	839	71k	16.210418	1419.3912		319
10.0.0.2	10.0.0.255	11	2,244	11	2,244	0	0	0.000000	1654.1266		10
10.0.0.2	224.0.0.252	1	66	1	66	0	0	60.741065	0.0000		
10.0.0.2	195.154.122.54	3	194	3	194	0	0	215.901114	9.0326		171
10.0.0.2	194.109.206.212	3	194	3	194	0	0	215.901176	9.0513		171
10.0.0.2	92.222.38.67	3	194	3	194	0	0	216.806008	9.0184		172
10.0.0.2	51.255.41.65	3	194	3	194	0	0	237.925369	9.0348		171
10.0.0.2	154.35.175.225	3	194	3	194	0	0	237.925576	9.0349		171
10.0.0.2	193.23.244.244	3	194	3	194	0	0	247.883467	9.0212		172
10.0.0.2	163.172.35.249	3	194	3	194	0	0	259.952832	9.0188		172
10.0.0.2	91.229.20.27	3	194	3	194	0	0	276.929698	9.0349		171
10.0.0.2	171.25.193.9	3	194	3	194	0	0	279.930849	9.0039		172
10.0.0.2	185.100.84.212	3	194	3	194	0	0	294.955250	9.0346		171
10.0.0.2	185.13.39.197	3	194	3	194	0	0	324.299171	9.0178		172
10.0.0.2	185.97.32.18	3	194	3	194	0	0	372.167057	9.0194		172
10.0.0.2	199.254.238.52	3	194	3	194	0	0	404.198997	9.0513		171
10.0.0.2	198.96.155.3	3	194	3	194	0	0	510.286236	9.0188		172
10.0.0.2	128.31.0.39	3	194	3	194	0	0	547.305240	9.0035		172
10.0.0.2	163.172.139.104	3	194	3	194	0	0	769.556609	9.0033		172
10.0.0.2	86.59.21.38	3	194	3	194	0	0	1017.396835	9.0183		172
10.0.0.2	188.166.23.127	3	194	3	194	0	0	1518.415863	9.0177		172

Podejrzane w tych wynikach jest to, że wszystkie nieznane połączenia mają takie same statystyki - ilość wysłanych pakietów, wielkość w bajtach. Pozwala nam to podejrzewać, że są to adresy wykorzystane tylko po to, aby sprawdzić poprawność łącza.

Microsoft Network Monitor 3.4

File Edit View Frames Capture Filter Experts Tools Help

New CaptureOpen CaptureSave AsCapture SettingsStartPauseStop

CaptureStart PagePanels

Network Conversations

My Traffic

System (0)

<Unknown>

System (6)

eth0 (720)

IPv4 (10.0.0.2 - 195.154.122.54) ConnID = 17

IPv4 (10.0.0.2 - 194.109.206.212) ConnID = 19

IPv4 (10.0.0.2 - 92.222.38.67) ConnID = 21

IPv4 (10.0.0.2 - 51.255.41.65) ConnID = 23

IPv4 (10.0.0.2 - 154.35.175.225) ConnID = 25

IPv4 (10.0.0.2 - 193.23.244.244) ConnID = 27

IPv4 (10.0.0.2 - 163.172.35.249) ConnID = 29

IPv4 (10.0.0.2 - 91.229.20.27) ConnID = 31

IPv4 (10.0.0.2 - 171.25.193.9) ConnID = 33

IPv4 (10.0.0.2 - 185.100.84.212) ConnID = 35

IPv4 (10.0.0.2 - 185.13.39.197) ConnID = 37

IPv4 (10.0.0.2 - 185.97.32.18) ConnID = 39

IPv4 (10.0.0.2 - 199.254.238.52) ConnID = 45

IPv4 (10.0.0.2 - 198.96.155.3) ConnID = 120

IPv4 (10.0.0.2 - 128.31.0.39) ConnID = 331

IPv4 (10.0.0.2 - 163.172.139.104) ConnID = 337

IPv4 (10.0.0.2 - 86.59.21.38) ConnID = 339

IPv4 (10.0.0.2 - 188.166.23.127) ConnID = 363

RunD32.exe (236)

Other Traffic

Display Filter

ApplyRemoveHistoryLoad Filter

FindAutoscroll

Frame Number	Time Date Local Adjusted	Time Offset	Process Name	Source	Destination	Protocol Name	Description	Conn ID
51	3:41:35 AM 5/31/2021	102.910353	taskhsvc.exe	FEVINT	195.154.122.54	TCP	TCP:Flags=.....S, SrcPort=49319, DstPort=HTTSP(443), PayloadLen=0, Seq=1603043091, Ack=0, Win=8192 (Negotia...	TCP:1...
52	3:41:35 AM 5/31/2021	102.910493	taskhsvc.exe	FEVINT	194.109.206.212	TCP	TCP:Flags=.....S, SrcPort=49320, DstPort=HTTSP(443), PayloadLen=0, Seq=3880879615, Ack=0, Win=8192 (Negotia...	TCP:2...
53	3:41:36 AM 5/31/2021	103.8146012	taskhsvc.exe	FEVINT	92.222.38.67	TCP	TCP:Flags=.....S, SrcPort=49321, DstPort=HTTSP(443), PayloadLen=0, Seq=2113326817, Ack=0, Win=8192 (Negotia...	TCP:2...
54	3:41:38 AM 5/31/2021	105.9210794	taskhsvc.exe	FEVINT	195.154.122.54	TCP	TCP:[SynRetranmit #52]Flags=.....S, SrcPort=49319, DstPort=HTTSP(443), PayloadLen=0, Seq=1603043091, Ack=...	TCP:1...
55	3:41:38 AM 5/31/2021	105.9377385	taskhsvc.exe	FEVINT	194.109.206.212	TCP	TCP:[SynRetranmit #52]Flags=.....S, SrcPort=49320, DstPort=HTTSP(443), PayloadLen=0, Seq=3880879615, Ack=...	TCP:2...
56	3:41:39 AM 5/31/2021	106.6284179	taskhsvc.exe	FEVINT	92.222.38.67	TCP	TCP:[SynRetranmit #52]Flags=.....S, SrcPort=49321, DstPort=HTTSP(443), PayloadLen=0, Seq=2113326817, Ack=...	TCP:2...
59	3:41:44 AM 5/31/2021	111.9063709	taskhsvc.exe	FEVINT	195.154.122.54	TCP	TCP:[SynRetranmit #51]Flags=.....S, SrcPort=49319, DstPort=HTTSP(443), PayloadLen=0, Seq=1603043091, Ack=...	TCP:1...
60	3:41:44 AM 5/31/2021	111.9252654	taskhsvc.exe	FEVINT	194.109.206.212	TCP	TCP:[SynRetranmit #52]Flags=.....S, SrcPort=49320, DstPort=HTTSP(443), PayloadLen=0, Seq=3880879615, Ack=...	TCP:2...
61	3:41:45 AM 5/31/2021	112.7666986	taskhsvc.exe	FEVINT	92.222.38.67	TCP	TCP:[SynRetranmit #52]Flags=.....S, SrcPort=49321, DstPort=HTTSP(443), PayloadLen=0, Seq=2113326817, Ack=...	TCP:2...
62	3:41:57 AM 5/31/2021	124.8911776	taskhsvc.exe	FEVINT	154.35.175.225	TCP	TCP:Flags=.....S, SrcPort=49322, DstPort=9001, PayloadLen=0, Seq=417879185, Ack=0, Win=8192 (Negotiating a...	TCP:2...
63	3:41:57 AM 5/31/2021	124.8911776	taskhsvc.exe	FEVINT	154.35.175.225	TCP	TCP:Flags=.....S, SrcPort=49323, DstPort=HTTSP(443), PayloadLen=0, Seq=3512724317, Ack=0, Win=8192 (Negotia...	TCP:2...
66	3:42:00 AM 5/31/2021	127.9058965	taskhsvc.exe	FEVINT	154.35.175.225	TCP	TCP:[SynRetranmit #63]Flags=.....S, SrcPort=49323, DstPort=HTTSP(443), PayloadLen=0, Seq=3512724317, Ack=...	TCP:2...
67	3:42:00 AM 5/31/2021	127.9058954	taskhsvc.exe	FEVINT	51.255.41.65	TCP	TCP:[SynRetranmit #62]Flags=.....S, SrcPort=49322, DstPort=9001, PayloadLen=0, Seq=417879185, Ack=0, Win=...	TCP:2...
69	3:42:06 AM 5/31/2021	133.9219038	taskhsvc.exe	FEVINT	51.255.41.65	TCP	TCP:[SynRetranmit #63]Flags=.....S, SrcPort=49323, DstPort=9001, PayloadLen=0, Seq=417879185, Ack=0, Win=...	TCP:2...
70	3:42:06 AM 5/31/2021	133.9223401	taskhsvc.exe	FEVINT	154.35.175.225	TCP	TCP:[SynRetranmit #63]Flags=.....S, SrcPort=49323, DstPort=HTTSP(443), PayloadLen=0, Seq=3512724317, Ack=...	TCP:2...
71	3:42:07 AM 5/31/2021	134.8464674	taskhsvc.exe	FEVINT	193.23.244.244	TCP	TCP:Flags=.....S, SrcPort=49324, DstPort=HTTSP(443), PayloadLen=0, Seq=2645355665, Ack=0, Win=8192 (Negotia...	TCP:2...
72	3:42:10 AM 5/31/2021	137.8598329	taskhsvc.exe	FEVINT	193.23.244.244	TCP	TCP:[SynRetranmit #71]Flags=.....S, SrcPort=49324, DstPort=HTTSP(443), PayloadLen=0, Seq=2645355665, Ack=...	TCP:2...
73	3:42:16 AM 5/31/2021	143.8457745	taskhsvc.exe	FEVINT	193.23.244.244	TCP	TCP:[SynRetranmit #71]Flags=.....S, SrcPort=49324, DstPort=HTTSP(443), PayloadLen=0, Seq=2645355665, Ack=...	TCP:2...
76	3:42:19 AM 5/31/2021	146.8923056	taskhsvc.exe	FEVINT	163.172.35.249	TCP	TCP:Flags=.....S, SrcPort=49327, DstPort=HTTSP(443), PayloadLen=0, Seq=3074491319, Ack=0, Win=8192 (Negotia...	TCP:3...
77	3:42:22 AM 5/31/2021	149.8905604	taskhsvc.exe	FEVINT	163.172.35.249	TCP	TCP:[SynRetranmit #76]Flags=.....S, SrcPort=49327, DstPort=HTTSP(443), PayloadLen=0, Seq=3074491319, Ack=...	TCP:3...
78	3:42:28 AM 5/31/2021	155.9020639	taskhsvc.exe	FEVINT	163.172.35.249	TCP	TCP:[SynRetranmit #76]Flags=.....S, SrcPort=49327, DstPort=HTTSP(443), PayloadLen=0, Seq=3074491319, Ack=...	TCP:3...
79	3:42:36 AM 5/31/2021	163.8290196	taskhsvc.exe	FEVINT	91.229.20.27	TCP	TCP:Flags=.....S, SrcPort=49328, DstPort=9001, PayloadLen=0, Seq=2404143674, Ack=0, Win=8192 (Negotiating a...	TCP:3...
80	3:42:39 AM 5/31/2021	166.8247254	taskhsvc.exe	FEVINT	171.25.193.9	TCP	TCP:Flags=.....S, SrcPort=49329, DstPort=HTTSP(443), PayloadLen=0, Seq=2008325619, Ack=0, Win=8192 (Negotia...	TCP:3...
81	3:42:39 AM 5/31/2021	166.8442346	taskhsvc.exe	FEVINT	91.229.20.27	TCP	TCP:[SynRetranmit #79]Flags=.....S, SrcPort=49328, DstPort=9001, Seq=2404143674, Ack=0, Win=...	TCP:3...
84	3:42:40 AM 5/31/2021	169.8281134	taskhsvc.exe	FEVINT	171.25.193.9	TCP	TCP:[SynRetranmit #82]Flags=.....S, SrcPort=49329, DstPort=HTTSP(443), PayloadLen=0, Seq=2008325619, Ack=...	TCP:3...
85	3:42:40 AM 5/31/2021	172.8594170	taskhsvc.exe	FEVINT	91.229.20.27	TCP	TCP:[SynRetranmit #79]Flags=.....S, SrcPort=49328, DstPort=9001, PayloadLen=0, Seq=2404143674, Ack=0, Win=...	TCP:3...
86	3:42:48 AM 5/31/2021	175.8280699	taskhsvc.exe	FEVINT	171.25.193.9	TCP	TCP:[SynRetranmit #82]Flags=.....S,	
87	3:42:54 AM 5/31/2021	181.8291130	taskhsvc.exe	FEVINT	185.100.84.212	TCP	TCP:Flags=.....S, SrcPort=49331, DstPort=HTTSP(443), PayloadLen=0, Seq=2494942337, Ack=0, Win=8192 (Negotia...	TCP:3...

## Process Explorer

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
svchost.exe	< 0.01	4,676 K	8,888 K	880	Host Process for Windows S...	Microsoft Corporation
svchost.exe	< 0.01	15,700 K	27,396 K	904	Host Process for Windows S...	Microsoft Corporation
taskeng.exe		936 K	3,744 K	1316	Task Scheduler Engine	Microsoft Corporation
MicrosoftEdgeUpdate.exe		1,132 K	2,188 K	1932	Microsoft Edge Update	Microsoft Corporation
svchost.exe		1,456 K	4,364 K	1028	Host Process for Windows S...	Microsoft Corporation
svchost.exe	< 0.01	10,660 K	10,964 K	1164	Host Process for Windows S...	Microsoft Corporation
spoolsv.exe		4,464 K	9,024 K	1308	Spooler SubSystem App	Microsoft Corporation
svchost.exe		7,940 K	9,520 K	1424	Host Process for Windows S...	Microsoft Corporation
taskhost.exe		6,996 K	7,204 K	1504	Host Process for Windows T...	Microsoft Corporation
svchost.exe		2,676 K	6,128 K	1664	Host Process for Windows S...	Microsoft Corporation
svchost.exe		3,028 K	6,448 K	1692	Host Process for Windows S...	Microsoft Corporation
cyggrunsrv.exe		5,896 K	5,860 K	1824		
conhost.exe	< 0.01	564 K	2,344 K	1976	Console Window Host	Microsoft Corporation
sshd.exe		6,120 K	5,804 K	1992		
wlms.exe		576 K	2,600 K	2008	Windows License Monitoring...	Microsoft Corporation
spssvc.exe		1,952 K	7,052 K	1156	Microsoft Software Protectio...	Microsoft Corporation
svchost.exe		1,188 K	4,140 K	1580	Host Process for Windows S...	Microsoft Corporation
explorer.exe	3.10	36,284 K	47,352 K	2452	Windows Explorer	Microsoft Corporation
VBoxTray.exe	0.01	1,336 K	4,804 K	2564	VirtualBox Guest Additions Tr...	Oracle Corporation
cmd.exe		1,696 K	2,544 K	3016	Windows Command Processor	Microsoft Corporation
Procmon.exe	0.04	13,124 K	17,368 K	2392	Process Monitor	Sysinternals - www.sysinter...
procexp.exe	0.82	21,720 K	30,432 K	2732	Sysinternals Process Explorer	Sysinternals - www.sysinter...
Regshot-x86-ANSI.exe		96,632 K	102,228 K	812	Regshot 1.9.0 x86 ANSI	Regshot Team
notepad.exe		2,076 K	5,664 K	2092	Notepad	Microsoft Corporation
ed01ebfbc9eb5bbea545af4...	9.03	17,676 K	22,164 K	3540	DiskPart	Microsoft Corporation
@WanaDecryptor@.exe		1,424 K	5,420 K	1628	Load PerfMon Counters	Microsoft Corporation
taskhsvc.exe	0.06	6,812 K	10,068 K	2084		
@WanaDecryptor@.exe	0.10	1,740 K	6,684 K	1148	Load PerfMon Counters	Microsoft Corporation
SearchIndexer.exe	< 0.01	19,988 K	17,108 K	2784	Microsoft Windows Search I...	Microsoft Corporation
conhost.exe		788 K	3,300 K	3024	Console Window Host	Microsoft Corporation
WmiPrvSE.exe		1,924 K	5,152 K	3908	WMI Provider Host	Microsoft Corporation
conhost.exe		556 K	2,276 K	2172	Console Window Host	Microsoft Corporation
VSSVC.exe		1,436 K	5,252 K	3688	Microsoft® Volume Shadow ...	Microsoft Corporation
svchost.exe		988 K	3,988 K	2836	Host Process for Windows S...	Microsoft Corporation
wbengine.exe		1,332 K	5,244 K	3336	Microsoft® Block Level Bac...	Microsoft Corporation
vds.exe		2,052 K	6,392 K	3592	Virtual Disk Service	Microsoft Corporation
System Idle Process	75.46	0 K	24 K	0		
System	2.91	52 K	232 K	4		
Interrupts	8.28	0 K	0 K	n/a	Hardware Interrupts and DPCs	
smss.exe		224 K	804 K	252	Windows Session Manager	Microsoft Corporation
csrss.exe	0.01	1,184 K	3,452 K	344	Client Server Runtime Process	Microsoft Corporation
csrss.exe	0.15	1,184 K	5,280 K	396	Client Server Runtime Process	Microsoft Corporation
wininit.exe		876 K	3,308 K	404	Windows Start-Up Application	Microsoft Corporation
services.exe		3,356 K	7,184 K	496	Services and Controller app	Microsoft Corporation
svchost.exe	< 0.01	2,540 K	7,104 K	604	Host Process for Windows S...	Microsoft Corporation
VBoxService.exe	< 0.01	1,476 K	4,260 K	664	VirtualBox Guest Additions S...	Oracle Corporation
lsass.exe	0.01	2,428 K	8,024 K	504	Local Security Authority Proc...	Microsoft Corporation
lsm.exe	0.01	1,024 K	2,900 K	512	Local Session Manager Serv...	Microsoft Corporation
winlogon.exe		1,604 K	5,292 K	432	Windows Logon Application	Microsoft Corporation

Kolejno uruchamiane procesy zaobserwowane przy użyciu narzędzia Process Explorer:

procexp.exe	0.72	13,440 K	23,304 K	2732	Sysinternals Process Explorer	Sysinternals - www.sysinter...
Regshot-x86-ANSI.exe		48,580 K	52,312 K	812	Regshot 1.9.0 x86 ANSI	Regshot Team
ed01ebfbc9eb5bbea545af4...	Susp...	224 K	76 K	3540	DiskPart	Microsoft Corporation
SearchIndexer.exe	< 0.01	15,992 K	10,196 K	2784	Microsoft Windows Search I...	Microsoft Corporation
conhost.exe		788 K	3,300 K	3024	Console Window Host	Microsoft Corporation
WmiPrvSE.exe		1,736 K	4,976 K	3908	WMI Provider Host	Microsoft Corporation
conhost.exe	1.86	980 K	3,654 K	3620	COM Surrogate	Microsoft Corporation
System Idle Process	20.79	0 K	24 K	0		



ed01ebfbc9eb5bbea545af4	21.16	3,116 K	6,524 K	3540 DiskPart	Microsoft Corporation
icacds.exe	1.99	376 K	1,824 K	4088	Microsoft Corporation
SearchIndexer.exe	0.79	15,992 K	10,196 K	2784 Microsoft Windows Search I...	Microsoft Corporation
conhost.exe		788 K	3,300 K	3024 Console Window Host	Microsoft Corporation
WmiPrvSE.exe		1,736 K	4,976 K	3908 WMI Provider Host	Microsoft Corporation
dllhost.exe		980 K	3,654 K	3620 COM Surrogate	Microsoft Corporation
conhost.exe	0.50	544 K	2,280 K	2612 Console Window Host	Microsoft Corporation
System Idle Process	48.98	0 K	24 K	0	
System	0.78	52 K	568 K	4	




Regshot-x86-ANSI.exe		48,580 K	52,312 K	812 Regshot 1.9.0 x86 ANSI	Regshot Team
ed01ebfbc9eb5bbea545af4	6.01	5,728 K	9,380 K	3540 DiskPart	Microsoft Corporation
cmd.exe	1.78	1,652 K	2,412 K	3284 Windows Command Processor	Microsoft Corporation
cscript.exe	3.22	1,072 K	4,024 K	2668 Microsoft Windows Console Based ...	Microsoft Corporation
SearchIndexer.exe	2.59	16,012 K	10,312 K	2784 Microsoft Windows Search I...	Microsoft Corporation
SearchProtocolHost.exe	2.39	1,172 K	3,868 K	3224 Microsoft Windows Search P...	Microsoft Corporation
SearchFilterHost.exe	3.76	1,224 K	4,156 K	3972 Microsoft Windows Search F...	Microsoft Corporation
conhost.exe		788 K	3,300 K	3024 Console Window Host	Microsoft Corporation
WmiPrvSE.exe		1,736 K	4,976 K	3908 WMI Provider Host	Microsoft Corporation
dllhost.exe		980 K	3,654 K	3620 COM Surrogate	Microsoft Corporation
conhost.exe	1.22	544 K	2,300 K	3276 Console Window Host	Microsoft Corporation
System Idle Process	43.92	0 K	24 K	0	
System	1.49	52 K	568 K	4	
Internet	2.29	0 K	0 K	n/a Hardware Interrupts and DPCs	

## Wnioski:

- program wywołuje proces icacds.exe - jest to program systemowy, jednak może być czasem wykorzystywany przez malware do ukrycia w systemie (sprawdzimy to w dalszych etapach analizy)
- uruchamiany jest interpreter poleceń cmd.exe zatem należy się spodziewać, że malware będzie wykonywał swoje komendy w systemie
- zaobserwowaliśmy również program cscript.exe zatem analogicznie jak wyżej szukać należy skryptów w kodzie programu

## Process Monitor

Filtrujemy zdarzenia po nazwie programu:

Column	Relation	Value	Action
<input checked="" type="checkbox"/>  Process Name	is	ed01ebfbc9eb5bbea545af4d01bf5...	Include
<input checked="" type="checkbox"/>  Process Name	is	Procmon.exe	Exclude
<input checked="" type="checkbox"/>  Process Name	is	Processn.exe	Exclude

## attrib.exe

10:11:...	ed01ebfbc9eb5...	3540	CreateFile	C:\Windows\System32\attrib.exe	SUCCESS	Desired Access: Generic Read, D...
10:11:...	ed01ebfbc9eb5...	3540	QueryStandardI...	C:\Windows\System32\attrib.exe	SUCCESS	AllocationSize: 16,384, EndOfFile:...
10:11:...	ed01ebfbc9eb5...	3540	QueryStandardI...	C:\Windows\System32\attrib.exe	SUCCESS	AllocationSize: 16,384, EndOfFile:...
10:11:...	ed01ebfbc9eb5...	3540	ReadFile	C:\Windows\System32\attrib.exe	SUCCESS	Offset: 15,360, Length: 1,024, Pri...
10:11:...	ed01ebfbc9eb5...	3540	ReadFile	C:\Windows\System32\attrib.exe	SUCCESS	Offset: 12,288, Length: 4,096, I/O...
10:11:...	ed01ebfbc9eb5...	3540	CloseFile	C:\Windows\System32\attrib.exe	SUCCESS	
10:11:...	ed01ebfbc9eb5...	3540	CreateFile	C:\Windows\system32\ui\SwDPM.dll	PATH NOT FOUND	Desired Access: Read Attributes, ...
10:11:...	ed01ebfbc9eb5...	3540	QuerySecurityFile	C:\Windows\System32\attrib.exe	SUCCESS	Information: Owner, Group, DACL...
10:11:...	ed01ebfbc9eb5...	3540	QueryBasicInfor...	C:\Windows\System32\attrib.exe	SUCCESS	Creation Time: 7/13/2009 4:15:01...
10:11:...	ed01ebfbc9eb5...	3540	QuerySecurityFile	C:\Windows\System32\attrib.exe	SUCCESS	Information: Owner, Group, DACL...
10:11:...	ed01ebfbc9eb5...	3540	QueryBasicInfor...	C:\Windows\System32\attrib.exe	SUCCESS	Creation Time: 7/13/2009 4:15:01...

## icacds.exe

10:11:...	ed01ebfbc9eb5...	3540	CreateFile	C:\malware_analysis\bin\projekt\02_windows\02\vcacls.exe	NAME NOT FOUND	Desired Access: Read Attributes, ...
10:11:...	ed01ebfbc9eb5...	3540	CreateFile	C:\malware_analysis\bin\projekt\02_windows\02\vcacls.exe	NAME NOT FOUND	Desired Access: Read Attributes, ...
10:11:...	ed01ebfbc9eb5...	3540	CreateFile	C:\Windows\System32\vcacls.exe	SUCCESS	Desired Access: Read Attributes, ...
10:11:...	ed01ebfbc9eb5...	3540	QueryBasicInfor...	C:\Windows\System32\vcacls.exe	SUCCESS	CreationTime: 7/13/2009 4:15:32...
10:11:...	ed01ebfbc9eb5...	3540	CloseFile	C:\Windows\System32\vcacls.exe	SUCCESS	
10:11:...	ed01ebfbc9eb5...	3540	CreateFile	C:\Windows\System32\vcacls.exe	SUCCESS	Desired Access: Read Attributes, ...
10:11:...	ed01ebfbc9eb5...	3540	QueryBasicInfor...	C:\Windows\System32\vcacls.exe	SUCCESS	CreationTime: 7/13/2009 4:15:32...
10:11:...	ed01ebfbc9eb5...	3540	CloseFile	C:\Windows\System32\vcacls.exe	SUCCESS	
10:11:...	ed01ebfbc9eb5...	3540	CreateFile	C:\Windows\System32\vcacls.exe	SUCCESS	Desired Access: Read Data/List ...
10:11:...	ed01ebfbc9eb5...	3540	CreateFile Mapp...	C:\Windows\System32\vcacls.exe	FILE LOCKED WI...	Sync Type: SyncTypeCreateSecti...
10:11:...	ed01ebfbc9eb5...	3540	QueryStandardI...	C:\Windows\System32\vcacls.exe	SUCCESS	AllocationSize: 28,672, EndOfFile:...
10:11:...	ed01ebfbc9eb5...	3540	ReadFile	C:\Windows\System32\vcacls.exe	SUCCESS	Offset: 0, Length: 4,096, I/O Flag...
10:11:...	ed01ebfbc9eb5...	3540	ReadFile	C:\Windows\System32\vcacls.exe	SUCCESS	Offset: 25,600, Length: 1,536, I/O...
10:11:...	ed01ebfbc9eb5...	3540	CreateFile Mapp...	C:\Windows\System32\vcacls.exe	SUCCESS	Sync Type: SyncTypeOther

Jest to program służący do manipulacji listy kontroli dostępu do plików (Integrity Control Access Control List)

cryptsp.dll - jest to plik typu Cryptographic Service Provider API

10:11:...	ed01ebfbc9eb5...	3540	CreateFile	C:\malware_analysis\bin\projekt\02_windows\02\CRYPTSP.dll	NAME NOT FOUND	Desired Access: Read Attributes
10:11:...	ed01ebfbc9eb5...	3540	CreateFile	C:\Windows\System32\cryptsp.dll	SUCCESS	Desired Access: Read Attributes
10:11:...	ed01ebfbc9eb5...	3540	QueryBasicInfor...	C:\Windows\System32\cryptsp.dll	SUCCESS	CreationTime: 4/4/2021 10:06:4
10:11:...	ed01ebfbc9eb5...	3540	CloseFile	C:\Windows\System32\cryptsp.dll	SUCCESS	
10:11:...	ed01ebfbc9eb5...	3540	CreateFile	C:\Windows\System32\cryptsp.dll	SUCCESS	Desired Access: Read Data/List
10:11:...	ed01ebfbc9eb5...	3540	CreateFile Mapp...	C:\Windows\System32\cryptsp.dll	FILE LOCKED WI...	Sync Type: SyncTypeCreateSec
10:11:...	ed01ebfbc9eb5...	3540	CreateFile Mapp...	C:\Windows\System32\cryptsp.dll	SUCCESS	Sync Type: SyncTypeOther
10:11:...	ed01ebfbc9eb5...	3540	Load Image	C:\Windows\System32\cryptsp.dll	SUCCESS	Image Base: 0x753d0000, Image
10:11:...	ed01ebfbc9eb5...	3540	CloseFile	C:\Windows\System32\cryptsp.dll	SUCCESS	

Program otwiera klucz:

RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Microsoft Enhanced RSA and AES Cryptographic Provider\Image Path
RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Microsoft Enhanced RSA and AES Cryptographic Provider\Image Path
CreateFile	C:\Windows\System32\rsaenh.dll
QueryBasicInfor...	C:\Windows\System32\rsaenh.dll

Możemy zatem spodziewać się, że do szyfrowania lub komunikacji z C2 program wykorzystuje biblioteki Windowsa szyfrujące algorytmami RSA lub AES.

10:11:...	ed01ebfbc9eb5...	3540	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Microsoft Enhan...	SUCCESS	Type: REG_SZ, Length: 66, C
10:11:...	ed01ebfbc9eb5...	3540	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Microsoft Enhan...	SUCCESS	Type: REG_SZ, Length: 66, C
10:11:...	ed01ebfbc9eb5...	3540	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Microsoft Enhan...	SUCCESS	Type: REG_SZ, Length: 66, C
10:11:...	ed01ebfbc9eb5...	3540	RegQueryValue	HKLM\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Microsoft Enhan...	SUCCESS	Type: REG_SZ, Length: 66, C
10:11:...	ed01ebfbc9eb5...	3540	CreateFile	C:\Windows\System32\rsaenh.dll	SUCCESS	Desired Access: Read Attribut
10:11:...	ed01ebfbc9eb5...	3540	QueryBasicInfor...	C:\Windows\System32\rsaenh.dll	SUCCESS	CreationTime: 7/13/2009 4:3
10:11:...	ed01ebfbc9eb5...	3540	CloseFile	C:\Windows\System32\rsaenh.dll	SUCCESS	
10:11:...	ed01ebfbc9eb5...	3540	CreateFile	C:\Windows\System32\rsaenh.dll	SUCCESS	Desired Access: Read Data/L
10:11:...	ed01ebfbc9eb5...	3540	CreateFile Mapp...	C:\Windows\System32\rsaenh.dll	FILE LOCKED WI...	Sync Type: SyncTypeCreateS
10:11:...	ed01ebfbc9eb5...	3540	QueryStandardI...	C:\Windows\System32\rsaenh.dll	SUCCESS	AllocationSize: 245,760, EndC
10:11:...	ed01ebfbc9eb5...	3540	CreateFile Mapp...	C:\Windows\System32\rsaenh.dll	SUCCESS	Sync Type: SyncTypeOther
10:11:...	ed01ebfbc9eb5...	3540	CloseFile	C:\Windows\System32\rsaenh.dll	SUCCESS	
10:11:...	ed01ebfbc9eb5...	3540	CreateFile	C:\Windows\System32\rsaenh.dll	SUCCESS	Desired Access: Read Attribut
10:11:...	ed01ebfbc9eb5...	3540	QueryBasicInfor...	C:\Windows\System32\rsaenh.dll	SUCCESS	CreationTime: 7/13/2009 4:3
10:11:...	ed01ebfbc9eb5...	3540	CloseFile	C:\Windows\System32\rsaenh.dll	SUCCESS	
10:11:...	ed01ebfbc9eb5...	3540	CreateFile	C:\Windows\System32\rsaenh.dll	SUCCESS	Desired Access: Read Data/L
10:11:...	ed01ebfbc9eb5...	3540	CreateFile Mapp...	C:\Windows\System32\rsaenh.dll	FILE LOCKED WI...	Sync Type: SyncTypeCreateS
10:11:...	ed01ebfbc9eb5...	3540	QueryStandardI...	C:\Windows\System32\rsaenh.dll	SUCCESS	AllocationSize: 245,760, EndC
10:11:...	ed01ebfbc9eb5...	3540	CreateFile Mapp...	C:\Windows\System32\rsaenh.dll	SUCCESS	Sync Type: SyncTypeOther
10:11:...	ed01ebfbc9eb5...	3540	CloseFile	C:\Windows\System32\rsaenh.dll	SUCCESS	
10:11:...	ed01ebfbc9eb5...	3540	CreateFile	C:\Windows\System32\rsaenh.dll	SUCCESS	Desired Access: Read Attribut
10:11:...	ed01ebfbc9eb5...	3540	QueryBasicInfor...	C:\Windows\System32\rsaenh.dll	SUCCESS	CreationTime: 7/13/2009 4:3
10:11:...	ed01ebfbc9eb5...	3540	CloseFile	C:\Windows\System32\rsaenh.dll	SUCCESS	
10:11:...	ed01ebfbc9eb5...	3540	CreateFile	C:\Windows\System32\rsaenh.dll	SUCCESS	Desired Access: Read Data/L
10:11:...	ed01ebfbc9eb5...	3540	CreateFile Mapp...	C:\Windows\System32\rsaenh.dll	FILE LOCKED WI...	Sync Type: SyncTypeCreateS

Pliki kryptografii asymetrycznej:

12:29:...	ed01ebfbc9eb5...	2896	CreateFile	C:\Users\VEUser\Desktop\default_cert.pem.WNCRY	NAME NOT FOUND	Desired Access: R...
12:29:...	ed01ebfbc9eb5...	2896	CreateFile	C:\Users\VEUser\Desktop\default_cert.pem	SUCCESS	Desired Access: R...
12:29:...	ed01ebfbc9eb5...	2896	QueryStandardI...	C:\Users\VEUser\Desktop\default_cert.pem	SUCCESS	AllocationSize: 4,0...
12:29:...	ed01ebfbc9eb5...	2896	QueryBasicInfor...	C:\Users\VEUser\Desktop\default_cert.pem	SUCCESS	CreationTime: 4/6/...
12:29:...	ed01ebfbc9eb5...	2896	ReadFile	C:\Users\VEUser\Desktop\default_cert.pem	SUCCESS	Offset: 0, Length: 8...
12:29:...	ed01ebfbc9eb5...	2896	ReadFile	C:\Users\VEUser\Desktop\default_cert.pem	SUCCESS	Offset: 0, Length: 1...
12:29:...	ed01ebfbc9eb5...	2896	CreateFile Mapp...	C:\malware_analysis\bin\projekt\02_windows\02\taskd.exe	FILE LOCKED WI...	Sync Type: SyncTy...
12:29:...	ed01ebfbc9eb5...	2896	CreateFile Mapp...	C:\malware_analysis\bin\projekt\02_windows\02\taskd.exe	SUCCESS	Sync Type: SyncTy...

Podsumowując ten etap:

- ponownie potwierdza się fakt tworzenia wielu nowych plików przez program główny (spakowany), do których wpisywany jest payload dostarczony przez główny plik wykonywalny
- wiemy już, że program będzie wykorzystywał proces o nazwie tasksche.exe do zachowania trwałości w systemie
- wywoływany jest program icacls.exe - być może chodzi tu o zdobycie praw dostępu do plików w systemie (konieczne do zaszyfrowania tych plików)
- process attrib.exe - zapewnia dostęp i możliwość zmiany atrybutów plików (np Read-Only, Hidden, System, Directory)
- tworzone są pliki .bat (programy wsadowe), które malware wykorzystuje do wywoływania swoich komendy
- program otwiera biblioteki Windowsa zawierające funkcje kryptograficzne algorytmów RSA i AES - malware wykorzystywać je będzie do szyfrowania dysku oraz komunikacji z C2

### Zaawansowana analiza dynamiczna:

#### OllyDbg

Wyszukujemy wywołania procesów icacls oraz attrib w zrzucie pamięci

Address	Hex dump	ASCII
0040F4DC	73 63 68 65 2E 65 78 65	sche.exe
0040F4E4	00 00 00 00 54 61 73 6B	....Task
0040F4EC	53 74 61 72 74 00 00 00	Start...
0040F4F4	74 2E 77 6E 72 79 00 00	t.wnry..
0040F4FC	69 63 61 63 6C 73 20 2E	icacls .
0040F504	20 2F 67 72 61 6E 74 20	/grant
0040F50C	45 76 65 72 79 6F 6E 65	Everyone
0040F514	3A 46 20 2F 54 20 2F 43	:F /T /C
0040F51C	20 2F 51 00 61 74 74 72	/Q.attr
0040F524	69 62 20 2B 68 20 2E 00	ib +h ..
0040F52C	57 4E 63 72 79 40 32 6F	WNcry@2o
0040F534	6C 37 00 00 2F 69 00 00	l7../i..
0040F53C	01 00 00 00 08 00 00 00	0...0...
0040F544	02 00 00 00 04 00 00 00	0...0...

Możemy tutaj wyczytać z jakimi argumentami programy te są wywoływane, a następnie zweryfikować ich znaczenie w dokumentacji.

Mamy zatem wywołania:

- icacls /grant Everyone :F /T /C /Q

[/grant[:r] <sid>:[...]]

Grants specified user access rights. Permissions replace previously granted explicit permissions. Not adding the :r, means that permissions are added to any previously granted explicit permissions.

- o **F** - Full access



Performs the operation on all specified files in the current directory and its subdirectories.

/c

Continues the operation despite any file errors. Error messages will still be displayed.

/l

Performs the operation on a symbolic link instead of its destination.

/q

Suppresses success messages.

Oznacza to, że w wyniku tego polecenia przydzielone zostaną pełne prawa dostępu do wszystkich plików i katalogów znajdujących się poniżej podanej ścieżki.

- **attrib +h**

{+|-}h

Sets (+) or clears (-) the Hidden file attribute. If a file uses this attribute set, you must clear the attribute before you can change any other attributes for the file.

Celem tego polecenia jest przydzielenie atrybutu ukrycia pliku, zatem w wyniku tej operacji niektóre pliki mogą zostać ukryte dla użytkownika.

Analizę rozpoczniemy od zajrzenia w okno Text strings (PPM na widok debuggera i wybieramy Search for > All referenced strings). Wybieramy interesujące nas fragmenty i ustawiamy na nich breakpoints:

```
00401A86 PUSH ed01ebfb.0040F0F0 ASCII "CryptDestroyKey"
00401A93 PUSH ed01ebfb.0040F0E0 ASCII "CryptEncrypt"
00401AA0 PUSH ed01ebfb.0040F0D0 ASCII "CryptDecrypt"
00401AAD PUSH ed01ebfb.0040F0C4 ASCII "CryptGenKey"
00401B46 PUSH ed01ebfb.0040EB88 UNICODE "%s\\%s"
00401BFE PUSH ed01ebfb.0040F40C UNICODE "%s\\ProgramData"
00401C4D PUSH ed01ebfb.0040F3F8 UNICODE "%s\\Intel"
00401D3E PUSH ed01ebfb.0040F42C ASCII "cmd.exe /c \"%s\""
00401DB6 PUSH ed01ebfb.0040F43C ASCII "XIA"
00401E55 PUSH ed01ebfb.0040E010 ASCII "c.unry"
00401EB0 MOV DWORD PTR SS:[EBP-C],ed01ebfb.0040F40C ASCII "13AM4UW2dhxYgXe0epoHkHS0uy6NgaEb94"
00401EB7 MOV DWORD PTR SS:[EBP-8],ed01ebfb.0040F40C ASCII "12t9VDPgwue29NyMgw519p7AA8is.jr6SMw"
00401EBE MOV DWORD PTR SS:[EBP-4],ed01ebfb.0040F40C ASCII "115p7UMNgoj1pMvkpHijcRdfJNXj6LrLn"
00401F08 PUSH ed01ebfb.0040F4B4 ASCII "Global\\MsWinZonesCacheCounterMutexA"
00401F10 PUSH ed01ebfb.0040F4AC ASCII "%s%d"
00401F32 PUSH ed01ebfb.0040F4D8 ASCII "taskche.exe"
0040203B PUSH ed01ebfb.0040F538 ASCII "/"
00402061 MOV ESI,ed01ebfb.0040F4D8 ASCII "taskche.exe"
004020C9 MOV DWORD PTR SS:[ESP],ed01ebfb.0040F52E ASCII "WNcry@2ol7"
004020DC PUSH ed01ebfb.0040F520 ASCII "WNcry@2ol7"
004020E3 PUSH ed01ebfb.0040F4FC ASCII "attrib +h ."
00402125 PUSH ed01ebfb.0040F4F4 ASCII "icacis. /grant Everyone:F /T /C /Q"
00402145 PUSH ed01ebfb.0040F4E8 ASCII "t.unry"
0040228C PUSH ed01ebfb.0040EBE8 ASCII "TaskStart"
004022A1 PUSH ed01ebfb.0040F55C ASCII "kernel32.dll"
00404177 MOV DWORD PTR DS:[ESI+18],ed01ebfb.0040F40C ASCII "GetNativeSystemInfo"
ASCII "invalid literal/length code"
```

Pierwszy breakpoint, na który trafiamy to ten związany ze stringiem WNcry@2ol7, a więc miejsce gdzie payload malware'u jest rozpakowywany.

String "attrib +h ." okazuje się być wykorzystywany w funkcji CreateProcessA. Widok stosu przed samym jej wywołaniem:

0012F75C	00000000	ModuleFileName = NULL
0012F760	0040F520	CommandLine = "attrib +h ."
0012F764	00000000	pProcessSecurity = NULL
0012F768	00000000	pThreadSecurity = NULL
0012F76C	00000000	InheritHandles = FALSE
0012F770	00000000	CreationFlags = CREATE_NO_WINDOW
0012F774	00000000	pEnvironment = NULL
0012F778	00000000	CurrentDir = NULL
0012F77C	0012F78C	pStartupInfo = 0012F78C
0012F780	0012F7D0	pProcessInfo = 0012F7D0
0012F784	0012FEE8	
0012F788	771EDBAE	msvort.strchr
0012F78C	00000044	
0012F790	00000000	
0012F794	00000000	

W wyniku tej operacji zatem folder, z którego ten program jest wykonywany zostanie ukryty.

Analogiczny sposób jest wykorzystywany do wykonania kolejnej komendy. Ponownie wykonywane jest to przy użyciu funkcji CreateProcessA. Widok stosu:

0012F750	00000000	ModuleFileName = NULL
0012F754	0040F4FC	CommandLine = "icacls . /grant Everyone:F /T /C /Q"
0012F758	00000000	pProcessSecurity = NULL
0012F75C	00000000	pThreadSecurity = NULL
0012F760	00000000	InheritHandles = FALSE
0012F764	00000000	CreationFlags = CREATE_NO_WINDOW
0012F768	00000000	pEnvironment = NULL
0012F76C	00000000	CurrentDir = NULL
0012F770	0012F780	pStartupInfo = 0012F780
0012F774	0012F7C4	pProcessInfo = 0012F7C4
0012F778	0012FEE8	
0012F77C	771EDBAE	msvort.strchr
0012F780	00000044	
0012F784	00000000	
0012F788	00000000	

W wyniku tej operacji wszystkim użytkownikom systemu nadane zostaną uprawnienia do wykonywania programów w tym folderze (czyli działa niezależnie od uprawnień danego użytkownika).

String 'cmd.exe /c "%s"' prowadzi do miejsca gdzie tworzony jest serwis o nazwie fbqrawoirxak113 - pod tą samą nazwą stworzony był klucz rejestru gdzie zapisywany był program tasksche.exe



<pre> . 897D F8      MOV     DWORD PTR SS:[EBP-8],EDI . FF15 24804000 CALL    DWORD PTR DS:[&lt;&amp;ADVAPI32.OpenSCMan . 38C7        CMP     EAX,EDI . 8945 FC      MOV     DWORD PTR SS:[EBP-4],EAX . 75 07        JNZ     SHORT ed01ebfb.00401D12 . 33C0        XOR     EAX,EAX . E9 96000000 JMP     ed01ebfb.00401DA8 . 53          PUSH    EBX . 56          PUSH    ESI . BB FF010F00 MOV     EBX,0F01FF . BE ACF84000 MOV     ESI,ed01ebfb.0040F8AC . 53          PUSH    EBX . 56          PUSH    ESI . 50          PUSH    EAX . FF15 04804000 CALL    DWORD PTR DS:[&lt;&amp;ADVAPI32.OpenServic . 38C7        CMP     EAX,EDI . 8945 F4      MOV     DWORD PTR SS:[EBP-C],EAX . 74 17        JE      SHORT ed01ebfb.00401D45 . 57          PUSH    EDI . 57          PUSH    EDI . 50          PUSH    EAX . FF15 08804000 CALL    DWORD PTR DS:[&lt;&amp;ADVAPI32.StartServ . FF75 F4      PUSH    DWORD PTR SS:[EBP-C] . FF15 0C804000 CALL    DWORD PTR DS:[&lt;&amp;ADVAPI32.CloseServ . 6A 01        PUSH    1 . 5E          POP     ESI . EB 56        JMP     SHORT ed01ebfb.00401D9B . FF75 08      PUSH    DWORD PTR SS:[EBP+8] . 8085 F4FBFFFF LEA     EAX,DWORD PTR SS:[EBP-40C] . 68 2CF44000 PUSH    ed01ebfb.0040F42C . 50          PUSH    EAX . FF15 1C814000 CALL    DWORD PTR DS:[&lt;&amp;MSVCRT.sprintf&gt;] . 83C4 0C      ADD     ESP,0C . 8085 F4FBFFFF LEA     EAX,DWORD PTR SS:[EBP-40C] . 57          PUSH    EDI . 57          PUSH    EDI . 57          PUSH    EDI . 57          PUSH    EDI . 57          PUSH    EDI . E7          DISU     ENT </pre>	<pre> ADVAPI32.OpenSCManagerA ASCII "fbqrawoirwak113" ADVAPI32.OpenServiceA ADVAPI32.StartServiceA ADVAPI32.CloseServiceHandle &lt;%s&gt; format = "cmd.exe /c \"%s\"" s printf Password ServiceStartName pDependencies pTagId </pre>
--	---

Gdy malware zakończył już proces szyfrowania dysku tworzone są nowe wątki, w których program rozpoczyna nowe procesy

Threads							
Ident	Entry	Data block	Last error	Status	Priority	User time	System time
00000078	10005730	7FFD0000	ERROR_INVALID_PARAMETER (00000057)	Active	32 + 0	0.0000 s	0.0000 s
00000618	10004790	7FFD0000	ERROR_ALREADY_EXISTS (000000B7)	Active	32 + 0	0.0000 s	0.0000 s
00000660	100045C0	7FFD0000	ERROR_FILE_NOT_FOUND (00000002)	Active	32 + 0	0.0000 s	0.0000 s
00000664	100029E0	7FFD7000	ERROR_FILE_NOT_FOUND (00000002)	Active	32 + 0	4.7368 s	3.7353 s
00000710	10005300	7FFD0000	ERROR_RESOURCE_TYPE_NOT_FOUND (00000715)	Active	32 + 0	0.0300 s	0.0000 s
0000086C	10004990	7FFD0000	ERROR_INVALID_PARAMETER (00000057)	Active	32 + 0	0.0000 s	0.0000 s
00000A9C	004077BA	7FFDF000	ERROR_SUCCESS (00000000)	Active	32 + 0	27.9301 s	9.1231 s
00000C18	77C4FB8F	7FFDE000	ERROR_SUCCESS (00000000)	Active	32 + 0	0.0000 s	0.0000 s

Celem działania programu jest stałe wyświetlanie użytkownikowi informacji o fakcie zaszyfrowania dysku i propozycji wpłacenia okupu.

Jeśli w tym momencie ponownie uruchomimy system wyświetli nam się wspomniana informacja oraz program będzie widoczny w drzewie procesów. Program cyklicznie będzie sprawdzać czy powiadomienie o okupie znajduje się wciąż na ekranie (użytkownik może je zamknąć) i jeśli nie ponownie je wyświetli.

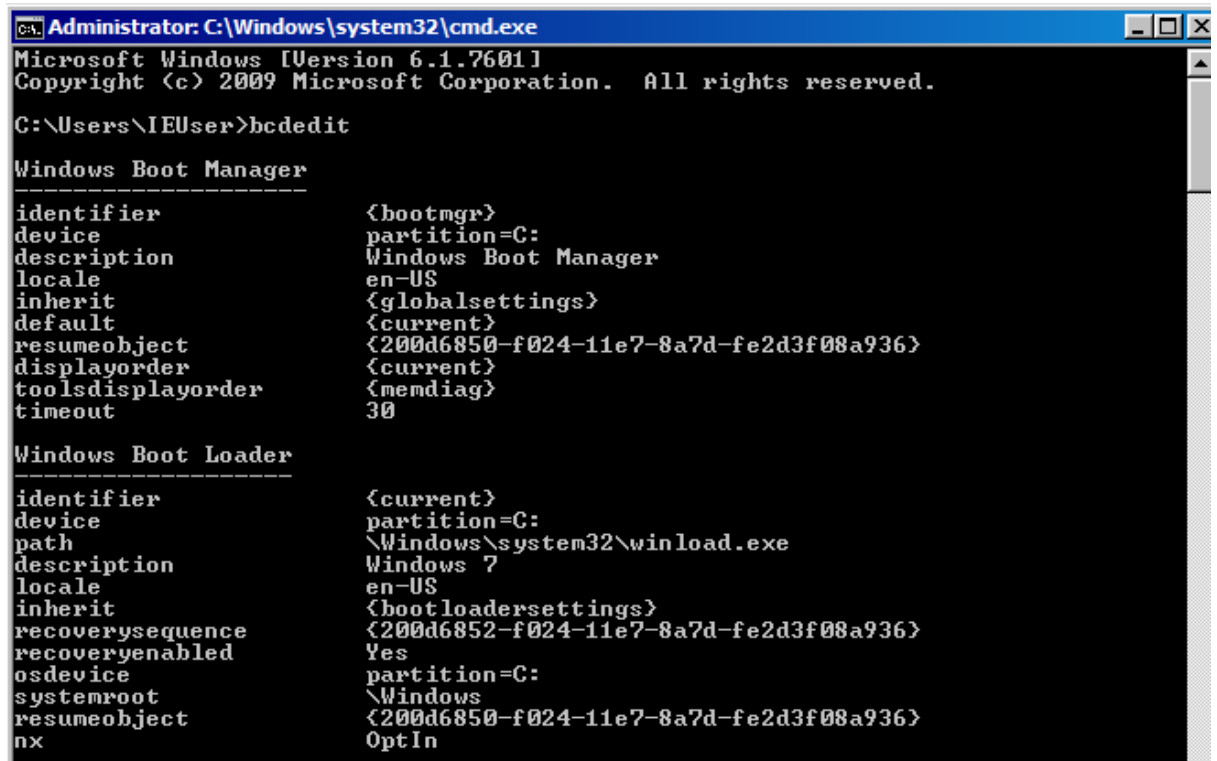
## WinDbg

[https://www.welivesecurity.com/2017/03/27/configure-windbg-kernel-debugging/?fbclid=IwAR1\\_nIVfAzwVppLkhYdygL2-M4kkyJMv5G-\\_5AuCFyX0mr3V7f23Y5wdSWo](https://www.welivesecurity.com/2017/03/27/configure-windbg-kernel-debugging/?fbclid=IwAR1_nIVfAzwVppLkhYdygL2-M4kkyJMv5G-_5AuCFyX0mr3V7f23Y5wdSWo)

Komendy:

Polecenie bcdedit pokazuje konfigurację boot managera oraz wszystkie wpisy w tablicy boot.

Running the command without parameters shows us the configuration of the boot manager and each of the entries in the boot table (initially there is just one entry). Then, the simplest way to generate a new entry with debugging enabled is to copy the first entry and change it



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\IEUser>bcdedit

Windows Boot Manager
-----
identifier          {bootmgr}
device              partition=C:
description         Windows Boot Manager
locale              en-US
inherit              {globalsettings}
default              {current}
resumeobject        {200d6850-f024-11e7-8a7d-fe2d3f08a936}
displayorder        {current}
toolsdisplayorder   {memdiag}
timeout             30

Windows Boot Loader
-----
identifier          {current}
device              partition=C:
path                \Windows\system32\winload.exe
description         Windows 7
locale              en-US
inherit              {bootloadersettings}
recoverysequence    {200d6852-f024-11e7-8a7d-fe2d3f08a936}
recoveryenabled     Yes
osdevice            partition=C:
systemroot          \Windows
resumeobject        {200d6850-f024-11e7-8a7d-fe2d3f08a936}
nx                  OptIn
```

Najprostszym sposobem aby wygenerować nowy wpis będzie skopiowanie już istniejącego wpisu i dodanie do niego opcji debuggowania. Wykonujemy tą operację korzystając z polecenia bcdedit i na koniec sprawdzamy czy pojawił się nowy wpis w tablicy:

```

C:\Users\IEUser>bcdedit /copy {current} /d "DebugOn"
The entry was successfully copied to {200d6854-f024-11e7-8a7d-fe2d3f08a936}.

C:\Users\IEUser>bcdedit /debug {200d6854-f024-11e7-8a7d-fe2d3f08a936} on
The operation completed successfully.

C:\Users\IEUser>bcdedit

Windows Boot Manager
-----
identifier                {bootmgr}
device                    partition=C:
description                Windows Boot Manager
locale                    en-US
inherit                    {globalsettings}
default                    {current}
resumeobject               {200d6850-f024-11e7-8a7d-fe2d3f08a936}
displayorder               {current}
toolsdisplayorder          {memdiag}
timeout                    30

Windows Boot Loader
-----
identifier                 {current}
device                     partition=C:
path                       \Windows\system32\winload.exe
description                 Windows 7
locale                     en-US
inherit                     {bootloadersettings}
recoverysequence           {200d6852-f024-11e7-8a7d-fe2d3f08a936}
recoveryenabled             Yes
osdevice                   partition=C:
systemroot                 \Windows
resumeobject               {200d6850-f024-11e7-8a7d-fe2d3f08a936}
nx                          OptIn

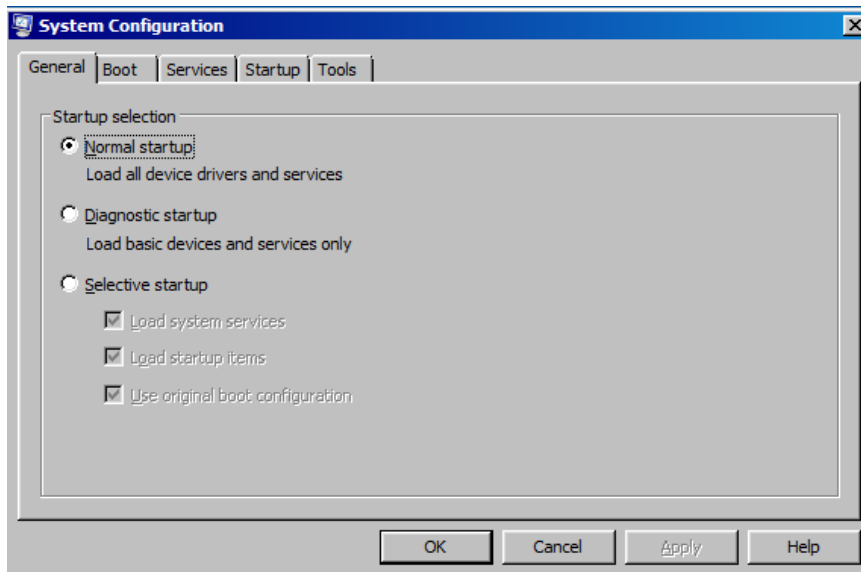
Windows Boot Loader
-----
identifier                 {200d6854-f024-11e7-8a7d-fe2d3f08a936}
device                     partition=C:
path                       \Windows\system32\winload.exe
description                 DebugOn
locale                     en-US
inherit                     {bootloadersettings}
recoverysequence           {200d6852-f024-11e7-8a7d-fe2d3f08a936}
recoveryenabled             Yes
osdevice                   partition=C:
systemroot                 \Windows
resumeobject               {200d6850-f024-11e7-8a7d-fe2d3f08a936}
nx                          OptIn
debug                       Yes

```

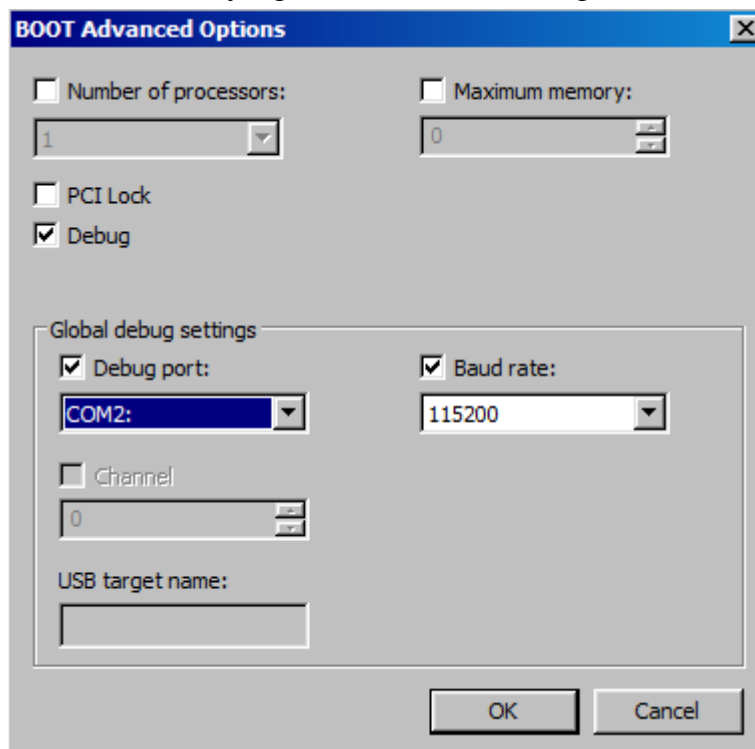
Wpis na samym dole to ten dodany przez nas.

Następnie znajdujemy program msconfig.exe i otwieramy go



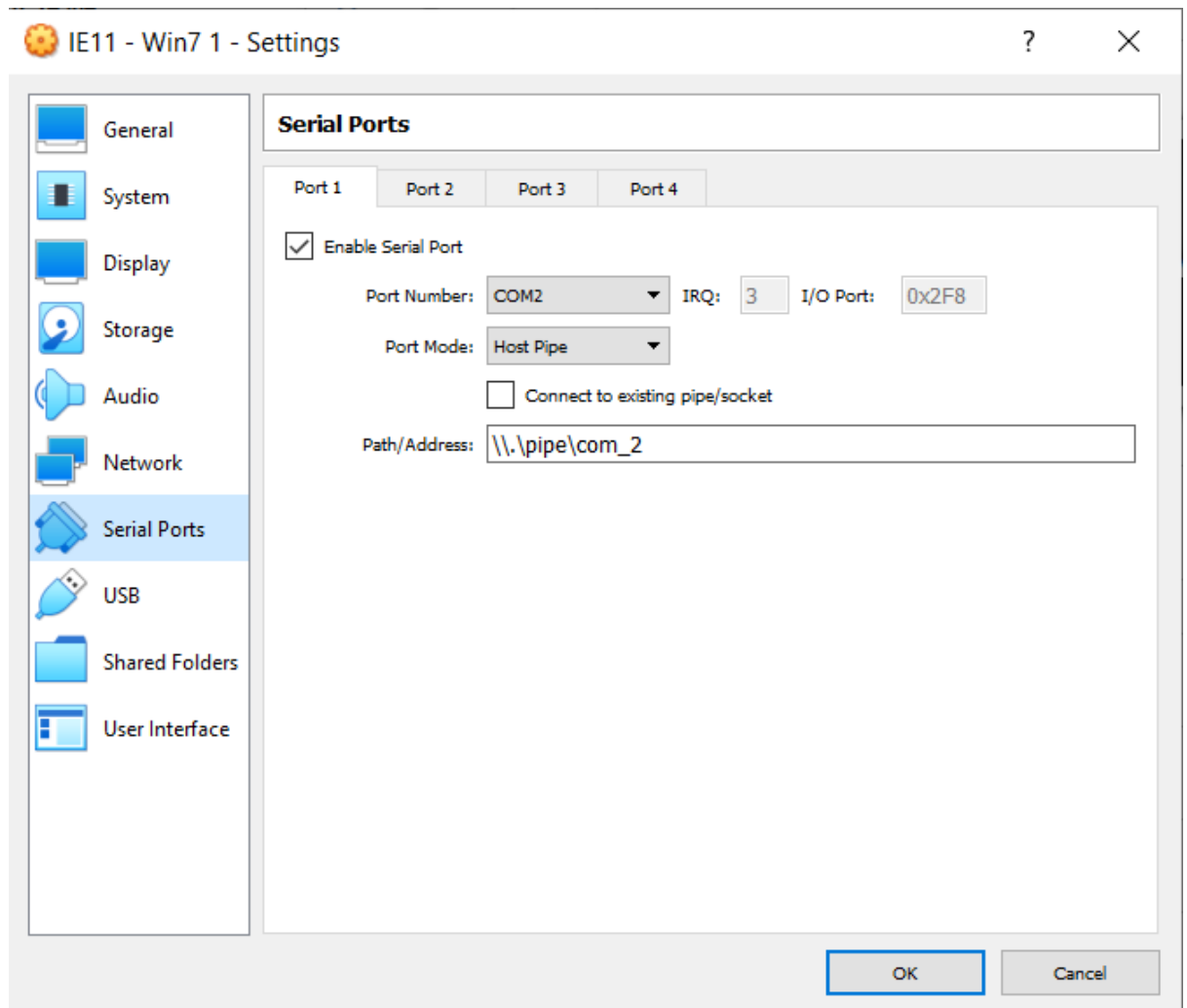


W zakładce “Boot” wybieramy opcja zaawansowane i w nowym oknie ustawiamy wartości dla portu szeregowego, przez którego nasz host będzie komunikował się z guestem w celu debugowania:

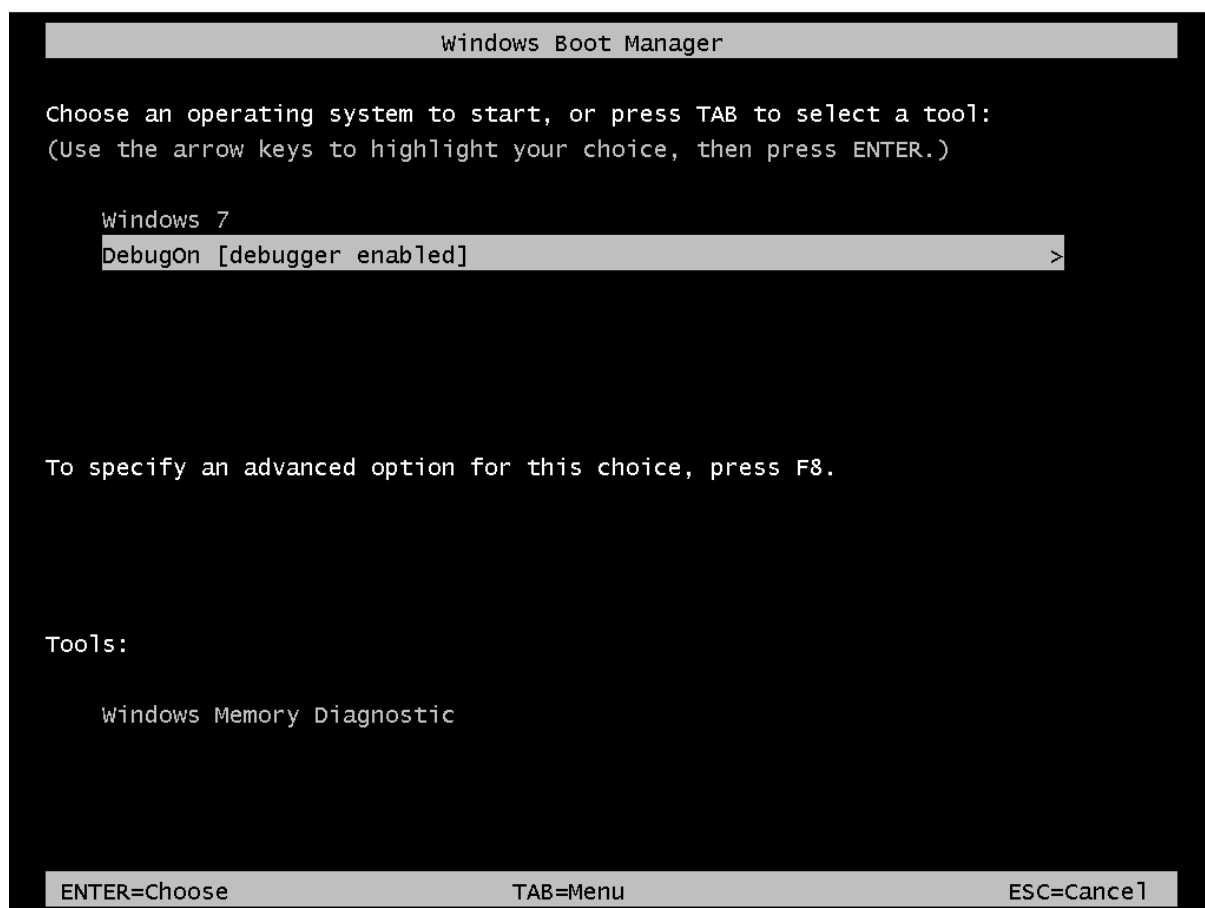


Zatwierdzamy wybór i wyłączamy maszynę

Konfiguracja dla Windowsa 7 na VirtualBox:



Następnie ponownie uruchamiamy maszynę, tym razem jednak przy starcie systemu wybieramy opcję "Debug on", którą stworzyliśmy w boot table.



Uznaliśmy, że nie ma konieczności korzystania z narzędzia WinDbg. OllyDbg dał już wystarczająco dużo informacji, a dla tego malware'u nie ma konieczności debugowania z poziomu jądra systemu.

### Podsumowanie analizy:

1. Badany plik jest spakowanym plikiem wykonywalnym będącym jednym payloadem znanego malware'u typu ransomware pod nazwą WannaCry. Analizowany plik identyfikowany jest skrótem MD5: ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa
2. Po uruchomieniu program rozpakowuje się na hasło *WNcry@2o17*
3. Spakowany program zawiera:
  - b.wnry — bitmapa ustawiana jako tło pulpitu ofiary z informacją o zaszyfrowaniu dysku
  - c.wnry - dane do komunikacji z C2 (adresy Tor, numery portfeli bitcoinowych)
  - r.wnry - plik tekstowy z żądaniem okupu
  - s.wnry - archiwum ZIP zawierające program Tor do zainstalowania na komputerze ofiary

- t.wnry - zaszyfrowana biblioteka DLL zawierająca funkcjonalności szyfrujące
  - u.wnry - główna część programu deszyfrującego
  - taskdl.exe - usuwa znalezione pliki o rozszerzeniu WNCRY
  - taskse.exe - odpowiada za komunikację sieciową w celu zestawienia sesji RDP, oraz wykorzystania podatności SMB
  - msg - folder z informacją o ataku dla ofiary w kilku językach
4. Program tworzy również kilka dodatkowych programów
    - @WanaDecryptor@.exe - program służący do deszyfrowania plików
    - @Please\_Read\_Me@.txt - żądanie okupu
    - 00000000.pky - publiczny klucz RSA
    - 00000000.res - dane do komunikacji z C2
    - default\_cert.pem.WNCRY - zaszyfrowany plik zawierający certyfikat do komunikacji z C2
  5. Program po starcie i rozpakowaniu wykonuje dwie komendy na katalogu, w którym został uruchomiony:
    - attrib +h . - cały katalog i wszystkie jego pliki zostają ukryte
    - icacls . /grant Everyone:F /T /C /Q - wszyscy użytkownicy systemu dostają uprawnienia do wykonywania plików zawartych w tym katalogu
  6. Program modyfikuje klucze rejestrów w celu zachowania trwałości w systemie:  
 HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\fbqrawoirxak113:  
 ""C:\malware\_analysis\bin\projekt\02\_windows\02\tasksche.exe"
  7. Program tasksche.exe (który jest po prostu naszym analizowanym plikiem) ładuje zintegrowany klucz RSA, następnie odszyfrowuje plik t.wnry, który zawiera bibliotekę DLL, która zawiera komponenty szyfrujące.
  8. Analiza sieciowa nie wykazała nic podejrzanego, poza łączeniem się z wieloma adresami. Prawdopodobnie próbka została pozbawiona potencjalnie niebezpiecznych elementów
  9. Analizując procesy warto się przyjrzeć cacls.exe, cmd.exe, cscript.exe, icacls.exe, oraz cryptsp.dll. Malware dzięki nim może wykonywać własne skrypty, ukrywać się w systemie, manipulować dostępem, oraz szyfrować dane

Źródła:

[1] <https://www.avast.com/c-eternalblue>

[2] <https://blog.netop.com/wannacry-anatomy-of-a-ransomware-attack>