

Tasksche.exe

Thu thập thông tin về binary

(những phần như gathering information about binary, etc... bỏ qua không ghi, chủ yếu ghi các phần cho ta thông tin quan trọng)

wannacry_variant, wcry, ransomware

```
.data:0040F52C Str      db 'wNcry@2017',0      ; DATA XREF: WinMain(x,x,x,x)+E1fo
.data:0040F537                align 4
```

Bitcoin address found :

115p7UMMngoj1pMvkhHijcRdfJNXj6LrLn

12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw

13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94

```
.data:0040F440 a115p7ummngoj1p db '115p7UMMngoj1pMvkhHijcRdfJNXj6LrLn',0
.data:0040F440                                ; DATA XREF: sub_401E9E+20fo
.data:0040F463                                align 4
.data:0040F464 a12t9ydpgwueZ9n db '12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw',0
.data:0040F464                                ; DATA XREF: sub_401E9E+19fo
.data:0040F487                                align 4
.data:0040F488 a13am4vw2dhxygx db '13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94',0
.data:0040F488                                ; DATA XREF: sub_401E9E+12fo
```

Các file extension được nhắm tới để encrypt : .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



























```
.data:0040F125      align 4
.data:0040F128      dd offset aDoc      ; ".doc"
.data:0040F12C      dd offset aDocx     ; ".docx"
.data:0040F130      dd offset aDocb     ; ".docb"
.data:0040F134      dd offset aDocm     ; ".docm"
.data:0040F138      dd offset aDot      ; ".dot"
.data:0040F13C      dd offset aDotm     ; ".dotm"
.data:0040F140      dd offset aDotx     ; ".dotx"
.data:0040F144      dd offset aXls      ; ".xls"
.data:0040F148      dd offset aXlsx     ; ".xlsx"
.data:0040F14C      dd offset aXlsm     ; ".xlsm"
.data:0040F150      dd offset aXlsb     ; ".xlsb"
.data:0040F154      dd offset aXlw      ; ".xlw"
.data:0040F158      dd offset aXlt      ; ".xlt"
.data:0040F15C      dd offset aXlm      ; ".xlm"
.data:0040F160      dd offset aXlc      ; ".xlc"
.data:0040F164      dd offset aXltx     ; ".xltx"
.data:0040F168      dd offset aXltm     ; ".xltn"
.data:0040F16C      dd offset aPpt      ; ".ppt"
.data:0040F170      dd offset aPptx     ; ".pptx"
.data:0040F174      dd offset aPptm     ; ".pptm"
.data:0040F178      dd offset aPot      ; ".pot"
.data:0040F17C      dd offset aPps      ; ".pps"
.data:0040F180      dd offset aPpsm     ; ".ppsm"
.data:0040F184      dd offset aPpsx     ; ".ppsx"
.data:0040F188      dd offset aPpam     ; ".ppam"
.data:0040F18C      dd offset aPotx     ; ".potx"
.data:0040F190      dd offset aPotm     ; ".potm"
.data:0040F194      dd offset aPst      ; ".pst"
.data:0040F198      dd offset aOst      ; ".ost"
.data:0040F19C      dd offset aMsg      ; ".msg"
.data:0040F1A0      dd offset aEml      ; ".eml"
.data:0040F1A4      dd offset aEdb      ; ".edb"
```

Các country/language được nhắm tới và thông báo đòi tiền chuộc : m_bulgarian, m_chinese (simplified), m_chinese (traditional), m_croatian, m_czech, m_danish, m_dutch, m_english, m_filipino, m_finnish, m_french, m_german, m_greek, m_indonesian, m_italian, m_japanese, m_korean, m_latvian, m_norwegian, m_polish, m_portuguese, m_romanian, m_russian, m_slovak, m_spanish, m_swedish, m_turkish, m_vietnamese

chạy qua binwalk, ta có thể thấy file tasksche.exe, một phần trong PE's resources, được compress bởi ZIP DEFLATE algorithm, thực tế khi vút vào IDA, ta cũng có thể đoán được phần nào nếu bắt gặp 2 strings được đánh dấu bên dưới

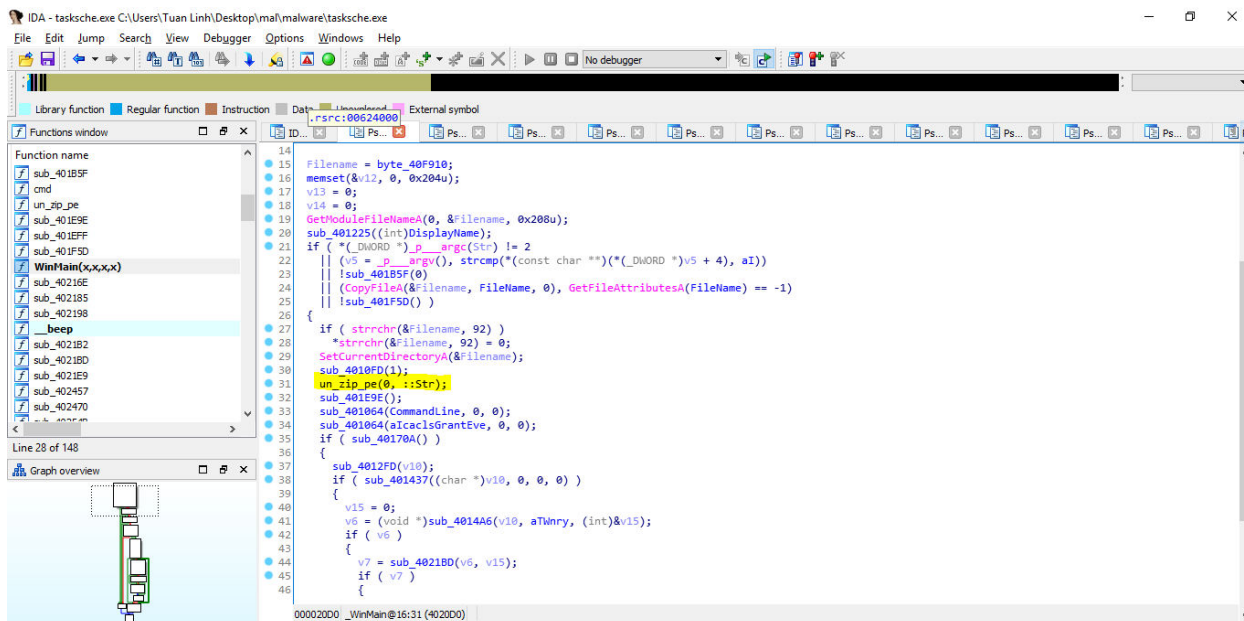
DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	Microsoft executable, portable (PE)
52811	0xCE48	Copyright string: "Copyright 1995-1998 Mark Adler "
53332	0xD054	CRC32 polynomial table, little endian
54368	0xD460	Copyright string: "Copyright 1998 Gilles Vollant "
65776	0x190F9	Zip archive data, encrypted at least v2.0 to extract, compressed size: 14164, uncompressed size: 1440054, name: b.wnry
79076	0x13868	Zip archive data, encrypted at least v2.0 to extract, compressed size: 177, uncompressed size: 780, name: c.wnry
80189	0x1393D	Zip archive data, encrypted at least v2.0 to extract, compressed size: 9404, uncompressed size: 47870, name: msg/m_bulgarian.wnry
89643	0x15E28	Zip archive data, encrypted at least v2.0 to extract, compressed size: 11044, uncompressed size: 54359, name: msg/m_chinese (simplified).w
nr		
100748	0x1898C	Zip archive data, encrypted at least v2.0 to extract, compressed size: 11633, uncompressed size: 79346, name: msg/m_chinese (traditional).
wnry		
112443	0x1873B	Zip archive data, encrypted at least v2.0 to extract, compressed size: 8905, uncompressed size: 39070, name: msg/m_croatian.wnry
121397	0x1DA35	Zip archive data, encrypted at least v2.0 to extract, compressed size: 9079, uncompressed size: 40512, name: msg/m_czech.wnry
130522	0x1FDDA	Zip archive data, encrypted at least v2.0 to extract, compressed size: 8688, uncompressed size: 37045, name: msg/m_danish.wnry
139257	0x21FF9	Zip archive data, encrypted at least v2.0 to extract, compressed size: 8694, uncompressed size: 36987, name: msg/m_dutch.wnry
147997	0x2421D	Zip archive data, encrypted at least v2.0 to extract, compressed size: 8700, uncompressed size: 36973, name: msg/m_english.wnry
156745	0x26449	Zip archive data, encrypted at least v2.0 to extract, compressed size: 8795, uncompressed size: 37580, name: msg/m_filipino.wnry
165589	0x286D5	Zip archive data, encrypted at least v2.0 to extract, compressed size: 8786, uncompressed size: 38377, name: msg/m_finnish.wnry
174423	0x2A957	Zip archive data, encrypted at least v2.0 to extract, compressed size: 8841, uncompressed size: 38437, name: msg/m_french.wnry
183311	0x2CC0F	Zip archive data, encrypted at least v2.0 to extract, compressed size: 8787, uncompressed size: 37181, name: msg/m_german.wnry
192145	0x2EE91	Zip archive data, encrypted at least v2.0 to extract, compressed size: 9554, uncompressed size: 49044, name: msg/m_greek.wnry
201745	0x31411	Zip archive data, encrypted at least v2.0 to extract, compressed size: 8691, uncompressed size: 37196, name: msg/m_indonesian.wnry
210487	0x33637	Zip archive data, encrypted at least v2.0 to extract, compressed size: 8735, uncompressed size: 36883, name: msg/m_italian.wnry
219270	0x35886	Zip archive data, encrypted at least v2.0 to extract, compressed size: 11242, uncompressed size: 81844, name: msg/m_japanese.wnry
230561	0x384A1	Zip archive data, encrypted at least v2.0 to extract, compressed size: 11209, uncompressed size: 91501, name: msg/m_korean.wnry
241817	0x3B099	Zip archive data, encrypted at least v2.0 to extract, compressed size: 9023, uncompressed size: 41169, name: msg/m_latvian.wnry
250888	0x3D408	Zip archive data, encrypted at least v2.0 to extract, compressed size: 8707, uncompressed size: 37577, name: msg/m_norwegian.wnry
259645	0x3F63D	Zip archive data, encrypted at least v2.0 to extract, compressed size: 8950, uncompressed size: 39896, name: msg/m_polish.wnry
268642	0x41962	Zip archive data, encrypted at least v2.0 to extract, compressed size: 8752, uncompressed size: 37917, name: msg/m_portuguese.wnry
277445	0x438C5	Zip archive data, encrypted at least v2.0 to extract, compressed size: 9499, uncompressed size: 52161, name: msg/m_romanian.wnry
286993	0x46111	Zip archive data, encrypted at least v2.0 to extract, compressed size: 9419, uncompressed size: 47108, name: msg/m_russian.wnry
296460	0x4860C	Zip archive data, encrypted at least v2.0 to extract, compressed size: 9124, uncompressed size: 41391, name: msg/m_slovak.wnry
306531	0x4A9DF	Zip archive data, encrypted at least v2.0 to extract, compressed size: 8727, uncompressed size: 37381, name: msg/m_spanish.wnry
314406	0x4CC26	Zip archive data, encrypted at least v2.0 to extract, compressed size: 8771, uncompressed size: 38483, name: msg/m_swedish.wnry
323225	0x4EE99	Zip archive data, encrypted at least v2.0 to extract, compressed size: 9084, uncompressed size: 42582, name: msg/m_turkish.wnry
332357	0x51245	Zip archive data, encrypted at least v2.0 to extract, compressed size: 11224, uncompressed size: 93778, name: msg/m_vietnamese.wnry
343632	0x53E50	Zip archive data, encrypted at least v2.0 to extract, compressed size: 484, uncompressed size: 864, name: r.wnry
344152	0x54058	Zip archive data, encrypted at least v2.0 to extract, compressed size: 3009375, uncompressed size: 3038286, name: s.wnry

Un-zip with 7-zip without password, I found that the creation date was pretty old :3 (Maybe it was changed *intended*)

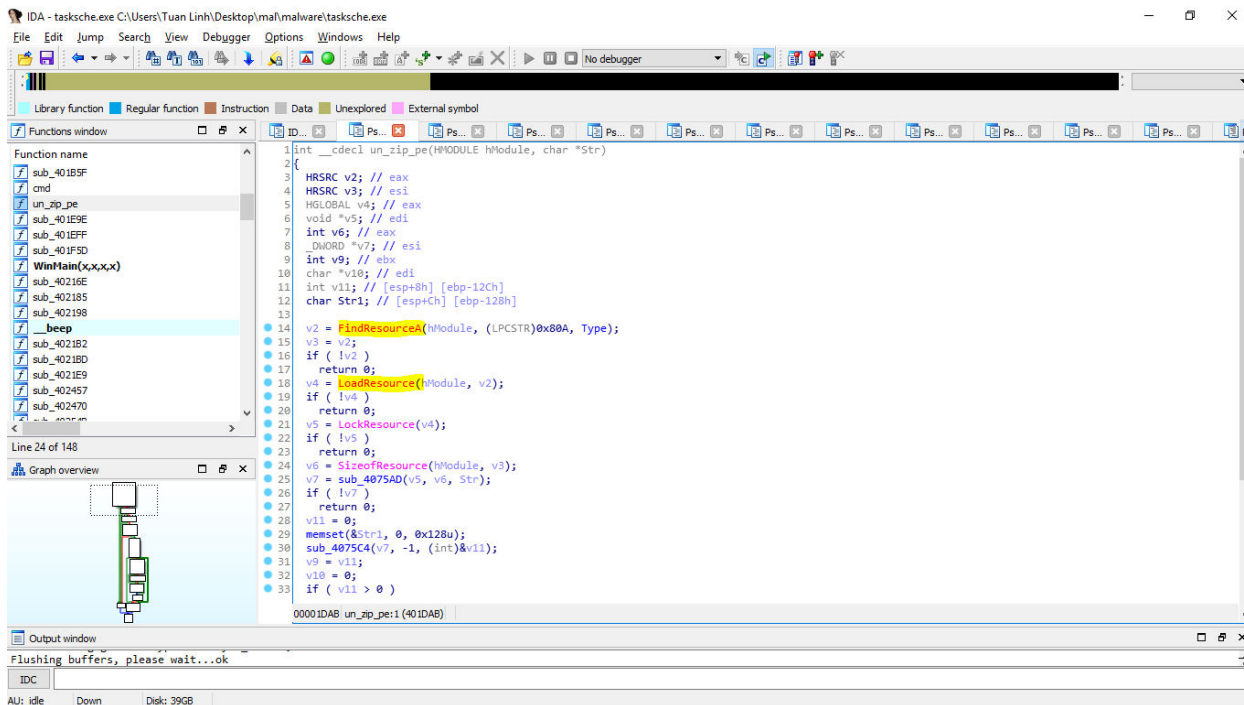
 m_bulgarian.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_chinese (simplified).wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_chinese (traditional).wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_croatian.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_czech.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_danish.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_dutch.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_english.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_filipino.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_finnish.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_french.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_german.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_greek.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_indonesian.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_italian.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_japanese.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_korean.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_latvian.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_norwegian.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_polish.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_portuguese.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_romanian.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_russian.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_slovak.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_spanish.wnry	11/20/2010 4:16 AM	WNRy File	0 KB
 m_swedish.wnry	11/20/2010 4:16 AM	WNRy File	0 KB

Phân tích code :

First things first, thấy ngay hàm WinMain(), bắt đầu với nó thôi 😊



Function này load resource => unzip nó với pass là Str = 'WNCry@2017'

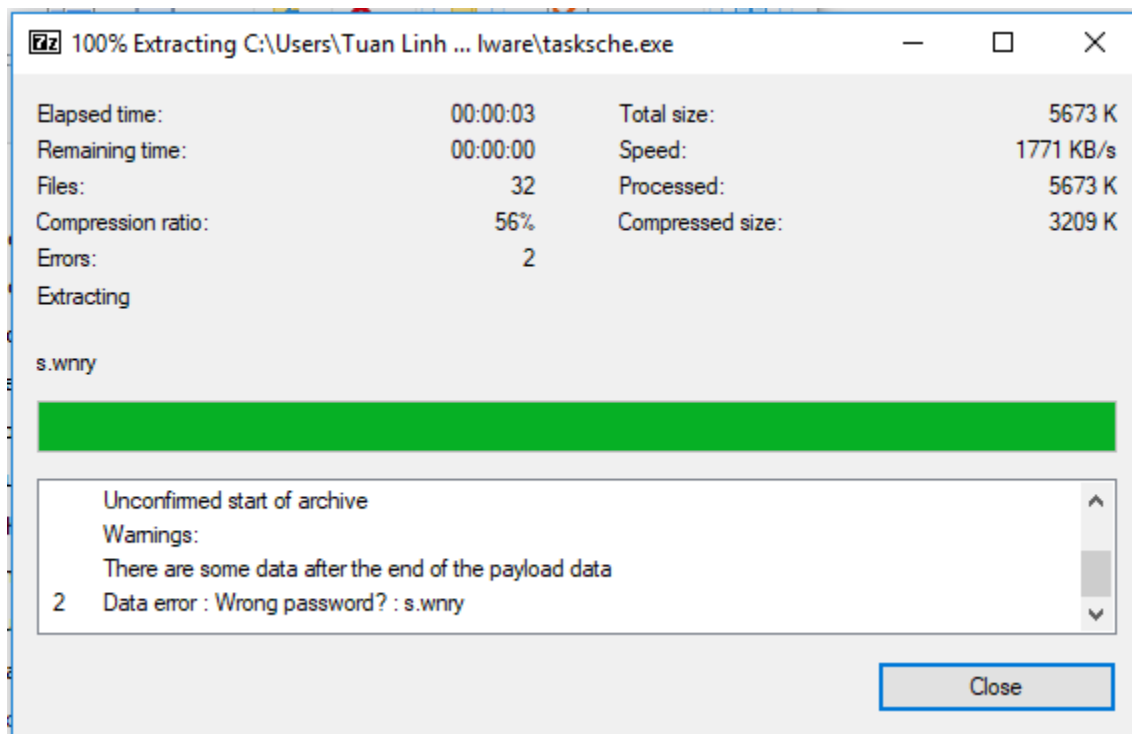


```

.data:0040F52C Str          db 'wNcry@2017',0          ; DATA XREF: WinMain(x,x,x,x)+E1fo
.data:0040F537          align 4

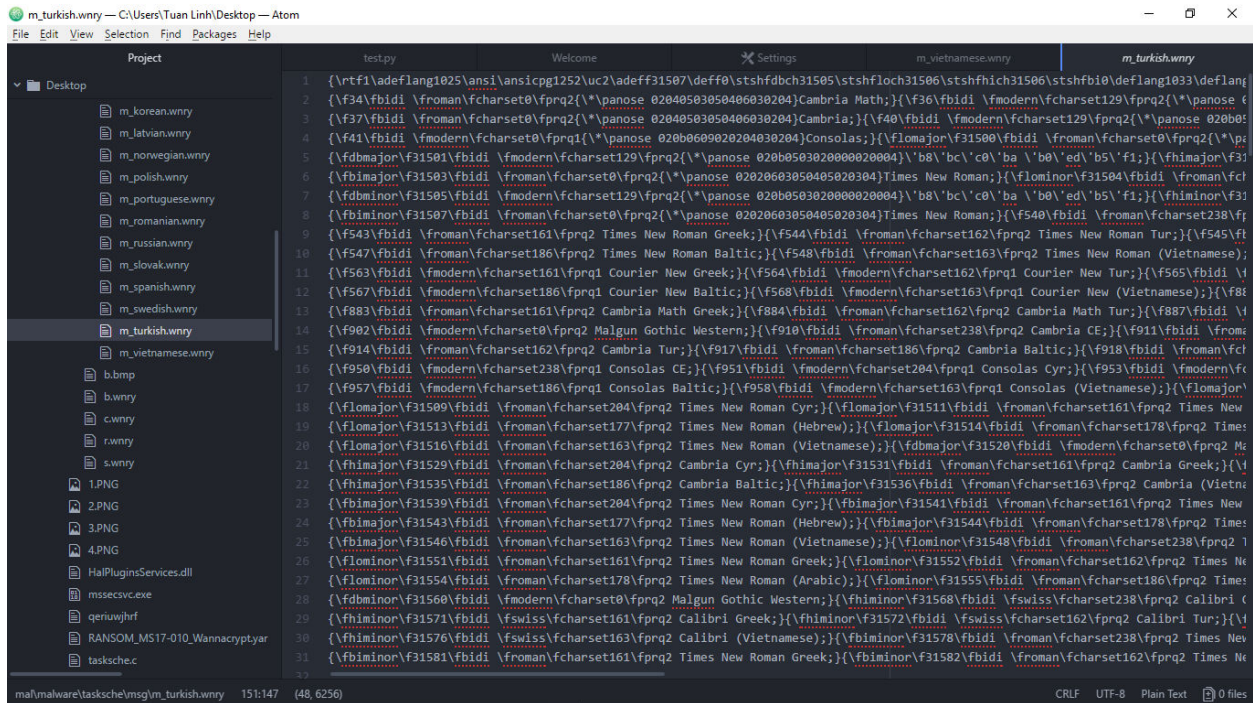
```

Thử xem :

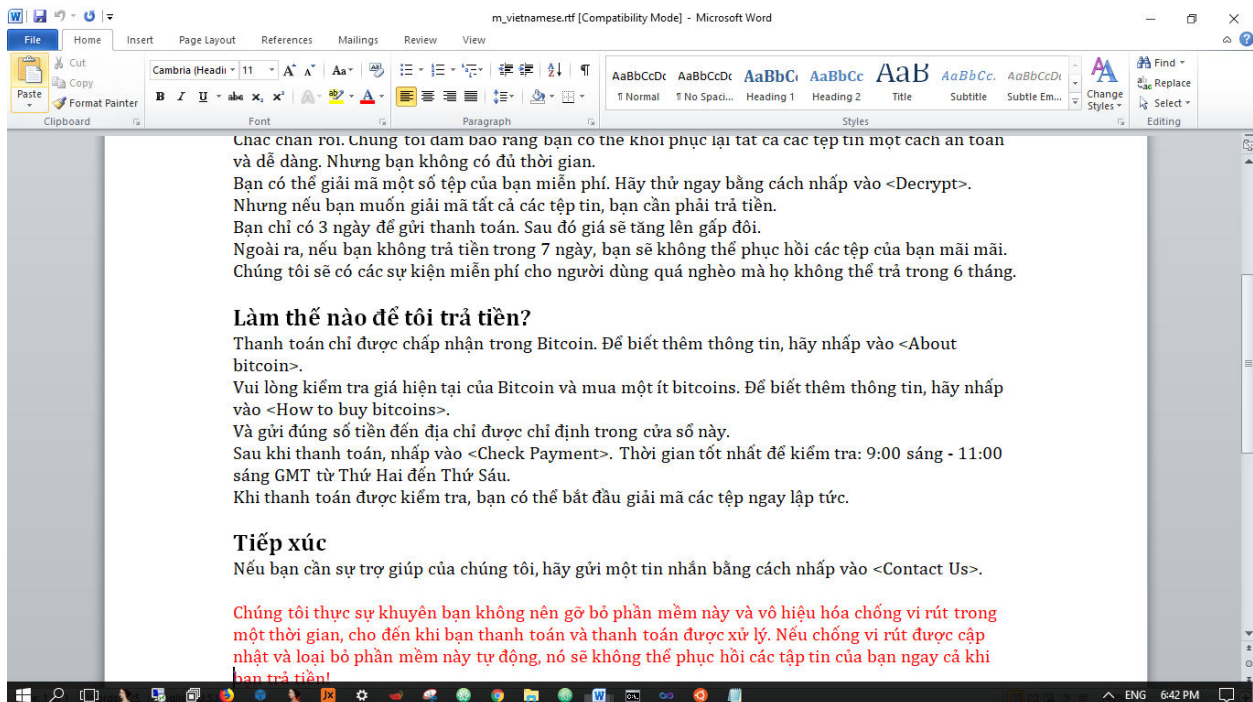


Yuppppppppp! Nhưng vẫn chưa decompress được s.wnry ☹ Nhưng để sau vậy, ngó thử mấy file vừa decrypt trước.

Ngó thử mấy file đòi tiền chuộc, xem Vietnamese trước vậy ☺

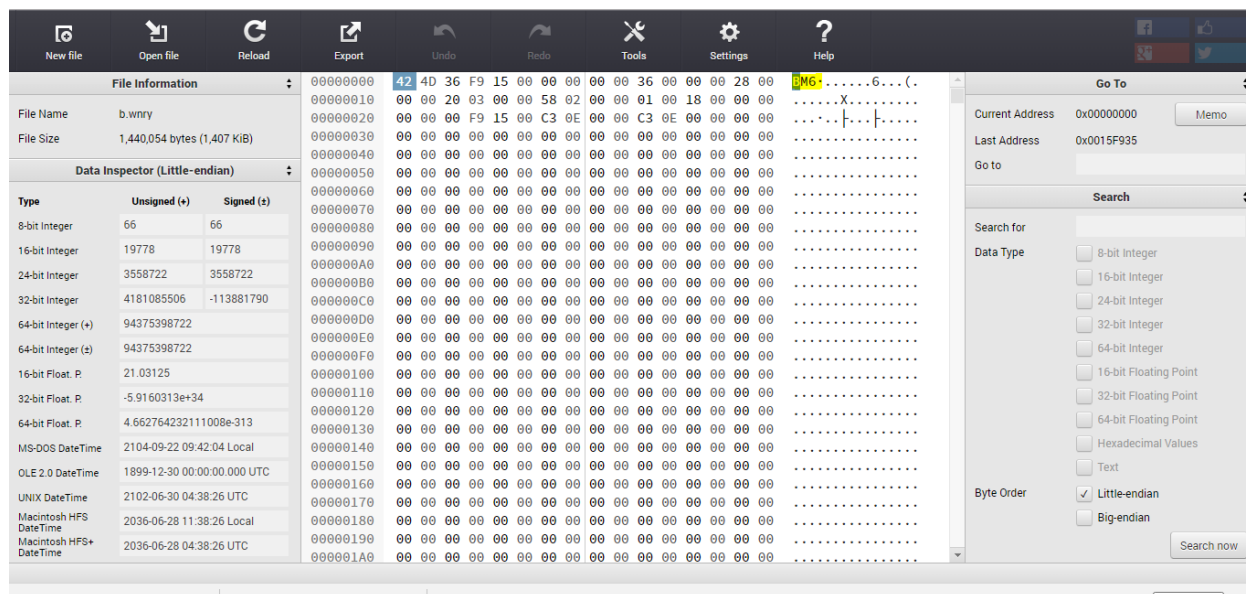


Kinh nghiệm của một người chơi forensic cho mình biết đây là file rtf, sửa lại extension và bật lên xem :

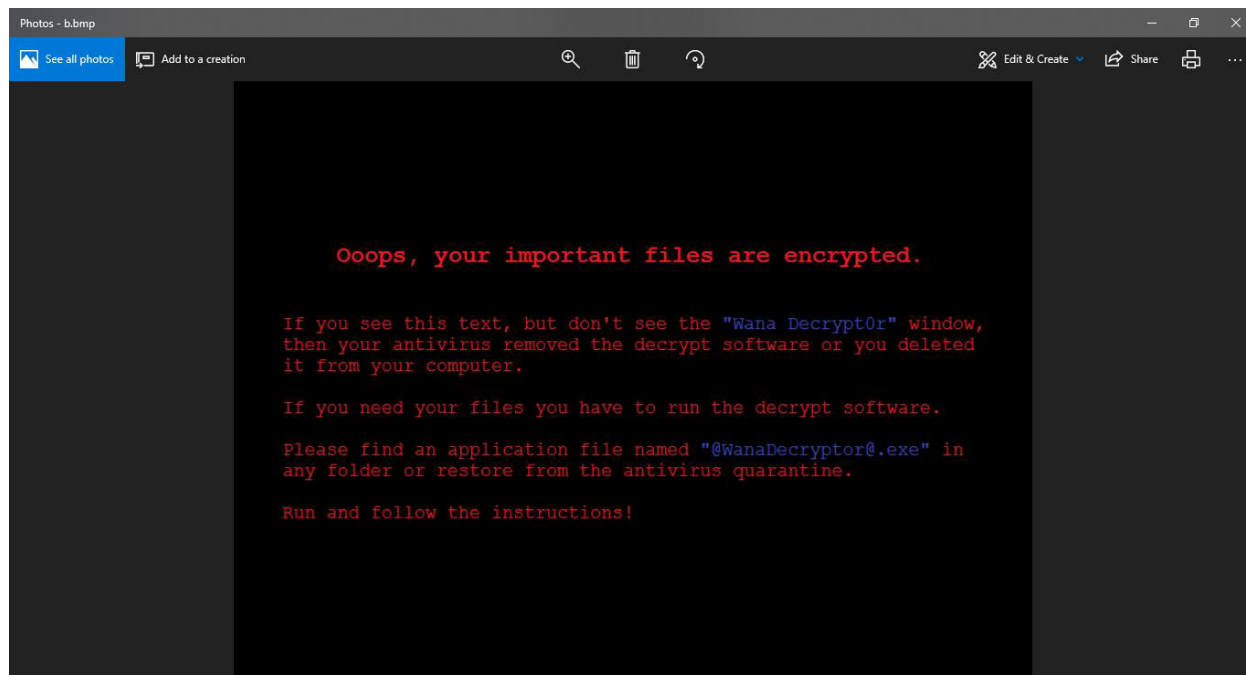


Thử các file khác xem :

B.wnry



Và lần nữa, kinh nghiệm của một người chơi forensic lại cho mình biết đây là file ảnh bitmap BMP qua 2 bit hex đầu tiên (BM). Thực tế bạn có thể sai tool để nhận dạng, nhưng đọc và nhớ signature của một số file thông dụng thì tốt hơn (PE,MK,BM,PNG,etc...) (https://www.garykessler.net/library/file_sigs.html , trang này thực sự rất tốt và có sig của nhiều file thông dụng, nó giúp mình rất nhiều hồi mới chơi forensic)



Lại đời tiền nữa...

C.wnry

The screenshot shows a hex editor interface with the following components:

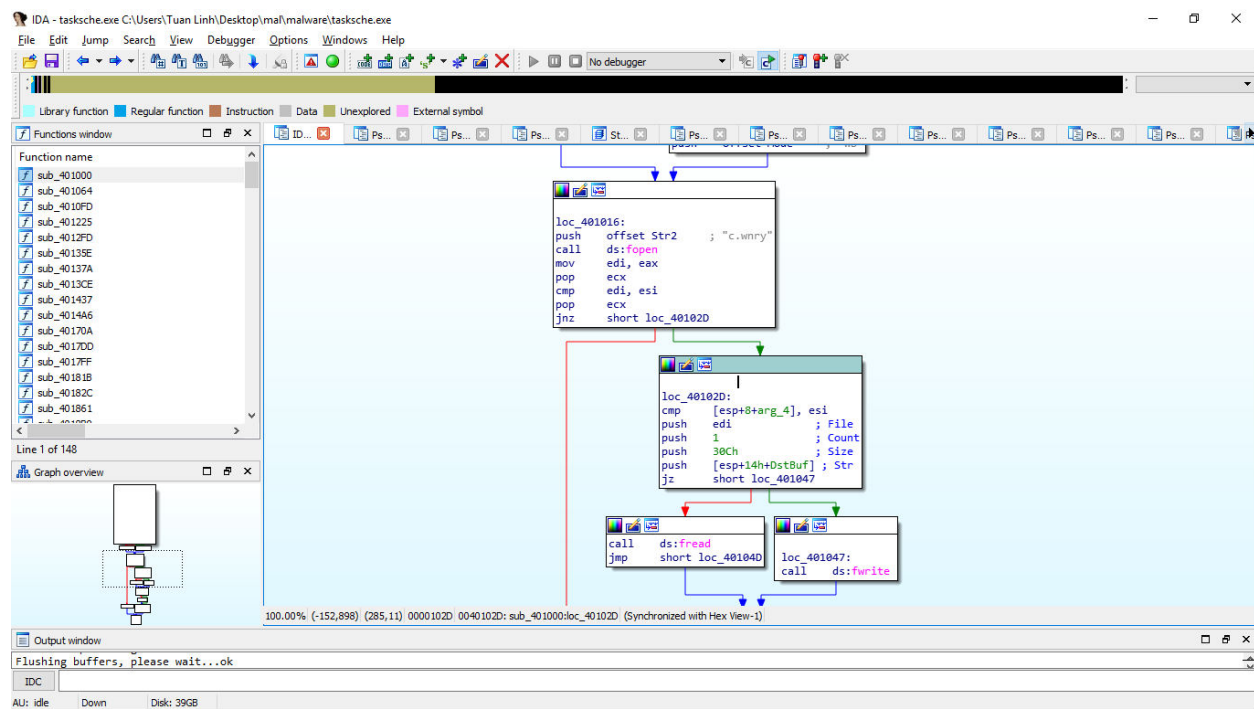
- Menu Bar:** New file, Open file, Reload, Export, Undo, Redo, Tools, Settings, Help.
- File Information:** File Name: c.wnry, File Size: 780 bytes.
- Data Inspector (Little-endian):** Shows file metadata including timestamps and file size.
- Hex View:** Displays the raw bytes of the file. The string of TOR addresses is visible in the hex view.

Cho vào hex, ta thấy ngay được 5 địa chỉ TOR sau :

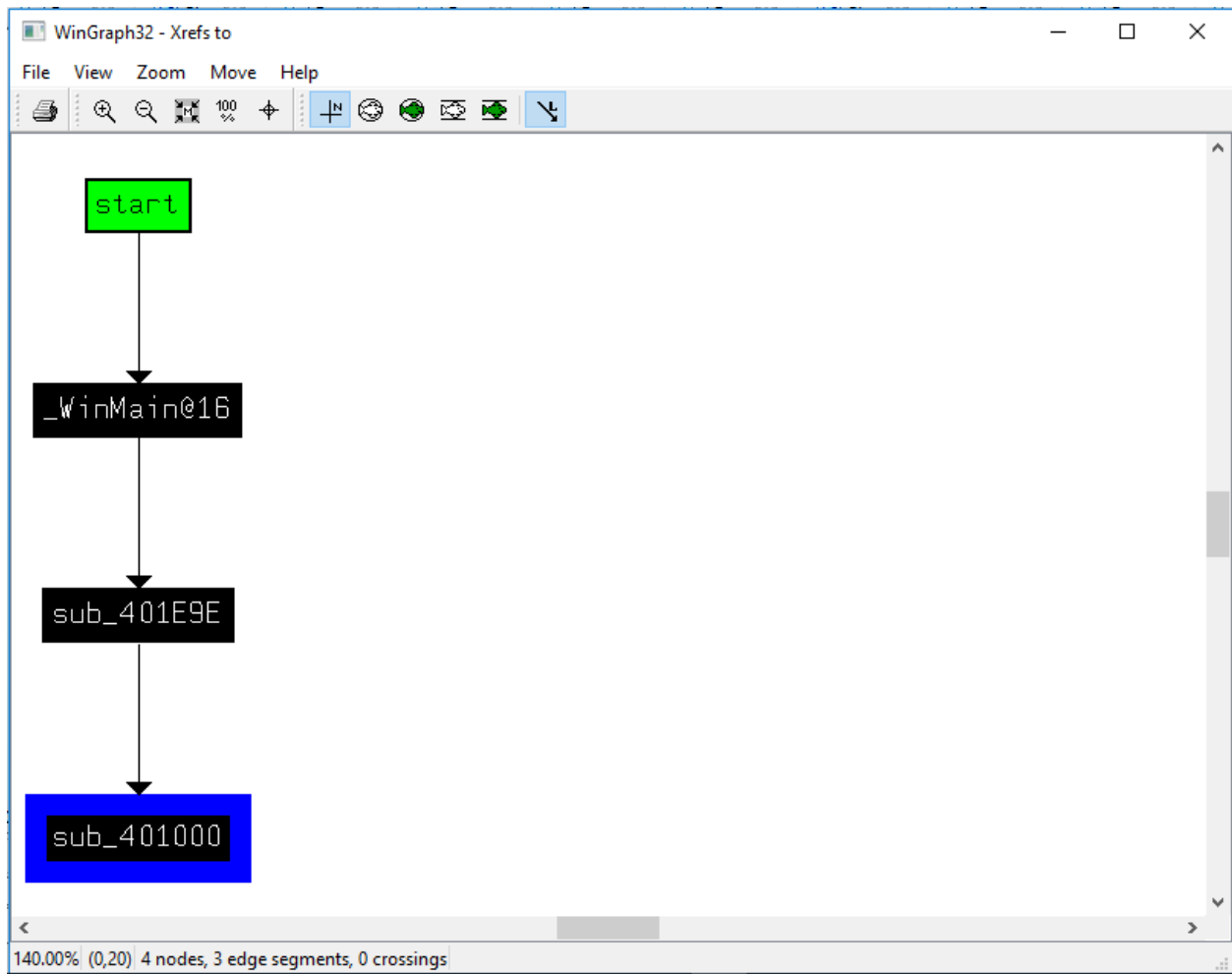
- gx7ekbenv2riucmf.onion
- 57g7spgrzlojinas.onion
- xxlvbrloxvriy2c5.onion
- 76jdd2ir2embyv47.onion
- cwwnhwhlz52ma.onion

Bên dưới nữa là link để tải + cài tor package.

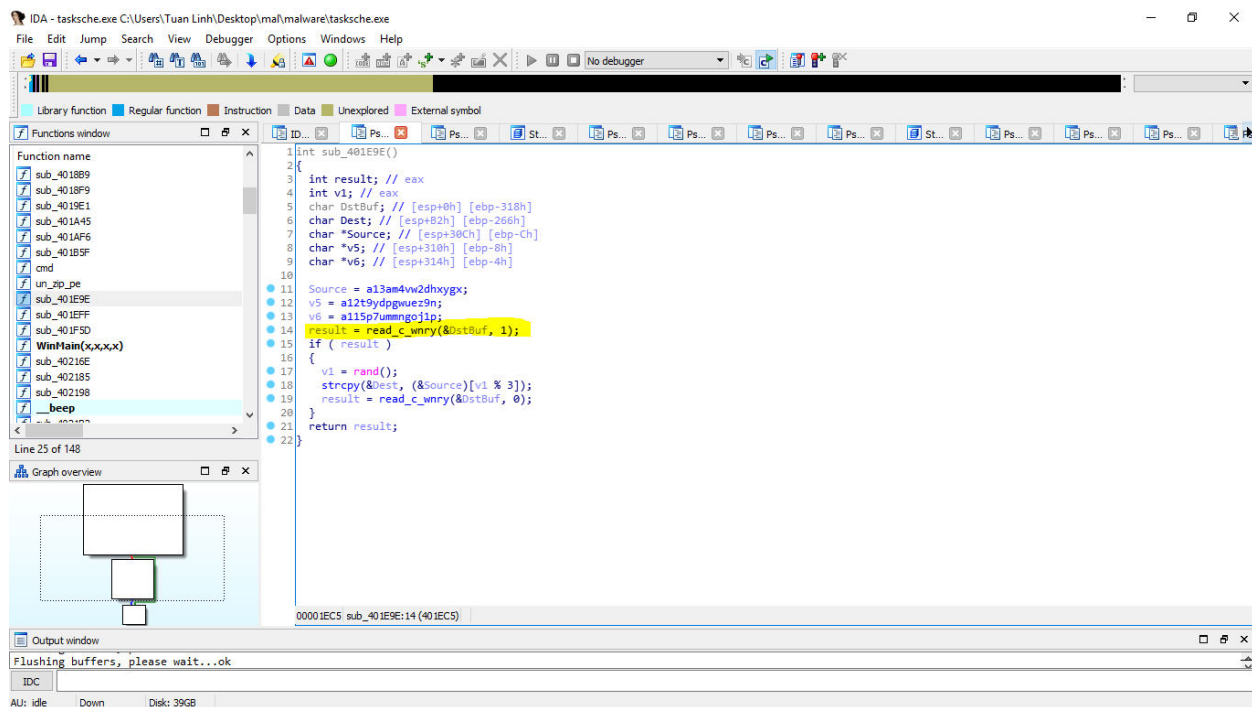
Nó có thể là địa chỉ của C&C panel. Dựa vào từ khóa “c.wnry” ta dễ dàng tìm được function đọc file này



Trace function này bằng Xref :



sub_401E9E



DstBuf (sau khi read c_wnry) được lưu vào results

Đọc code ta thấy nó chọn ngẫu nhiên 1 trong 3 địa chỉ ví bitcoin từ array Source và lưu vào *Dest

`v1 = rand();`

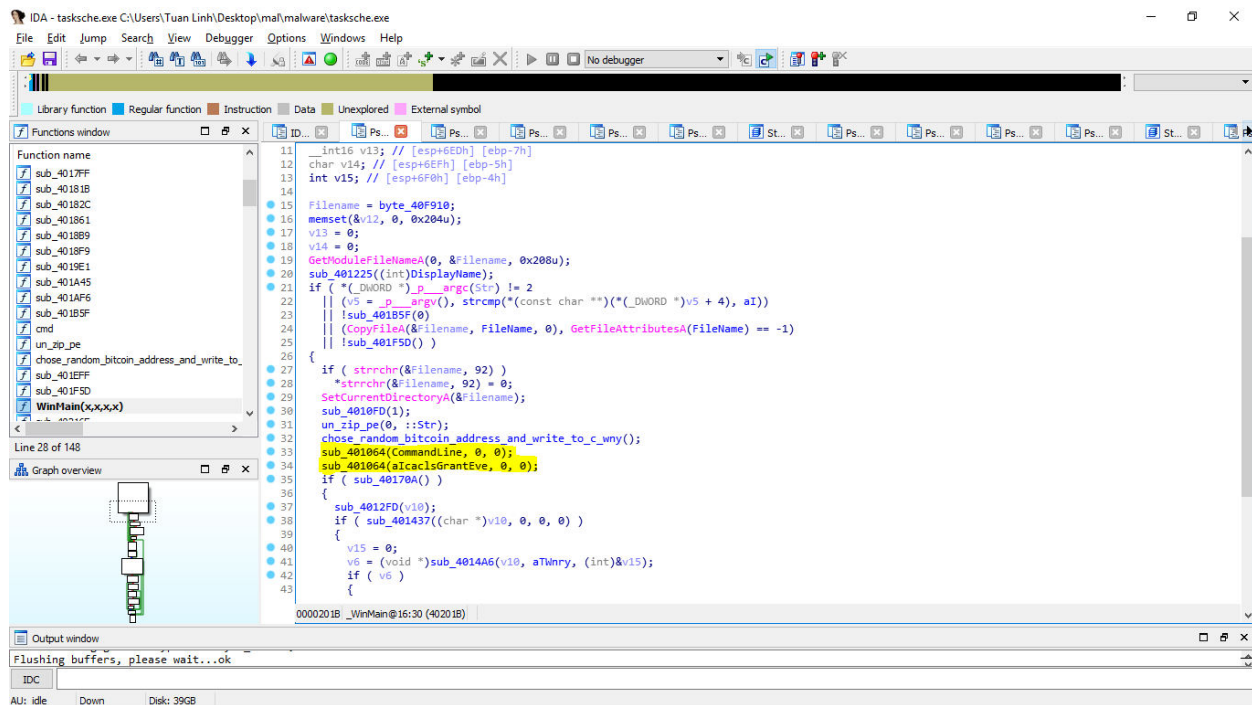
`strcpy(&Dest, (&Source)[v1 % 3]);`

Như vậy function này chức năng chủ yếu là chọn ra 1 trong 3 ví bitcoin ở trên.

Ở đây thì một điều thú vị xảy ra, lúc đầu không chú ý ☺ file c.wnry thì hàng loạt byte đầu là NULL, sau khi fread đã đọc đủ số byte nó cần để ghi bitcoin address vào (những byte này là NULL nếu lần đầu malware chạy) nên hàm fread() sẽ trả về NULL, như vậy trong lần đọc đầu tiên result = NULL, chính vì vậy mới có

If(result) => chọn trong 3 bitcoin address => chạy lại hàm read_cwnry() lần nữa => lưu vào trong c.wnry. Thực tế hàm này ko phải để đọc C&C panel url như mình nghĩ lúc đầu, mà nó dùng để lưu địa chỉ ví bitcoin vào c.wnry

Tới đây thì mọi thứ trở lại WinMain() ☺



Thôi thì trước khi quay lại phân tích mấy file kia, mình giải thích nhanh cái hàm **sub_401064** này vậy

```
int __cdecl sub_401064(LPSTR lpCommandLine, DWORD dwMilliseconds, LPDWORD lpExitCode)
{
    struct _STARTUPINFOA StartupInfo; // [esp+8h] [ebp-54h]
    struct _PROCESS_INFORMATION ProcessInformation; // [esp+4Ch] [ebp-10h]

    StartupInfo.cb = 68;
    memset(&StartupInfo.lpReserved, 0, 0x40u);
    ProcessInformation.hProcess = 0;
    ProcessInformation.hThread = 0;
    ProcessInformation.dwProcessId = 0;
    ProcessInformation.dwThreadId = 0;
    StartupInfo.wShowWindow = 0;
    StartupInfo.dwFlags = 1;
    if ( !CreateProcessA(0, lpCommandLine, 0, 0, 0, 0x8000000u, 0, 0, &StartupInfo, &ProcessInformation) )
        return 0;
    if ( dwMilliseconds )
    {
        if ( WaitForSingleObject(ProcessInformation.hProcess, dwMilliseconds) )
            TerminateProcess(ProcessInformation.hProcess, 0xFFFFFFFF);
        if ( lpExitCode )
            GetExitCodeProcess(ProcessInformation.hProcess, lpExitCode);
    }
    CloseHandle(ProcessInformation.hProcess);
    CloseHandle(ProcessInformation.hThread);
    return 1;
}
```

Ngộ ra cái gì đó thú vị ☺ như một RE-er nổi tiếng từng nói (hình như lão lena151 từng nói trong mấy tuts về cracking thì phải, cơ mà cũng 2 năm hơn rồi...ko nhớ rõ lắm ☺) msdn luôn là bạn của RE-er

[https://msdn.microsoft.com/en-us/library/windows/desktop/ms682425\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms682425(v=vs.85).aspx)

Với :

```
BOOL WINAPI CreateProcess(  
    _In_opt_ LPCTSTR lpApplicationName,  
    _Inout_opt_ LPTSTR lpCommandLine,  
    _In_opt_ LPSECURITY_ATTRIBUTES lpProcessAttributes,  
    _In_opt_ LPSECURITY_ATTRIBUTES lpThreadAttributes,  
    _In_ BOOL bInheritHandles,  
    _In_ DWORD dwCreationFlags,  
    _In_opt_ LPVOID lpEnvironment,  
    _In_opt_ LPCTSTR lpCurrentDirectory,  
    _In_ LPSTARTUPINFO lpStartupInfo,  
    _Out_ LPPROCESS_INFORMATION lpProcessInformation  
);
```

Thì :

The *lpApplicationName* parameter can be **NULL**. In that case, the module name must be the first white space-delimited token in the *lpCommandLine* string

Như vậy lúc này có thể hiểu attrib + h được chia thành ['attrib', '+h'], lúc này chạy trên cmd sẽ là attrib và parameter +h

Nói đơn giản thì đây là mẹo sài CreateProcess để sử dụng cmd, 2 lệnh sau được execute trên cmd :

- **Attrib + h**

```
attrib +h config.sys
```

Add the hidden attribute to the config.sys file, causing it to be not be seen by the average user.

⇒ thêm attribute “hidden” cho các file ở current directory và sub directory

- **icacs . /grant Everyone:F /T /C /Q**

icacs là tool dùng để modify NTFS file system's permissions. Ở đây permissions được đặt cho **Everyone** với các permission sau :

- o A sequence of simple rights:

F (full access)

M (modify access)

RX (read and execute access)

R (read-only access)

W (write-only access)

/t

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 versus its destination.

/q

Suppresses success messages.

Q.wnry

The screenshot displays a hex editor interface with the following components:

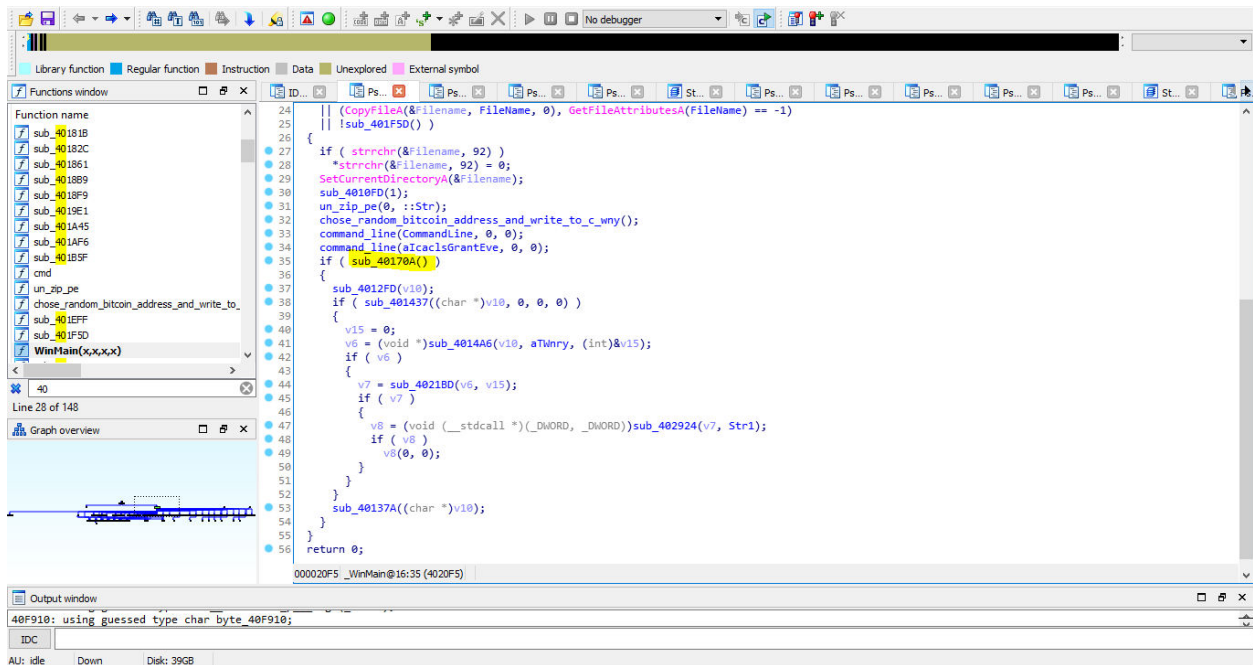
- File Information:**
 - File Name: r.wnry
 - File Size: 864 bytes
- Data Inspector (Little-endian):**

Type	Unsigned (*)	Signed (s)
8-bit Integer	81	81
16-bit Integer	14929	14929
24-bit Integer	2112081	2112081
32-bit Integer	538982993	538982993
64-bit Integer (*)	8386098704551000657	
64-bit Integer (s)		8386098704551000657
16-bit Float: P	0.7895508	
32-bit Float: P	1.3571822e-19	
64-bit Float: P	3.98827195877531e+252	
MS-DOS DateTime	Invalid Date	
OLE 2.0 DateTime	Invalid Date	
UNIX DateTime	1987-01-30 05:29:53 UTC	
Macintosh HFS DateTime	1921-01-29 12:29:53 Local	
Macintosh HFS+ DateTime	1921-01-29 05:29:53 UTC	
- Hex View:** Displays a grid of hexadecimal values (e.g., 3A 20 20 57 68 61 74) and their corresponding ASCII characters (e.g., .What's.wrong). The address 00000051 is highlighted.
- Search Panel:**
 - Go To:** Current Address (0x00000000), Last Address (0x0000035F).
 - Search for:** A text input field.
 - Data Type:** Radio buttons for 8-bit Integer, 16-bit Integer, 24-bit Integer, 32-bit Integer, 64-bit Integer, 16-bit Floating Point, 32-bit Floating Point, 64-bit Floating Point, Hexadecimal Values, Text.
 - Byte Order:** Checkboxes for Little-endian (checked) and Big-endian.
 - Search now:** A button to execute the search.

Đọc qua thấy mấy chỗ ghi %s, có lẽ sau đó program sẽ đọc và ghi đè file này sau, file này hẳn là file .txt đòi tiền chuộc.

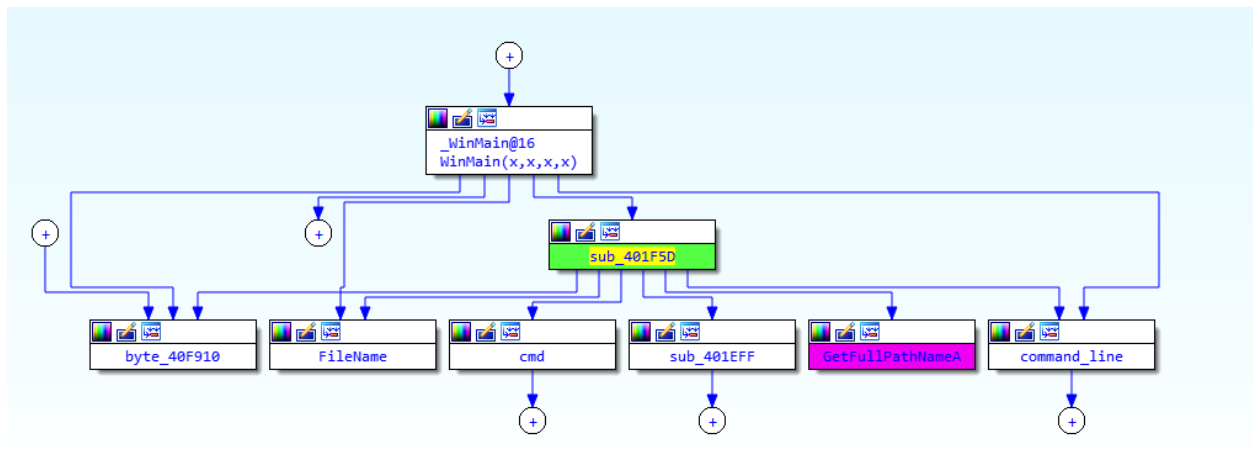
⇒ **Tổng kết : Thông qua phân tích các file được zip và dấu trong resource của PE này, ta thu được thông tin về message đòi tiền chuộc, địa chỉ C&C server của con malware này**

Tiếp tục trở lại phân tích các function trong WinMain() :



Sub_401F5D: execute malware + check mutex

Ta xem qua bằng graph trên IDA để có cái nhìn tổng quát về hàm này



Có thể phán đoán nó làm gì đó dùng cmd, làm gì đó với path của file, **sub_401EFF** là cái chi chi?

```

signed int __cdecl sub_401EFF(int a1)
{
    int v1; // esi
    HANDLE v2; // eax
    char Dest; // [esp+4h] [ebp-64h]

    sprintf(&Dest, aSD, aGlobalMswinzon, 0);
    v1 = 0;
    if ( a1 <= 0 )
        return 0;
    while ( 1 )
    {
        v2 = OpenMutexA(0x100000u, 1, &Dest);
        if ( v2 )
            break;
        Sleep(1000u);
        if ( ++v1 >= a1 )
            return 0;
    }
    CloseHandle(v2);
    return 1;
}

```

a1 = 60

aSD = '%s%d'

aGlobalMswinzon = 'Global\MsWinZonesCacheCounterMutexA'

OpenMutex function

Opens an existing named mutex object.

Syntax

C++

```
HANDLE WINAPI OpenMutex(  
    _In_ DWORD    dwDesiredAccess,  
    _In_ BOOL     bInheritHandle,  
    _In_ LPCTSTR  lpName  
);
```

Return value

If the function succeeds, the return value is a handle to the mutex object.

If the function fails, the return value is **NULL**. To get extended error information, call [GetLastError](#).

If a named mutex does not exist, the function fails and [GetLastError](#) returns **ERROR_FILE_NOT_FOUND**.

Như vậy nghĩa là con malware này sẽ check mutex

“Global\MsWinZonesCacheCounterMutexA0”, nhớ kĩ là **A0** mà không phải **A**, vì kia là “%s%d”.

Nếu search mạng thì thấy đa phần các mẫu WannaCry là A mà không phải A0.

Nếu success, tức là không có phần mềm nào sử dụng mutex đó, tức là $v2 \neq 0 \Rightarrow$ break vòng lặp while()

Nếu fail, sleep(1000) . Ở đây $++v1 \leq a1$ tức là $v1 += 1$ rồi mới so sánh với $a1 \Rightarrow$ lặp 60 lần * 1000 = 60000s để kiểm tra mutex.

\Rightarrow Ta có thể đổi tên **sub_()** này thành **check_mutex()**


```

300: sub_401F5D()
{
    CHAR Buffer; // [esp+4h] [ebp-208h]
    char v2; // [esp+5h] [ebp-207h]
    __int16 v3; // [esp+209h] [ebp-3h]
    char v4; // [esp+208h] [ebp-1h]

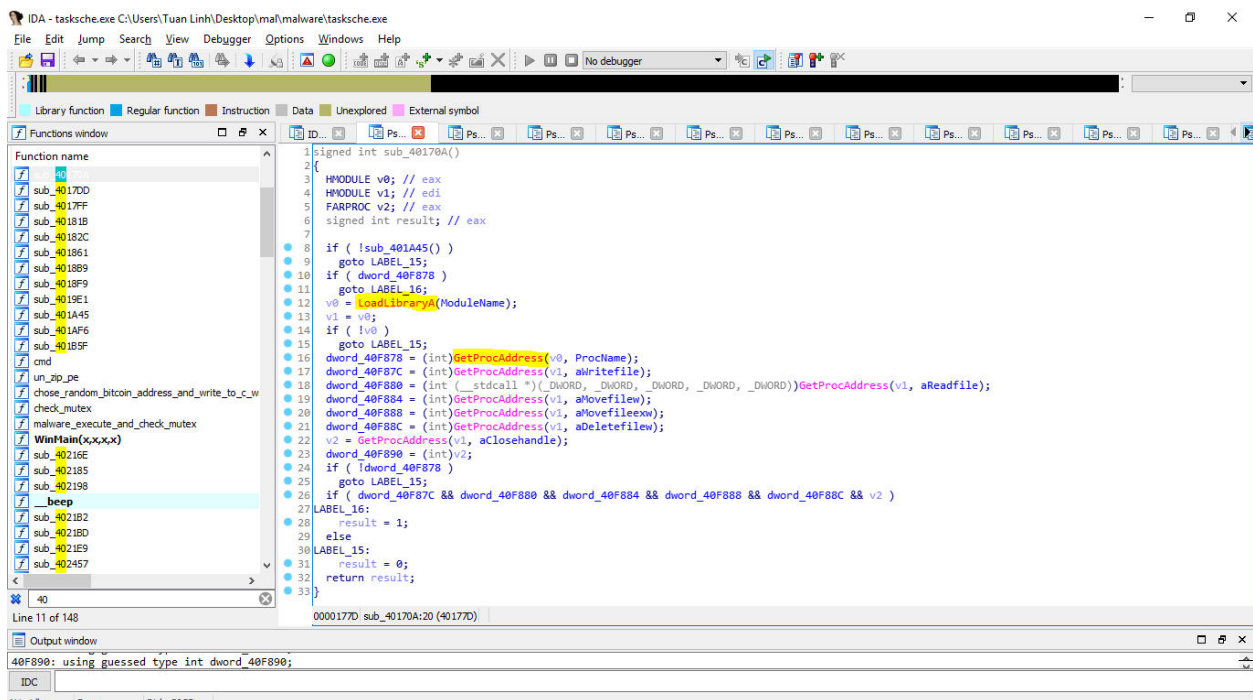
    Buffer = byte_40F910;
    memset(&v2, 0, 0x204u);
    v3 = 0;
    v4 = 0;
    GetFullPathNameA(fileName, 520u, &Buffer, 0);
    return cmd((int)&Buffer) && check_mutex(60) || command_line(&Buffer, 0, 0) && check_mutex(60);
}

```

[https://msdn.microsoft.com/en-us/library/windows/desktop/aa364963\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa364963(v=vs.85).aspx)

Căn bản thì hàm này không làm gì khác ngoài việc execute con malware này và kiểm tra mutex để chắc chắn rằng không có 2 con chạy cùng 1 lúc

Sub_40170A(): load dll



Chú ý 2 cái func bôi vàng, từ sau hôm feed team bài Step_To_Step trong VNPT Secathon vẫn ám ảnh 2 cái func này. Nhìn cái nhận ra ngay là gọi func trong DLL, chả biết nên vui hay nên buồn ☹

Start-up method :

Lúc đầu cứ ngỡ là không có gì thú vị, nhìn qua :

```
wscat(&Dest, &Source);
v12 = 0;
while ( 1 )
{
    if ( v12 )
        RegCreateKeyW(HKEY_CURRENT_USER, &Dest, &phkResult);
    else
        RegCreateKeyW(HKEY_LOCAL_MACHINE, &Dest, &phkResult);
    if ( phkResult )
    {
        if ( a1 )
        {
            GetCurrentDirectoryA(0x207u, &Buffer);
            v1 = strlen(&Buffer);
            v2 = RegSetValueEx(phkResult, ValueName, 0, 1u, (const BYTE *)&Buffer, v1 + 1) == 0;
        }
        else
        {
            cbData = 519;
            v3 = RegQueryValueExA(phkResult, ValueName, 0, 0, (LPBYTE)&Buffer, &cbData);
            v2 = v3 == 0;
            if ( !v3 )
                SetCurrentDirectoryA(&Buffer);
        }
        RegCloseKey(phkResult);
        if ( v2 )
            break;
    }
    if ( ++v12 >= 2 )
        return 0;
}
return 1;
}
```

000010FD sub_4010FD:55 (4010FD)

Đầu tiên thì nó thử reg key ở HKEY_CURRENT_USER hoặc HKEY_LOCAL_MACHINE, nếu được thì bắt đầu set registry value với lpValueName = ValueName = "wd" và lpData (byte) = Buffer = địa chỉ path hiện tại của nó (process đang chạy qua lệnh GetCurrentDirectoryA())

```
LONG WINAPI RegSetValueEx(
    _In_ HKEY hKey,
    _In_opt_ LPCTSTR lpValueName,
    _Reserved_ DWORD Reserved,
    _In_ DWORD dwType,
    _In_ const BYTE *lpData,
    _In_ DWORD cbData
);
```

⇒ Đơn giản chỉ là run file in start up với method là tạo registry key. Nhưng nếu nhìn kĩ hơn thì ta sẽ thấy điều thú vị :

Function được gọi là **RegCreateKeyW(HKEY_CURRENT_USER, &Dest, &phkResult);**

Vậy Dest là gì?

Nhìn : `qmemcpy(&Dest, &off_40E04C, 20u);`

⇒ Copy 20 byte `off_40E04C` vào `Dest`

```
.data:0040E04C off_40E04C      dd offset unk_6F0053  
.data:0040E050      dd offset unk_740066  
.data:0040E054      dd offset unk_610077  
.data:0040E058      dd offset unk_650072
```

Đại khái mà nói, nó copy các giá trị trên vào Dest, và tạo `lbSubKey = Dest`

```
.rsrc:006F0053 unk_6F0053      db  90h      |          ; DATA XREF: .data:off_40E04C↑o  
.rsrc:006F0054      db  0EAh    ; ẽ  
.rsrc:006F0055      db  53h    ; S
```

⇒ Kết luận : vẫn chả có gì thú vị