**Thu thập thông tin về binary**

**Taskche.exe :**

**#wannacry\_variant, #wcry, ransomware**

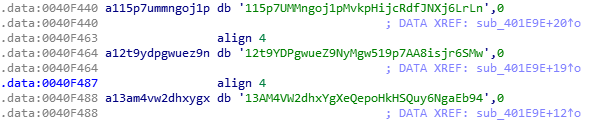


**Bitcoin address found :**

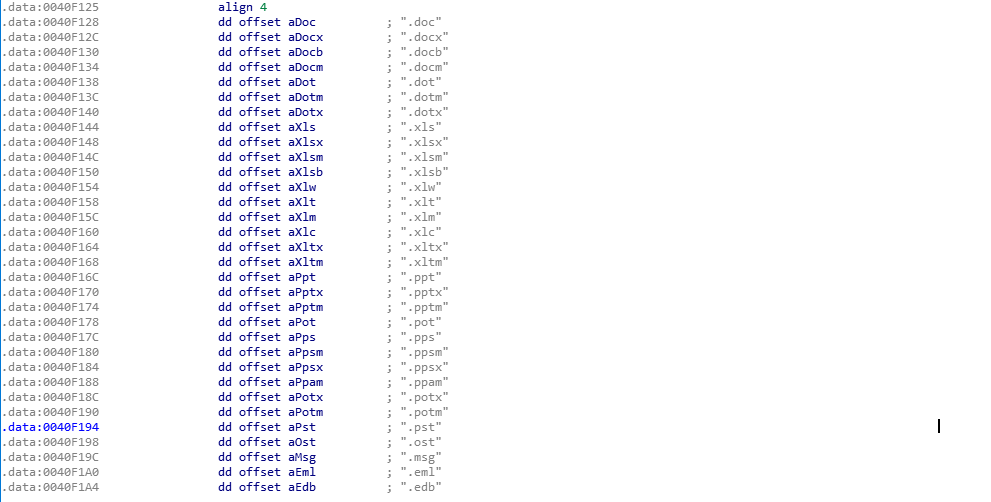
**115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn**

**12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw**

**13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94**

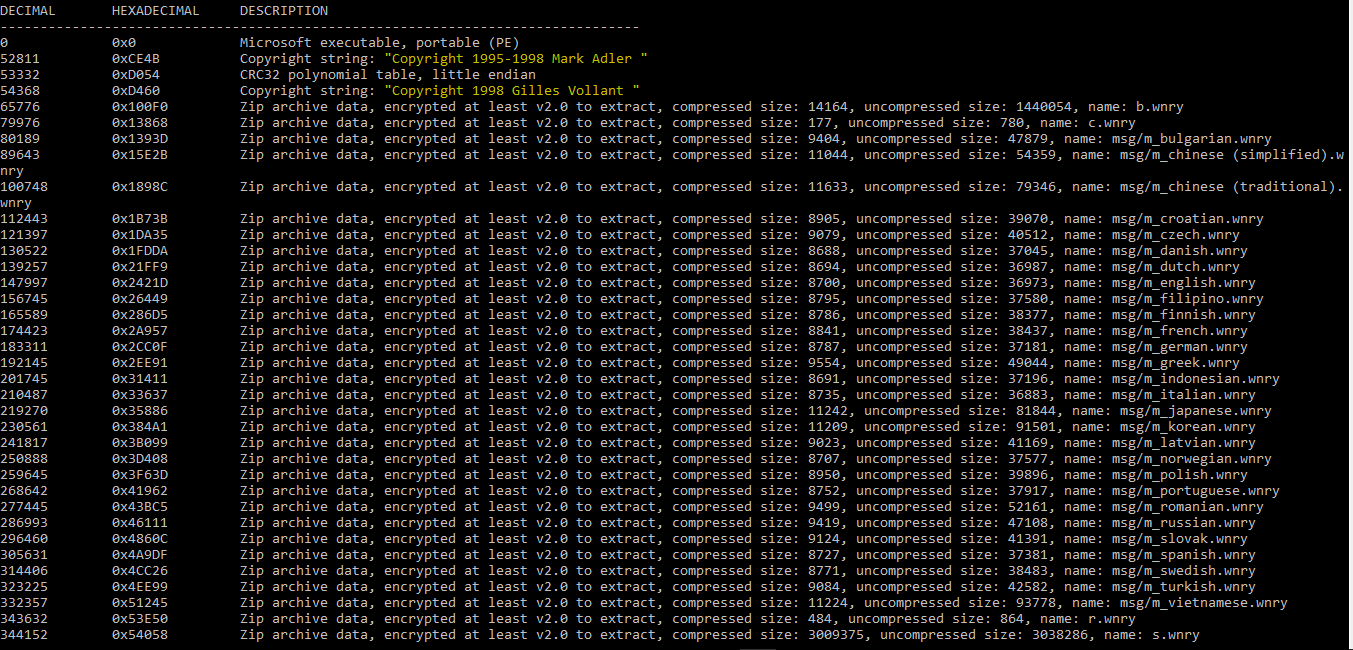


**Các file extension được nhắm tới để encrypt :** .doc, .docx, .xls, .xlsx, .ppt, .pptx, .pst, .ost, .msg, .eml, .vsd, .vsdx, .txt, .csv, .rtf, .123, .wks, .wk1, .pdf, .dwg, .onetoc2, .snt, .jpeg, .jpg, .docb, .docm, .dot, .dotm, .dotx, .xlsm, .xlsb, .xlw, .xlt, .xlm, .xlc, .xltx, .xltm, .pptm, .pot, .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

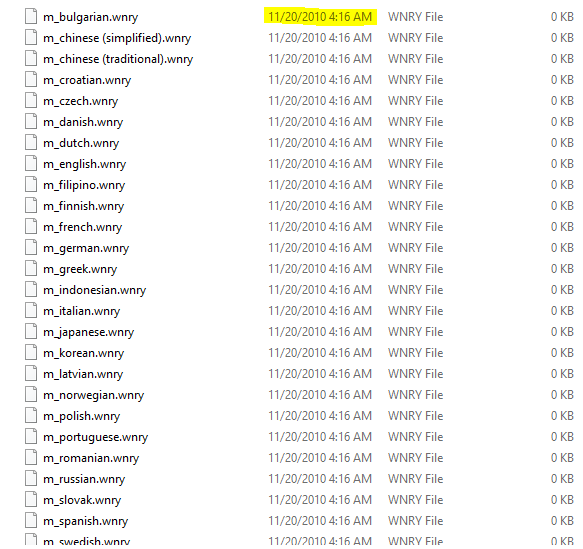


**Các country/language được nhắm tới và thông báo đòi tiền chuộc :** m\_bulgarian, m\_chinese (simplified), m\_chinese (traditional), m\_croatian, m\_czech, m\_danish, m\_dutch, m\_english, m\_filipino, m\_finnish, m\_french, m\_german, m\_greek, m\_indonesian, m\_italian, m\_japanese, m\_korean, m\_latvian, 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 vứt vào IDA, ta cũng có thể đoán được phần nào nếu bắt gặp 2 strings được đánh dấu bên dưới

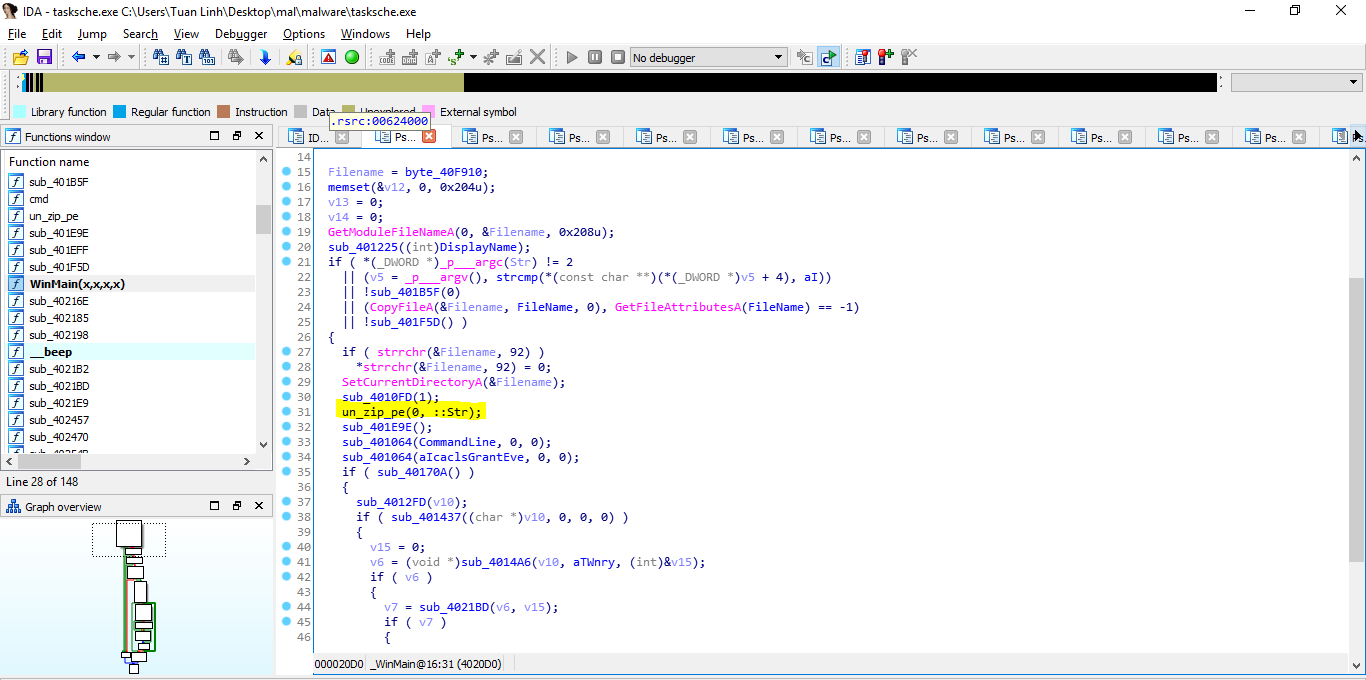


Un-zip with 7-zip without password, I found that the creation date was pretty old :3 (Maybe it was changed \*intended\*)

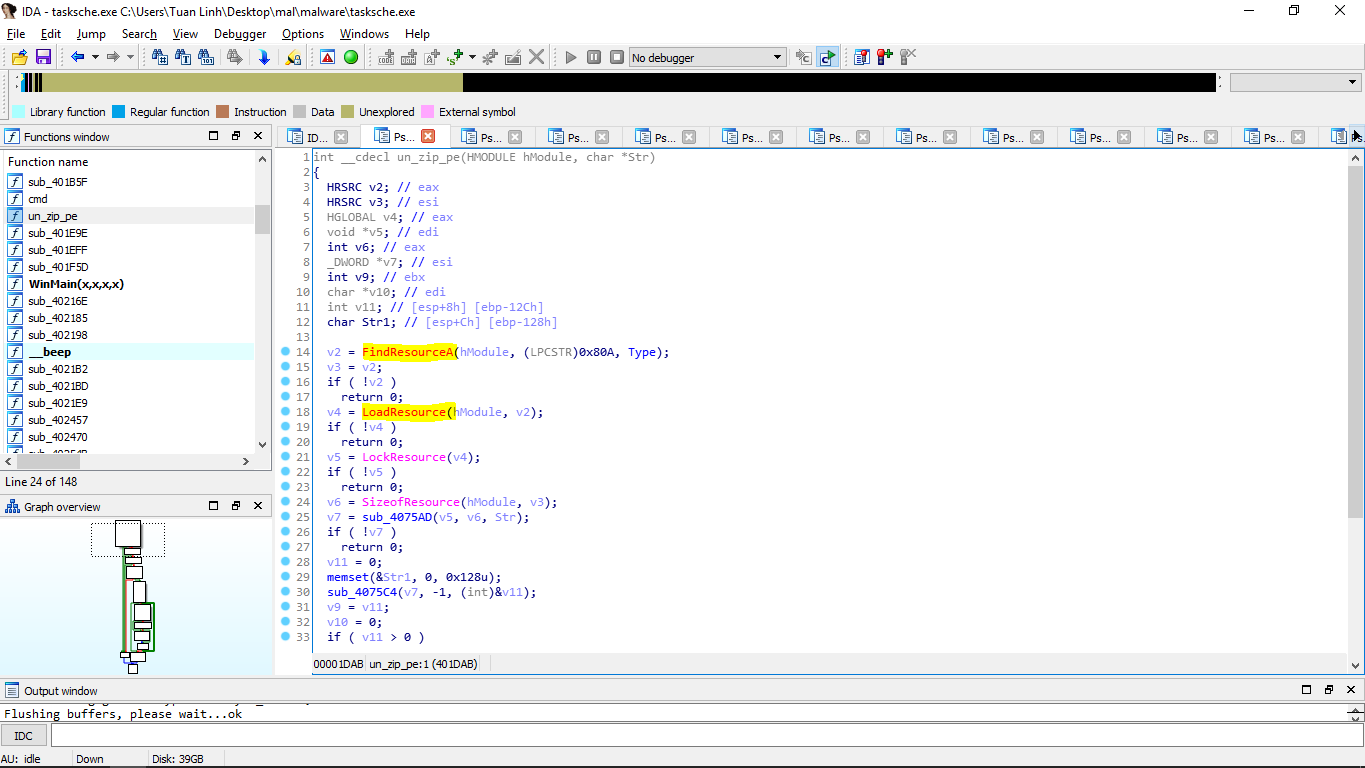


**Phân tích code :**

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

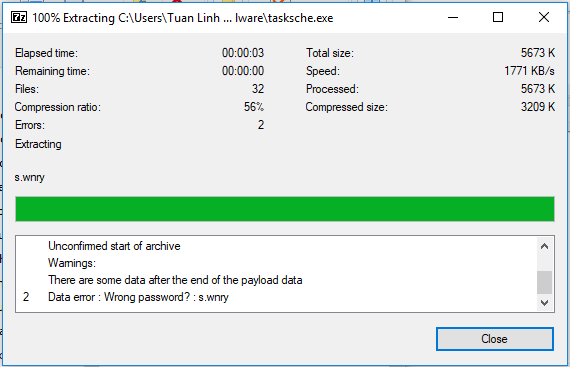


Function này load resource => unzip nó với pass là Str = ‘WNcry@20l7’



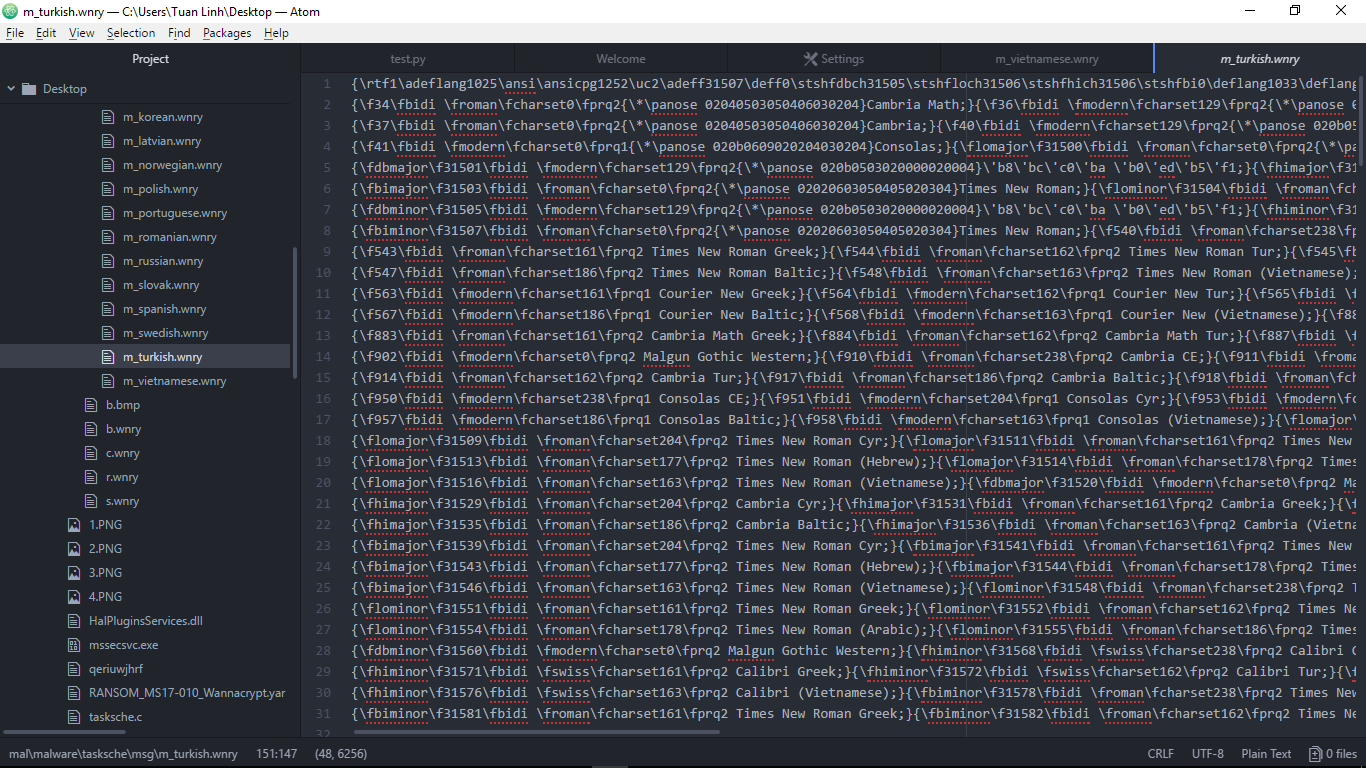


Thử xem :

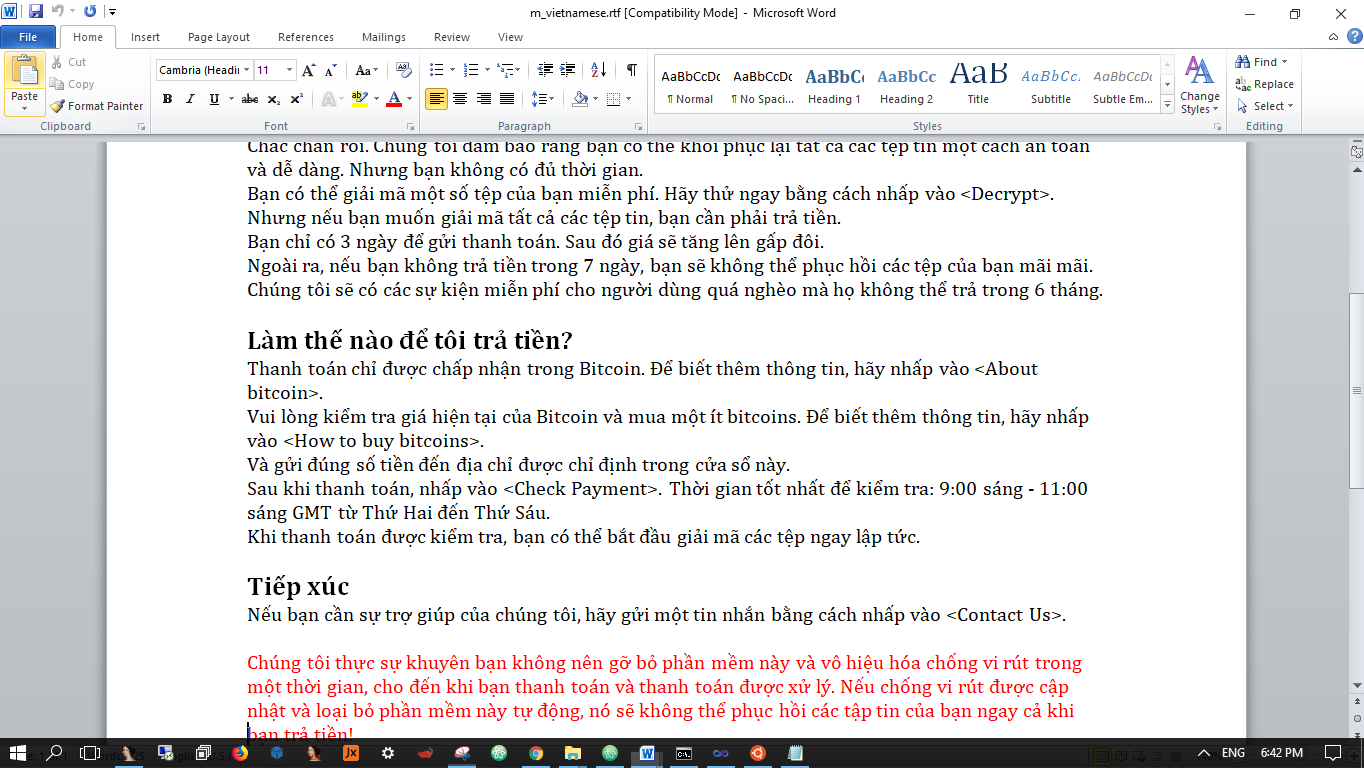


Yuppppppppp! Nhưng vẫn chưa decompress được s.wnry ☹ Nhưng để sau vậy, ngó thử mấy file vừa decrypt trước.

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

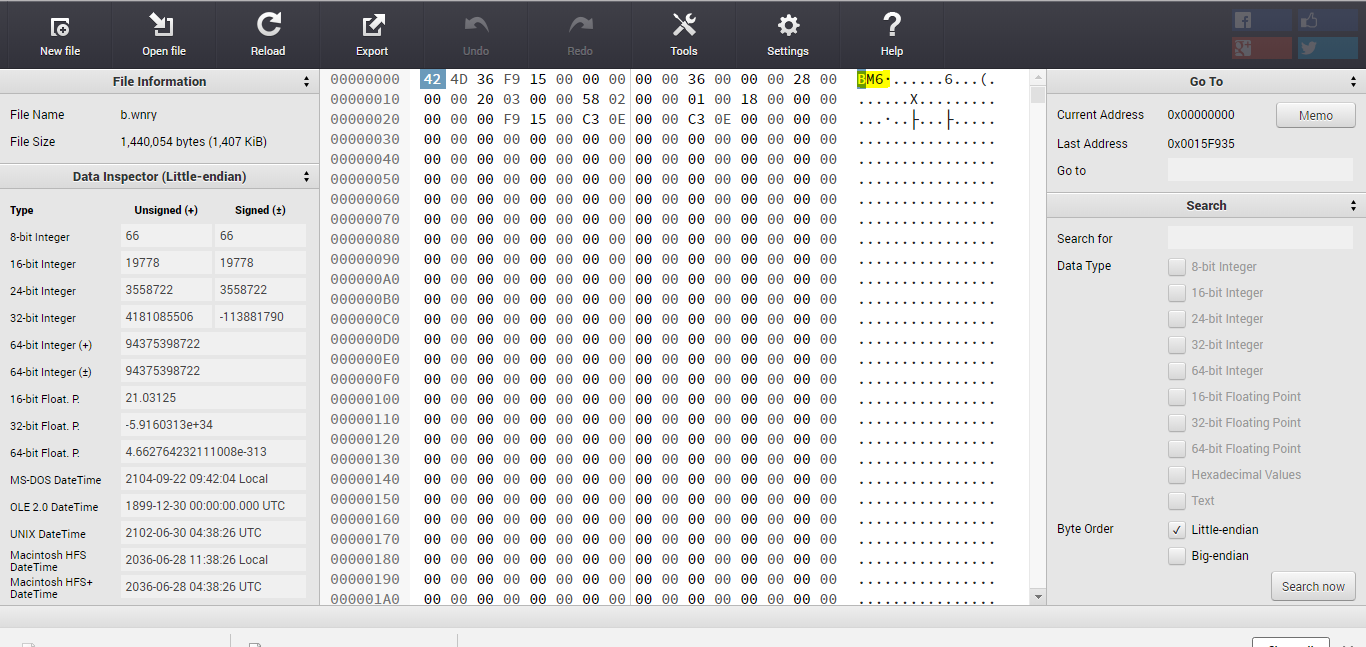


Kinh nghiệm của một người chơi forensic cho mình biết đây là file rtf, sửa lại extension và bật lên xem :

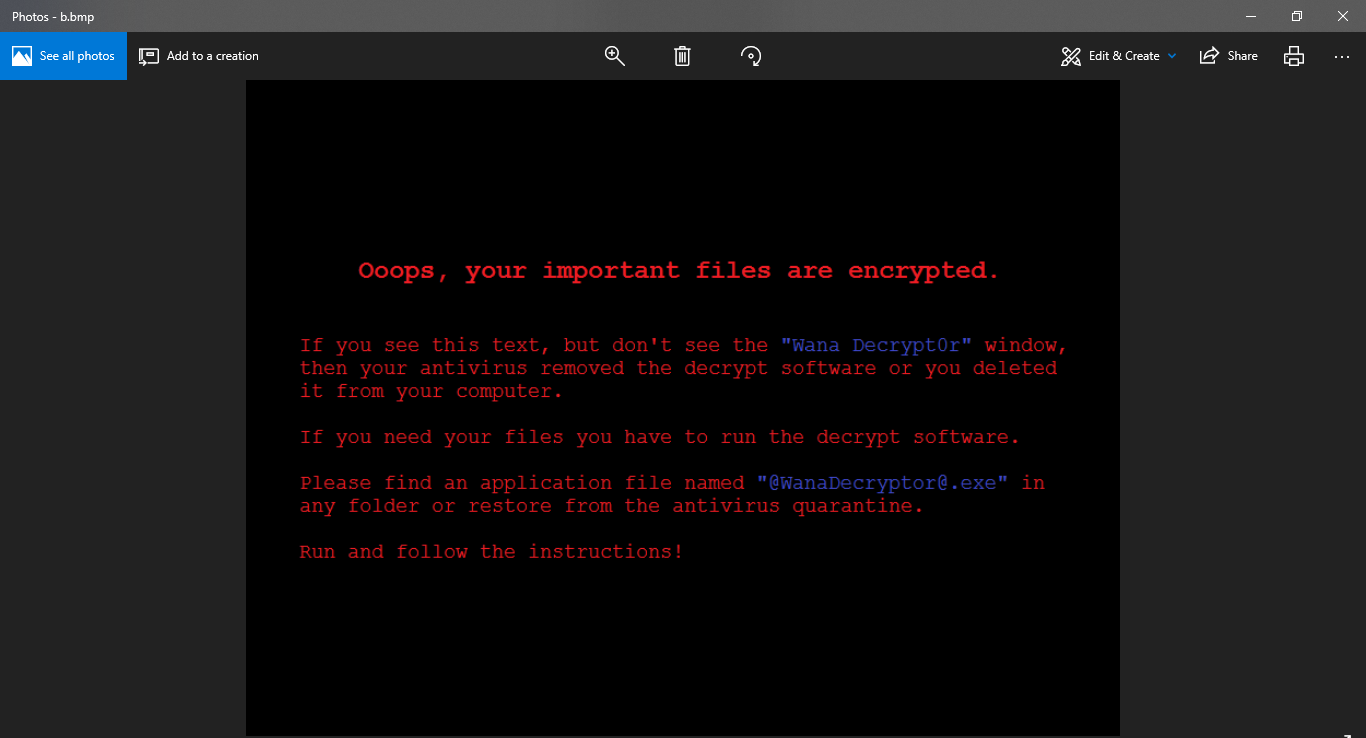


Thử các file khác xem :

**B.wnry**

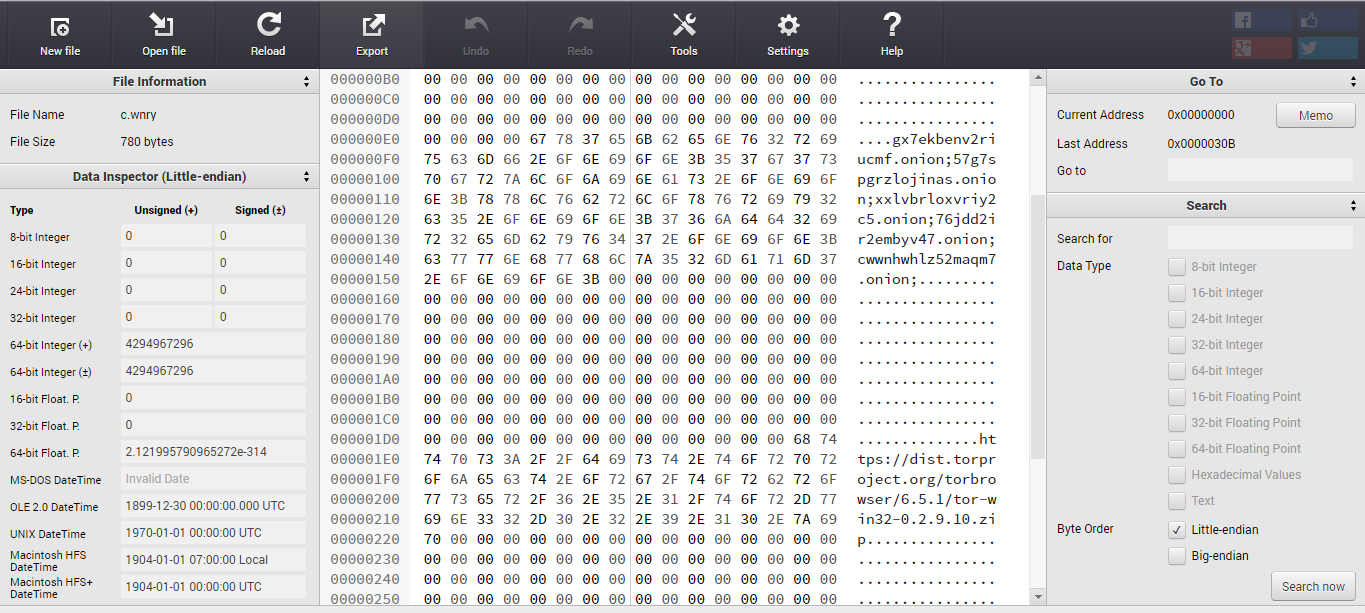


Và lần nữa, kinh nghiệm của một người chơi forensic lại cho mình biết đây là file ảnh bitmap BMP qua 2 bit hex đầu tiên (BM). Thực tế bạn có thể sài tool để nhận dạng, nhưng đọc và nhớ signature của một số file thông dụng thì tốt hơn (PE,MK,BM,PNG,etc…) (<https://www.garykessler.net/library/file_sigs.html>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



Lại đòi tiền nữa…

**C.wnry**

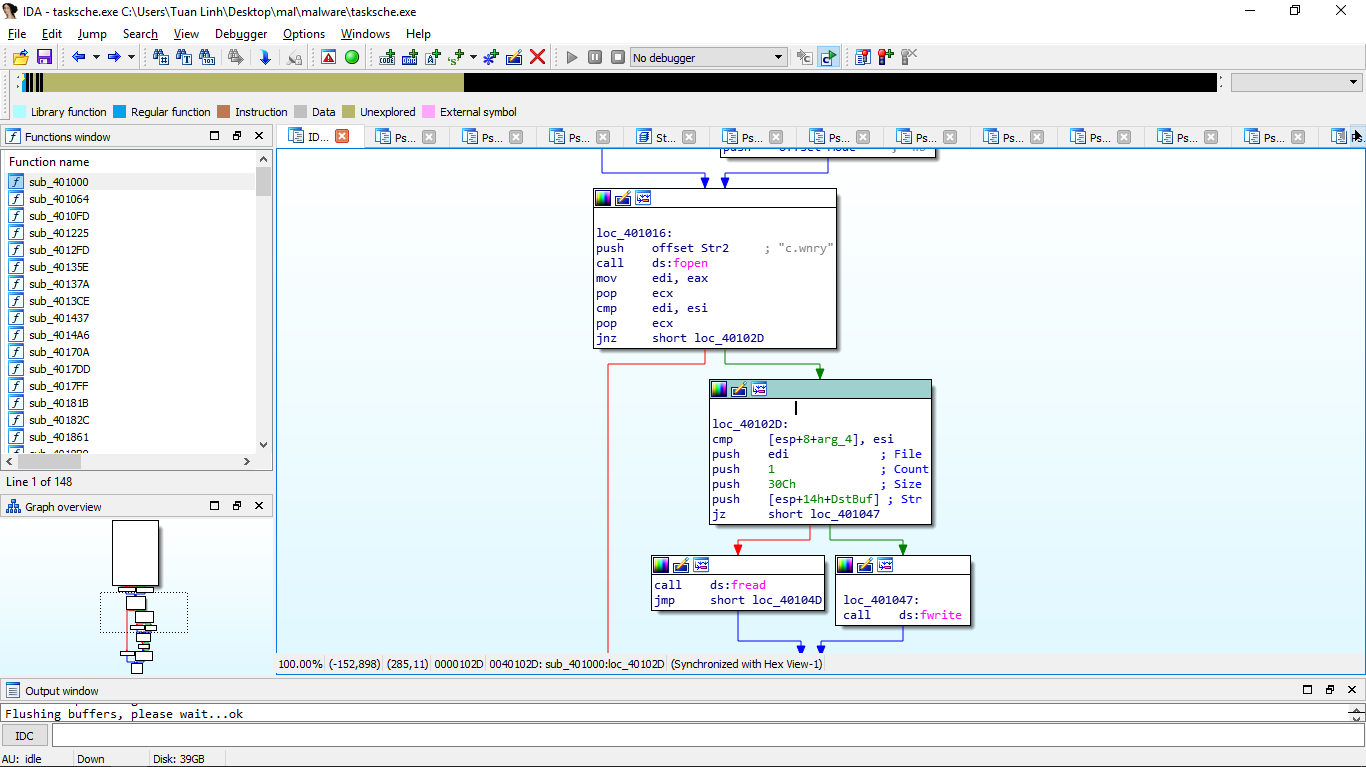


Cho vào hex, ta thấy ngay được 5 địa chỉ TOR sau :

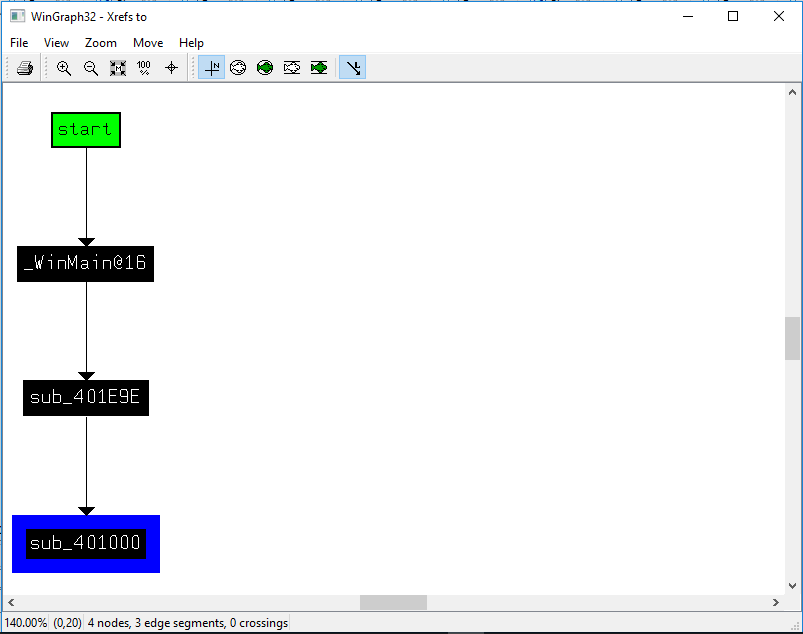
* gx7ekbenv2riucmf.onion
* 57g7spgrzlojinas.onion
* xxlvbrloxvriy2c5.onion
* 76jdd2ir2embyv47.onion
* cwwnhwhlz52ma.onion

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

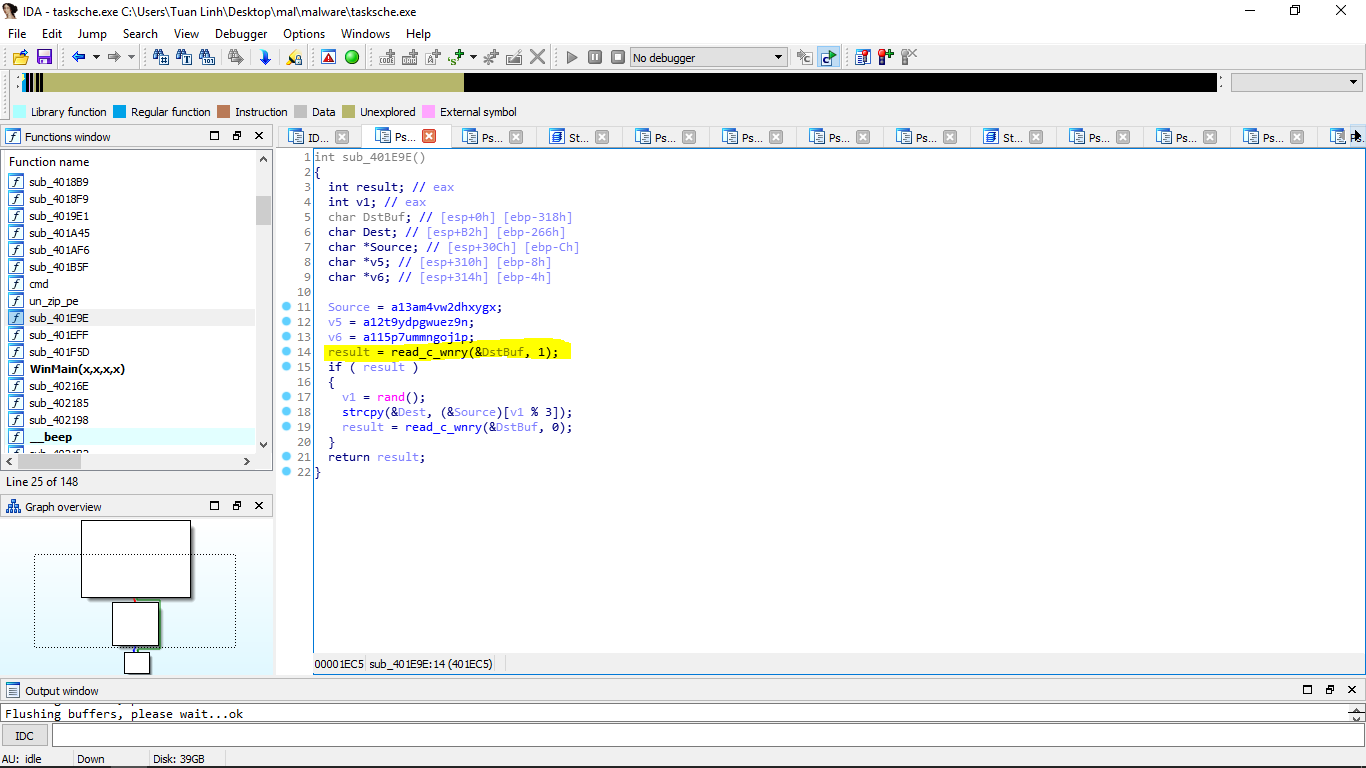
Nó có thể là địa chỉ của C&C panel. Dựa vào từ khóa “c.wnry” ta dễ dàng tìm được function đọc file này



Trace function này bằng Xref :



Xem function 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 arry Source và lưu vào \*Dest

v1 = rand();

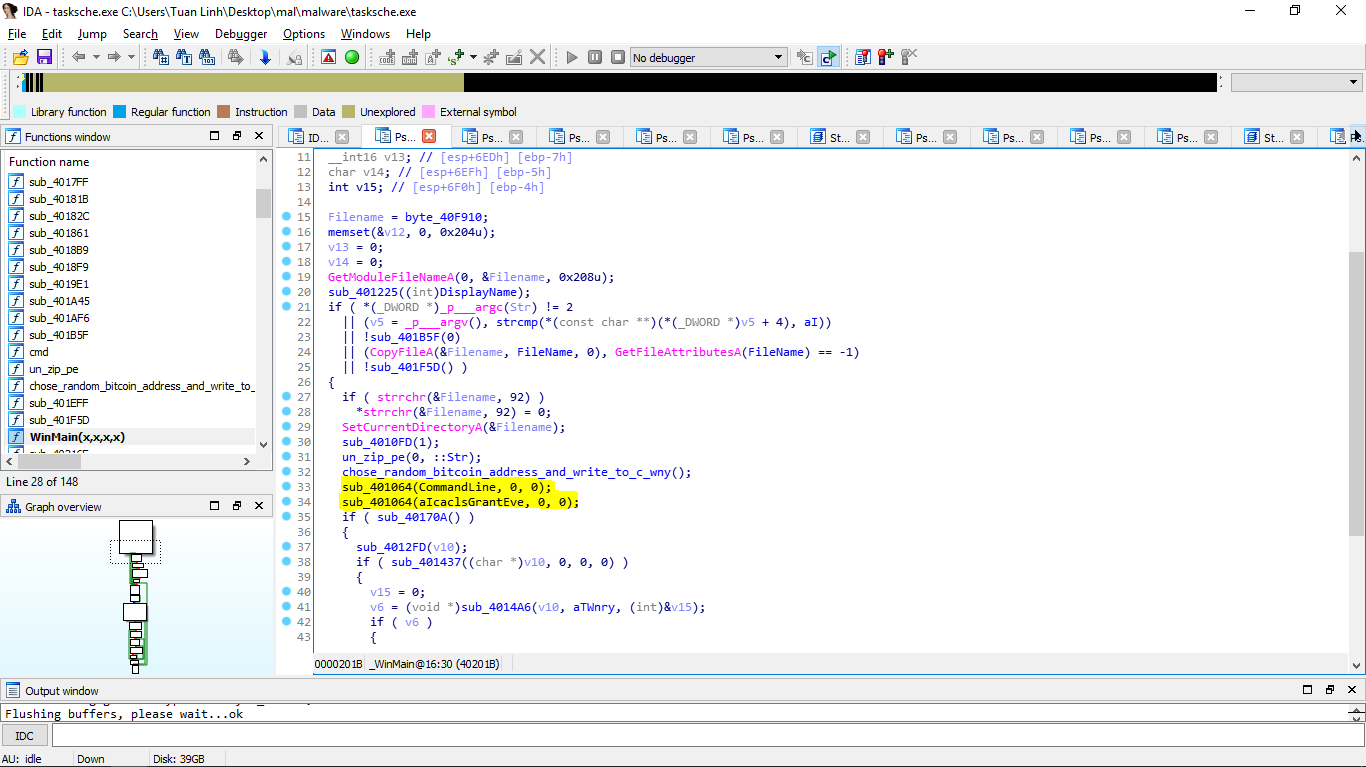
strcpy(&Dest, (&Source)[v1 % 3]);

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 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



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>

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

);

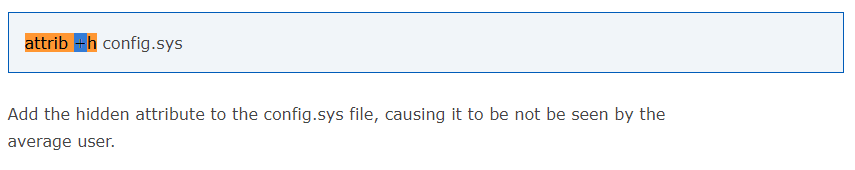
Thì :

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

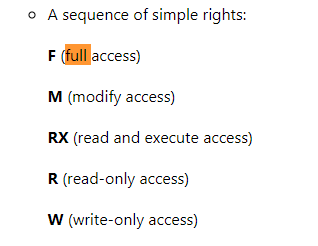
Nói đơn giản thì đây là mẹo sài CreateProcess để sử dụng cmd, 2 lệnh sau được execute trên cmd :

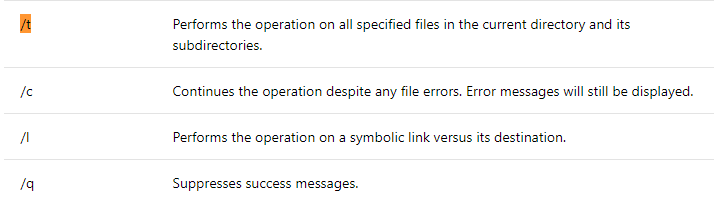
* **Attrib + h**



* thêm attribute “hidden” cho các file ở current directory và sub directory
* **icacls . /grant Everyone:F /T /C /Q**

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

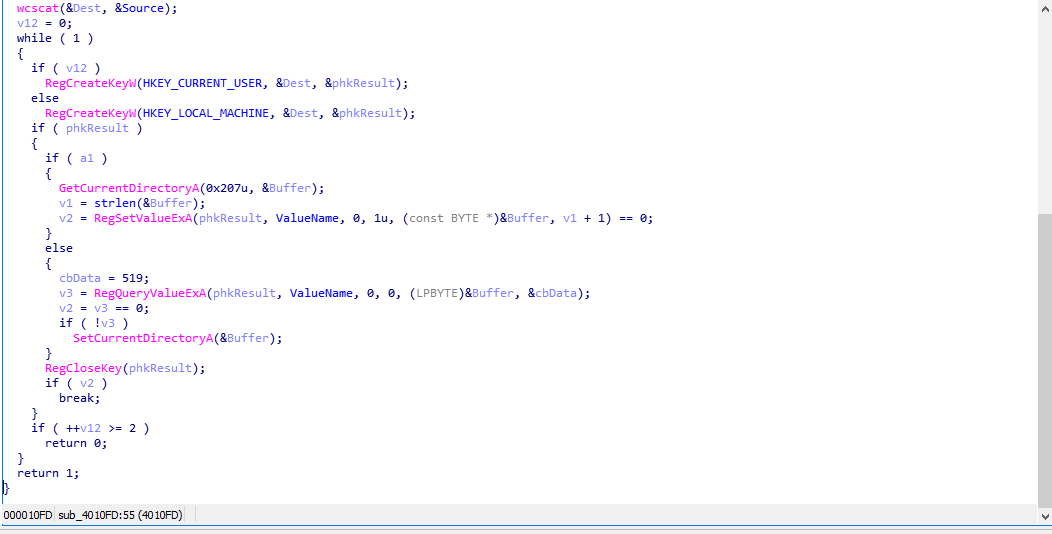




**Một vài function thú vị :**

* Start-up method :

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



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



Đại khái mà nói, nó copy các giá trị Unicode vào Dest, và tạo lbSubKey = Dest = kí tự Unicode



Nghe có vẻ quen quen