#KULGRAM
29 Mei 2017

Reverse Engineering “Malware”

Tutor : Yohanes Nugroho

Ini sebenernya materi untuk pemula banget, dasar untuk melakukan reverse engineering malware yang sederhana dan nanti diakhir akan dibahas malware challange dari flare-on CTF. untuk link soal bisa didownload dialamat ini

http://flare-on.com/files/Flare-On3_Challenges.zip

Kalo untuk mengetahui lebih banyak soal-soal dari CTF yang diadakan Flare-On, silahkan mampir ketulisan yang penulis pernah buat di :

http://blog.compactbyte.com/2016/10/03/tantangan-flare-on/

Jika tertarik dunia reverse engineering, khususnya CTF yang diadakan oleh flare-on, silahkan mencoba untuk menyelesaikan soal-soal yang tahun lalu dan coba untuk mengikuti challange dari Flare On yang tahun sekarang (2017). Jika berhasil menyelesaikan tantangan yang diberikan, akan ada kesempatan diwawancarai oleh FireEye.

Sebisa mungkin di Kulgram #ReversingID ini gunakan tools / aplikasi yang legal, biasanya tools yang dipakai oleh Reverser Professional adalah IDA Pro terbaru dengan decompiler. Dengan adalahnya decompiler kita bisa mendapatkan source code yang hampir mendekati source code yang asli.

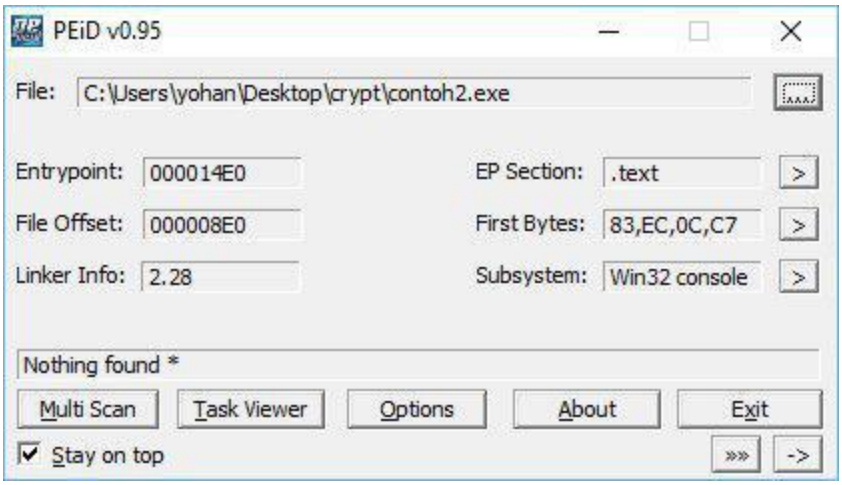
Pada kesempatan kali ini, kita akan membahas bagaimana menggunakan aplikasi IDA yang freeware. Silahkan download langsung diwebsite resminya

Untuk mempermudah akan saya berikan binari dan sourcenya (anggap saja hasil dari decompile)

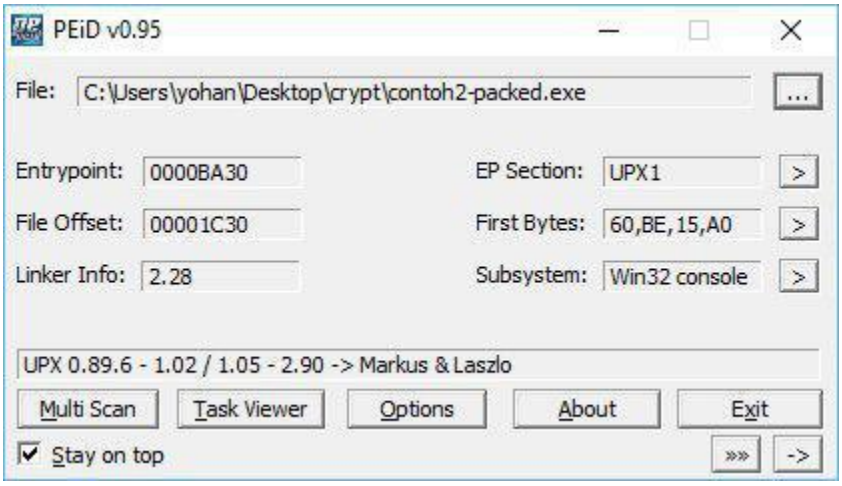
https://drive.google.com/open?id=0B-sySrTTra5nMmxyclVfelFZYjA

Soal flare on yang dibahas pada kesempatan kali ini,kurang lebih sama dengan source code yang saya berikan.

Untuk pemula, sebelum melakukan reversing, ada baiknya menggunakan Tools PEiD terlebih dahulu untuk file ini berjenis apa.

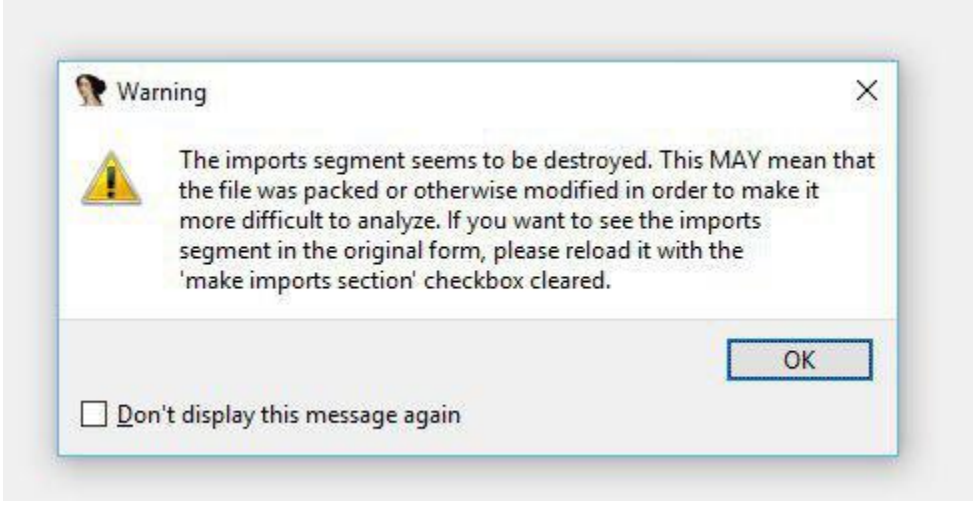


Gambar 01: contoh aplikasi yang tidak di pack



Gambar 02: contoh gambar yang dipack menggunkan UPX

Mengetahui file tersebut di pack atau tidak, sangatlah penting karena akan memudahkan kita ketika membuka IDA.



Gambar 03: contoh IDA membuka file yang di pack

PEiD digunakan untuk membongkar file yang dipack dengan sederhana, untuk membuka file yang dipack dengan metode tertentu silahkan dicari pack tersebut(terkadang hasur manual).

Pengetahuan dasar

Ransomeware adalah malware yang melakukan enkripsi terhadap file dan meminta tebusan. Dalam aksinya, terkadang memanfaatkan exploit untuk melakukan penyebaran. Baik dilakukan melalui jaringan atau email.

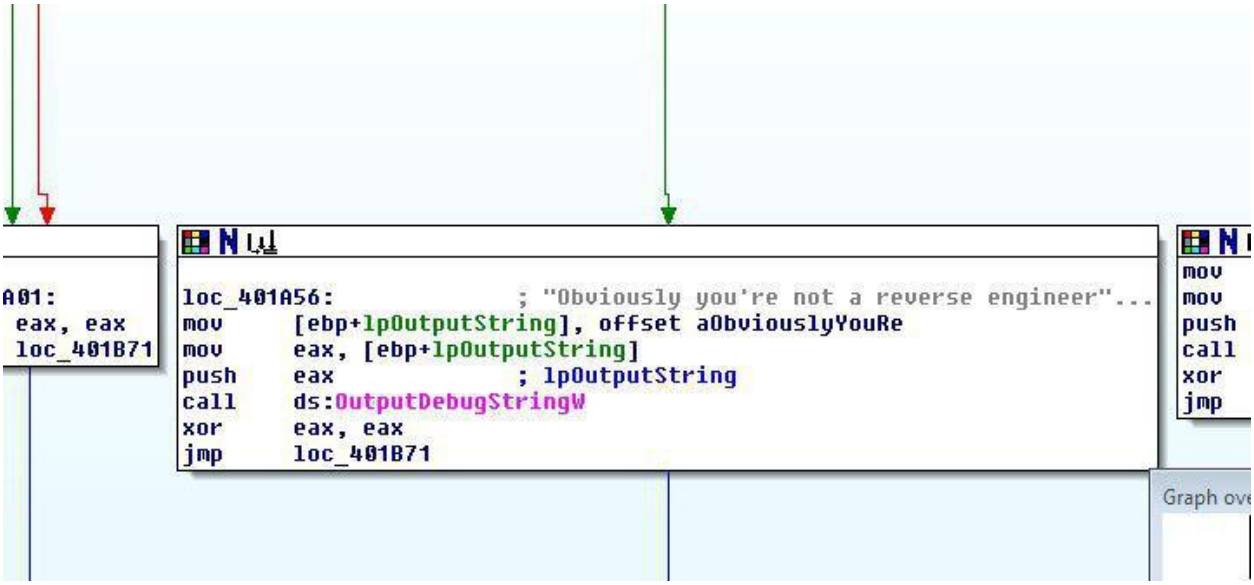
Sekarang, akan dibahas sedikit mengenaik file contoh2.c.
Program ini sangat sederhana:

- Mencari file *.doc yang ada di dekstop
- Membuat file *.encr sebagai enkripsinya

Sengaja tidak dihapus file aslinya dengan tujuan apabila file tersebut dijalankan, cukup hapus saja file *.encr (karena metode belajar)

Challange flare on no 2 ini adalah DudeLocker, tugas kita adalah mengembalikan file BussinessPapers.doc yang sudah ter-enkripsi oleh malware ini.

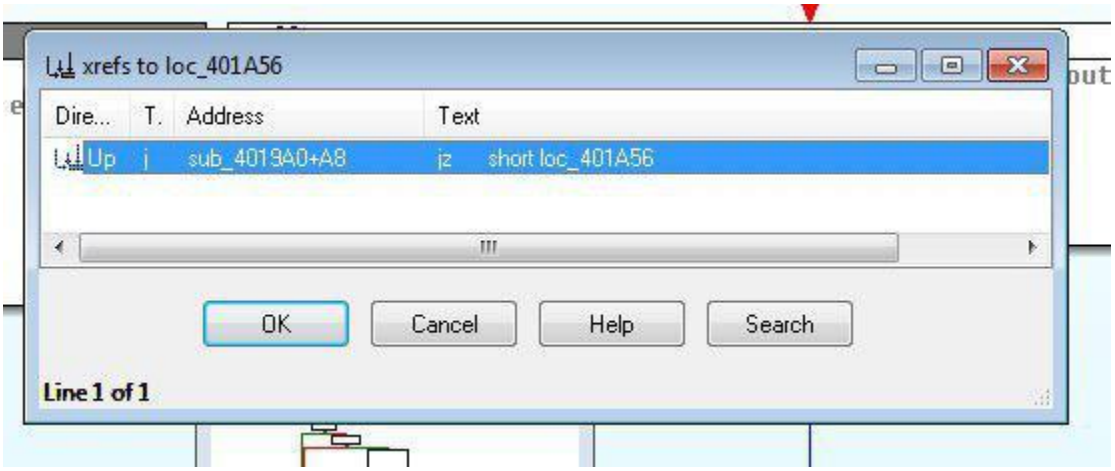
Ada baiknya menggunakan VM (Virtual Machine) untuk melakukan analisa malware dan gunakan fasilitas snapshoot



Gambar 04: hasil debug

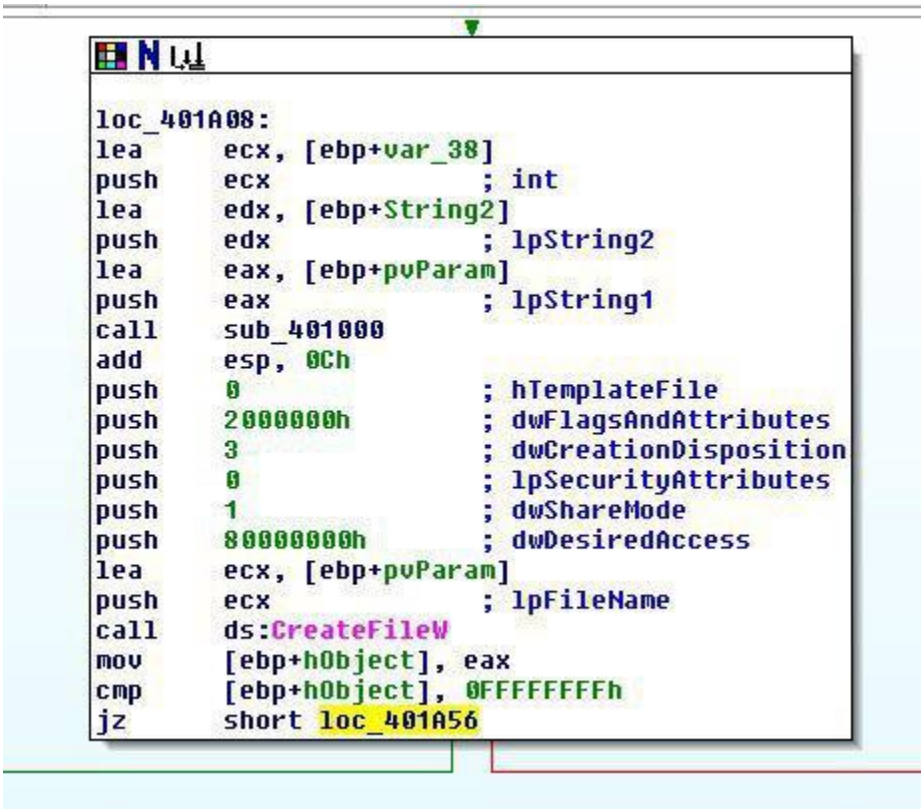
Jika dijalankan dengan debugger, kita bisa melihat string ini. Di IDA kita bisa melihat dari mana asal panah hijau tersebut.

Kita bisa menekan tombol (x) untuk melakukan *cross reference* untuk melihat asalnya.



Gambar 05: *cross reference*

Ternyata hasilnya dari sini :



Gambar 06: hasil *cross reference*

Penjelasan :
Jika file tertentu tidak berhasil dibuka / diciptakan, maka akan dianggap bukan reverse “Obviously you're not a reverse engineer..."

bagaimana kita bisa tahu file apa yang berusaha dibuka dengan CreateFileW? ada beberapa cara:

1. Kita baca assemblynya (atau hasil dekompilasinya jika ada)
2. Kita jalankan program tersebut dengan debugger, dan lihat nilainya apa sebelum pemanggilaan CreateFile
3. Kita memakai program process monitor (https://technet.microsoft.com/en-us/sysinternals/processmonitor.aspx)

dalam kasus ini saya pakai saja Process Monitor

Dengan memfilter Process Name (DudeLocker.exe) dan Operation (CreateFile):

Time ...	Process Name	PID	Operation	Path	Result	Detail
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows\Prefetch\DUDELOCKER.EXE-F5310042.pf	NAME NOT FOUND	Desired Access: G...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows	SUCCESS	Desired Access: E...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows\System32\wow64.dll	SUCCESS	Desired Access: R...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows\System32\wow64.dll	SUCCESS	Desired Access: R...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows\System32\wow64win.dll	SUCCESS	Desired Access: R...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows\System32\wow64win.dll	SUCCESS	Desired Access: R...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows\System32\wow64cpu.dll	SUCCESS	Desired Access: R...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows\System32\wow64cpu.dll	SUCCESS	Desired Access: R...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows\System32\wow64log.dll	NAME NOT FOUND	Desired Access: R...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows	SUCCESS	Desired Access: R...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Users\yohanes\Desktop\2	SUCCESS	Desired Access: E...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows\SysWOW64\sechost.dll	SUCCESS	Desired Access: R...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows\SysWOW64\sechost.dll	SUCCESS	Desired Access: R...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows\SysWOW64\imm32.dll	SUCCESS	Desired Access: R...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows\SysWOW64\imm32.dll	SUCCESS	Desired Access: R...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows\SysWOW64\imm32.dll	SUCCESS	Desired Access: R...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows\SysWOW64\imm32.dll	SUCCESS	Desired Access: R...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows\SysWOW64\imm32.dll	SUCCESS	Desired Access: R...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows\SysWOW64\imm32.dll	SUCCESS	Desired Access: R...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Windows\Globalization\Sorting\SortDefault.nls	SUCCESS	Desired Access: G...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Users\yohanes\Desktop	SUCCESS	Desired Access: R...
5:13:3...	DudeLocker.exe	3112	CreateFile	C:\Users\yohanes\Desktop\Briefcase	NAME NOT FOUND	Desired Access: G...

Gambar 07: Procmon

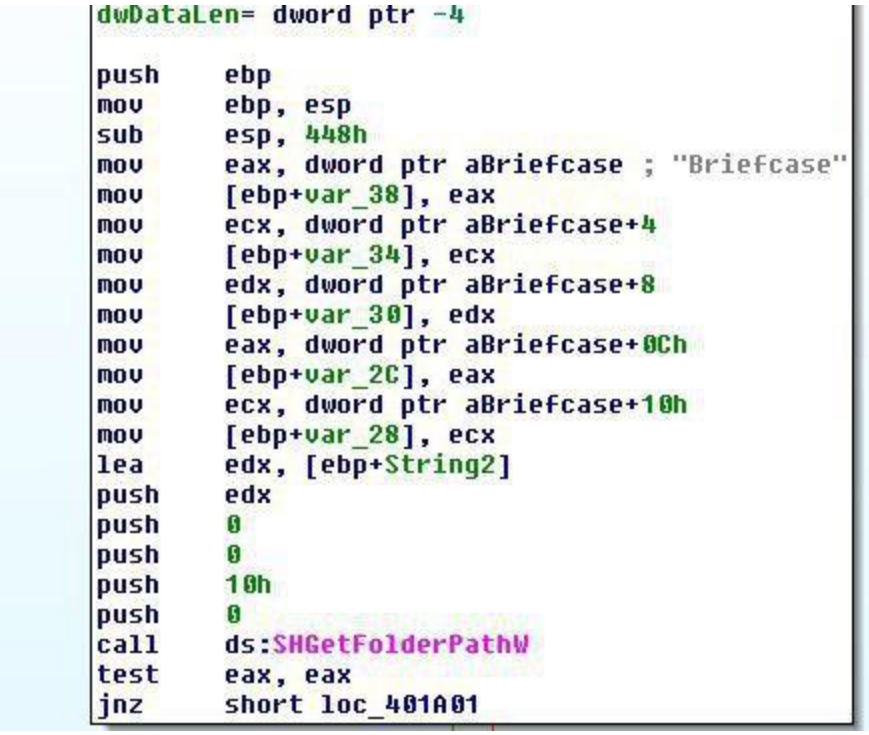
Ternyata program tersebut mencari folder yang bernama Briefcase yang ada di desktop. Jika kita melakukan disassemble lagi kita bisa melihat string yang menuju pada Briefcase

```
.data:00403027                  db    0
.data:00403028 aBriefcase:                              ; DATA XREF: sub_4019A0+9↑r
.data:00403028                                          ; sub_4019A0+11↑r ...
.data:00403028                  unicode 0, <Briefcase>,0
```

Gambar 08: hasil disassemble

Dan lebih tepatnya digabungkan ke Desktop

Copyright ReversingID 2017

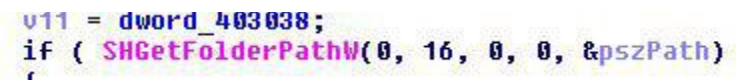


Gambar 08: infromasi SHGetFolderPathW

Jika kita tidak mengerti / mengetahui tentang fungsi yang ada, kita bisa mencarinya di dokumentasi microsoft

https://msdn.microsoft.com/en-us/library/windows/desktop/bb762181(v=vs.85).aspx

Di parameter kedua, nilai yang diberikan adalah 10h atau 16 desimal
darimana kita tahu bahwa ini adalah Folder Desktop? kita harus memetakan kembali ke nilai CSIDL
sayangnya di sini: https://msdn.microsoft.com/en-us/library/windows/desktop/bb762494(v=vs.85).aspx tidak ditemukan, jadi harus scroll ke bawah untuk mencari tahu header file apa yang dipakai, dalam kasus ini Shlobj.h
Sebelum masuk ke file itu: jangan dipikir dengan decompiler ini akan lebih mudah, karena hasil decompilernya:



Gambar 09: hasil decompiler

Jika kita sudah menginstall SDK WIndows, maka file shlobj.h itu bisa dicari, jika belum, kita bisa cari di internet, contohnya saya temukan di :
https://gist.github.com/dingetje/820689

```
#define CSIDL_MYMUSIC   13
#define CSIDL_MYVIDEO   14
#define CSIDL_DESKTOPDIRECTORY  16
#define CSIDL_DRIVES    17
#define CSIDL_NETWORK   18
#define CSIDL_NETHOOD   19
#define CSIDL_FONTS     20
```

Gambar 10: Desktop Directory Number

CSIDL_DESKTOPDIRECTORY nilainya 16
berarti fungsi ShGetFolderPathW tadi dipakai untuk mencari direktori Desktop
ini contoh yang sangat kecil, dan biasanya kita tidak perlu mencari tahu fungsi yang jelas bisa dilihat nilai kembaliannya di debugger
tapi teknik mencari konstanta tadi sering kali perlu dipakai untuk mengetahui parameter fungsi-fungsi enkripsi
Misalnya seperti ini:

```
if (!CryptCreateHash(hProvider, CALG_SHA_256, 0, 0, &hHash))
cbData = lstrlen(password) * sizeof(TCHAR);
```

Gambar 11: contoh penggunaan

Saya tidak akan membahas lebih dalam lagi mengenai DudeLocker ini, tapi beberapa hal bisa saya beritahukan:

- coba masukkan file .doc ke dalam Desktop/Briefcase (buat dulu direktori Briefcase)
- coba dekrip file .doc yang ada

setidaknya ada dua strategi untuk mendekrip filenya dalam kasus ini:

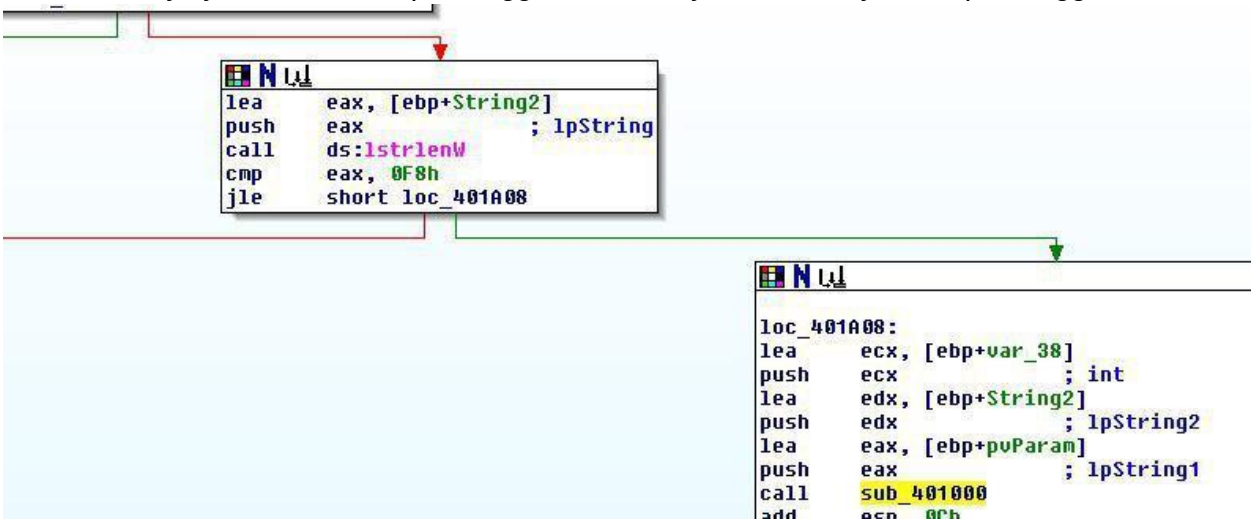
- memahami proses enkripsi (simetrik), dan menulis kode program untuk dekrip filenya
- mempatch executablenya supaya bisa mendekrip file yang ada

Jika masih bingung dengan metode enkripsi menggunakan CryptoAPI, silakan kunjungi link di source code contoh2.c
http://etutorials.org/Programming/secure+programming/Chapter+5.+Symmetric+Encryption/5.25+Using+Symmetric+Encryption+with+Microsoft+s+CryptoAPI/
(ada bug kecil di kodenya, untuk koreksinya silakan baca komentar terakhir di situs itu)

Pertanyaan

1.
Q: Di program itu udah ada fungsi dekripnya kah mas?
A: nggak ada, tapi kalau sudah tahu cara enkripnya, bisa diakali

2.
Q: saya mau tanya,gmn cara baca dari decompiler tersebut bahwa file yang di cari ada briefcase pada desktop korban ? sedangkan ini hanya tertulis briefcase
A: Sebenarnya jika ditrace ada pemanggilan berikutnya, sebentar ya. ada pemanggilan ini:



jika ditelusurin subrutin itu

```
; int __cdecl sub_401000(LPWSTR lpString1,LPCWSTR lpString2,int)
sub_401000 proc near

String2= word ptr -4
lpString1= dword ptr  8
lpString2= dword ptr  0Ch
arg_8= dword ptr  10h

push    ebp
mov     ebp, esp
push    ecx
mov     eax, dword_4030F0
mov     dword ptr [ebp+String2], eax
mov     ecx, [ebp+lpString2]
push    ecx             ; lpString2
mov     edx, [ebp+lpString1]
push    edx             ; lpString1
call    ds:lstrcpyW
lea     eax, [ebp+String2]
push    eax             ; lpString2
mov     ecx, [ebp+lpString1]
push    ecx             ; lpString1
call    ds:lstrcatW
mov     edx, [ebp+arg_8]
push    edx             ; lpString2
mov     eax, [ebp+lpString1]
push    eax             ; lpString1
call    ds:lstrcatW
mov     esp, ebp
pop     ebp
retn
sub_401000 endp
```

ada pemanggilan lstrcpy dan lstrcat
https://msdn.microsoft.com/en-us/library/windows/desktop/ms647490(v=vs.85).aspx
https://msdn.microsoft.com/en-us/library/windows/desktop/ms647487(v=vs.85).aspx

Sebagai catatan, hasil dekompilasinya seperti ini dengan IDA Pro

```
1 LPWSTR __cdecl sub_401000(LPWSTR lpString1, LPCWSTR lpString2, LPCWSTR a3)
2 {
3   WCHAR String2[2]; // [sp+0h] [bp-4h]@1
4
5   *(_DWORD *)String2 = dword_4030F0;
6   lstrcpyW(lpString1, lpString2);
7   lstrcatW(lpString1, String2);
8   return lstrcatW(lpString1, a3);
9 }
```

tidak beda jauh dari assemblynya, kita tetap perlu paham fungsi yang dipanggil
untuk yang sudah belajar C dan tahu fungsi strcat, strcpy tentunya sudah paham

Ini saya pakaikan decompiler IDA Pro untuk bagian enkripsinya:

Terima Kasih
ReversingID, Yohanes Nugroho, Yang ngerangkum tulisan ini diawal dan semua.