

TUGAS
KEAMANAN JARINGAN KOMPUTER



Nama : Dede Triseptiawan
Nim : 09011181320001

SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
2017

Analisa malware

Analisa malware adalah suatu aktivitas yang kerap dilakukan oleh sejumlah praktisi keamanan teknologi informasi untuk mendeteksi ada atau tidaknya komponen sub program atau data yang bertujuan jahat dalam sebuah file elektronik.

Ada dasarnya malware adalah sebuah program, yang disusun berdasarkan tujuan tertentu dengan menggunakan logika dan algoritma yang relevan dengannya. Oleh karena itulah maka model analisa yang biasa dipergunakan untuk mengkaji malware sangat erat kaitannya dengan ilmu dasar komputer, yaitu: bahasa pemrograman, algoritma, struktur data, dan rekayasa piranti lunak.

Pada saat ini, sudah tersedia berbagai macam software yang dapat digunakan untuk melindungi komputer / jaringan komputer dari serangan malware, diantaranya : antivirus, firewall, ids, internet protection dan lain-lain. Namun kecanggihan dari software tersebut, umumnya dapat dilewati menggunakan teknik-teknik tertentu sehingga software tersebut tidak dapat mendeteksi adanya aktivitas malicious program yang sedang berjalan.

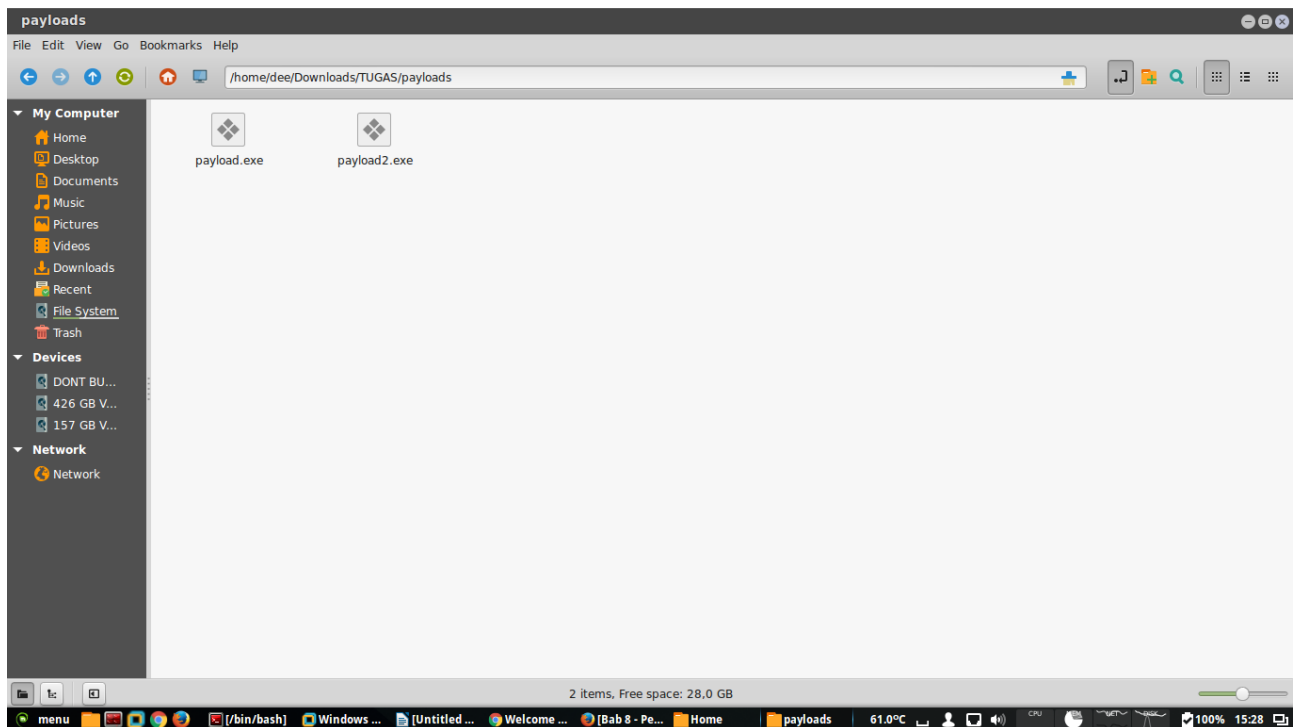
Ada dua metode untuk melakukan analisa terhadap malware, yaitu :

1. **Dynamic Analysis** : Merupakan metode yang digunakan untuk melakukan analisa terhadap malware dengan mengamati kinerja sistem yang dapat terlihat dari perilaku sistem sebelum malware dijalankan dengan perilaku sistem setelah malware tersebut dijalankan pada sistem tersebut. Metode dynamic analysis umumnya menggunakan software virtual seperti VirtualBox, VMWare dan lain-lain, sehingga apabila malware yang dijalankan tersebut ternyata merusak sistem, maka sistem utama tidak mengalami kerusakan akibat malware tersebut.
2. **Static Analysis** : Merupakan metode yang digunakan untuk melakukan analisa malware dengan cara mengamati secara langsung kode sumber (source code) malware tersebut. Dalam mengamati kode sumber malware, terdapat teknik yang umumnya digunakan, yaitu Reverse Engineering.

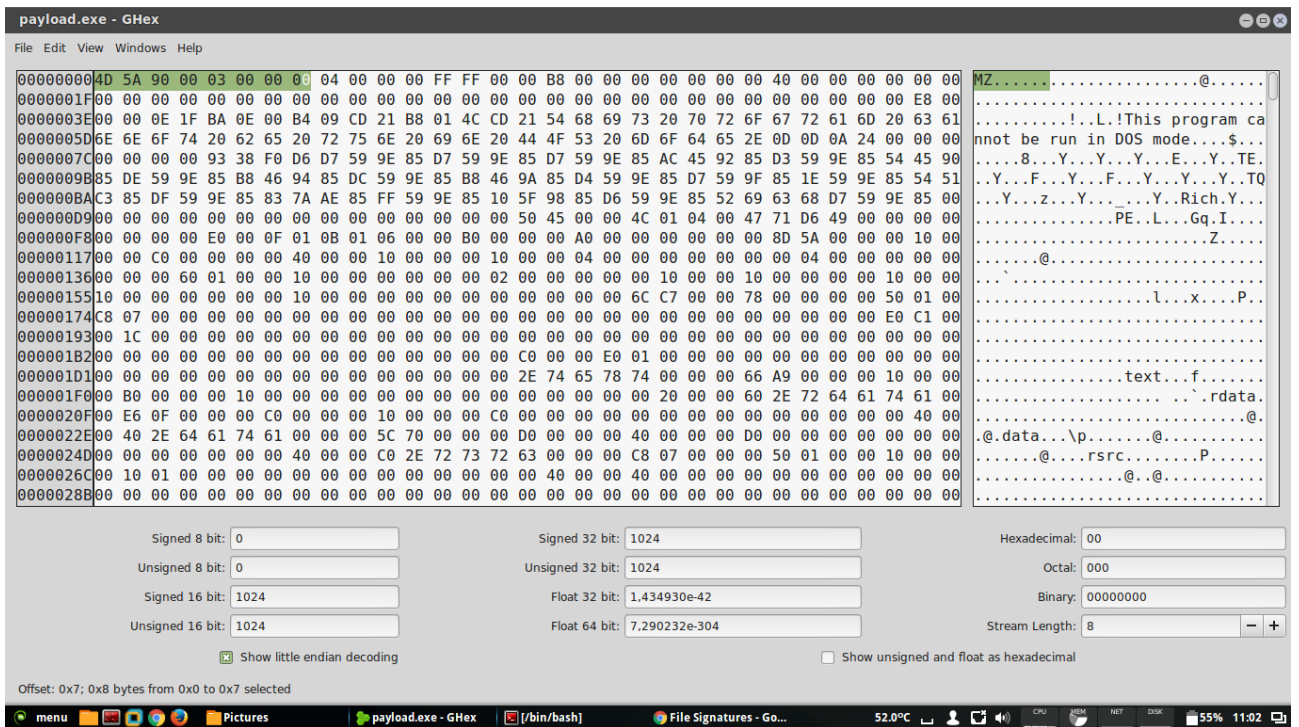
Sebagai uji coba, digunakan bahan yang telah diberikan berupa payload.exe, dan payload2.exe . Dan tools yang digunakan untuk menganalisa bahan yang diberikan berupa ghex, hexdump, strings, ollydbg, dan ida pro.Ghex berguna untuk debugging masalah dengan

kode, dan untuk memuat data dari file, melihat dan mengedit hex dan ascii. Hexdump fungsinya sama dengan ghex untuk melihat kode hex yang ada pada file tersebut, beda dengan ghex tampilan berbentuk gui. Strings berfungsi untuk melihat program/ perintah-perintah dalam sebuah file. Ollydbg berfungsi untuk debugging / debug dan melakukan dumping, serta melakukan pengintaian proses apa saja yang terjadi pada sebuah program. Ida pro adalah piranti bantuan untuk melakukan debug dengan melihat kode program dalam bentuk assembler.

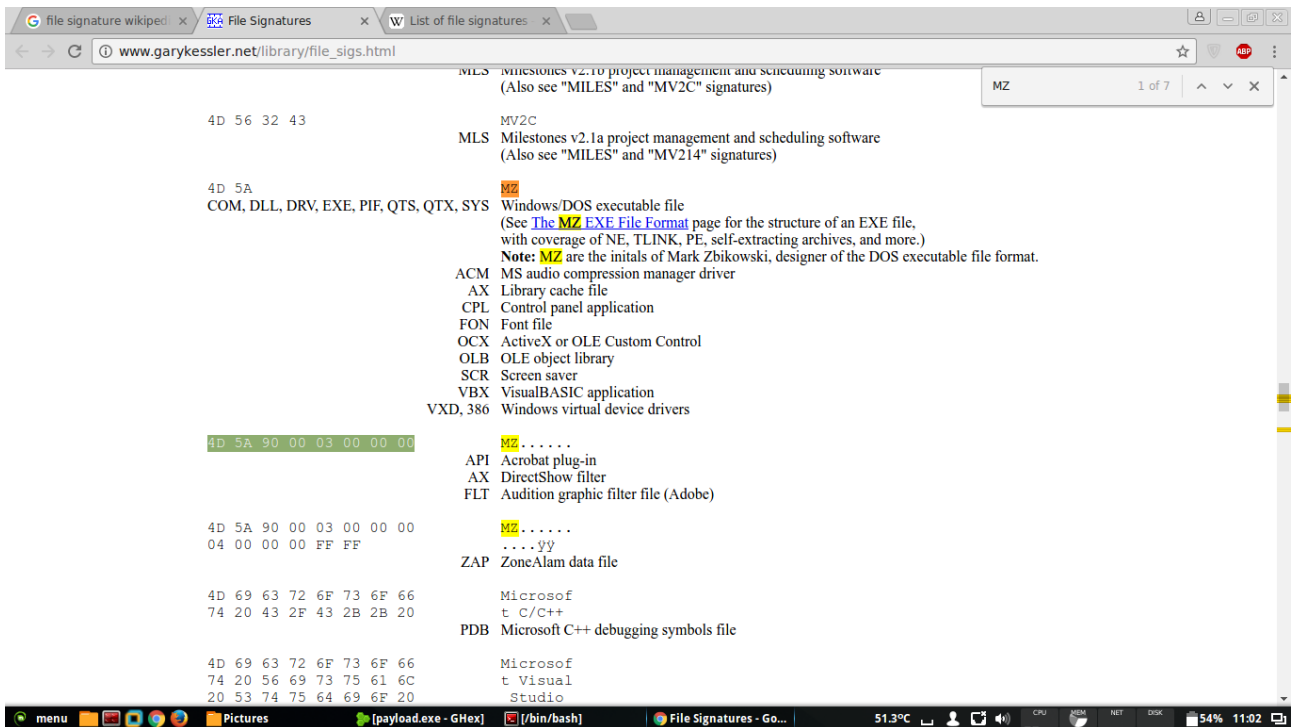
Pertama yang dilakukan membuka file yang diberikan



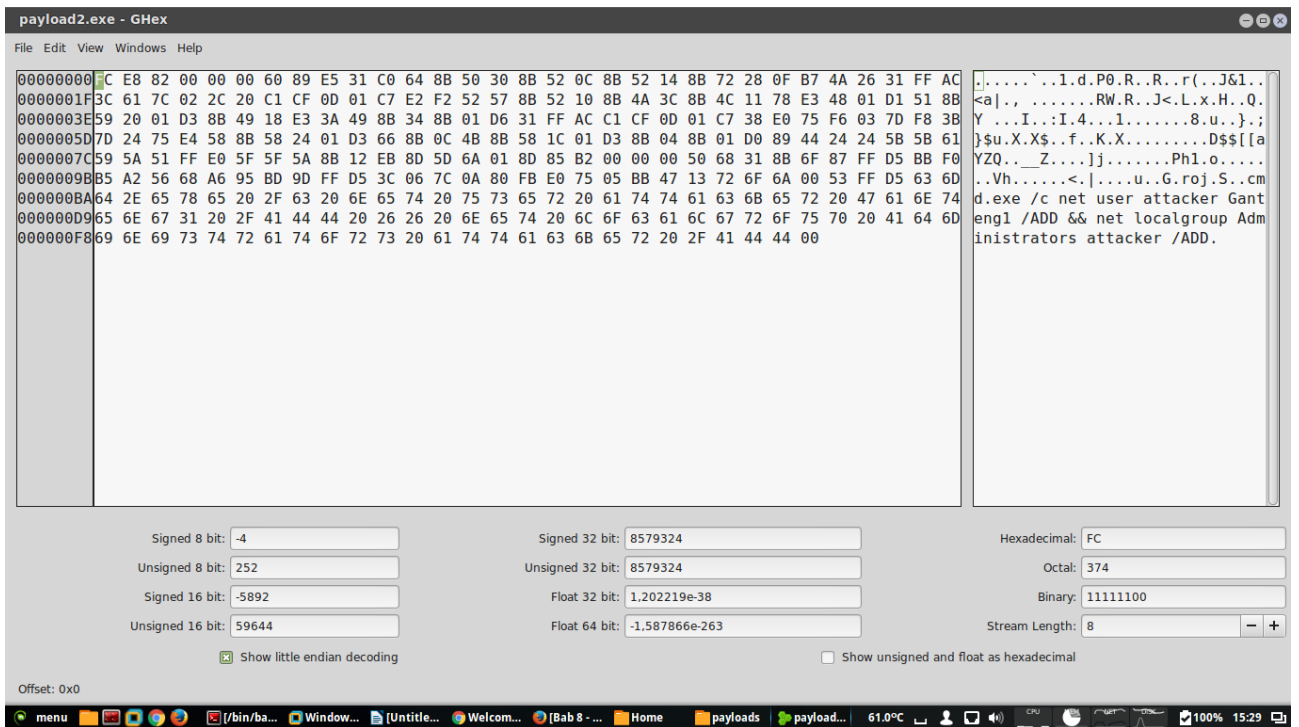
Terdapat 2 file pada gambar di atas berupa payload.exe dan payload2.exe. kemudian kita coba terlebih dahulu payload.exe. dengan cara klik kanan pada file tersebut dan open with dengan ghex.



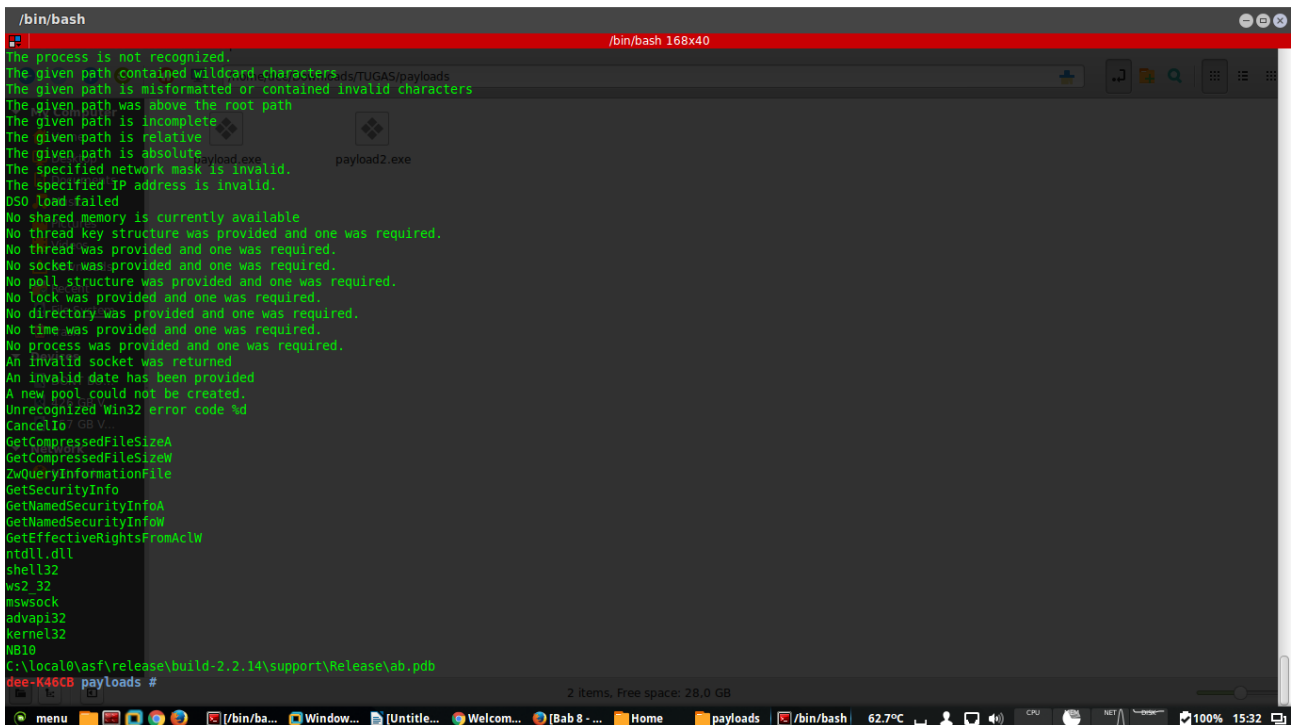
Berikut tampilan pada ghex, terdapat kode yang diblok berupa 4D 5A 90 00 03 00 00 00 dan kode sebelumnya MZ..... kemudian kita cari di list of file signature



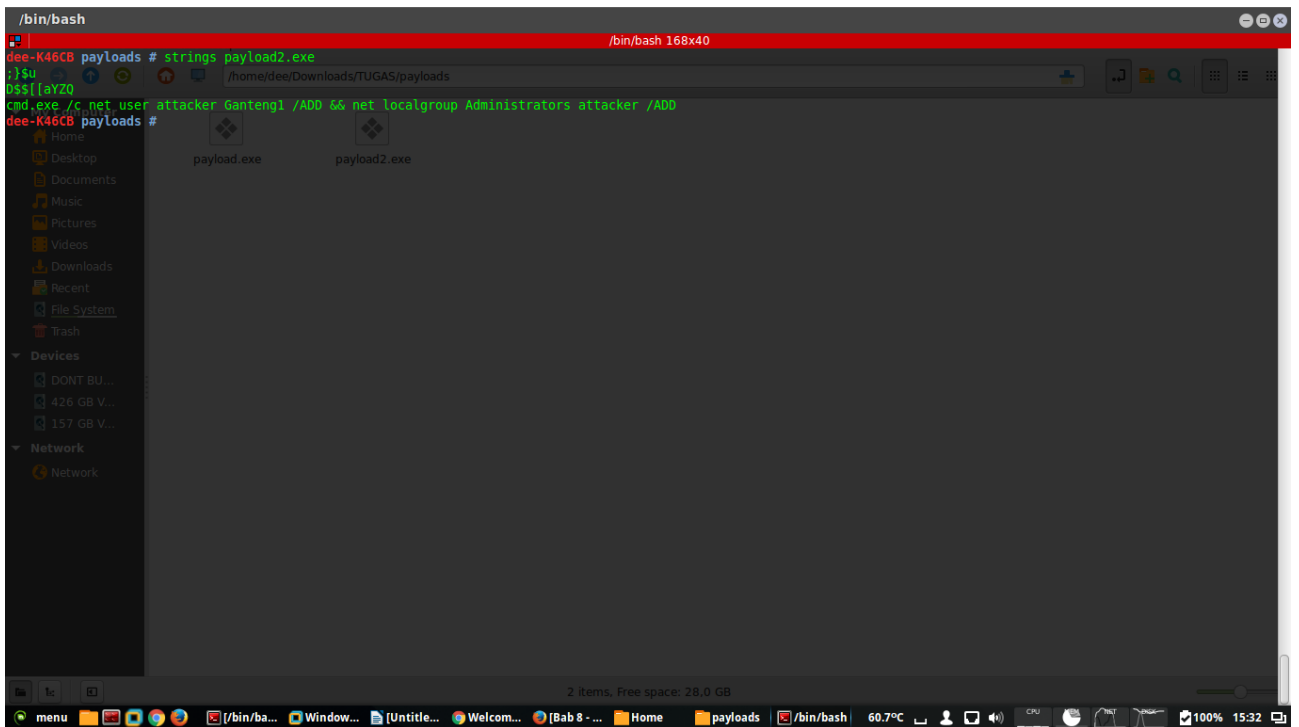
File MZ..... merupakan jenis file yang hanya tersedia untuk windows dan berjenis aplikasi API(acrobat plug-in), AX(directshow filter) dan FLT(audition graphic filter file(adobe)). Kemudian kita coba file payload2.exe



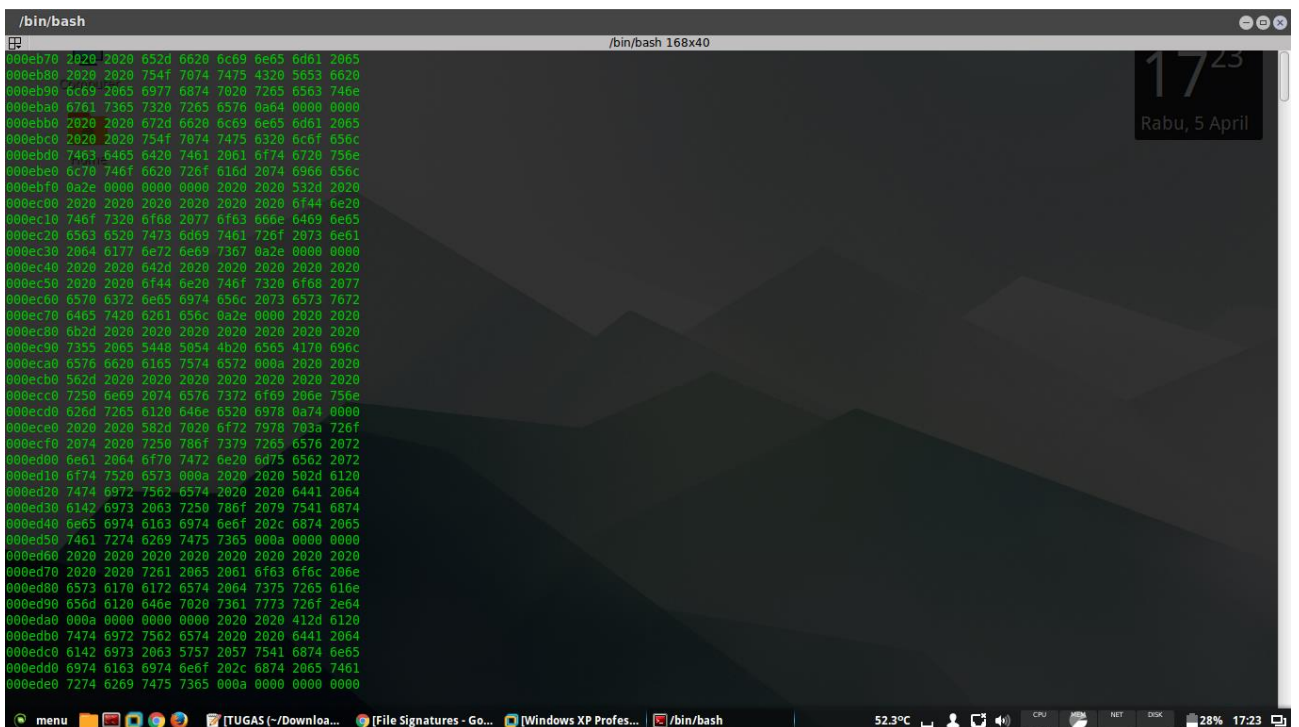
Berikut pada gambar diatas hasil kode hex pada file payload2.exe. kemudian kita lakukan dengan perintah di terminal strings payload.exe



Maka akan keluar seperti gambar diatas, kemudian kita lakukan ke file selanjutnya



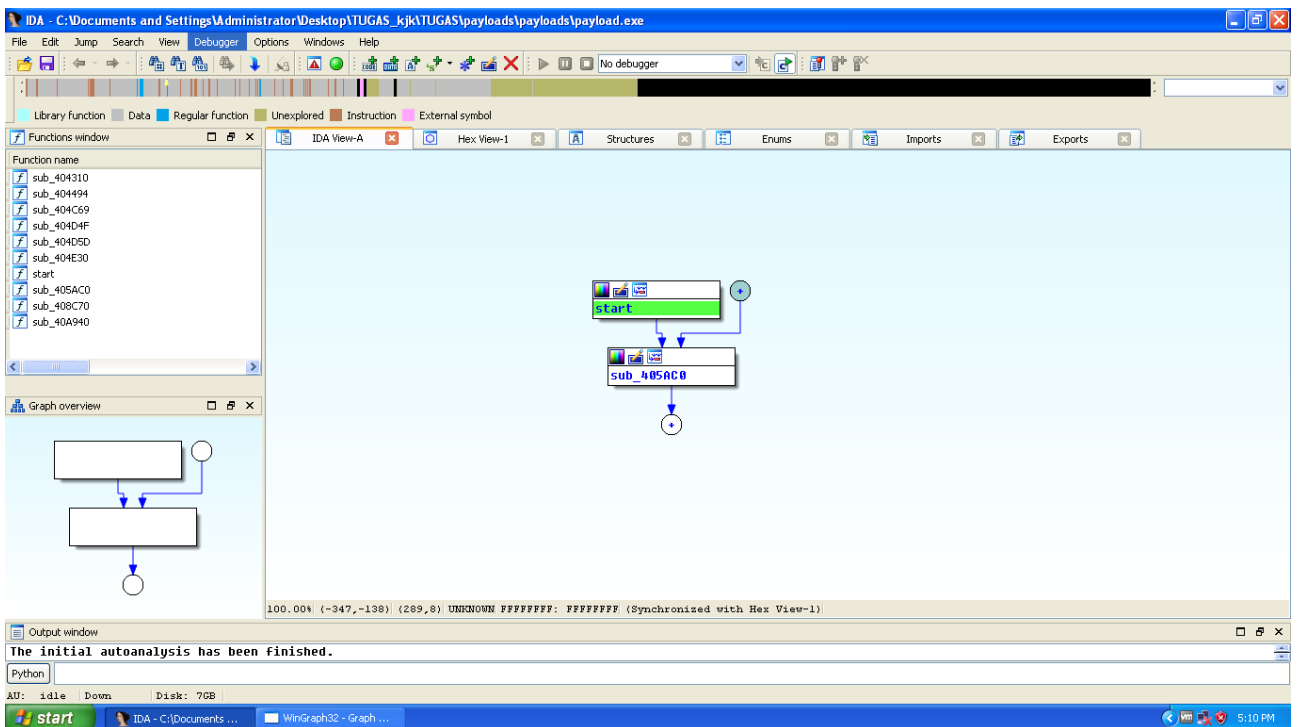
Maka akan tampil seperti gambar diatas. Kemudian kita lakukan perintah pada terminal dengan perintah hexdump payload.exe

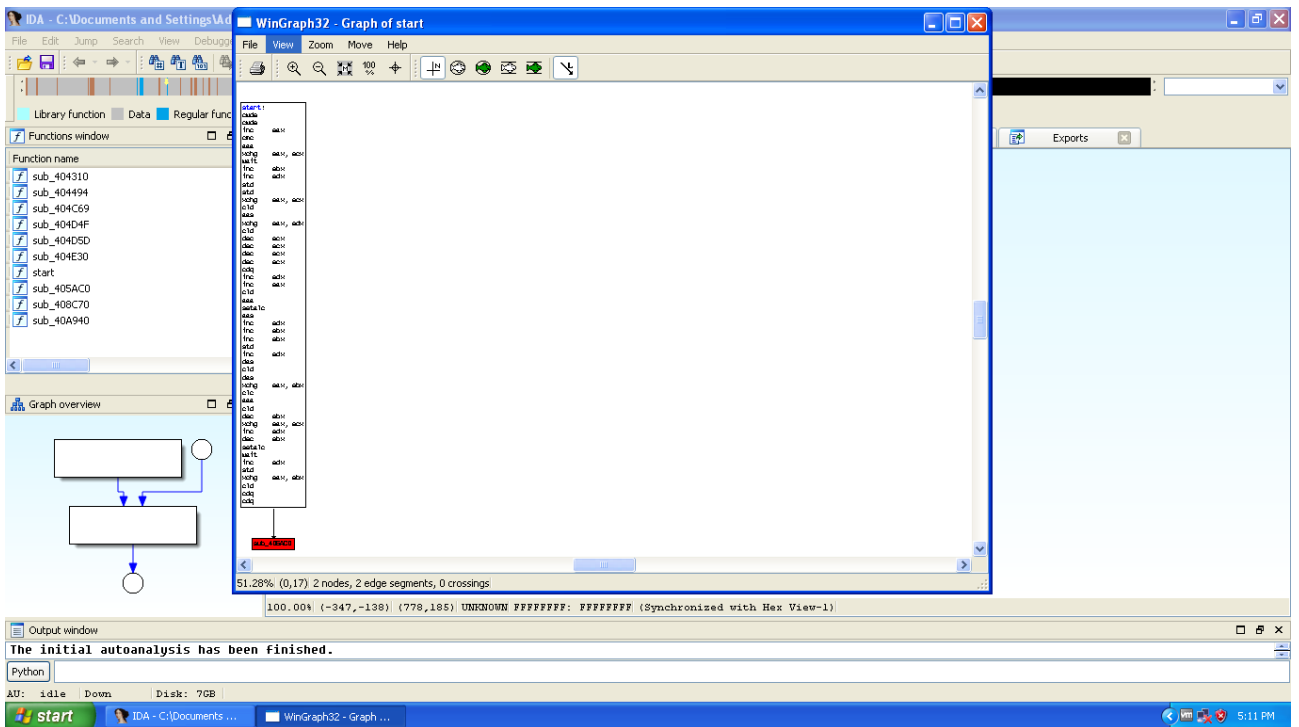


Maka akan keluar seperti gambar diatas, kemudian kita lakukan ke file satunya.

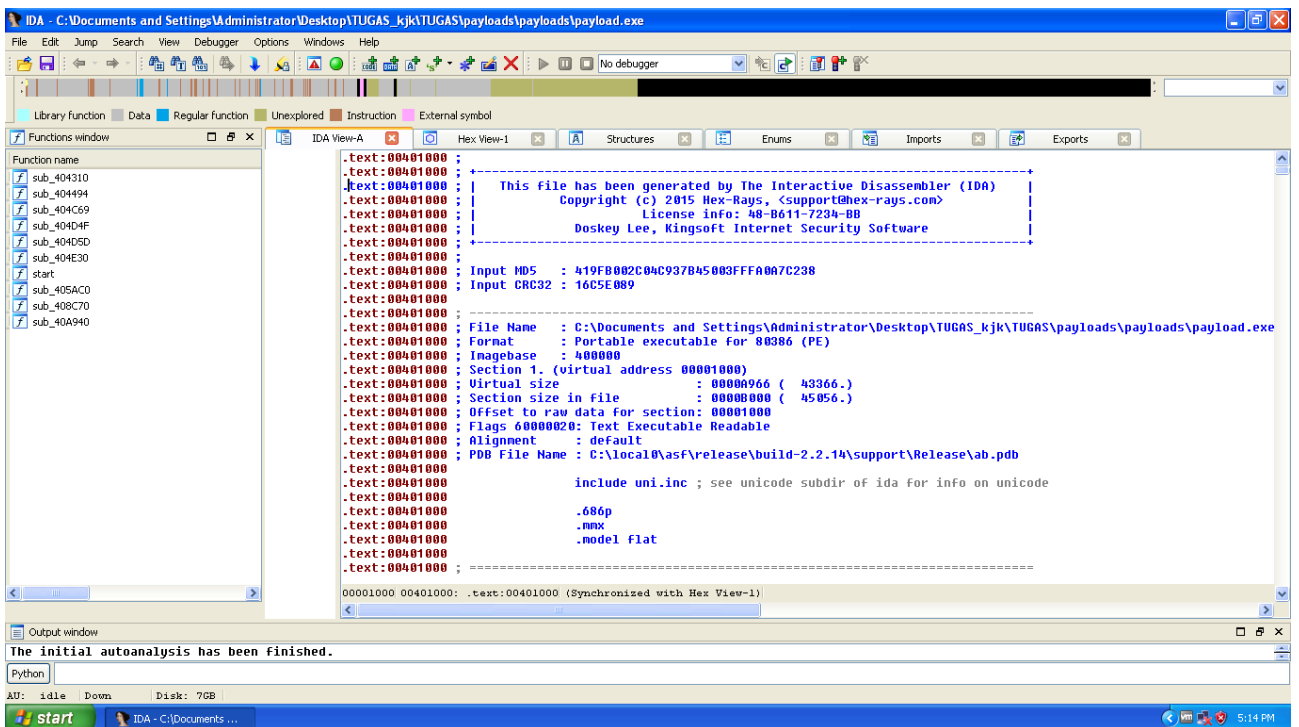
```
/bin/bash
deedee-K46CB ~/Downloads/TUGAS/payloads $ hexdump payload2.exe
00000000 88fc 0082 0080 8960 31e5 64c0 568b 8b30
00000010 0c52 528b 8b14 2872 b70f 264a ff31 3cac
00000020 7461 2c02 c120 0dcf c701 72e2 5752 528b
00000030 8b10 3c4a 4c8b 7811 48e3 d181 8b51 2059
00000040 d301 498b e318 493a 349b 0180 31de acff
00000050 c7e1 010d 38e7 75e0 03f6 787d 7d3b 7524
00000060 58e4 588b 0124 66d3 0c8b 8b4b 1c58 d301
00000070 048b 018b 8980 2444 5024 015b 5a59 ff51
00000080 57e0 5a5f 128b 04eb 6a5d 0401 b285 0000
00000090 5080 3160 6f8b ff87 bbd5 b5f0 56a2 a668
000000a0 bd93 ff9d 3cd5 7c06 800a e9fb 0975 470b
000000b0 7213 686f 5380 05ff 6d63 2864 7865 2065
000000c0 632f 6e20 7465 7520 6373 2072 7401 6174
000000d0 6863 7265 4720 6a61 6374 676e 2831 412f
000000e0 4444 2620 2026 656e 2874 6f6e 6163 676c
000000f0 6f72 7075 4120 6d64 6e69 7369 7274 7461
00001000 726f 2073 7461 6174 6b63 7265 2f20 4441
00001100 0044
00001120
deedee-K46CB ~/Downloads/TUGAS/payloads $
```

Maka akan keluar file hex seperti diatas. Kemudian kita buka tools virtual machine pada hal ini menggunakan vmware dan mengguakan windows xp sebagai sistem operasi untuk menjalankan tools ollydbg dan idapro. Dan berikut hasil untuk payload.exe

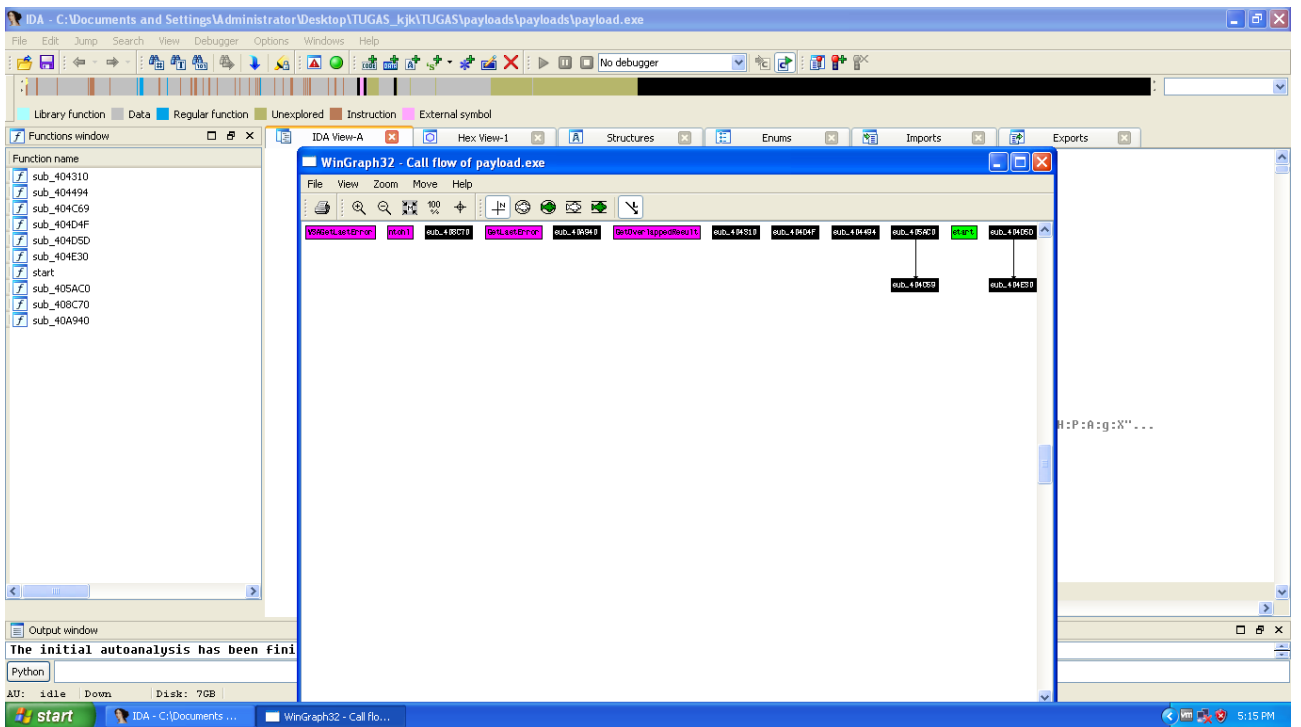




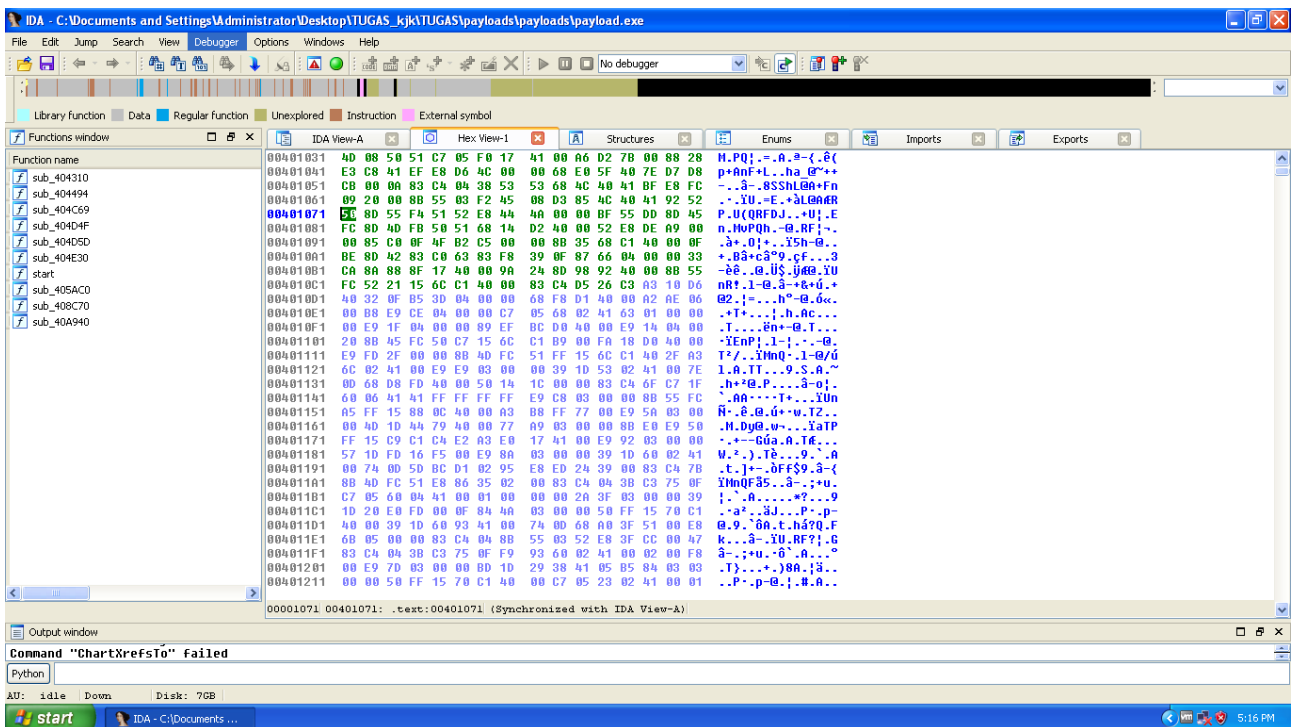
Pada gambar diatas merupakan alur (flowchart) graph yang tersedia pada aplikasi ida pro



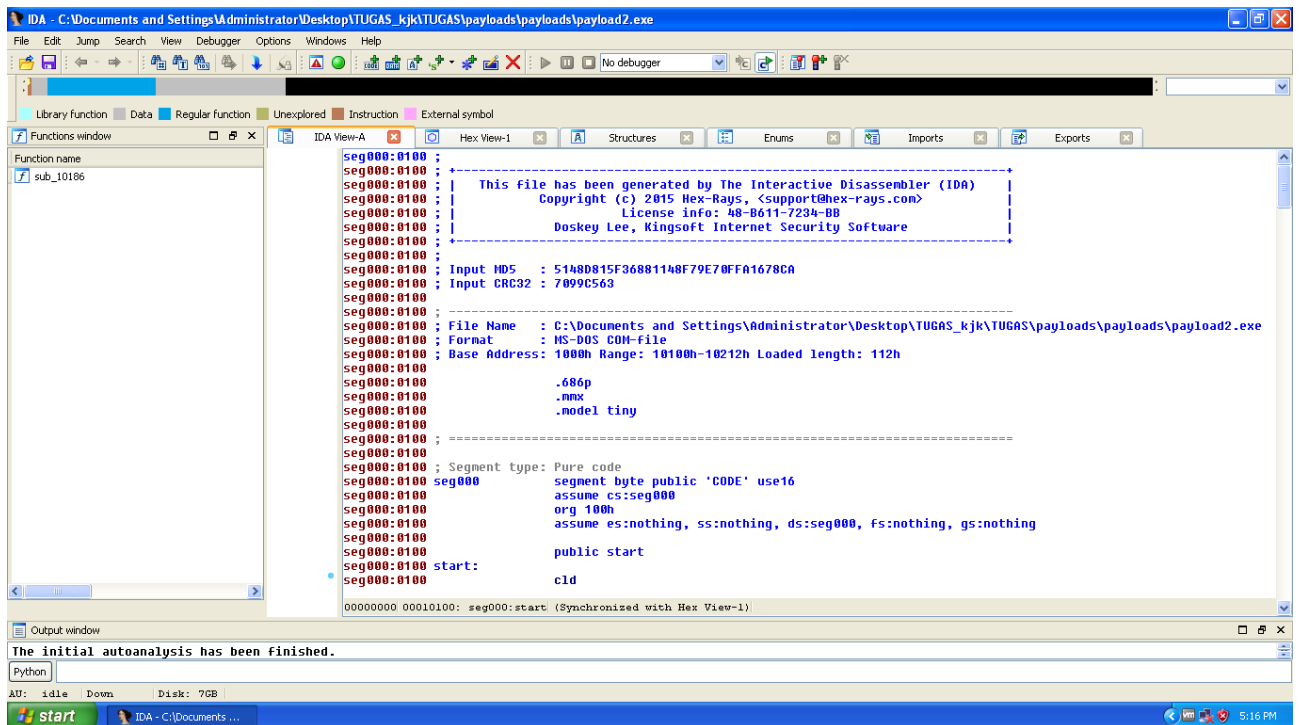
Pada gambar diatas merupakan hasil convert file tersebut dalam assembly



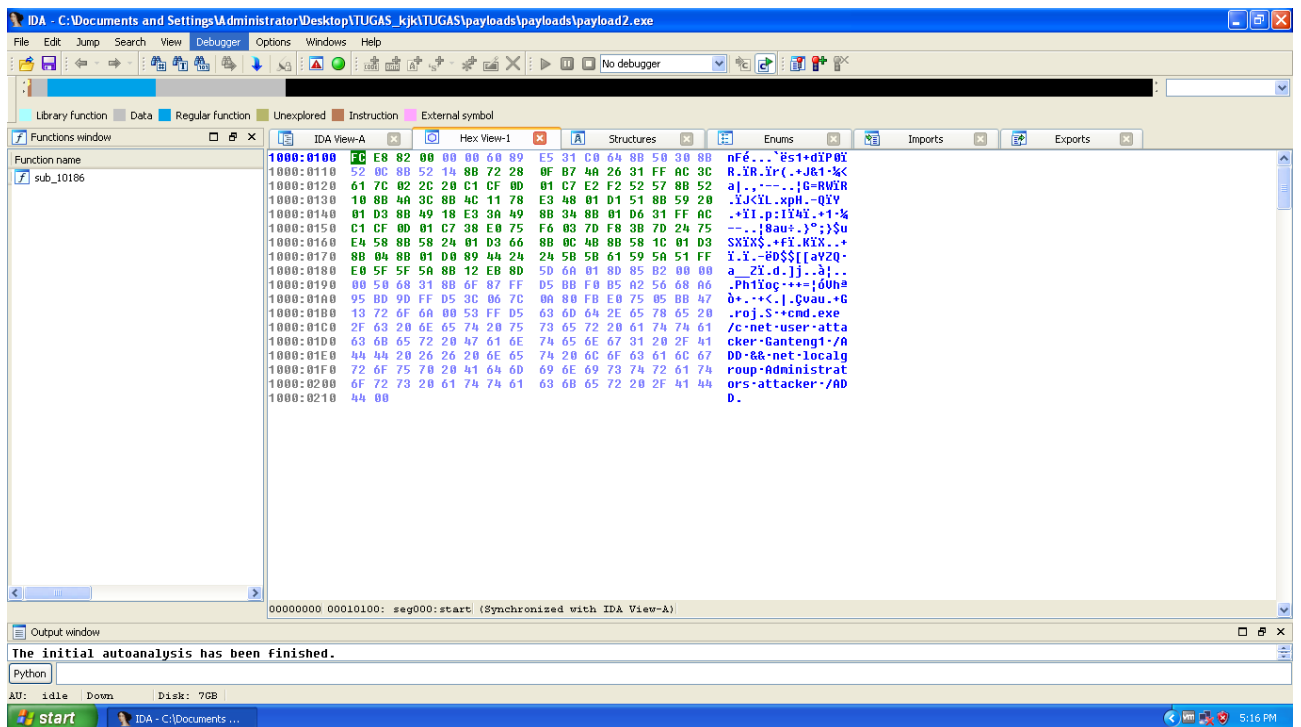
Berikut pada gambar diatas hasil graph alur pada payload.exe



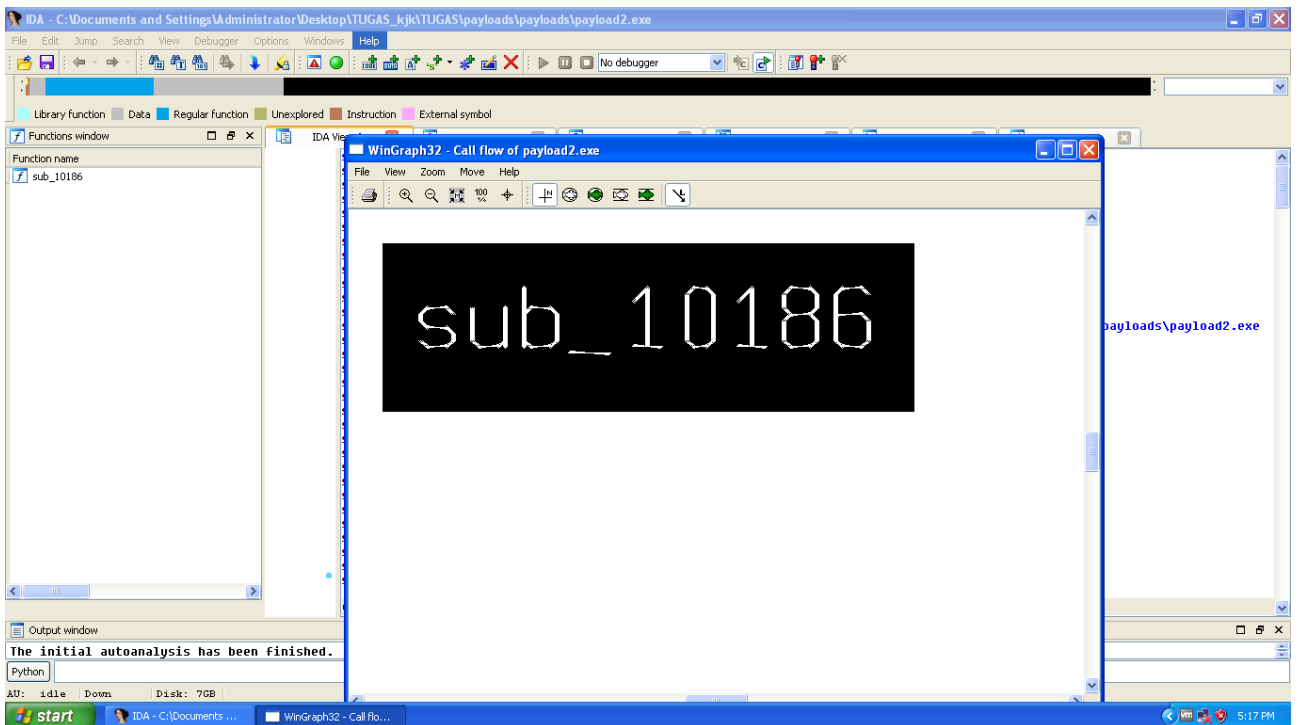
Pada gambar diatas merupakan hasil hex pada aplikasi ida pro. Kemudian kita lakukan pada file satunya payload2.exe



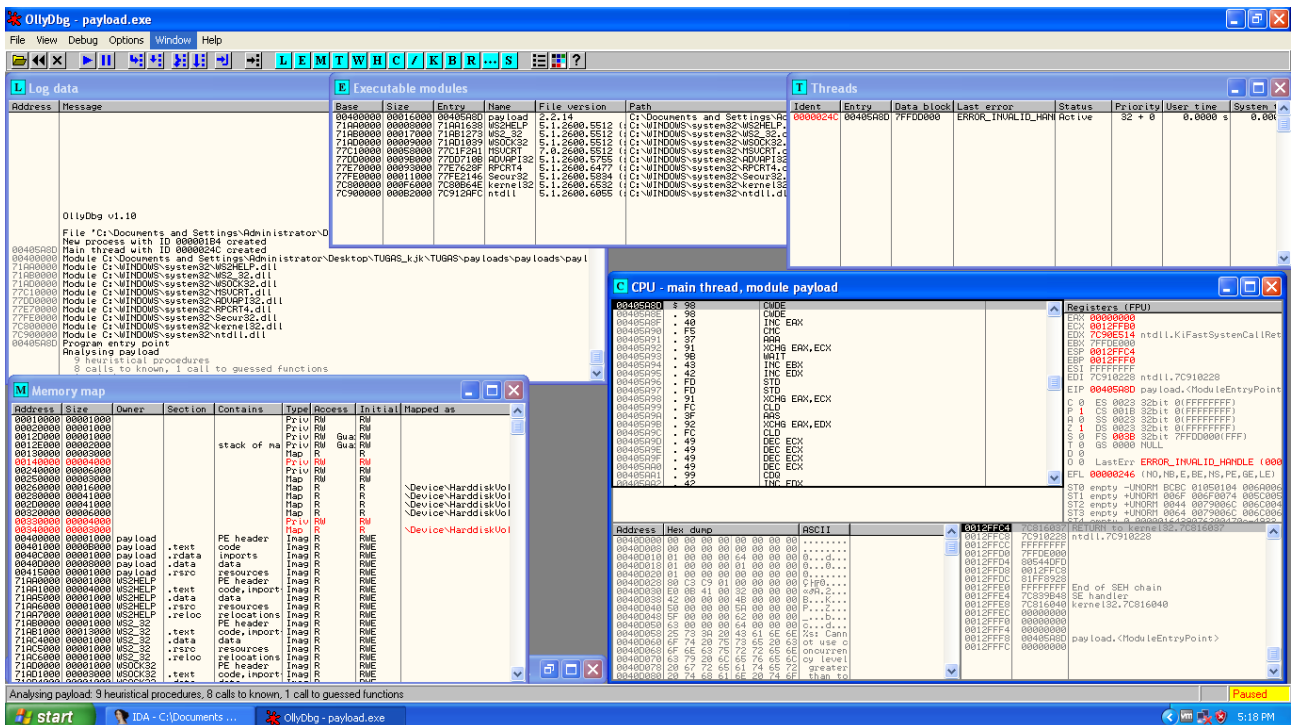
Pada gambar diatas merupakan hasil convert assembly pada file payload2.exe



Pada gambar diatas merupakan hasil hex pada file payload2.exe



Pada gambar diatas merupakan hasil graph pada payload2.exe. kemudian kita gunakan aplikasi ollydbg



Pada gambar diatas merupakan hasil dari file payload.exe

The screenshot displays the OllyDbg interface with the following components:

- Log data:** Shows the program's execution flow, including file operations and module loading.
- Executable modules:** Lists loaded modules such as ntndm.exe, kernel32.dll, and user32.dll.
- Memory map:** Provides a detailed view of memory addresses, sizes, and access permissions for various sections.
- CPU - main thread, module ntndm:** Displays the current assembly instructions, including:
 - 000F449: 68 4816000F PUSH ntndm.0F001648
 - 000F450: EB 73400000 CALL ntndm.0F01B8C8
 - 000F451: 8F 94000000 MOV EDI, 94
 - 000F452: 8BC3 CALL EBX
 - 000F453: E8 6FC00000 CALL ntndm.0F01C000
 - 000F454: 895E MOV EDI, ESP
 - 000F455: 925E MOV DWORD PTR DS:[ESI], EDI
 - 000F456: 8E PUSH EDI
 - 000F457: F15 2C10000F CALL DWORD PTR DS:[!KERNEL32.GetVersion@Version1
 - 000F458: 8B4E MOV EDI, DWORD PTR DS:[ESI+10]
 - 000F459: 43 0B50400F MOV DWORD PTR DS:[F0A59831], EDI
 - 000F45A: 8B4E MOV EDI, DWORD PTR DS:[ESI+4]
 - 000F45B: 9040 MOV DWORD PTR DS:[F0A59831], ECX
 - 000F45C: 8B4E MOV EDI, DWORD PTR DS:[ESI+8]
 - 000F45D: 9116 MOV DWORD PTR DS:[F0A59833], EDX
 - 000F45E: 8B7A MOV EDI, DWORD PTR DS:[ESI+4C]
 - 000F45F: 91E5 MOV EDI, FFFF
 - 000F460: 895E MOV DWORD PTR DS:[F0658BC], EDI
 - 000F461: 5F5E JMP EBZ
- Registers (FPU):** Shows the state of registers, with EIP pointing to 000F449.
- Hex dump:** Shows the memory dump for the instruction at address 000F449.

Pada gambar diatas merupakan hasil dari file payload2.exe