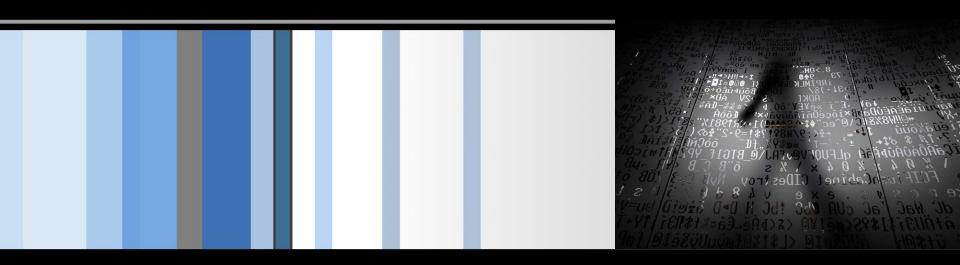
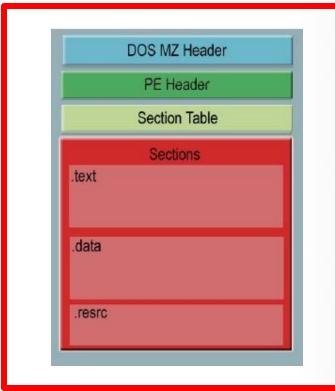
# CSC 471 Modern Malware Analysis PE Structure (2)

Si Chen (schen@wcupa.edu)



#### Portable Executable (PE) file

- A Portable Executable (PE) file is the standard binary file format for an Executable (.exe) or DLL under Windows NT, Windows 95, and Win32.
- Derived from COFF (Common Object File Format) in UNIX platform, and it is not really "portable".





Now here is the kicker. Even though this specification is spelled out by Microsoft, compilers/linkers chose to ignore some parts of it.

To make things even worse, the Microsoft loader doesn't enforce a good portion of this specification and instead makes assumptions if things start getting weird.

So even though the spec outlined here says a particular field is supposed to hold a certain value, the compiler/linker or even a malicious actor could put whatever they want in there and the program will likely still run...

#### Portable Executable (PE) file

- PE formatted files include:
  - .exe, .scr (executable)
  - .dll, .ocx, .cpl, drv (library)
  - .sys, .vxd (driver files)
  - .obj (objective file)





- All PE formatted files can be executed, except obj file.
  - .exe, .scr can be directly executed inside Shell (explorer.exe)
  - others can be executed by other program/service
- PE refers to 32 bit executable file, or PE32. 64 bit executable file is named as PE+ or PE32+. (Note that it is not PE64).

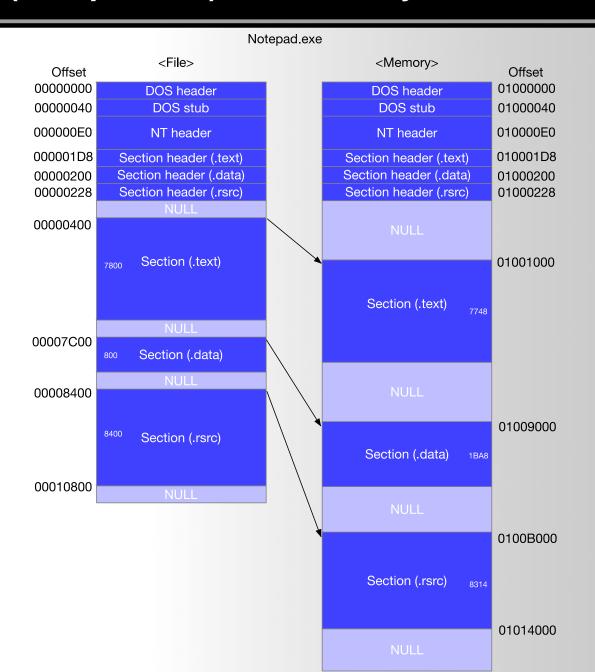


#### PE Example – Notepad.exe

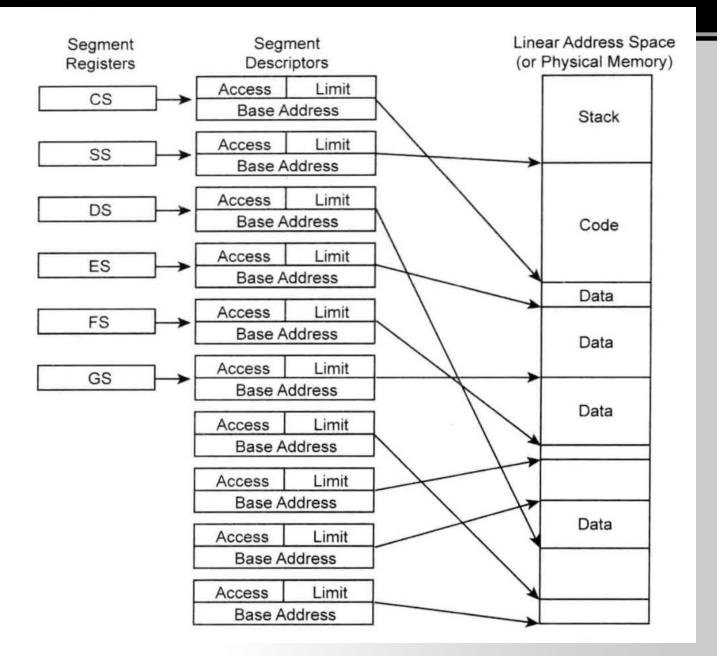
```
M<mark>Z</mark>É.......
0000000
           4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF
00000010
                                   00
                                      40 00
                                                00
                                                   00
                         00
                            00
                               00
                                             00
                                                          00
                                                             00
                                                                   ∃.....@...
00000020
                               00
                                   00
                                      00 00
                     00
                         00
                            00
                                             00
                                                00
                                                   00
                                                          00 00
00000030
                     00
                         00
                            00
                               00
                                   00
                                      00 00 00
                                                00 E8
                                                          00 00
                                                                   . . . . . . . . . . . . <del>.</del> . . .
00000040
                                      21 B8
                                                                      ..-|.=!a.L=!Th
                  BA 0E
                         00
                            В4
                               09
                                   CD
                                             01 4C CD
                                                       21 54 68
00000050
                     70
                            6F
                               67
                                   72
                                      61 6D 20 63 61 6E 6E 6F
                                                                   is.program.canno
                                                                   t.be.run.in.DOS.
00000060
                  62 65
                            72 75
                                  6E 20 69 6E
                                                20 44 4F
           6D 6F 64 65 2E
                                                                   mode....$.....
00000070
                            OD OD OA 24 00 00
                                                00 00
                                                                   Ñm.¢ß.x╚ß.x╚ß.x╚
00000080
           A5 6D 16 9B E1 0C 78 C8 E1 0C 78 C8 E1 0C 78 C8
                                                                   ./8 La.x La.x La.x L
00000090
                  38 C8 E0 0C 78 C8 E1 0C 78 C8 E0
                                                                   ./a L≥.x Lβ.v L#.x L
000000A0
                  61 C8 F2 OC 78 C8 E1 OC 79 C8 23
                                                                   v/= L<sub>α.x</sub>L;/d L≥.xL
000000B0
                  3D C8 E0
                            0C 78
                                  C8
                                      3B 2F 64 C8 F2
                                                                   ./E╚α.x╚Richß.x╚
000000C0
           1B 2F 45 C8 E0 0C 78
                                  C8
                                      52 69 63 68 E1 0C 78 C8
00000D0
                                      00 00 00
               00 00 00 00
                            00
                               00
                                   00
                                                00 00
                                                       00 00 00
000000E0
                                  00
                                      50 45 00
                                                                   .......PE..L...
                                                00 4C 01 03 00
               00
                  00
                     00
                         00
                            00
                               00
                                                                   .\ddot{a}};....\alpha...
                                      00 00 00
00000F0
                     3B
                         00
                            00
                               00
                                   00
                                                00 E0
                                                       00
                                                             01
00000100
                     00
                         00
                            6E
                               00
                                   00
                                      00 A6
                                             00
                                                00
                                                   00
                                                                   ....n...a.....
                                                             00
00000110
                                                                   00
                            10
                               00
                                   00
                                      00 80
                                             00
                                                00
                                                   00
                                                          00 01
00000120
                                      05 00 01 00 05
               10
                     00
                         00
                            02 00
                                   00
                                                                   . . . . . . . . . . . . . . . .
00000130
                                                                   . . . . . . . . . 0 . . . . .
                                      00 30 01 00 00
                  00
                     00
                         00
                            00
                               00
                                   00
                                                          00 00
00000140
                  01 00
                         02
                            00
                               00
                                   80
                                      00 00 04 00 00
                                                      10 01 00
00000150
                                      00 00
                                                00
                  10
                     00
                         00
                            10
                               00
                                   00
                                             00
                                                   10
                                                       00
                                                          00
                                                             00
                                                                   .......m.. L
00000160
                            00
                                   00
                                      20 6D
                     00
                               00
                                             00
                                                00 C8
                                                          00 00
                                                                   .á..Hë.......
00000170
                                      00 00
                     00
                         48
                            89
                               00
                                   00
                                             00
                                                00
                                                   00
                                                       00
                                                          00
                                                             00
00000180
                     00
                         00
                            00
                               00
                                   00
                                      00 00
                                             00
                                                00
                                                   00
                                                       00
                                                          00
                                                             00
00000190
                  00 00 1C 00 00 00 00 00 00 00 00
```



#### Load PE file (Notepad.exe) into Memory









#### VA & RVA

- VA (Virtual Address): The address is called a "VA" because Windows creates a distinct VA space for each process, independent of physical memory. For almost all purposes, a VA should be considered just an address. A VA is not as predictable as an RVA because the loader might not load the image at its preferred location.
- RVA (Relative Virtual Address): The address of an item after it is loaded into memory, with the base address of the image file subtracted from it. The RVA of an item almost always differs from its position within the file on disk (file pointer).

#### RVA + ImageBase = VA

In 32bit Windows OS, each process has 4GB virtual memory which means the range of VA is: **00000000 - FFFFFFF** 



#### **DOS Header**

```
struct DOS Header
// short is 2 bytes, long is 4 bytes
     char signature[2] = { 'M', 'Z' };
     short lastsize;
     short nblocks;
     short nreloc;
     short hdrsize;
     short minalloc;
     short maxalloc;
    void *ss; // 2 byte value
    void *sp; // 2 byte value
     short checksum;
    void *ip; // 2 byte value
    void *cs; // 2 byte value
     short relocpos;
     short noverlay;
     short reserved1[4];
     short oem id;
     short oem info;
     short reserved2[10];
 }
```

The first 2 letters are **always** the letters "**MZ**", the initials of Mark Zbikowski, who created the first linker for DOS. To some people, the first few bytes in a file that determine the type of file are called the "**magic number**,"

short reserved2[10]; long e\_lfanew; // Offset to the 'PE\0\0' signature relative to the beginning of the file



#### **DOS Header**

long 
$$\rightarrow$$
 32 bit  $\rightarrow$  ? Byte

E0 00 00 00 value for e\_lfanew  $\rightarrow$  ?



#### **DOS Header**

e\_lfanew → 000000E0



#### **DOS** stub

```
..°..'.Í!,.LÍ!Th
00000040
                                                            54
                             B4
                                Π9
                                          В8
                                              01
                                                  4C
                                                     CD
00000050
                                                            6E
                                                                6F
                                              20
                                                                    is program canno
                                                                    t be run in DOS
00000060
                                75
                                    6E
                                       20
                                           69
                                              6E
                                                            53
                                                                20
00000070
                                                                    mode....$.....
                      65
                         2 E
                             DD
                                DD
                                    \cap A
                                       24
                                              00
                                                     00
                                                               nn
00000080
                                                                    ì...[;"ä5ò"ä5ò"ä5ò
                                                               F2
                                                                    kë:ò@ä5òkëUò@ä5ò
00000090
                                           EB
                                              55
                                                     Α9
000000000
                                       A8
                                          E4
                                                                    këhò»ä5ò¨ä4òcä5ò
                         BB
                                                               F2
000000B0
                                                                    këkò@ä5òkëjò¿ä5ò
                                           EB
                                              6A
                                                     BF
                                                                    këoò@ä5òRich~ä5ò
00000000
                                              63
                                                     A8
00000000
           00
                         00
                                00
                                       00
                                              00
                                                  00
                                                     00
                                                            00
                                                               00
```

https://virtualconsoles.com/online-emulators/dos/

```
C:\>notepad.exe
This program cannot be run in DOS mode.
```



#### **NT Header**

#### **IMAGE\_NT\_HEADERS32** structure

12/04/2018 • 2 minutes to read

Represents the PE header format.

#### **Syntax**

#### **Members**

Signature

A 4-byte signature identifying the file as a PE image. The bytes are "PE\0\0".

FileHeader

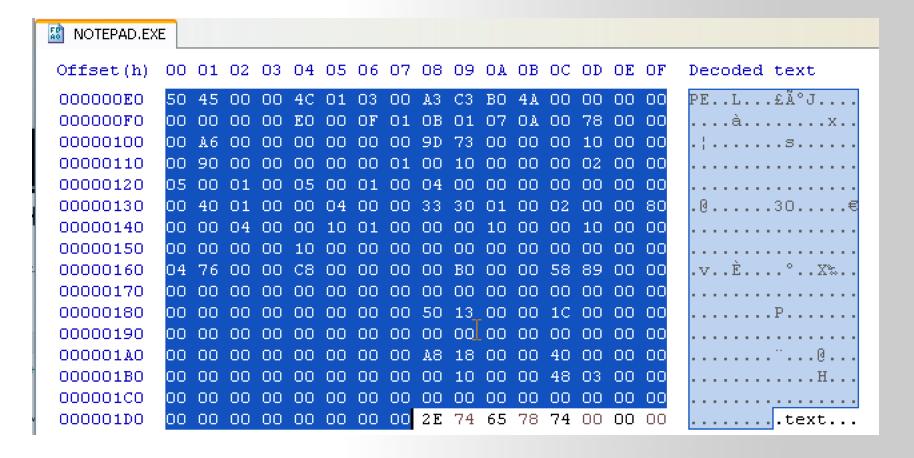
An <u>IMAGE FILE HEADER</u> structure that specifies the file header.

OptionalHeader

An  $\underline{\mathsf{IMAGE\_OPTIONAL\_HEADER}}$  structure that specifies the optional file header.

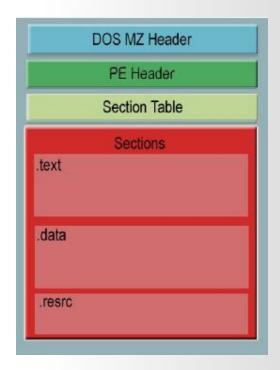


#### **NT Header**





# **Section Header**



Name	Privilege
.code	Executable, read
.data	Non-Executable, read/write
.resource	Non-Executable, read



# IMAGE\_SECTION\_HEADER structure

12/04/2018 • 4 minutes to read

Represents the image section header format.

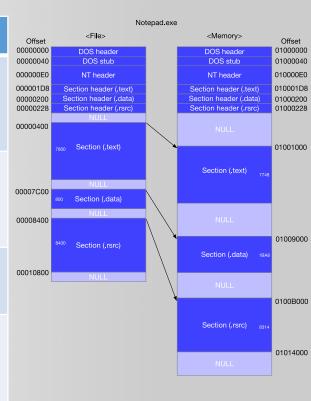
#### **Syntax**

```
1 Copy
C++
typedef struct _IMAGE_SECTION_HEADER {
  BYTE Name[IMAGE_SIZEOF_SHORT_NAME];
  union {
    DWORD PhysicalAddress;
    DWORD VirtualSize;
  } Misc;
  DWORD VirtualAddress;
  DWORD SizeOfRawData;
  DWORD PointerToRawData;
  DWORD PointerToRelocations;
  DWORD PointerToLinenumbers;
  WORD NumberOfRelocations;
  WORD NumberOfLinenumbers;
  DWORD Characteristics;
} IMAGE SECTION HEADER, *PIMAGE SECTION HEADER;
```



#### **Section Header**

Members	Meaning
VirtualSize	The total size of the section when loaded into memory, in bytes.
VirtualAddress	The address of the first byte of the section when loaded into memory (RVA)
SizeOfRaw Data	The size of the section data on disk, in bytes.
PointerToRawData	The address of the first byte of the section on disk.
Characteristics	The characteristics of the image.



https://docs.microsoft.com/enus/windows/desktop/api/winnt/ns-winnt-\_image\_section\_header



## **Section Header**

000001D0	00 00 00 00 00 00 00 00 2E 74 65 78 74 00 00 00text	
000001E0	48 77 00 00 00 10 00 00 00 78 00 00 00 04 00 00 Hwx	
000001F0	00 00 00 00 00 00 00 00 00 00 00 20 00 0	
00000200	2E 64 61 74 61 00 00 00 A8 1B 00 00 00 90 00 00 data"	
00000210	00 08 00 00 00 7C 00 00 00 00 00 00 00 00 00 <mark> </mark>	
00000220	00 00 00 00 40 00 00 CO 2E 72 73 72 63 00 00 00@À.rsrc	
00000230	58 89 00 00 00 B0 00 00 00 8A 00 00 00 84 00 00 x≈°š	
00000240	00 00 00 00 00 00 00 00 00 00 40 00 00 40	



#### Inspecting PE Header Information in Linux

```
import pefile
import sys

malware_file = sys.argv[1]

pe = pefile.PE(malware_file)
for section in pe.sections:
print "Name: %s VirtualSize: %s VirtualAddr: %s SizeofRawData: %s PointerToRawData: %s" %

(section.Name, hex(section.Misc_VirtualSize), hex(section.VirtualAddress), section.SizeOfRawData, section.PointerToRawData)
```

```
root@localhost python display_sections.py a99c01d5748b1bfd203fc1763e6612e8

Name: .text VirtualSize: 0x7378 VirtualAddr: 0x1000 SizeofRawData: 29696 PointerToRawData: 1024

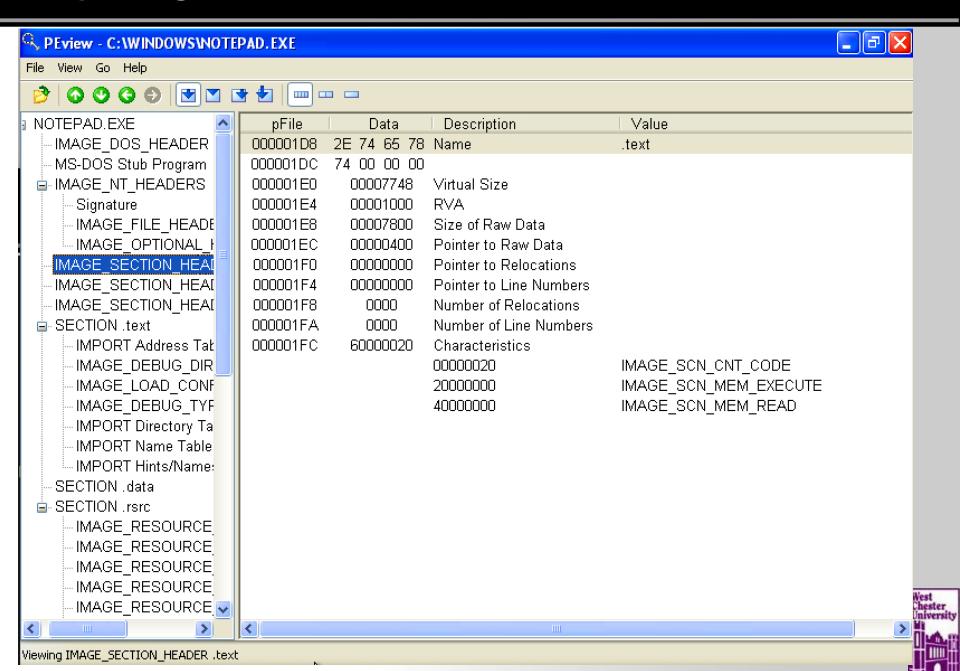
Name: .rdata VirtualSize: 0x261c VirtualAddr: 0x9000 SizeofRawData: 10240 PointerToRawData: 30720

Name: .data VirtualSize: 0x2cac VirtualAddr: 0xc000 SizeofRawData: 3584 PointerToRawData: 40960

Name: .rsrc VirtualSize: 0x1b4 VirtualAddr: 0xf000 SizeofRawData: 512 PointerToRawData: 44544
```



#### **Inspecting PE Header Information**



#### Inspecting file imports with pefile library

```
import pefile
     import sys
 3
     malware_file = sys.argv[1]
 4
     pe = pefile.PE(malware_file)
     if hasattr(pe, 'DIRECTORY ENTRY IMPORT'):
 6
          for entry in pe.DIRECTORY_ENTRY_IMPORT:
              print "%s" % entry.dll
 8
              for imp in entry.imports:
 9
10
                  if imp.name != None:
11
                      print "\t %s" % (imp.name)
                  else:
12
                      print "\tord(%s)" % (str(imp.ordinal))
13
              print "\n"
14
```



#### Inspecting file export with pefile library

```
import pefile
import sys

malware_file = sys.argv[1]

pe = pefile.PE(malware_file)
if hasattr(pe, 'DIRECTORY_ENTRY_EXPORT'):

for exp in pe.DIRECTORY_ENTRY_EXPORT.symbols:
    print "%s" % exp.name
```



#### Inspecting PE Header Information in Linux

```
import pefile
import sys

malware_file = sys.argv[1]

pe = pefile.PE(malware_file)
for section in pe.sections:
print "Name: %s VirtualSize: %s VirtualAddr: %s SizeofRawData: %s PointerToRawData: %s" %

(section.Name, hex(section.Misc_VirtualSize), hex(section.VirtualAddress), section.SizeOfRawData, section.PointerToRawData)
```

```
root@localhost python display_sections.py a99c01d5748b1bfd203fc1763e6612e8

Name: .text VirtualSize: 0x7378 VirtualAddr: 0x1000 SizeofRawData: 29696 PointerToRawData: 1024

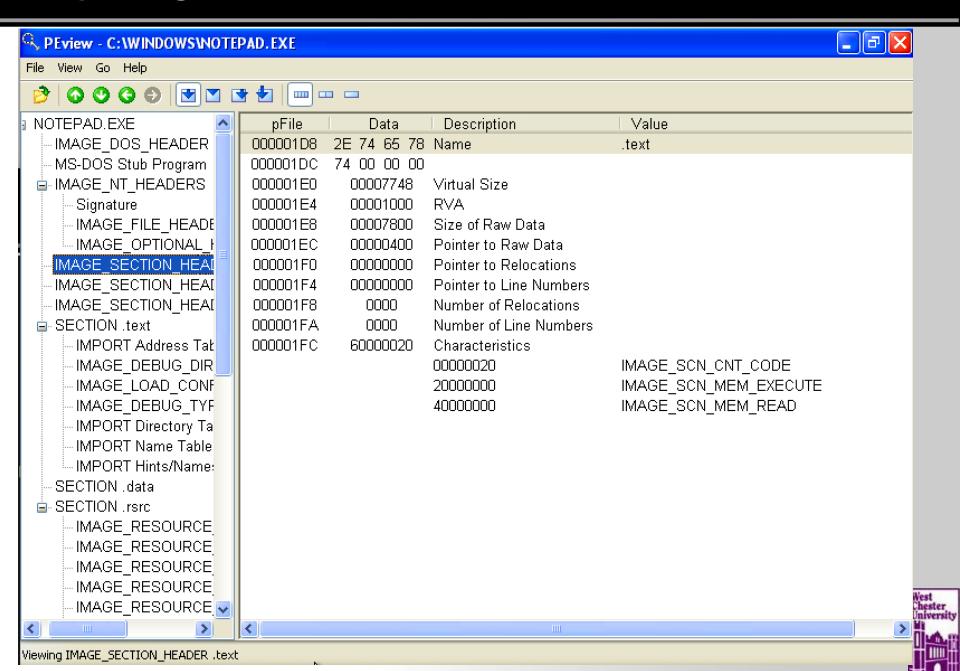
Name: .rdata VirtualSize: 0x261c VirtualAddr: 0x9000 SizeofRawData: 10240 PointerToRawData: 30720

Name: .data VirtualSize: 0x2cac VirtualAddr: 0xc000 SizeofRawData: 3584 PointerToRawData: 40960

Name: .rsrc VirtualSize: 0x1b4 VirtualAddr: 0xf000 SizeofRawData: 512 PointerToRawData: 44544
```



#### **Inspecting PE Header Information**





#### **Notepad.exe Process**

.text .data .rsrc kernel32.dll user32.dll gdi32.dll shell32.dll advapi32.dll ntdll32.dll

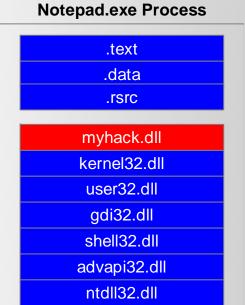


An executable file links to (or loads) a DLL in one of two ways:

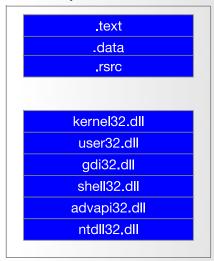
- Explicit Linking (run-time dynamic linking)
  - the executable using the DLL must make function calls to explicitly load and unload the DLL, and to access the DLL's exported functions.
- 1. Call LoadLibrary() (or a similar function) to load the DLL and obtain a module handle.
- 2. Call GetProcAddress() to obtain a function pointer to each exported function that the application wants to call.
- 3. Call FreeLibrary() when done with the DLL.
  - Implicit Linking (load-time dynamic linking)
    - The operating system loads the DLL when the executable using it is loaded.
- 1. A header file (.H file) containing the declarations of the exported functions and/or C++ classes.
- 2. An import library (.LIB files) to link with. The linker creates the import library when the DLL is built.
- 3. The actual DLL (.DLL file).







#### **Notepad.exe Process**



Implicit Linking (load-time dynamic linking)



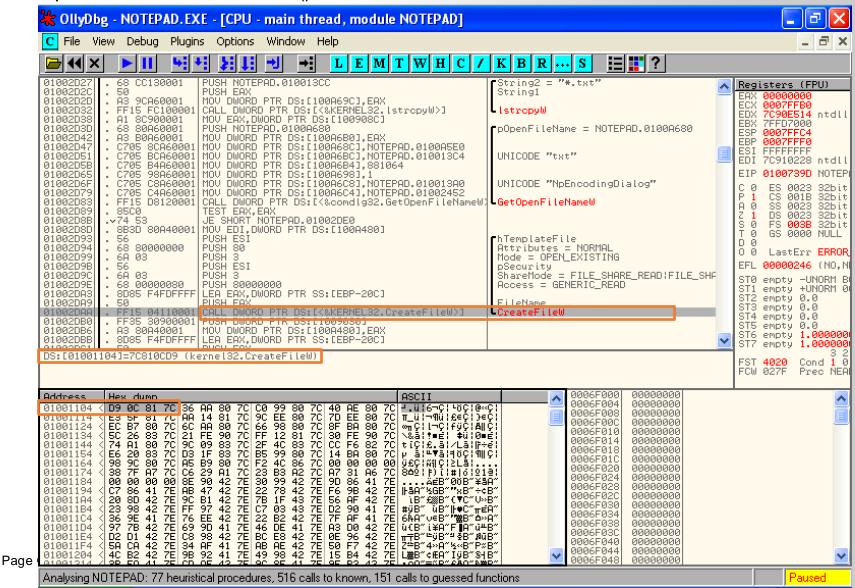
An executable file links to (or loads) a DLL in one of two ways:

- Explicit Linking (run-time dynamic linking) → DLL Injection
  - the executable using the DLL must make function calls to explicitly load and unload the DLL, and to access the DLL's exported functions.
- 1. Call LoadLibrary() (or a similar function) to load the DLL and obtain a module handle.
- 2. Call GetProcAddress() to obtain a function pointer to each exported function that the application wants to call.
- 3. Call FreeLibrary() when done with the DLL.
  - Implicit Linking (load-time dynamic linking) → IAT Table
    - The operating system loads the DLL when the executable using it is loaded.
- 1. A header file (.H file) containing the declarations of the exported functions and/or C++ classes.
- 2. An import library (.LIB files) to link with. The linker creates the import library when the DLL is built.
- 3. The actual DLL (.DLL file).



#### Implicit Linking and IAT (Import Address Table)

Notepad.exe Call CreateFileW() → Call 0x01001104 → Call 0x7C810CD9





#### Implicit Linking and IAT (Import Address Table)

Notepad.exe Call CreateFileW() → Call 0x01001104 → Call 0x7C810CD9

Function Name	IAT Address	Real Address
CreateFileW()	0x01001104	0x7C810CD9

When the application was first compiled, it was designed so that all API calls will **NOT** use **direct hardcoded addresses** but rather work through a function pointer.

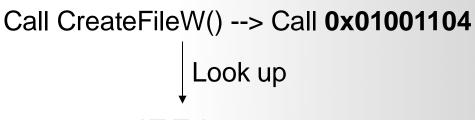
This was accomplished through the use of **an import address table**. This is a table of function pointers filled in by the windows loader as the dlls are loaded.



# Why IAT?



Support different Windows Version (9X, 2K, XP, Vista, 7, 8, 10)



XP

IAT Table

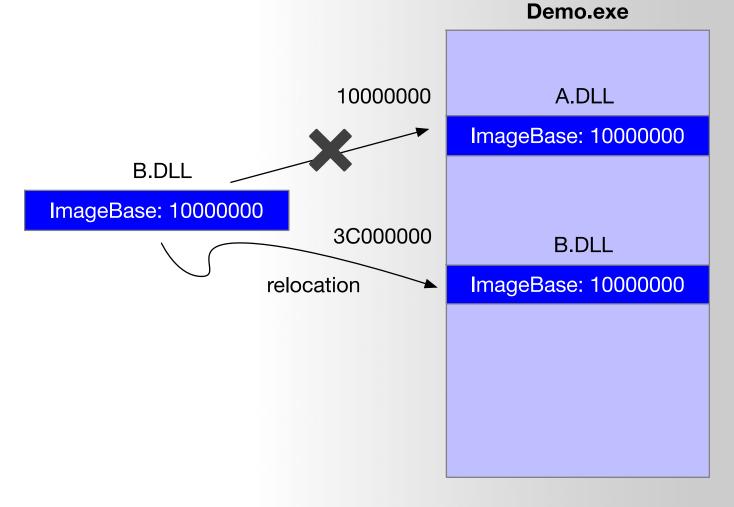
<b>Function Name</b>	IAT Address	Real Address
CreateFileW()	0x01001104	0x7C810CD9

#### Windows 7

Function Name	IAT Address	Real Address
CreateFileW()	0x01001104	0x7C81FFFF
•••		

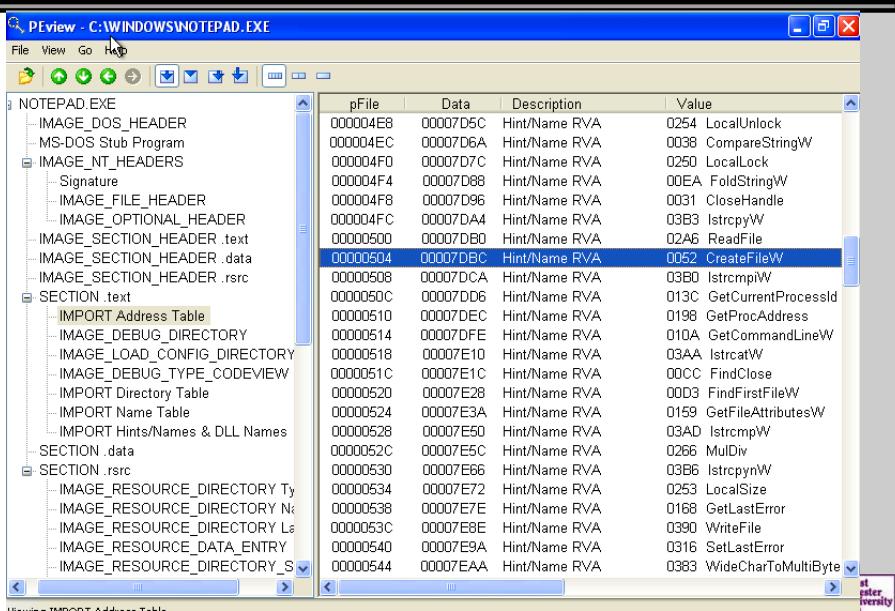


#### Support DLL Relocation





#### Look up IAT Table with PEview



Page '

Viewing IMPORT Address Table

🎁 start















#### **Import Directory Table**

The Import Directory Table contains entries for every DLL which is loaded by the executable. Each entry contains, among other, Import Lookup Table (ILT) and Import Address Table (IAT)





#### Inspecting file imports with pefile library

```
import pefile
import sys

malware_file = sys.argv[1]
pe = pefile.PE(malware_file)
if hasattr(pe, 'DIRECTORY_ENTRY_IMPORT'):
    for entry in pe.DIRECTORY_ENTRY_IMPORT:
        print "%s" % entry.dll
        for imp in entry.imports:
            if imp.name != None:
                print "\t %s \t %s" % (hex(imp.address), imp.name)
        else:
                print "\tord(%s)" % (str(imp.ordinal))
        print "\n"
```



KERNEL32.dll 0x1000c000 GetModuleFileNameW 0x1000c004 OutputDebugStringW 0x1000c008 CloseHandle 0x1000c00c CreateThread 0x1000c010 WriteConsoleW 0x1000c014 CreateFileW 0x1000c018 UnhandledExceptionFilter 0x1000c01c SetUnhandledExceptionFilter 0x1000c020 GetCurrentProcess 0x1000c024 TerminateProcess 0x1000c028 IsProcessorFeaturePresent 0x1000c02c QueryPerformanceCounter 0x1000c030 GetCurrentProcessId 0x1000c034 GetCurrentThreadId 0x1000c038 GetSystemTimeAsFileTime 0x1000c03c InitializeSListHead 0x1000c040 IsDebuggerPresent 0x1000c044 GetStartupInfoW 0x1000c048 GetModuleHandleW 0x1000c04c InterlockedFlushSList 0x1000c050 RtlUnwind 0x1000c054 GetLastError 0x1000c058 SetLastError 0x1000c05c EnterCriticalSection 0x1000c060 LeaveCriticalSection 0x1000c064 DeleteCriticalSection 0x1000c068 Initialize Critical Section And Spin Count0x1000c06c TlsAlloc 0x1000c070 TlsGetValue 0x1000c074 TlsSetValue 0x1000c078 TlsFree 0x1000c07c FreeLibrary GetProcAddress 0x1000c080 0x1000c084 LoadLibraryExW 0x1000c088 RaiseException 0x1000c08c ExitProcess 0x1000c090 GetModuleHandleExW 0x1000c094 HeapAlloc 0x1000c098 HeapFree 0x1000c09c FindClose 0x1000c0a0 FindFirstFileExW 0x1000c0a4 FindNextFileW 0x1000c0a8 IsValidCodePage 0x1000c0ac GetACP 0x1000c0b0 Get0EMCP 0x1000c0b4 GetCPInfo 0x1000c0b8 GetCommandLineA 0x1000c0bc GetCommandLineW 0x1000c0c0 MultiByteToWideChar 0x1000c0c4 WideCharToMultiByte 0x1000c0c8 GetEnvironmentStringsW 0x1000c0cc FreeEnvironmentStringsW 0x1000c0d0 GetStdHandle 0x1000c0d4 GetFileType 0x1000c0d8 LCMapStringW 0x1000c0dc GetProcessHeap 0x1000c0e0 GetStringTypeW 0x1000c0e4 HeapSize 0x1000c0e8 HeapReAlloc SetStdHandle 0x1000c0ec 0x1000c0f0 FlushFileBuffers 0x1000c0f4 WriteFile 0x1000c0f8 GetConsoleCP GetConsoleMode 0x1000c0fc 0×1000c100 SetFilePointerEx 0x1000c104 DecodePointer urlmon.dll

# Real-world Case Study



(Trojan.Win32.Dllhijack.a)

root@li254-249	python enum_exports.py 16d6b0e2c77da2776a88dd88c7cfc672
0x100011e0	CreateDatabaseQueryObject 1
0x100011e0	DataImporterMain 2
0x100011e0	FlashboxMain 3
0×100010d0	Kugou <u>M</u> ain 4

#### KuGou

From Wikipedia, the free encyclopedia

KuGou (Chinese: 酷狗音乐) is a Chinese music streaming and download service established in 2004 and owned by Tencent Music.<sup>[1][2]</sup> It is the largest music streaming service in the world, with more than 450 million monthly active users.<sup>[2]</sup> KuGou is the largest online music service in China, with a market share of 28%.<sup>[1]</sup> It has more than 800 million users.<sup>[1]</sup> A merger between China Music Corporation and Tencent's QQ Music was announced on July 15, 2016.<sup>[1][3]</sup> The services are expected to continue being offered separately.<sup>[1]</sup> Together with Kuwo, another online music service also owned by Tencent Music and the third largest one in China,<sup>[1]</sup> KuGou holds a music award ceremony, the KU Music Asian Music Awards,<sup>[4]</sup> also known as Cool Music Asia Festival Award.<sup>[5]</sup>

#### References [edit]

- 1. ^ a b c d e f Zen Soo (July 15, 2016). "Tencent to merge QQ Music service with China Music Corp to create streaming giant" &. South China Morning Post. Retrieved August 20, 2016.
- 3. \* Millward, Steven (July 15, 2016). "In China, 'Spotify' is free" & . Tech In Asia. Retrieved August 20, 2016.
- 4. ^ Kim Dong-Joo (March 31, 2016). "Kang Ta & SHINee garner awards at 'KU MUSIC ASIAN MUSIC AWARDS'" & . sg.style.yahoo.com. Retrieved August 20, 2016.
- 5. A "FTISLAND Wins "Asia's Popular Band" Award at Cool Music Asia Festival Award" &. Soompi. April 23, 2015. Retrieved August 20, 2016.

#### External links [edit]

#### KuGou



Developer(s)

**Tencent Music** 

Initial release

2004; 15 years ago

Operating system Android, iOS, Web,

Windows

Type

Music streaming

Website

www.kugou.com 🗗

 $\mathbf{III} \cap \mathbf{III}$ 

```
Dump of assembler code for function kugou!FlashboxMain:
   0x100011e0 <+0>:
                         xor
                                eax,eax
   0x100011e2 <+2>:
                         ret
   0x100011e3 <+3>:
                         nop
   0x100011e4 <+4>:
                         nop
   0x100011e5 <+5>:
                         nop
   0x100011e6 <+6>:
                         nop
   0x100011e7 <+7>:
                         nop
   0x100011e8 <+8>:
                         nop
   0x100011e9 <+9>:
                         nop
   0x100011ea <+10>:
                         nop
   0x100011eb <+11>:
                         nop
   0x100011ec <+12>:
                         nop
   0x100011ed <+13>:
                         nop
   0x100011ee <+14>:
                         nop
   0x100011ef <+15>:
                         nop
```



### Incident Response

Risk Assessment

Remote Access Uses network protocols on unusual ports

Network Behavior Contacts 2 domains and 2 hosts. View the network section for more details.

 https://www.hybridanalysis.com/sample/037203d274cb66bad34559c0f426e9e1bf91a048155
 240581f4aa554be17925c?environmentId=100



## Ofd6e3fb1cd5ec397ff3cdbaac39d80c

```
root@li254-249
                      python enum exports.py 0fd6e3fb1cd5ec397ff3cdbaac39d80c
0x10002628
                 AheadLib LpkPresent
0x10002634
                 AheadLib ScriptApplyDigitSubstitution
                                                          37
0×10002640
                 AheadLib ScriptApplyLogicalWidth
                                                          38
0x1000264c
                 AheadLib ScriptBreak
0×10002658
                 AheadLib ScriptCPtoX
                 AheadLib ScriptCacheGetHeight
0×10002664
0×10002670
                                                  42
                 AheadLib ScriptFreeCache
0x1000267c
                 AheadLib ScriptGetCMap
                                                  43
0x10002688
                 AheadLib ScriptGetFontProperties
0x10002694
                 AheadLib_ScriptGetGlyphABCWidth
0×1000271f
                 AheadLib ScriptGetLogicalWidths
0x1000272b
                 AheadLib ScriptGetProperties
                 AheadLib ScriptIsComplex
0x10002737
                                                  48
0x10002743
                                                  49
                 AheadLib ScriptItemize
0×10003091
                 AheadLib ScriptJustify
                                                  50
0x1000309d
                 AheadLib ScriptLayout
0x100030a9
                 AheadLib ScriptPlace
                                         52
0x100030b5
                 AheadLib ScriptRecordDigitSubstitution
0x100030c1
                 AheadLib ScriptShape
0x100030cd
                                                  55
                 AheadLib ScriptStringAnalyse
0x100030d9
                 AheadLib ScriptStringCPtoX
                                                  56
0x100030e5
                 AheadLib ScriptStringFree
0x100030f1
                 AheadLib ScriptStringGetLogicalWidths
                                                          58
0x100030fd
                 AheadLib ScriptStringGetOrder
0×10003109
                 AheadLib ScriptStringOut
                                                  60
                 AheadLib ScriptStringValidate
0×10003115
                                                  61
0×10003121
                 AheadLib ScriptStringXtoCP
                                                  62
0x1000312d
                                                          63
                 AheadLib ScriptString pLogAttr
0x10003139
                 AheadLib ScriptString pSize
                                                  64
                                                          65
0×10003145
                 AheadLib ScriptString pcOutChars
0×10003151
                                                  66
                 AheadLib ScriptTextOut
0x1000315d
                 AheadLib ScriptXtoCP
                                         67
0x10003169
                                                  68
                 AheadLib UspAllocCache
0×10003175
                 AheadLib UspAllocTemp
                                         69
0×10003181
                 AheadLib UspFreeMem
                                          70
0x100023cf
                 LpkDllInitialize
                                         311
0x100023db
                 LpkDrawTextEx
                                 411
0×1001d040
                 LpkEditControl
0x100023f3
                                611
                 LpkExtTextOut
0x100023ff
                 LpkGetCharacterPlacement
0x100025f3
                 LpkGetTextExtentExPoint
                                                  811
0x100023b7
                 LpkInitialize
                                 111
0×10002604
                 LpkPSMTextOut
                                 911
0×10002628
                 LpkPresent
0x100023c3
                 LpkTabbedText0ut
                                          1011
0×10002610
                 LpkUseGDIWidthCache
0x100023cf
                 MemCode LpkDllInitialize
0x100023db
                 MemCode LpkDrawTextEx
0x100023e7
                 MemCode LpkEditControl
0x100023f3
                 MemCode LpkExtTextOut
0x100023ff
                 MemCode LpkGetCharacterPlacement
0x100025f3
                 MemCode LpkGetTextExtentExPoint
0x100023b7
                 MemCode LpkInitialize
0×10002604
                 MemCode LpkPSMTextOut
                                         79
0x100023c3
                 MemCode LpkTabbedTextOut
                                                  80
                                                  81
0×10002610
                 MemCode LpkUseGDIWidthCache
0x1000261c
                 MemCode ftsWordBreak
0x10002634
                 ScriptApplyDigitSubstitution
0×10002640
                 ScriptApplyLogicalWidth
```

```
gdb-peda$ disas ScriptBreak
Dump of assembler code for function drc!ScriptBreak:
    0x1000264c <+0>: push 0x1001d804
    0x10002651 <+5>: call 0x1000233d
    0x10002656 <+10>: jmp eax
End of assembler dump.
```

```
gdb-peda$ disas LpkPresent
Dump of assembler code for function drc!LpkPresent:
    0x10002628 <+0>: push 0x1001d7c0
    0x1000262d <+5>: call 0x1000233d
    0x10002632 <+10>: jmp eax
End of assembler dump.
```



## 6a764e4e6db461781d080034aab85aff & cc3c6c77e118a83ca0513c25c208832c

root@li254-249	python enum_exports.py 6a764e4e6db	461781d080034aab85aff	root@li254-249	<pre>python enum exports.py cc</pre>	3c6c77e118a83ca0513c25c208832c
0×10004f00	AheadLib ScriptApplyDigitSubstitution	36	0×10001100	LpkPresent $\overline{1}$	
0×10004f10	AheadLib ScriptApplyLogicalWidth	37	0×10001120	ScriptApplyDigitSubstitution	2
0×10004f20	AheadLib ScriptBreak 38		0×10001140	ScriptApplyLogicalWidth	3
0x10004f30	AheadLib_ScriptCPtoX 39		0×10001160	ScriptBreak 4	
0×10004f40	AheadLib ScriptCacheGetHeight 40		0×10001180	ScriptCPtoX 5	
0×10004f50	AheadLib ScriptFreeCache 41		0x100011a0		
0×10004f60	AheadLib ScriptGetCMap 42				
0×10004f70	AheadLib ScriptGetFontProperties	43	0×100011c0	ScriptFreeCache 7	
0×10004f80	AheadLib ScriptGetGlyphABCWidth	44	0×100011e0	ScriptGetCMap 8	
0×10004f90	AheadLib ScriptGetLogicalWidths	45	0×10001200	ScriptGetFontProperties	9
0×10004fa0	AheadLib ScriptGetProperties 46		0×10001220	ScriptGetGlyphABCWidth	10
0×10004fb0	AheadLib ScriptIsComplex 47		0×10001240	ScriptGetLogicalWidths	11
0x10004fc0	AheadLib ScriptItemize 48		0×10001260	ScriptGetProperties 12	
0×10004fd0	AheadLib ScriptJustify 49		0×10001280	ScriptIsComplex 13	
0x10004fe0	AheadLib ScriptLayout 50		0x100012a0	ScriptItemize 14	
0×10004ff0	AheadLib ScriptPlace 51		0×100012c0	ScriptJustify 15	
0×10005000	AheadLib ScriptRecordDigitSubstitution	52	0x100012e0	ScriptLayout 16	
0×10005010	AheadLib ScriptShape 53		0×10001300	ScriptPlace 17	
0×10005020	AheadLib ScriptStringAnalyse 54		0×10001320	ScriptRecordDigitSubstitution	18
0×10005030	AheadLib ScriptStringCPtoX 55		0×10001340	ScriptShape 19	
0×10005040	AheadLib ScriptStringFree 56		0×10001360	ScriptStringAnalyse 20	
0×10005050	AheadLib ScriptStringGetLogicalWidths	57	0×10001380	ScriptStringCPtoX 21	
0×10005060	AheadLib ScriptStringGetOrder 58		0x100013a0	ScriptStringFree 22	
0×10005070	AheadLib ScriptStringOut 59		0x100013a0	ScriptStringTree 22 ScriptStringGetLogicalWidths	23
0×10005080	AheadLib ScriptStringValidate 60			, 3	25
0×10005090	AheadLib ScriptStringXtoCP 61		0x100013e0		
0x100050a0	AheadLib ScriptString pLogAttr	62	0×10001400	ScriptStringOut 25	
0×100050b0	AheadLib ScriptString pSize 63		0×10001420	ScriptStringValidate 26	
0×100050c0	AheadLib_ScriptString_pcOutChars	64	0×10001440	ScriptStringXtoCP 27	
0×100050d0	AheadLib_ScriptTextOut 65		0×10001460	ScriptString_pLogAttr 28	
0x100050e0	AheadLib_ScriptXtoCP 66		0×10001480	ScriptString_pSize 29	
0×100050f0	AheadLib_UspAllocCache 67		0×100014a0	ScriptString_pcOutChars	30
0×10005100	AheadLib_UspAllocTemp 68		0×100014c0	ScriptTextOut 31	
0×10005110	AheadLib_UspFreeMem 69		0x100014e0	ScriptXtoCP 32	
0x10004ef0	AheadLib_mmLpkPresent 70		0×10001890	ServiceMain 36	
0x10004e50	LpkDllInitialize 311		0×10001500	UspAllocCache 33	
0x10004e60	LpkDrawTextEx 411		0×10001520	UspAllocTemp 34	
0x1000e92c	LpkEditControl 71		0×10001540	UspFreeMem 35	
0x10004e80	LpkExtTextOut 611				
0x10004e90	LpkGetCharacterPlacement 711				
0x10004ea0	LpkGetTextExtentExPoint 811				
0x10004e30	LpkInitialize 111				
0x10004ec0	LpkPSMTextOut 911				
0x10004ef0	LpkPresent 1				
0x10004e40	LpkTabbedTextOut 211				
0x10004ed0	LpkUseGDIWidthCache 1011				
0x10004e50	MemCode_LpkDllInitialize 72				
0x10004e60	MemCode_LpkDrawTextEx 73				
0x10004e70	MemCode_LpkEditControl 74				
0x10004e80	MemCode_LpkExtTextOut 75	76			
0x10004e90	MemCode_LpkGetCharacterPlacement	76			
0x10004ea0	MemCode_LpkGetTextExtentExPoint	77			
0x10004e30	MemCode_LpkInitialize 78				West Chester University
0×10004ec0	MemCode_LpkPSMTextOut 79				Chester
0×10004e40	MemCode_LpkTabbedTextOut 80				Wh
0x10004ed0	MemCode_LpkUseGDIWidthCache 81				ini.a. m
0x10004ee0	MemCode_ftsWordBreak 82				U Parkini
0×10004f00	ScriptApplyDigitSubstitution 2				
0×10004f10	ScriptApplyLogicalWidth 3				

## e0bed0b33e7b6183f654f0944b607618

#### e0bed0b33e7b6183f654f0944b607618

```
root@li254-249
                      python enum exports.py e0bed0b33e7b6183f654f0944b607618
0x100165f0
                 LsaApCallPackage
                 LsaApCallPackagePassthrough
0×10016610
                                                  2
0×10016600
                 LsaApCallPackageUntrusted
                 LsaApInitializePackage
0x100165e0
                 LsaApLogonTerminated
0x10016620
                                          5
                 LsaApLogonUserEx2
0×10016470
                                          6
0×10016560
                 SpInitialize
                 SpInstanceInit
0x100165d0
                                          8
0×10016570
                 SpLsaModeInitialize
                                          9
                 SpUserModeInitialize
0x100165c0
                                          10
```



## db8199eeb2d75e789df72cd8852a9fbb

(Rootkit.Win32.blackken.b)

## db8199eeb2d75e789df72cd8852a9fbb

```
      root@li254-249
      python enum_exports.py db8199eeb2d75e789df72cd8852a9fbb

      0x10006707
      ?Start@@YGKPAX@Z
      1

      0x10006707
      MakeCache
      2
```

Is this claim correct?

If two export functions share the same address, it's a malware.



## 1c1131112db91382b9d8b46115045097

### 1c1131112db91382b9d8b46115045097



## **EAT (Export Address Table)**

- Similar to IAT, EAT data is stored in IMAGE\_EXPORT\_DIRECTORY
- EAT contains an RVA that points to an array of pointers to (RVAs of) the functions in the module.



## Lab3

- Create your own anti-malware system based on heuristic analysis.
- Check course website

