Reverse Engineering: Towards Malware Analysis

Lecture - Basic Static Analysis Techniques

Computer Science Practice
SPRING 2023

Outline

- **Static Analysis**: Extract binary features ***without*** execution
- **Basic Static Analysis**: Inspect binaries ***without*** looking at their instructions
- Antivirus
- Hashing
- Strings
- Headers
- Functions
- Packers
- Most importantly generate leads for future analysis!

Antivirus Scanning

- A useful first step
- But AV products are not perfect:
  - Too often rely on signatures or heuristics
  - Easy to modify & evade
- Multiple scanners increase your chances of identifying known malware
  - So use VirusTotal: www.virustotal.com
- Be careful
  - OPSEC

Other Open Source Intelligence

Let me google that for you

- MD5, filename, and “interesting” string search
- No cheating in malware analysis - except the back of the book ;)

Hashing

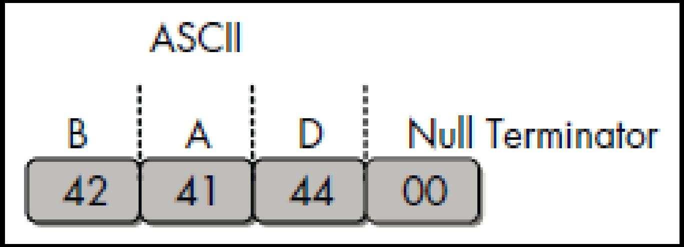
- Digital fingerprint
- Crypto calculation for unique value
- 1-bit change → large difference in value
- MD5 and SHA-1 are most popular
- Blacklist and whitelist
- Generate, search, share…
  - We use internally to track malware
  - So does the rest of the industry

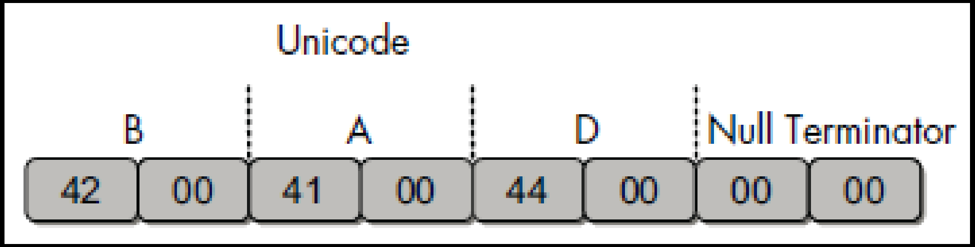
Strings

- Identifying sequence of characters
- Strings are used in source code to:
  - Print a message
  - Connect to URL or Domain
  - Copy a file
  - Registry key
- Strings program: print strings with 3 or more characters
  - Part of SysInternals: https://learn.microsoft.com/en-us/sysinternals/downloads/strings
- Many versions out there

Review of Strings

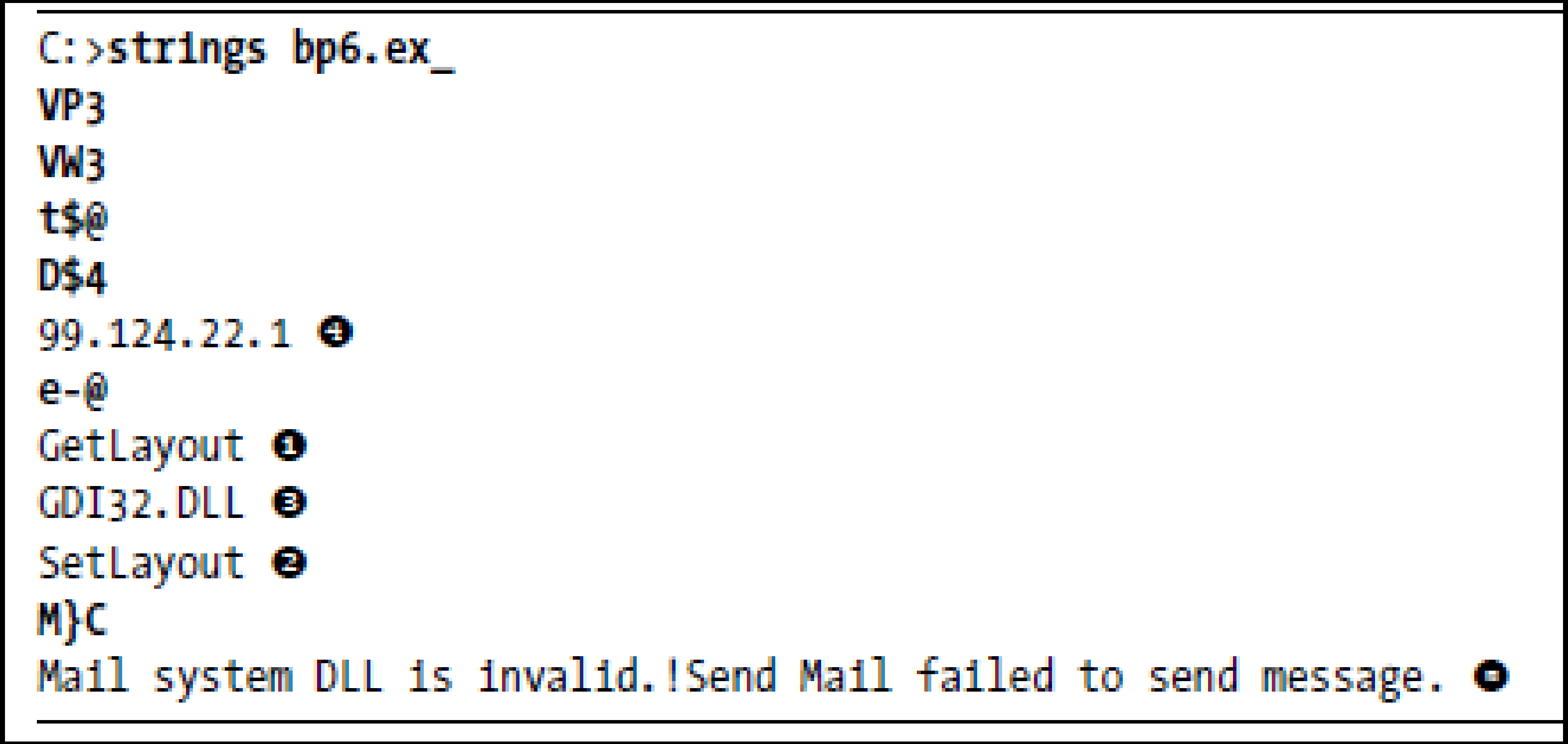
- ASCII and Unicode strings end with a null (0x00) terminator
- ASCII strings use 1 byte per character
- Typically, Unicode strings use 2 bytes per character
- Usually, Wide Character string == Unicode string
- Other string types?





Searching for Strings

- Not fool proof
- Requires human interpretation



```
C:>strings bp6.ex_
VP3
VW3
t$@
D$4
99.124.22.1 ❹
e-@
GetLayout ❶
GDI32.DLL ❸
SetLayout ❷
M}C
Mail system DLL is invalid.!Send Mail failed to send message. ❺
```

Be Careful!!!!!

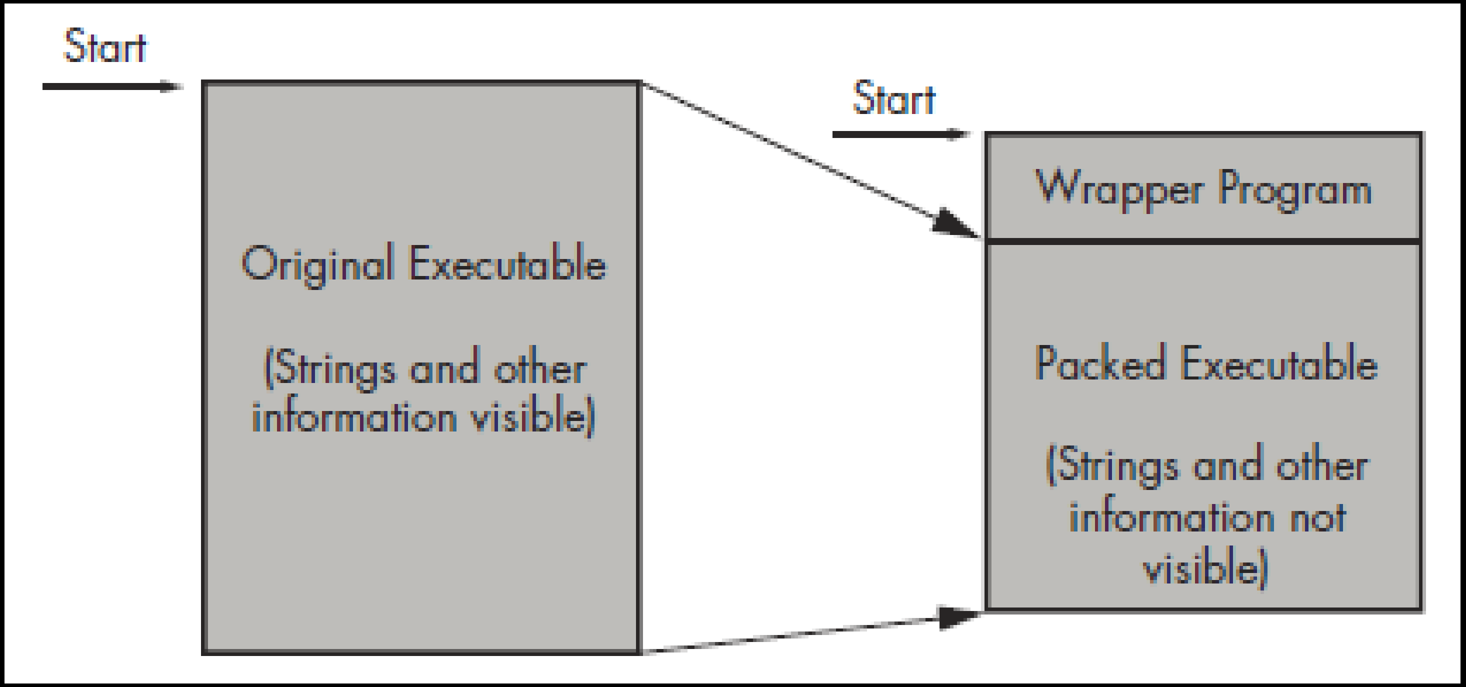
- Strings are great leads but…
  - May not be used at runtime
  - Can be modified before use

Packed & Obfuscated Malware

- Obfuscated
  - hiding the execution
- Packed
  - A subset of obfuscation
  - Compressed and not directly analyzable
- Packing goals: smaller, obfuscated or both
- Can protect against Basic Static Analysis techniques: strings & instructions become unreadable

Understanding Packed Files

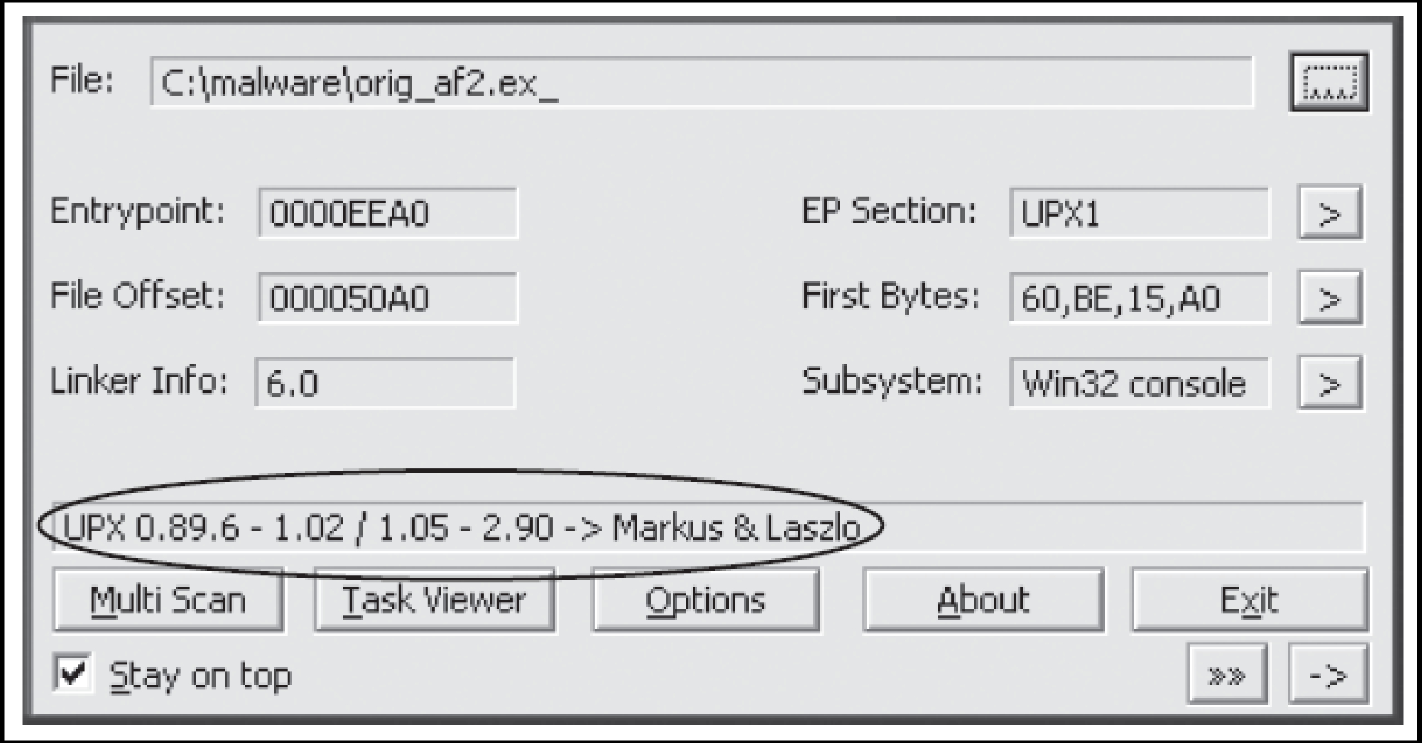
- Anatomy of a packed file: you'll see the wrapper only
- Static analysis of a packed file


Start
Original Executable
(Strings and other information visible)
Start
Wrapper Program
Packed Executable
(Strings and other information not visible)

Detecting Packers with PEiD

- PEiD
  - A tool to help you determine the packer or compiler used to create the binary
  - When successful PEiD will tell you the packer type and version
  - Plug-ins, e.g. Krypto Analyzer
- WARNING!
  - Many PEiD plugins will actually RUN the binary when attempting to determine the packer type
  - Use PEiD in your VM!

PEiD in action



```
File:        C:\malware\orig_af2.ex_

Entrypoint:  0000EEA0          EP Section:   UPX1
File Offset: 000050A0          First Bytes:  60,BE,15,A0
Linker Info: 6.0               Subsystem:    Win32 console

UPX 0.89.6 - 1.02 / 1.05 - 2.90 -> Markus & Laszlo
```

Packer: UPX (Ultimate Packer for eXecutables)

- Most common packer
- Used for compression mostly

- `upx -d` will do the trick
- Then continue with static analysis techniques

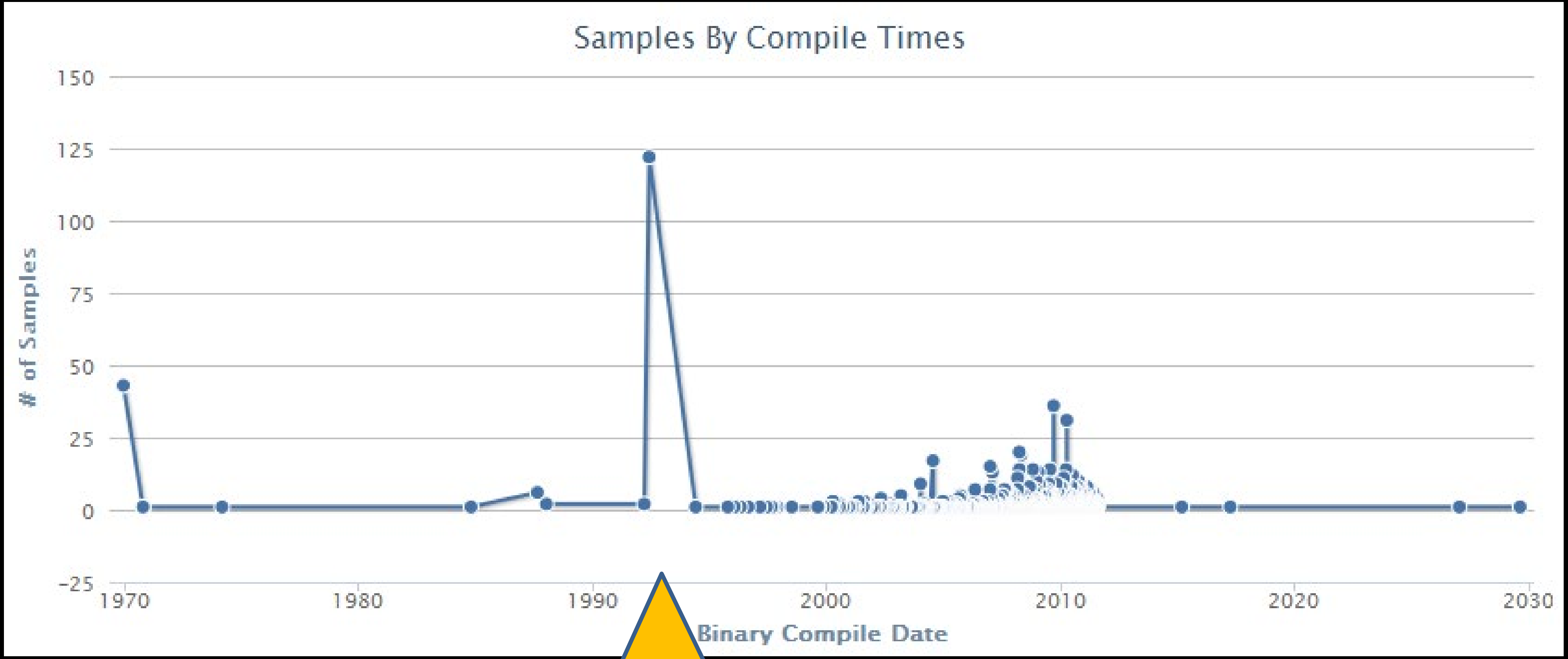
Portable Executable (PE) File Format

- Windows format for all executables
  - `EXE`
  - `DLL`
  - `SYS`
- PE header == lots of goodies for the Malware Analyst
  - Includes a list of linked libraries
- Libraries can be linked in several ways
  - Static: Rarely used for Windows
  - Dynamic: The most common method
  - Runtime
    - `LoadLibrary` & `GetProcAddress`
    - Often used by obfuscated malware
- **An In-Depth Look into the Win32 Portable Executable File Format**
  - http://msdn.microsoft.com/en-us/magazine/cc301805.aspx

PE Header Nuggets

- **Imports** – Functions from other libraries that are used; EXEs typically import functions
- **Exports** – Functions that are meant to be called by other programs; DLLs export functions
- **Timestamp** – Time when the program was compiled
- **Sections** – Names and sizes of parts of the file on disk and in memory
- **Subsystem** – Command line or GUI program?
- **Resources** – Strings, icons, menus, and other info used at runtime

Timestamp: Is the Information Accurate?



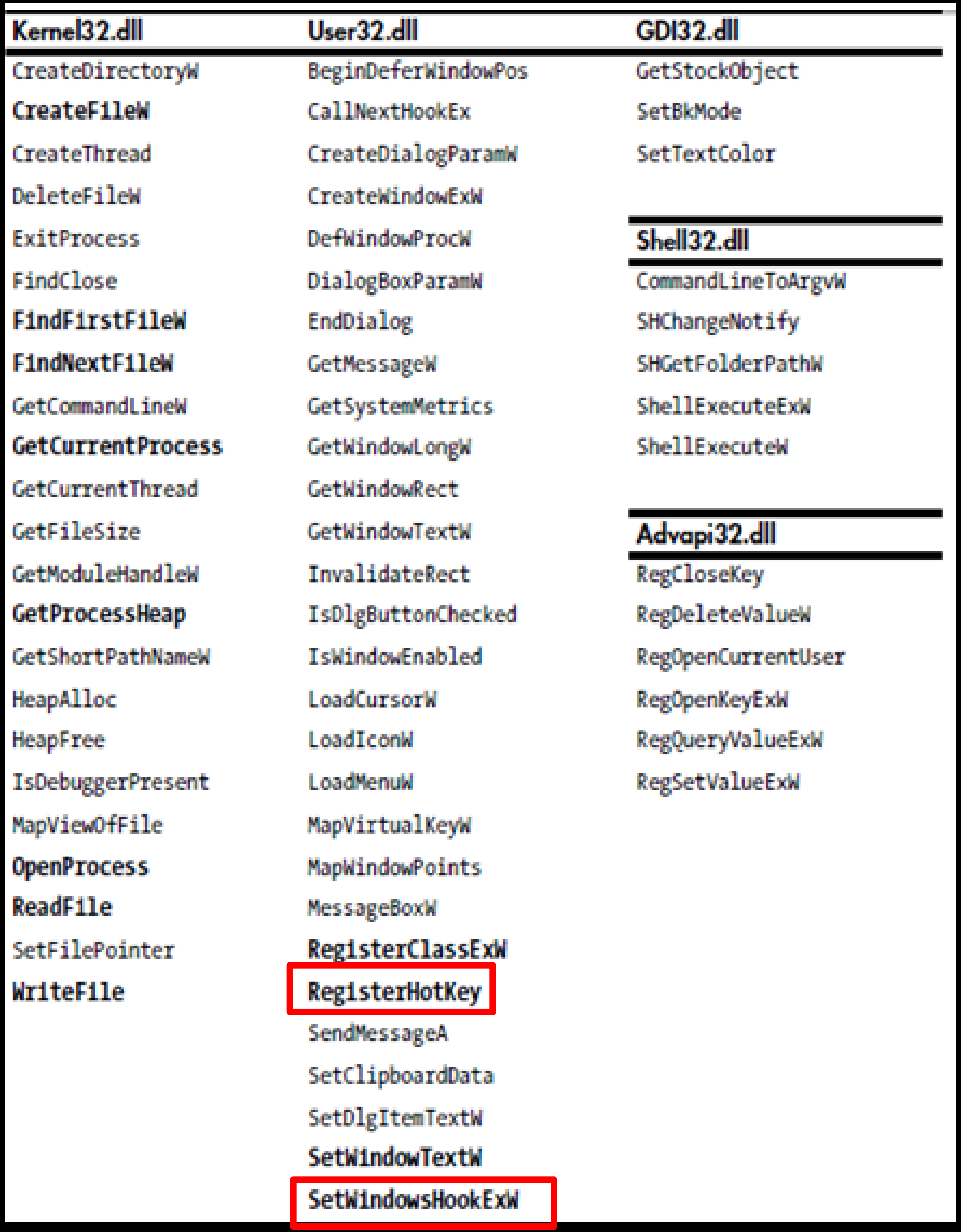
19 June 1992

Common DLLs

- `Kernel32.dll`:
  Core functionality, e.g. access & manipulation of memory, files, hardware
- `Advapi.dll`:
  Advanced core Windows components, e.g. Registry
- `User32.dll`: UI components
- `Gdi32.dll`: Graphics display & manipulation
- `Ntdll.dll`: Interface to the Windows kernel;
  typically *not* directly called by normal executables
- `WSock32.dll` & `WS2_32.dll`: Networking DLLs
- `Wininet.dll`: Networking applications, e.g. FTP, HTTP, NTP

Import Functions

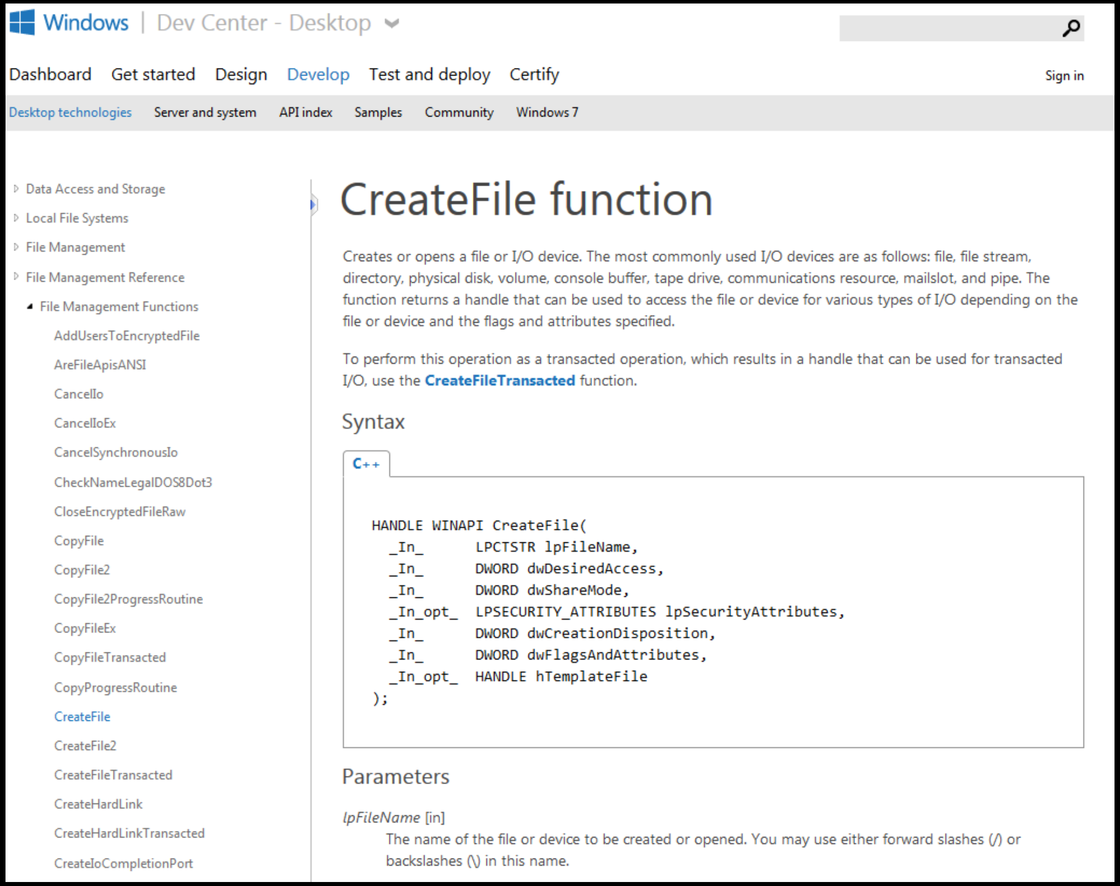
- Example 1: `Keylogger.exe`



Kernel32.dll	User32.dll	GDI32.dll
CreateDirectoryW	BeginDeferWindowPos	GetStockObject
CreateFileW	CallNextHookEx	SetBkMode
CreateThread	CreateDialogParamW	SetTextColor
DeleteFileW	CreateWindowExW	
ExitProcess	DefWindowProcW	**Shell32.dll**
FindClose	DialogBoxParamW	CommandLineToArgvW
FindFirstFileW	EndDialog	SHChangeNotify
FindNextFileW	GetMessageW	SHGetFolderPathW
GetCommandLineW	GetSystemMetrics	ShellExecuteExW
GetCurrentProcess	GetWindowLongW	ShellExecuteW
GetCurrentThread	GetWindowRect	
GetFileSize	GetWindowTextW	**Advapi32.dll**
GetModuleHandleW	InvalidateRect	RegCloseKey
GetProcessHeap	IsDlgButtonChecked	RegDeleteValueW
GetShortPathNameW	IsWindowEnabled	RegOpenCurrentUser
HeapAlloc	LoadCursorW	RegOpenKeyExW
HeapFree	LoadIconW	RegQueryValueExW
IsDebuggerPresent	LoadMenuW	RegSetValueExW
MapViewOfFile	MapVirtualKeyW	
OpenProcess	MapWindowPoints	
ReadFile	MessageBoxW	
SetFilePointer	**RegisterClassExW**	
WriteFile	**RegisterHotKey**	
	SendMessageA	
	SetClipboardData	
	SetDlgItemTextW	
	SetWindowTextW	
	SetWindowsHookExW	

More Information

- MSDN
- Appendix A of our textbook



CreateFile function

Creates or opens a file or I/O device. The most commonly used I/O devices are as follows: file, file stream, directory, physical disk, volume, console buffer, tape drive, communications resource, mailslot, and pipe. The function returns a handle that can be used to access the file or device for various types of I/O depending on the file or device and the flags and attributes specified.

To perform this operation as a transacted operation, which results in a handle that can be used for transacted I/O, use the **CreateFileTransacted** function.

Syntax

C++

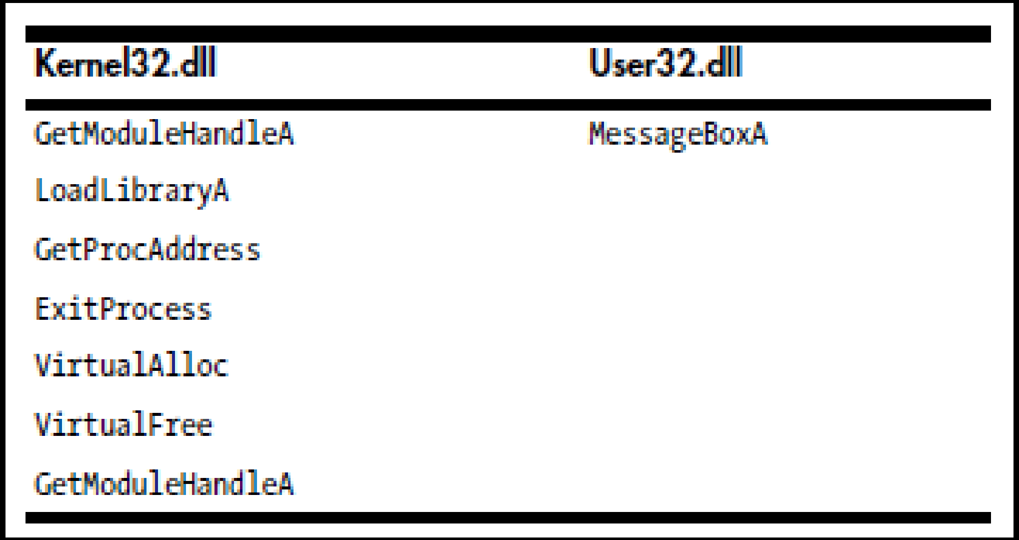
```
HANDLE WINAPI CreateFile(
  _In_      LPCTSTR lpFileName,
  _In_      DWORD dwDesiredAccess,
  _In_      DWORD dwShareMode,
  _In_opt_  LPSECURITY_ATTRIBUTES lpSecurityAttributes,
  _In_      DWORD dwCreationDisposition,
  _In_      DWORD dwFlagsAndAttributes,
  _In_opt_  HANDLE hTemplateFile
);
```

Parameters

lpFileName [in]
The name of the file or device to be created or opened. You may use either forward slashes (/) or backslashes (\) in this name.

Import Functions

- Example 2: `PackedProgram.exe`
- A dead-end
- *What can you infer/conclude?*



Kernel32.dll	User32.dll
GetModuleHandleA	MessageBoxA
LoadLibraryA	
GetProcAddress	
ExitProcess	
VirtualAlloc	
VirtualFree	
GetModuleHandleA	

Export Functions

- Example: `Keylogger.exe`
  - `LowLevelMouseProc`
  - `LowLevelKeyboardProc`

- Most important for DLLs you analyze
  - Provides functionality to other programs & code

PE File Headers & Sections

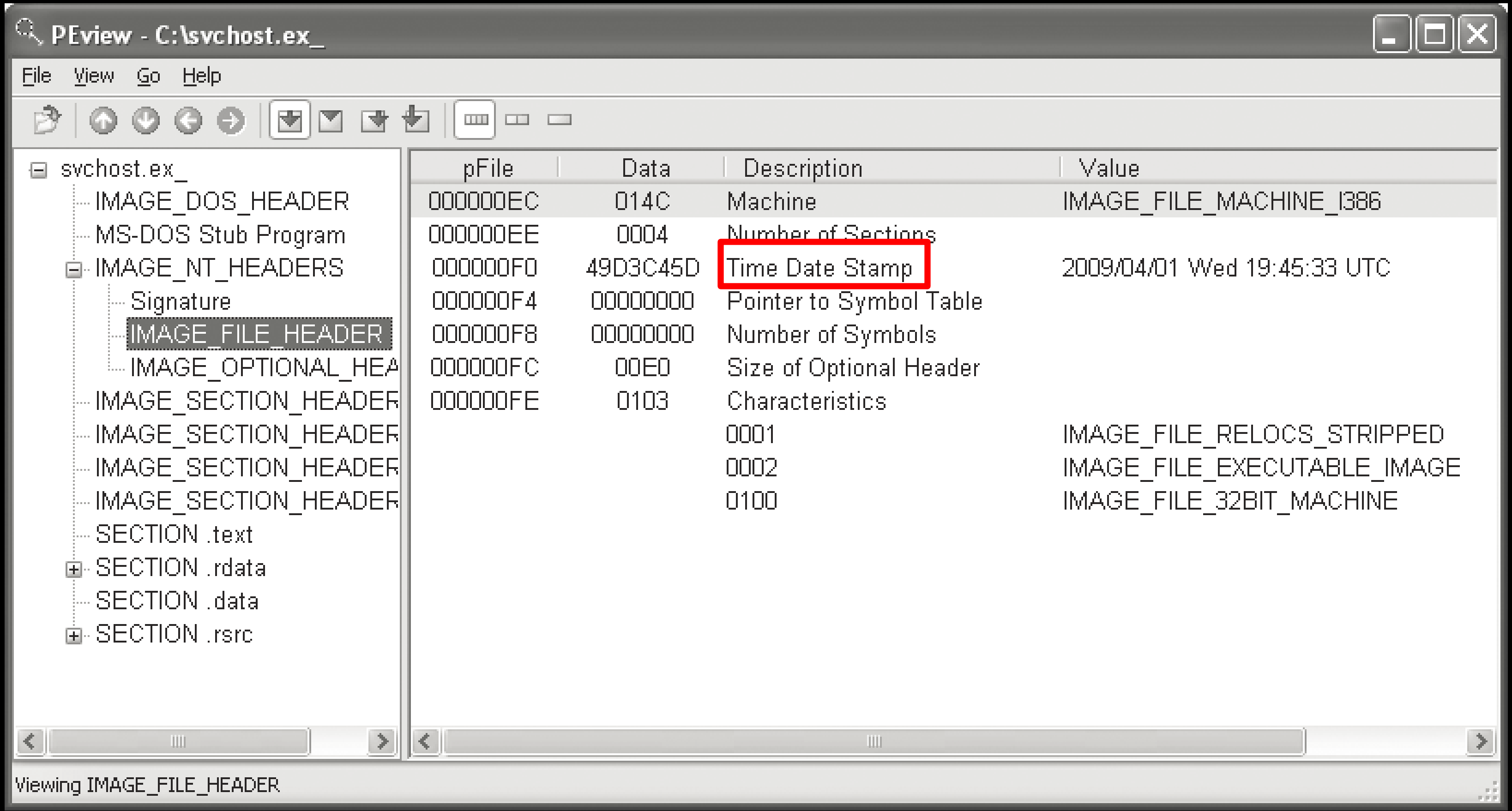
- PE file format
  - Headers
  - `.text`:
    the executable code (instructions)
  - `.rdata`:
    globally accessible read-only data, including imports & exports
  - `.data`: global data
  - `.rsrc`: resources, e.g. icons, images, menus, strings
- A sample screenshot later

PE Tools

- PEView
- Dependency Walker
- PE Explorer
- Resource Hacker
- CFF Explorer
- *So many out there…*

PEView

- Browse the PE header



```
svchost.ex_
  IMAGE_DOS_HEADER
  MS-DOS Stub Program
  IMAGE_NT_HEADERS
    Signature
    IMAGE_FILE_HEADER
    IMAGE_OPTIONAL_HEA
  IMAGE_SECTION_HEADER
  IMAGE_SECTION_HEADER
  IMAGE_SECTION_HEADER
  IMAGE_SECTION_HEADER
  SECTION .text
  SECTION .rdata
  SECTION .data
  SECTION .rsrc
```

pFile	Data	Description	Value
000000EC	014C	Machine	IMAGE_FILE_MACHINE_I386
000000EE	0004	Number of Sections	
000000F0	49D3C45D	Time Date Stamp	2009/04/01 Wed 19:45:33 UTC
000000F4	00000000	Pointer to Symbol Table	
000000F8	00000000	Number of Symbols	
000000FC	00E0	Size of Optional Header	
000000FE	0103	Characteristics	
		0001	IMAGE_FILE_RELOCS_STRIPPED
		0002	IMAGE_FILE_EXECUTABLE_IMAGE
		0100	IMAGE_FILE_32BIT_MACHINE

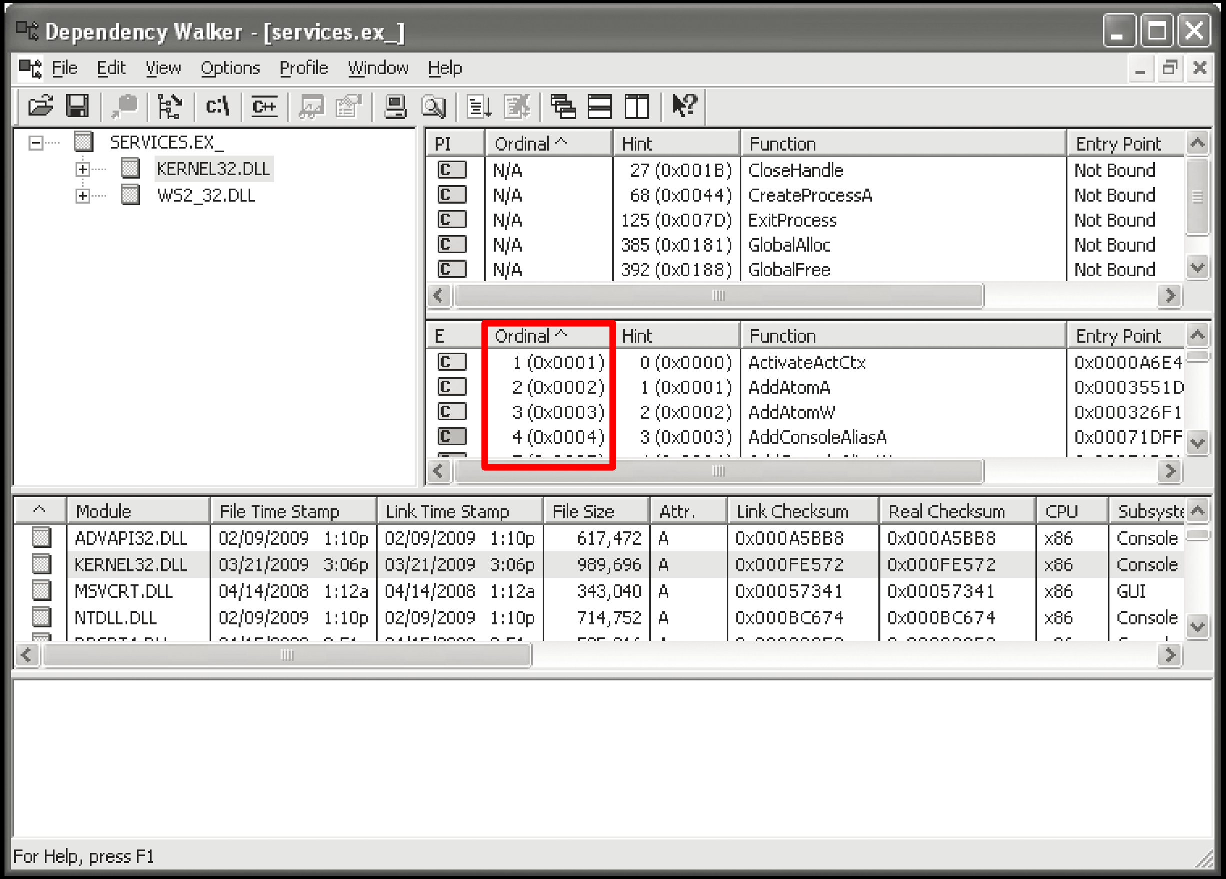
Viewing IMAGE_FILE_HEADER

PEView: Shown Content

- `Image_DOS_Header` & `MS-DOS_Stub program`: largely deprecated
- `IMAGE_NT_HEADERS`:
  - `Signature`: Always the same and can be ignored
  - `Image_FILE_HEADER`: Timestamp
  - `Image_OPTIONAL_HEADER`: A console or GUI program
- `IMAGE_SECTION_HEADERS`: section headers, including how much RAM & raw disk sizes the file requires
- Sections

Dependency Walker

- www.dependencywalker.com
  - Lists the dynamically linked functions in an executable



Dependency Walker - [services.ex_]
File Edit View Options Profile Window Help
SERVICES.EX_
KERNEL32.DLL
WS2_32.DLL
For Help, press F1

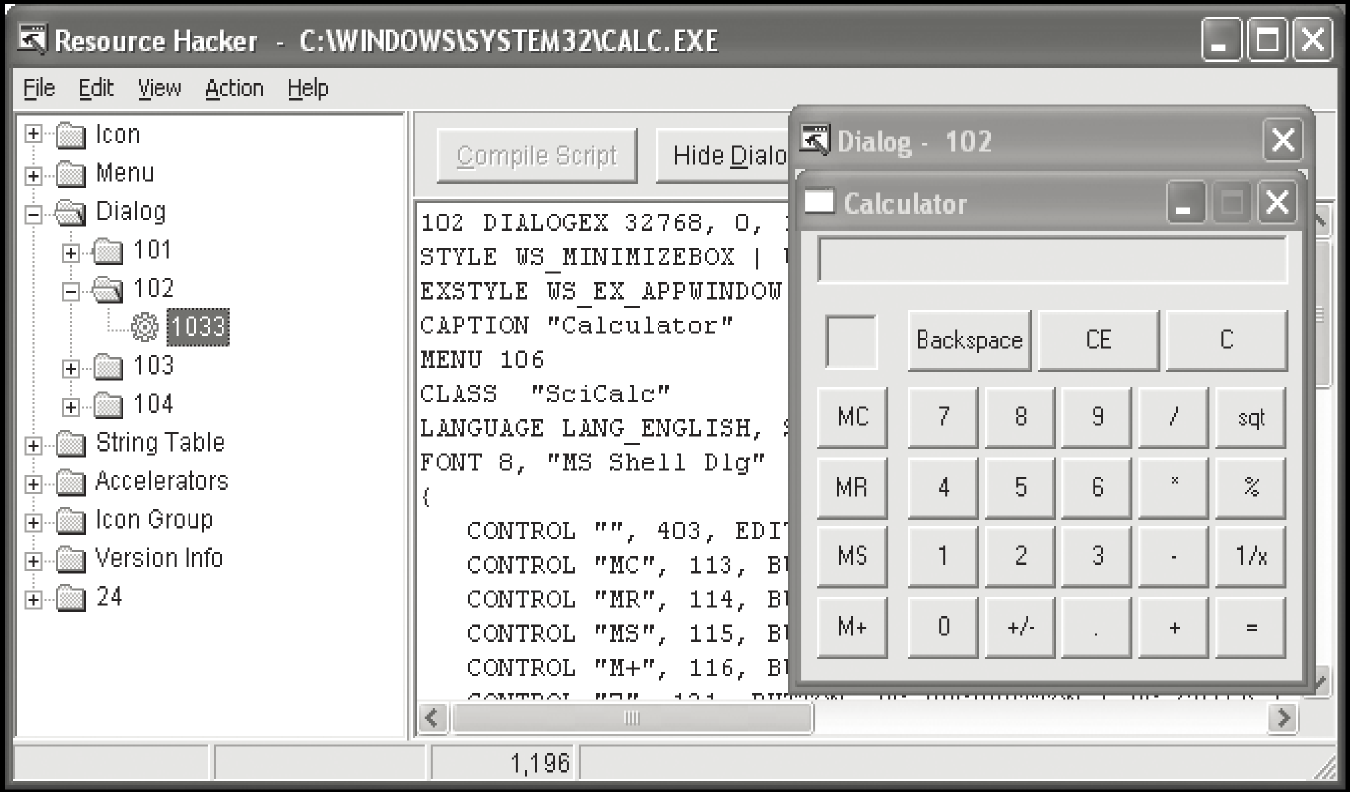
PI	Ordinal ^	Hint	Function	Entry Point
C	N/A	27 (0x001B)	CloseHandle	Not Bound
C	N/A	68 (0x0044)	CreateProcessA	Not Bound
C	N/A	125 (0x007D)	ExitProcess	Not Bound
C	N/A	385 (0x0181)	GlobalAlloc	Not Bound
C	N/A	392 (0x0188)	GlobalFree	Not Bound

E	Ordinal ^	Hint	Function	Entry Point
C	1 (0x0001)	0 (0x0000)	ActivateActCtx	0x0000A6E4
C	2 (0x0002)	1 (0x0001)	AddAtomA	0x0003551D
C	3 (0x0003)	2 (0x0002)	AddAtomW	0x000326F1
C	4 (0x0004)	3 (0x0003)	AddConsoleAliasA	0x00071DFF

^	Module	File Time Stamp	Link Time Stamp	File Size	Attr.	Link Checksum	Real Checksum	CPU	Subsyste
	ADVAPI32.DLL	02/09/2009 1:10p	02/09/2009 1:10p	617,472	A	0x000A5BB8	0x000A5BB8	x86	Console
	KERNEL32.DLL	03/21/2009 3:06p	03/21/2009 3:06p	989,696	A	0x000FE572	0x000FE572	x86	Console
	MSVCRT.DLL	04/14/2008 1:12a	04/14/2008 1:12a	343,040	A	0x00057341	0x00057341	x86	GUI
	NTDLL.DLL	02/09/2009 1:10p	02/09/2009 1:10p	714,752	A	0x000BC674	0x000BC674	x86	Console

Resource Hacker

- Allows us to view the `.rsrc` section
- Resource Hacker
  - www.angusj.com
- CFF Explorer can also be used



```
102 DIALOGEX 32768, 0,
STYLE WS_MINIMIZEBOX |
EXSTYLE WS_EX_APPWINDOW
CAPTION "Calculator"
MENU 106
CLASS  "SciCalc"
LANGUAGE LANG_ENGLISH,
FONT 8, "MS Shell Dlg"
{
   CONTROL "", 403, EDI
   CONTROL "MC", 113, B
   CONTROL "MR", 114, B
   CONTROL "MS", 115, B
   CONTROL "M+", 116, B
```

Basic Static Analysis: A Good Start

- Helps a lot!
- Provides many leads
- But it’s not enough in itself
- Can come back to these techniques later

- Always start with Basic Static Analysis techniques!!!
  - If you don’t, you may miss something obvious

?