

Advanced Function Obfuscation

Implementing Your Own Custom win32 API functions

No Obfuscation

Function Obfuscation

Function Obfuscation + String
Parameter Encryption

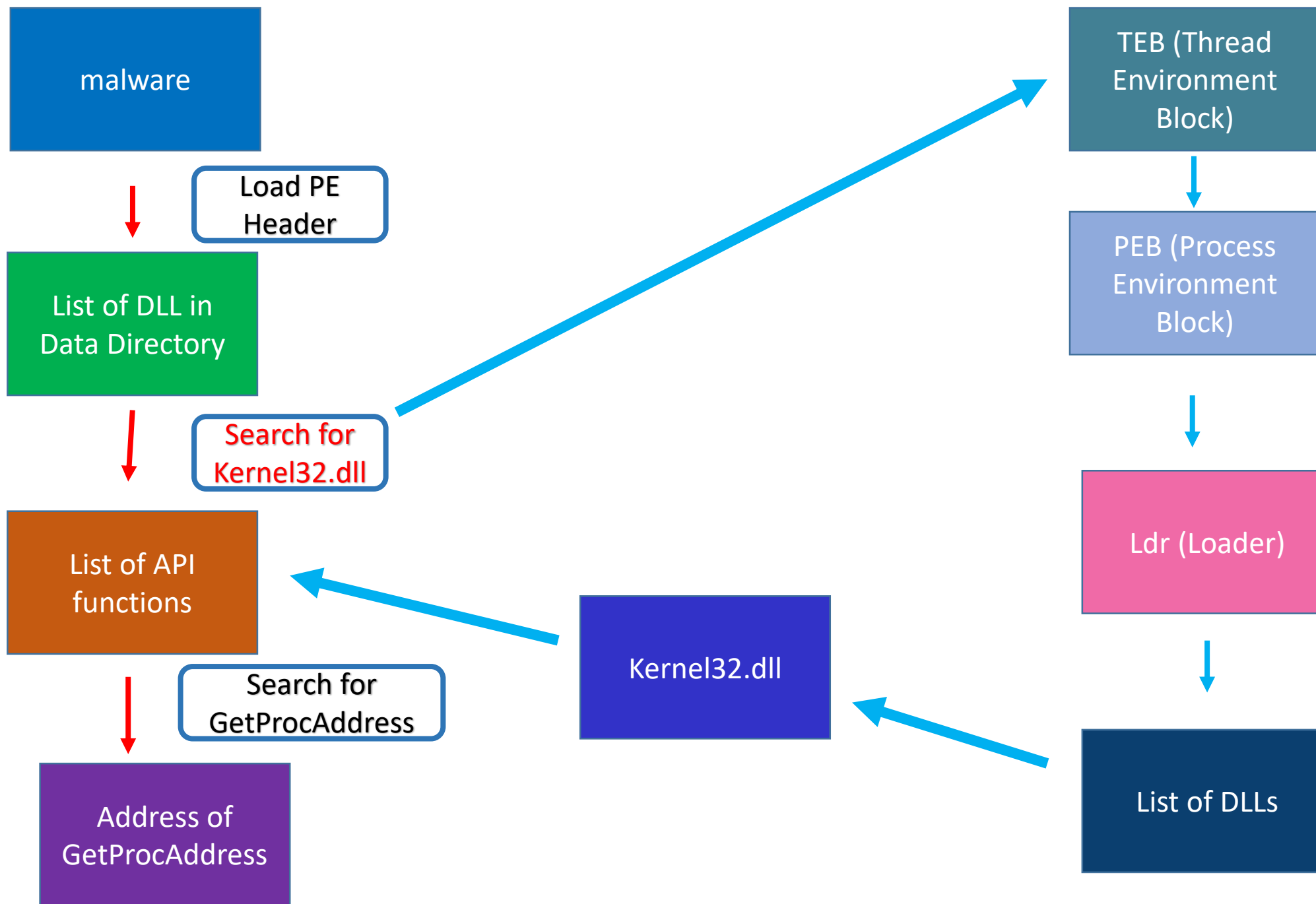
Custom API Function Obfuscation +
String Parameter Encryption

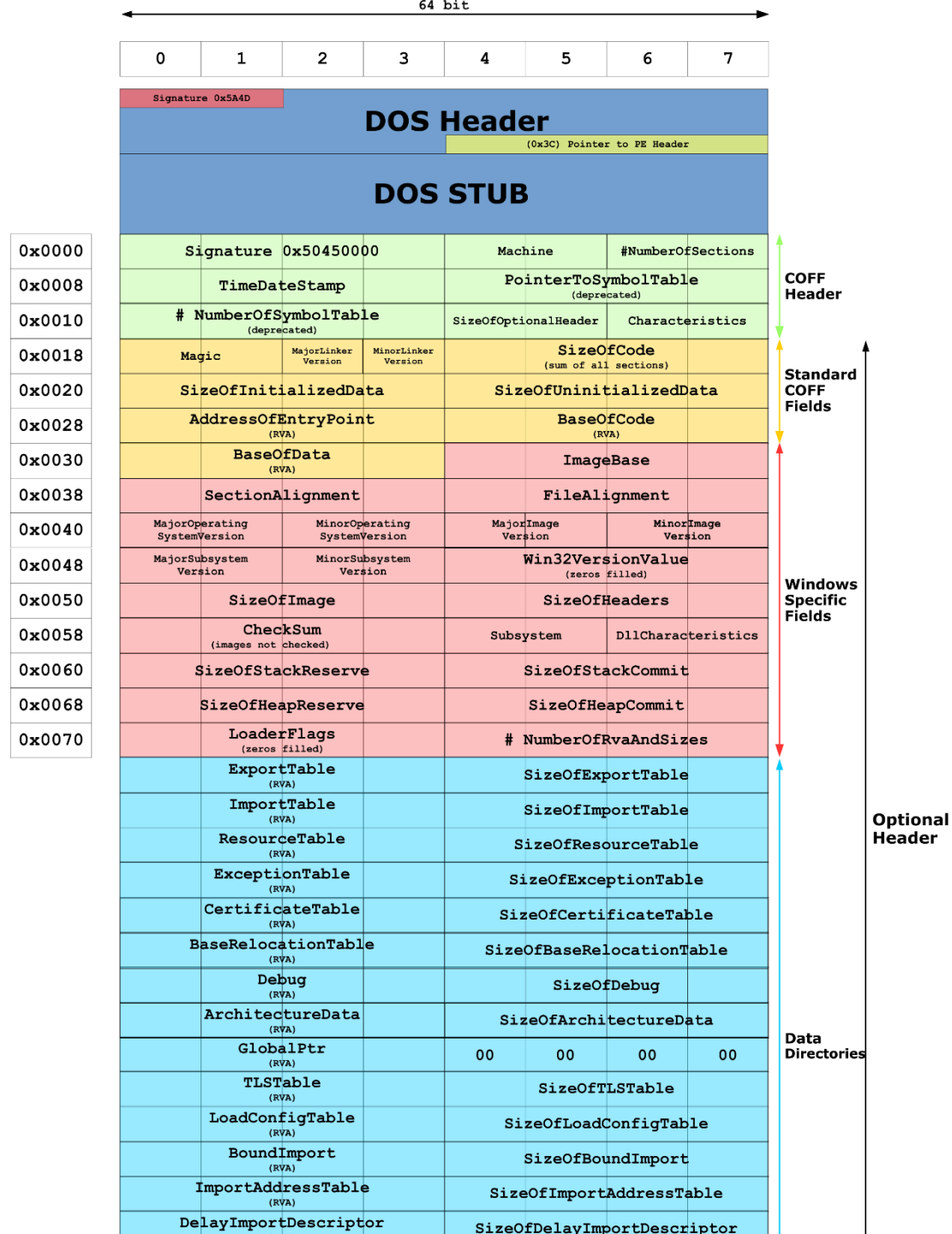
VirtualAlloc()

pVirtualAlloc = GetProcAddress(GetModuleHandle("Kernel32.dll"), "VirtualAlloc")

pVirtualAlloc = GetProcAddress(GetModuleHandle("Kernel32.dll"), strVirtualAlloc)

pVirtualAlloc = myGetProcAddress(myGetModuleHandle("Kernel32.dll"), strVirtualAlloc)





winnt.h

PE Header of Kernel32.dll (Optional Header Tab)

The image shows a side-by-side comparison of a C header file and a PE file's internal structure.

Left Panel (winnt.h): The code defines the `IMAGE_EXPORT_DIRECTORY` structure. A red box highlights the following fields:

```
typedef struct _IMAGE_EXPORT_DIRECTORY {
    DWORD Characteristics;
    DWORD TimeDateStamp;
    WORD MajorVersion;
    WORD MinorVersion;
    DWORD Name;
    DWORD Base;
    DWORD NumberOfFunctions;
    DWORD NumberOfNames;
    DWORD AddressOfFunctions; // RVA from base of image
    DWORD AddressOfNames;    // RVA from base of image
    DWORD AddressOfNameOrdinals; // RVA from base of image
} IMAGE_EXPORT_DIRECTORY, *PIMAGE_EXPORT_DIRECTORY;
```

Right Panel (PE-bear v0.5.0): The PE viewer shows the structure of `kernel32.dll`. The **Optional Header** tab is selected. The **Data Directory** table at the bottom lists various directories. A red box highlights the **Export Directory** entry, which is pointed to by a red arrow from the `IMAGE_EXPORT_DIRECTORY` structure in the header file.

Offset	Name	Value	Value
138	Size of Image	11F000	
13C	Size of Headers	800	
140	Checksum	123DF0	
144	Subsystem	3	Windows console
146	DLL Characteristics	140	
		40	DLL can move
		100	Image is NX compatible
148	Size of Stack Reserve	40000	
150	Size of Stack Commit	1000	
158	Size of Heap Reserve	100000	
160	Size of Heap Commit	1000	
168	Loader Flags	0	
16C	Number of RVAs and Sizes	10	
	Data Directory	Address	Size
170	Export Directory	9FFFC	ACF7
178	Import Directory	F8/6C	1F4
180	Resource Directory	116000	528
188	Exception Directory	10C000	9570
190	Security Directory	0	0
198	Base Relocation Table	117000	7ACC
1A0	Debug Directory	9BC58	38
1A8	Architecture Specific Data	0	0
1B0	RVA of GlobalPtr	0	0
1B8	TLS Directory	0	0
1C0	Load Configuration Directory	0	0
1C8	Bound Import Directory in headers	2E0	408
1D0	Import Address Table	9C000	1C98
1D8	Delay Load Import Descriptors	0	0

winnt.h

PE Header of Kernel32.dll (Exports Tab)

The image displays two side-by-side windows. The left window is a Notepad++ editor showing the `winnt.h` header file. The right window is the PE-bear v0.5.0 application showing the PE Header of `kernel32.dll` in the Exports tab. A red box highlights the `IMAGE_EXPORT_DIRECTORY` structure in `winnt.h`, and a red arrow points from this box to the Exports tab in PE-bear.

winnt.h Header File (Red Boxed Section):

```
#define IMAGE_SIZEOF_ARCHIVE_MEMBER_HDR 60

//
// DLL support.
//

//
// Export Format
//

//@[comment("MVI_tracked")]
typedef struct _IMAGE_EXPORT_DIRECTORY {
    DWORD Characteristics;
    DWORD TimeDateStamp;
    WORD MajorVersion;
    WORD MinorVersion;
    DWORD Name;
    DWORD Base;
    DWORD NumberOfFunctions;
    DWORD NumberOfNames;
    DWORD AddressOfFunctions; // RVA from base of image
    DWORD AddressOfNames; // RVA from base of image
    DWORD AddressOfNameOrdinals; // RVA from base of image
} IMAGE_EXPORT_DIRECTORY, *PIMAGE_EXPORT_DIRECTORY;

//
// Import Format
//

//@[comment("MVI_tracked")]
typedef struct _IMAGE_IMPORT_BY_NAME {
    WORD Hint;
    CHAR Name[1];
} IMAGE_IMPORT_BY_NAME, *PIMAGE_IMPORT_BY_NAME;
```

PE-bear v0.5.0 [C:/Windows/System32/kernel32.dll]

Exports Tab:

Offset	Name	Value	Meaning
9F5FC	Characteristics	0	
9F600	TimeDateStamp	554D6C15	Saturday, 09.05.2015 02:08:21 UTC
9F604	MajorVersion	0	
9F606	MinorVersion	0	
9F608	Name	A3684	KERNEL32.dll
9F60C	Base	1	
9F610	NumberOfFunctions	570	
9F614	NumberOfNames	570	
9F618	AddressOfFunctions	A0024	
9F61C	AddressOfNames	A15E4	
9F620	AddressOfNameOrdinals	A2BA4	

Exported Functions [1392 entries]

Offset	Ordinal	Function RVA	Name RVA	Name	Forwarder
9F624	1	AA1C0	A3691	AcquireSRWLoc...	NTDLL.RtlAcquireSRWLockExclus
9F628	2	AA1E1	A36A9	AcquireSRWLoc...	NTDLL.RtlAcquireSRWLockShare
9F62C	3	3C80	A36BE	ActivateActCtx	
9F630	4	66A10	A36CD	AddAtomA	
9F634	5	669B0	A36D6	AddAtomW	
9F638	6	6AC80	A36DF	AddConsoleAli...	
9F63C	7	6ACF0	A36F0	AddConsoleAli...	
9F640	8	AA1FF	A3701	AddDllDirectory	api-ms-win-core-libraryloader-l1
9F644	9	4B690	A3711	AddIntegrityLa...	
9F648	A	95600	A3737	AddLocalAltern...	
9F64C	B	8FAD0	A3756	AddLocalAltern...	
9F650	C	48AC0	A3775	AddRefActCtx	

Thank you