

information used by the code

The diagram illustrates the process of loading user32.dll into memory, showing how imports are resolved and how strings are loaded.

Imports structures

Address	Name	Hint
0x203c		0 ^{INT*}
0x2078	kernel32.dll	0 ^{ExitProcess} <small>Hint_Name</small>
0x2068		0 ^{IAT*}
0x2044		0 ^{INT*}
0x2085	user32.dll	0 ^{MessageBoxA} <small>Hint_Name</small>
0x2070		0 ^{IAT*}

All addresses here are RVA's.

Consequences

after loading,
 0x402068 will point to kernel32.dll's ExitProcess
 0x402070 will point to user32.dll's MessageBoxA

Strings

a simple PE executable\Hello world!\n

This is the whole file, however, most PE files contain more elements. Explanations are simplified, for conciseness.

version 1, 3rd May 2012

INT Import Name Table
Null-terminated list of pointers to Hint, Name structures

IAT Import Address Table
Null-terminated list of pointers
On file it is a copy of the INT
After loading it points to the imported APIs

HINT
Index in the exports table of a DLL to be imported
Not required but provides a speed-up by reducing look-ups