

注册表注入步骤：

下面示例过程需要在Windows 32位或Windows XP下
测试的DLL文件源码RegeditInjection.cpp

```
// RegeditInjection.dll

#include <windows.h>
#include <tchar.h>


#define DEF_PROCESS_NAME "cmd.exe" // 目标进程 cmd.exe

BOOL WINAPI DllMain(HINSTANCE hinstDll, DWORD dwReason, LPVOID lpvRevered) {
    char szPath[MAX_PATH] = {0, };
    char *p = NULL;

    GetModuleFileNameA(NULL, szPath, MAX_PATH);
    p = strrchr(szPath, '\\');

    switch( dwReason ) {
        case DLL_PROCESS_ATTACH:
            if( !_stricmp(p + 1, DEF_PROCESS_NAME) )
                MessageBox(NULL, TEXT("Hello cmd!!!"), TEXT("info"), MB_OK); // 被进程加载时弹出MessageBox("Dll Inject Success!!!")
            break;
        case DLL_PROCESS_DETACH:
            if( !_stricmp(p + 1, DEF_PROCESS_NAME) )
                MessageBox(NULL, TEXT("Goodbye cmd!!!"), TEXT("info"), MB_OK); // 被进程卸载时弹出MessageBox("Dll unInject Ok!!!")
            break;
    }
    return TRUE;
}
```

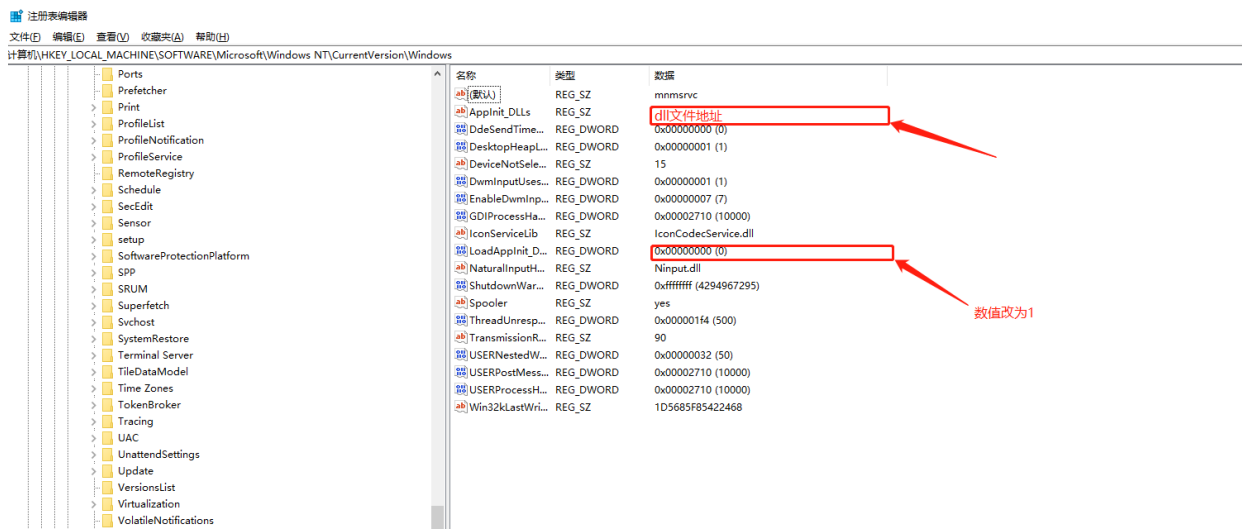
1.使用编译器编译成DLL文件

 RegeditInjection.dll	2019/10/8 19:12	应用程序扩展	36 KB
--	-----------------	--------	-------

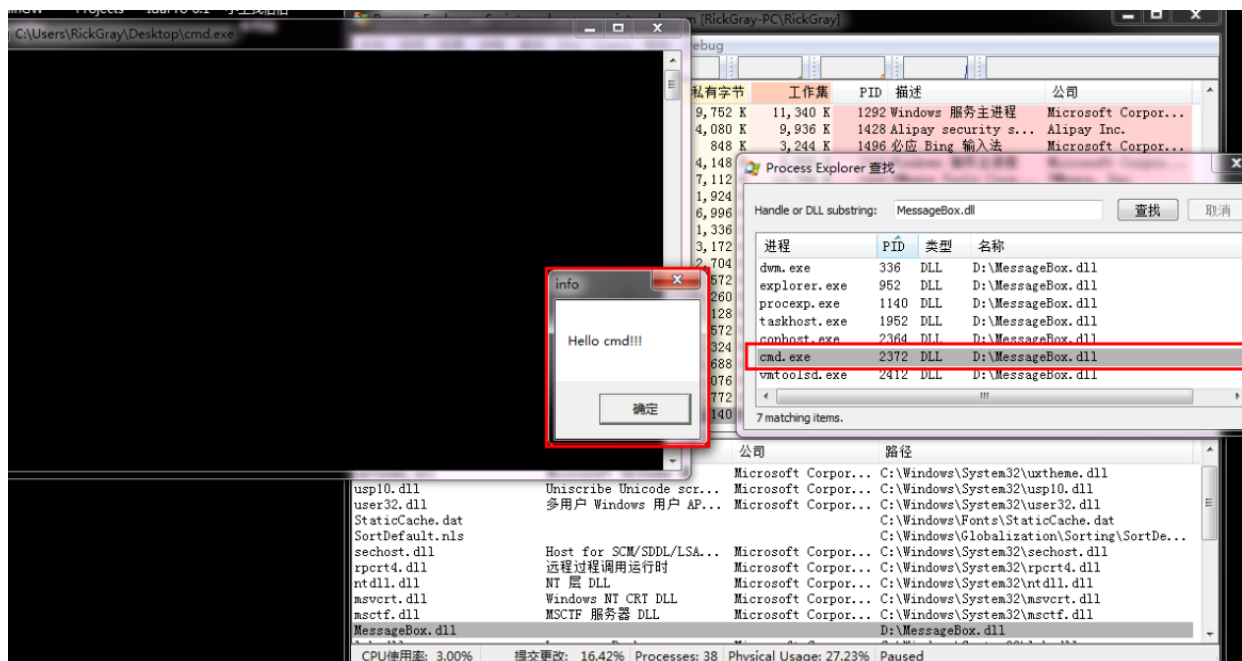
2.打开regedit.exe，进入如下路径：

**HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Windows**

编辑修改AppInit_DLLs表项的值为我们编译的RegeditInjection.dll所在的路径地址



3.注册表项修改完毕后，**重启系统**，使修改生效。重启完毕后，我们使用**Process Explorer**查看**RegeditInjection.dll**是否被注入进程。



4.从上图红色框框所标识的部分来看，运行cmd.exe时因为加载了user32.dll，所以也同时加载了我们自己写的MessageBox.dll，在DllMain()运行时，检测到当前进程为“cmd.exe”因此弹出了MessageBox()，说明注册表DLL注入成功。
若我们关闭cmd.exe，会弹出如下窗口

