2021/02/23_PE_第6课_导入表的应用

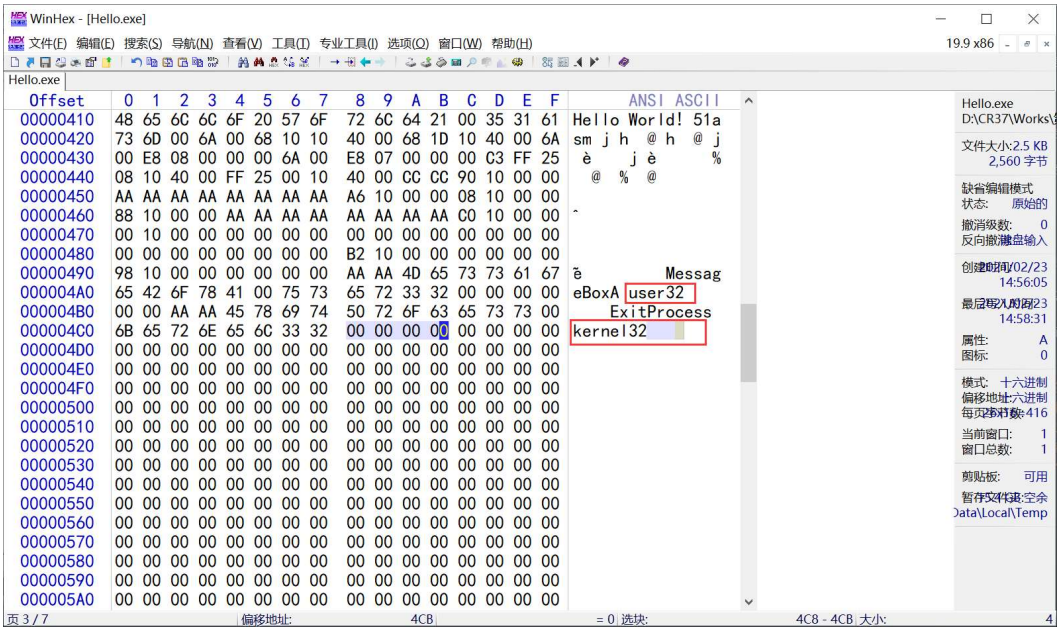
笔记本： PE
创建时间： 2021/2/23 星期二 10:06
作者： ileemi

地址导入表可有可无，但是操作系统会参考地址导入表，对于一个PE的可执行文件来说，最重要的就是导入表（必须存在）。获取地址导入表应该从导入表中获取，不应该直接从去获取地址导入表。

导入表的应用

- 反调试
- 导入表注入
- 免杀
- IAT HOOK

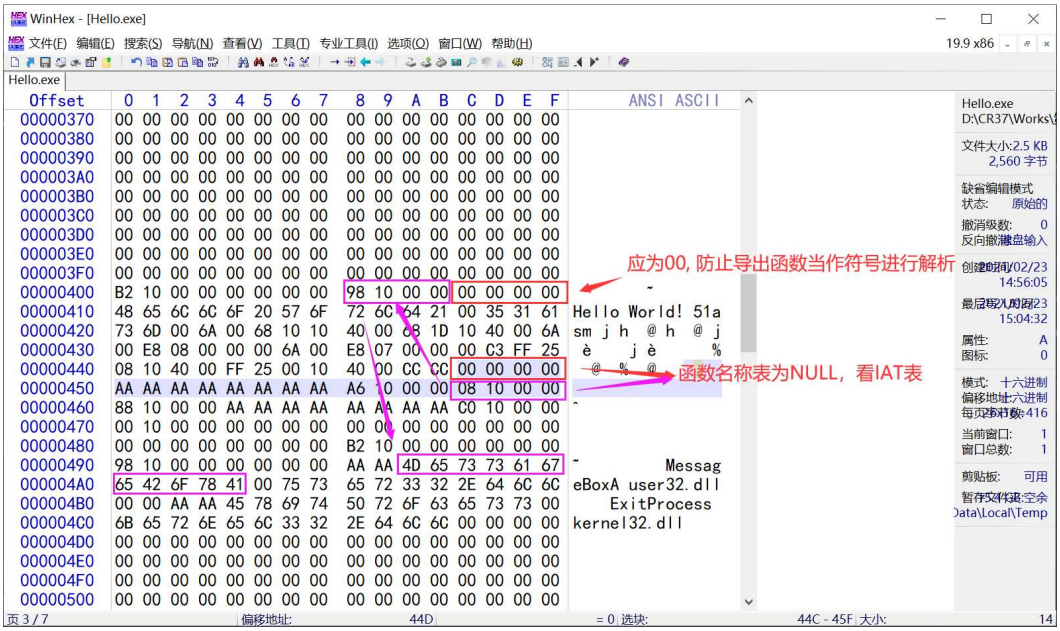
> 可执行程序中导出函数所在的库名称可以不添加 ".dll" 后缀，程序运行的时候操作系统会自动添加。



导入表的函数名称表可以为空

导入表的成员1，为函数名称表（可以为空），可用于反调试，但是操作系统会读取导入表的成员5（IAT表），将其指向的地址上的值当作函数名去解析（将函数名的地址回填到IAT表指向的地址上）。

当通过导入表的函数名称表解析导出函数时，且导入表的函数名称表为空时就需要通过导入表的成员5（IAT表）去解析。当作函数名称表去使用（获取函数名）。



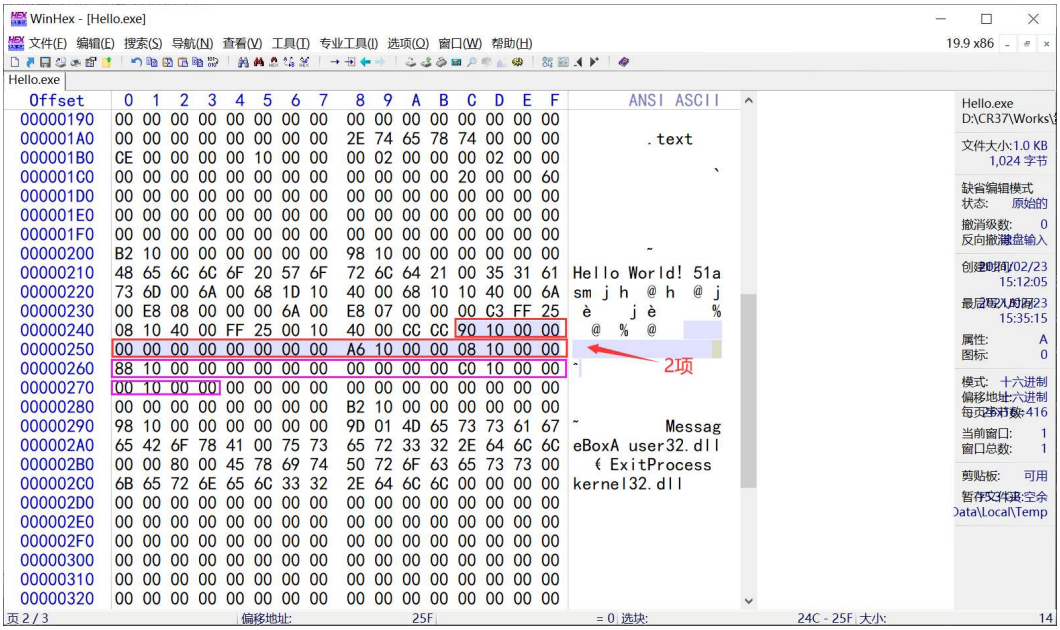
导入表注入

直接修改导入表中的参数4（库的名称），可实现代码注入。但是需要伪造一个对应的 ".dll" 文件。

> 缺点：直接修改可执行文件中API所使用的库名，会导致伪造的动态库不好编写（源库中存在很多导出函数的情况下）。

使用远程线程进行代码注入容易被杀毒软件拦截。但是通过修改导入表进行代码注入，杀毒软件就不容易检测到。

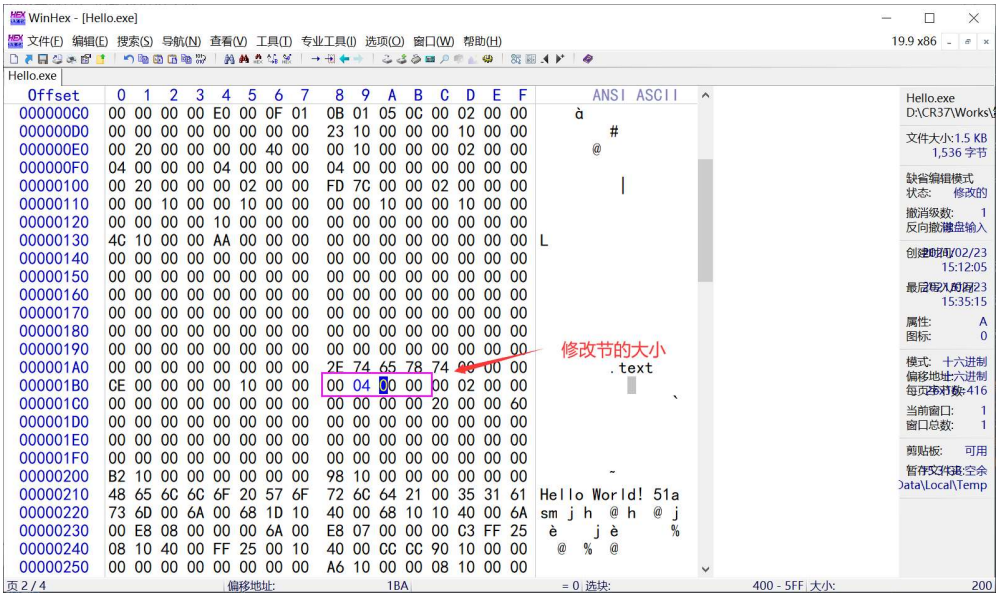
在可执行文件原有的导入表项后可添加一个新的导入表项，但是添加新的导入表项会受对应的内存空间大小的影响（话需要改文件内部的偏移值）。



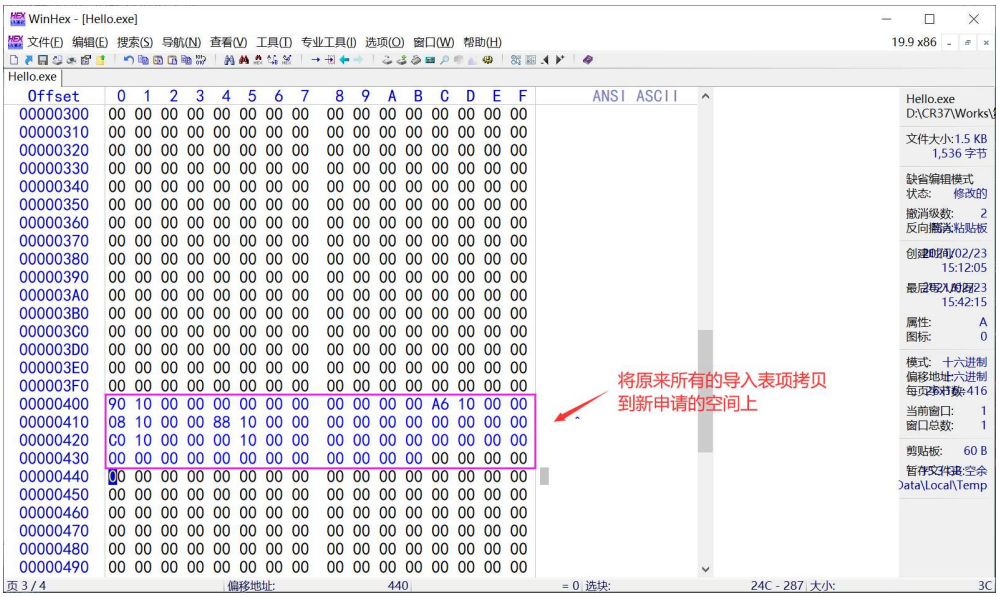
> 可行办法就是，在添加一个节，将已存在的导入表项移动到新的节中，并添加新的导入表项。或者扩展原来节的大小（并将原来所有的导入表项都拷贝到新申请的内存空间上）。导入表注入的隐秘性比较高。

操作步骤：

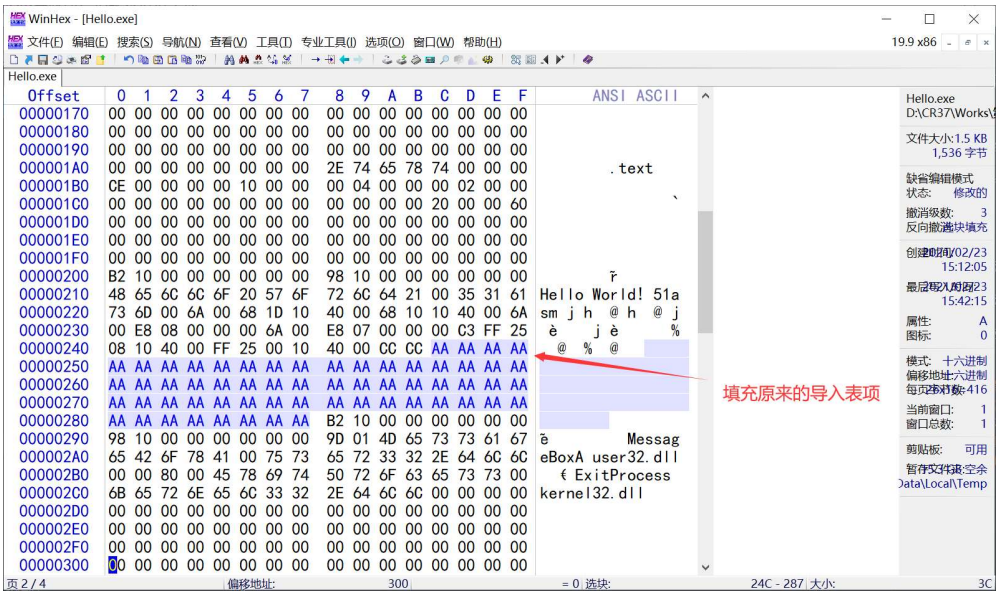
- 修改节的大小：



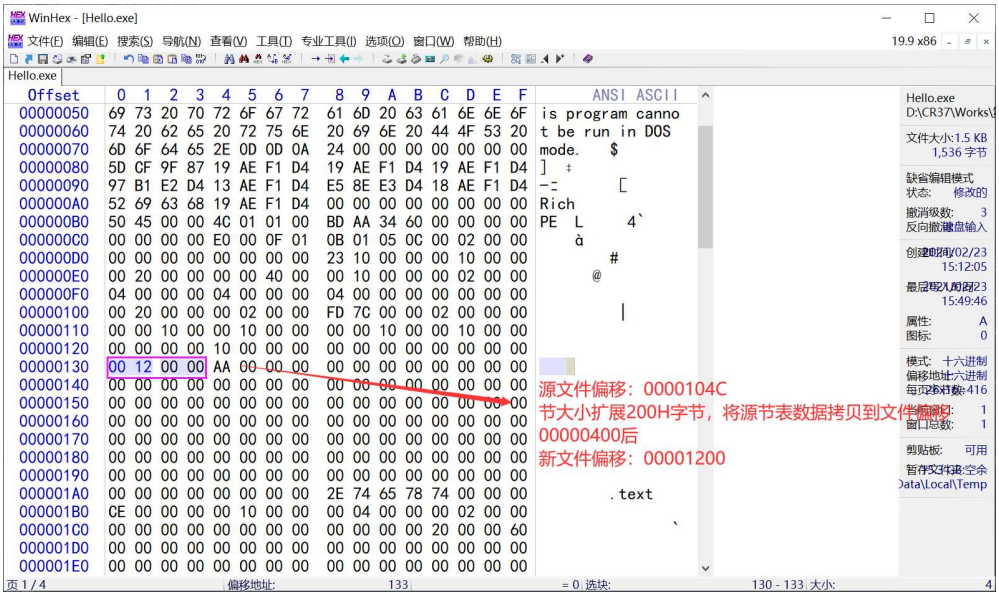
- 将原来所有的导入表项都拷贝到新申请的内存空间上：



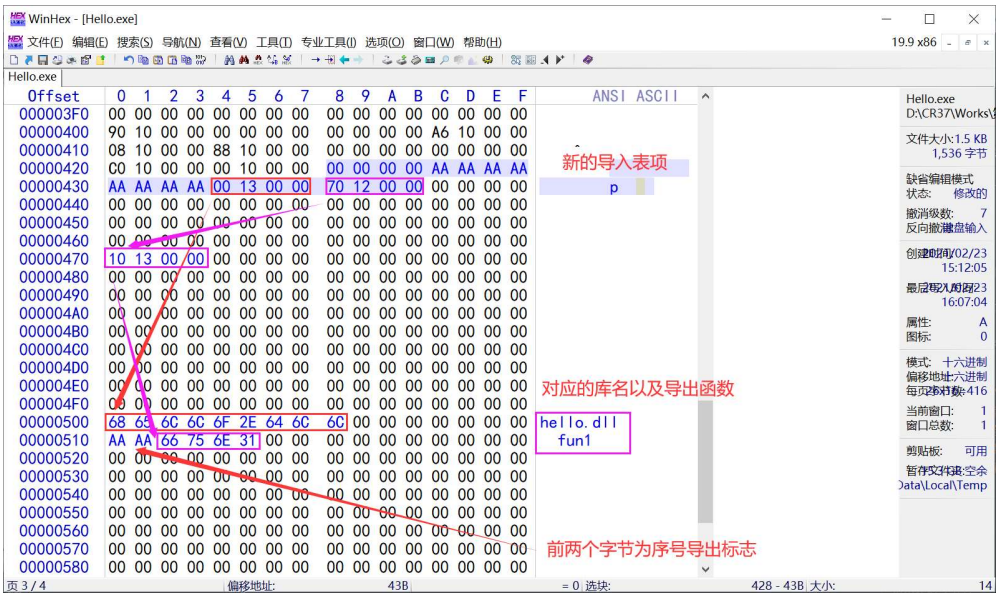
- 移动后，原来的导入表项就可以将其数据进行抹除。



- 修改指向导入表项的文件偏移：



- 添加新的表项（添加对应的库名以及对应的导出函数名）：



- 创建一个动态库并导出对应的函数，之后双击可执行程序即可。

缺点：怕数字签名检测，可对可执行文件中的模块进行绕过。

IAT Hook

遍历导入表找到目标API在文件中的偏移地址，当可执行文件运行的时候，将偏移地址+程序的基址（00400000），得到的内存地址上会保存目标API的地址，将其地址进行保存，指向新的API，新的API调用后，在调回到源API的地址完成API Hook。不在需要重定位。

相对于内敛Hook的缺点：当动态加载的API，该方法就无法获取。

对抗内存dump

软件的代码通过网络进行传输（传输一个PE文件），模拟操作系统加载PE文件，最后跳转到 OEP。

LoadPE

通过可执行程序去加载指定的可执行程序到内存中并执行目标进程。相当于在目标进程外套上一个壳。

加外部的可执行程序的模块基址移动到指定的外置，方便不合目标进程冲突。可使用 link 的 "base" 的编译选项进行指明。但是外部程序加载完成后，内存中的数据已经没有用，可以被覆盖