**《软件逆向工程》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | | **2021级信息安全02** | | | **姓名** | **于璐** |
| **实验题目** | **PE壳设计与验证** | | | | | |
| **实验时间** | **2021/11/28** | | **实验地点** | **DS3401** | | |
| **实验成绩** |  | | **实验性质** | **☑验证性 □设计性 □综合性** | | |
| 教师评价：  ☑算法/实验过程正确； ☑源程序/实验内容提交； ☑程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； ☑报告规范；  其他：  评价教师签名： 杨吉云 | | | | | | |
| 1. 实验目的 2. 验证PE文件结构构建 3. 验证PE文件壳的基本原理 | | | | | | |
| 1. 实验项目内容 2. 实现简单的跳转壳，在PE文件中添加新节，在新节其中加入跳转至原入口的指令，实现对原程序的启动。 3. 实现PE文件加密壳，对原程序代码节内容异或运算，在原程序中加入新节实现对原代码节内容解密并启动运行。 | | | | | | |
| 1. 实验过程或算法（源程序） 2. **实现跳转壳**   首先修改节表的数量,增加一个新的节，我们将numberOfSection的大小油3改为4。并将它命名为.newsec意思为新的节,然后按照他的上一个节即.data节的偏移和大小,设置新的节的偏移,我们只要定义新的节的大小为最小对齐粒度的1倍即可。  1  插入新的节  将新增节表项的VirtualSize和SizeOfRawData都设置为200h。因为前一节即.data节的VirtualAddress为3000h且.data节只会占据一个内存页，所以新增节的VirtualAddress为4000h。同理.data节的PointerToRawData为800h且.data节只占用了一个扇区大小，所以新增节的PointerToRawData为A00h。最后将Characteristics字段设置为 60000020h（修改为和.text节一样的），定义该新增节是可读、可执行且包含代码的节。其余字段默认不修改即可。  2  修改新的节的属性  3  添加新增节的内容  前两步只是增加了一个节表项，文件中还没有该节的数据，所以需要增加该节的数据。在010editor中选择 Edit→Insert/Overwrite→Insert Bytes即可往PE文件中插入字节数据，如下图所示。  4  查看新增节的内容，观察到文件中从A00h开始填充了200h字节的0  5  查看新节  **修改SizeOfImage**  新增节显然会使PE加载到内存时所需要的空间增大，由于之前设置新增节的大小为200h个字节，所以加载到内存只需要1页的空间即1000h个字节，所以将SizeOfImage增加1000h。原来的PE文件有文件头、.text、.rdata、.data四个部分，分别占用1页内存，SizeOfImage大小为4000h，直接将其修改为5000h即可。  6  至此新增节完成，此时运行exe文件，程序仍然可以运行，显示出弹窗如下图所示：  7  **实现跳转壳**  跳转壳需要写入到前一步新增的节中，要让程序启动时首先执行壳中的代码需要修改PE头中的AddressOfEnterPoint字段为新增节的起始RVA地址，即让程序加载到内存时，EIP的值为（404000h），即新节的地址，而不是.text节。即同之前设置节表项时的VirtualAddress（4000h）值。于此同时将原来的AddressOfEntryPoint（**1020h**）值保存下来，写跳转壳代码时使用。  8  此时在OD中运行程序，观察到EIP的值为404000h，即程序入口点修改成功。如下图所示：  9  **实现跳转：**  在新增节的内存地址处进行反汇编，写入代码**jmp 401020**，即跳转到原来的**程序执行入口点**。  10  **调试**  按下F7,观察到程序来到.text节处，EIP的值从404000h到401020h，跳转壳实现成功。  11  保存加了跳转壳的文件为lab3\_jmp,运行程序，messagebox出现，实验成功：  12   1. **实现加密壳**   增加节的步骤和实现跳转壳一样。然后：  在010editor中对.text节进行异或加密，选中.text节的所有数据并点击Tools→Hex Operations→Binary Xor...：将.text节中的每个字节都和0x12进行异或。  1  异或之后，在010editor中观察.text节的内容，观察到写入成功：  2  **同任务一进行新增节操作并修改AddressOfEntryPoint为新增节的起始RVA，并将.text节的属性设置为可写**。在ollyDbg中写入汇编jmp 401020，然后按下F7，此时并不保存，观察到原本来的程序执行入口处全是乱码：  3  在Ollydbg中写入异或解密的程序：  5  **代码解释：**  这里我们四个字节四个字节的写入，而不是一个字节一个字节的写入。  首先将ecx和ebx的值都设置为1000，然后想办法将ebx的值指向内存中text节的起始RVA 401000h，将ECX的值设置为text节的最后RVA 402000h，然后进行循环，将【ebx】的值取出来和0x12121212进行异或，然后写入到内存里面，之后，将ebx的值+4，判断循环结束的条件为ebx的值是否小于ecx，如果小于则进行循环，否则解密完成，执行jmp 401020h，程序跳转到，text节继续执行代码。  然后将修改保存，选中所有的修改进行保存。新的另存为lab4\_2\_myxor，点击运行，观察到出现弹窗，程序执行正确，加密壳实现！  4 | | | | | | |
| 1. 实验结果及分析和（或）源程序调试过程   1.实验结果分析  1.1实验一、用OD打开跳转壳  222  并且运行lab3\_jmp文件，可以正常加载出弹窗  12  1.2实验二、用OD打开加密壳  111  并且运行加密壳文件可以正常打开弹窗  4  至此，实验完成。  2.实验心得和体会  在本节实验上，我学会了写加密壳和跳转壳，对PE文件的结构也有了新的理解，我认为在010editor下，可以更加清晰的观察和学习PE文件的结构，节的含义等，因为在这个软件下可以更好的显示出数据结构，同时我对节表项40个字节的含义有了更加深刻的了解，理解了他们的含义。 | | | | | | |