第三章 软件调试基础

1 PE文件格式

一般有可执行文件（exe&dll）

PE文件格式将可执行文件分成了若干个数据节，不同类型的资源被放在不同的节中

一般来说，有以下的数据节：

- rsrc：存放**程序的资源**，如图标、菜单等
- text：存放着二**进制的机器代码**
- idata：可执行文件所使用的动态链接库**等外来函数与文件的信息**，即输入表
- data：初始化的数据块

加壳

全称应该是可执行程序资源压缩，是保护文件的常用手段。加壳过的程序可以直接运行，但是不能查看源代码。要经过脱壳才可以查看源代码。

加壳可以很好的防止对程序的非法修改和静态反编译

加壳工具：压缩壳和加密壳

- 压缩壳的特点是减小软件体积大小，加密保护不是重点
- 加密壳种类比较多，不同的壳侧重点不同，一些壳单纯保护程序，另一些壳提供额外的功能，如提供注册机制、使用次数、时间限制等

2 虚拟内存

用户在用户模式运行；操作系统在内核模式运行，在内核模式可以访问所有的内存和硬件，使用所有的处理器指令。

物理内存&虚拟内存

一般用户模式下，看到的都是虚拟内存，程序进行虚地址到实地址的转换的过程我们称为**程序的再定位**

在运行PE文件时，操作系统会自动加载该文件到内存，并为其映射出4GB的虚拟存储空间，然后继续运行，这就形成了所谓的进程空间，在这个空间中定位的地址称为**虚拟内存地址（Virtual Address，VA）**。

我们在PE文件中看到的指令是相对于磁盘文件而言的，文件偏移的话，我们还需要知道这条指令在内存中所处的位置，就是虚拟内存地址；在调试的时候看到虚拟内存的话，我们也需要找回该指令的机器码

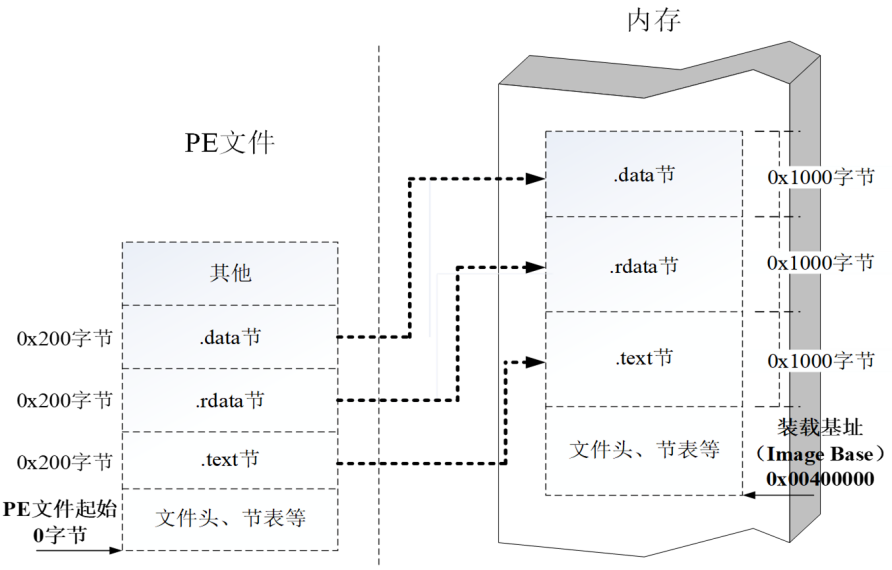
几个重要概念

- 相对虚拟地址：内存地址相对于映射基址的偏移量
- 文件偏移地址：数据在PE文件中的地址叫文件偏移地址，这是文件在磁盘上存放时相对于文件开头的偏移

- 装载基址：PE装入内存时的基地址。默认情况下，EXE文件在内存中的基地址是**0x00400000**，DLL文件是**0x10000000**
- 虚拟内存地址：PE文件中的指令被装入内存后的地址

虚拟内存地址、映射基址、相对虚拟内存地址三者之间有如下关系：

$$VA = Image\ Base + RVA$$



PE文件的数据节的大小永远是0x200的整数倍；内存中的节总是0x1000的整数倍

我们可以使用Lord PE来查看内存地址和数据在PE文件中的地址

3 调试分析工具

- OllyDbg——动态调试
- IDA PRO——逆向分析

4 软件破解示例

给一个简单的密码程序

```
#include <iostream>
using namespace std;
#define password "12345678"
bool verifyPwd(char * pwd)
{
    int flag;
    flag=strcmp(password, pwd);
    return flag==0;
}
void main()
{
    bool bFlag;
    char pwd[1024];
```

```
    printf("please input your password:\n");
    while (1)
    {
        scanf("%s",pwd);
        bFlag=verifyPwd(pwd);
        if (bFlag)
        {
            printf("passed\n");
            break;
        }else{
            printf("wrong password, please input again:\n");
        }
    }
}
```

我们需要破解debug模式下的exe程序

我们可以利用逻辑条件的修改、或者直接将判定密码的语句给置空就可以了