壳

保护PE文件（EXE、DLL）

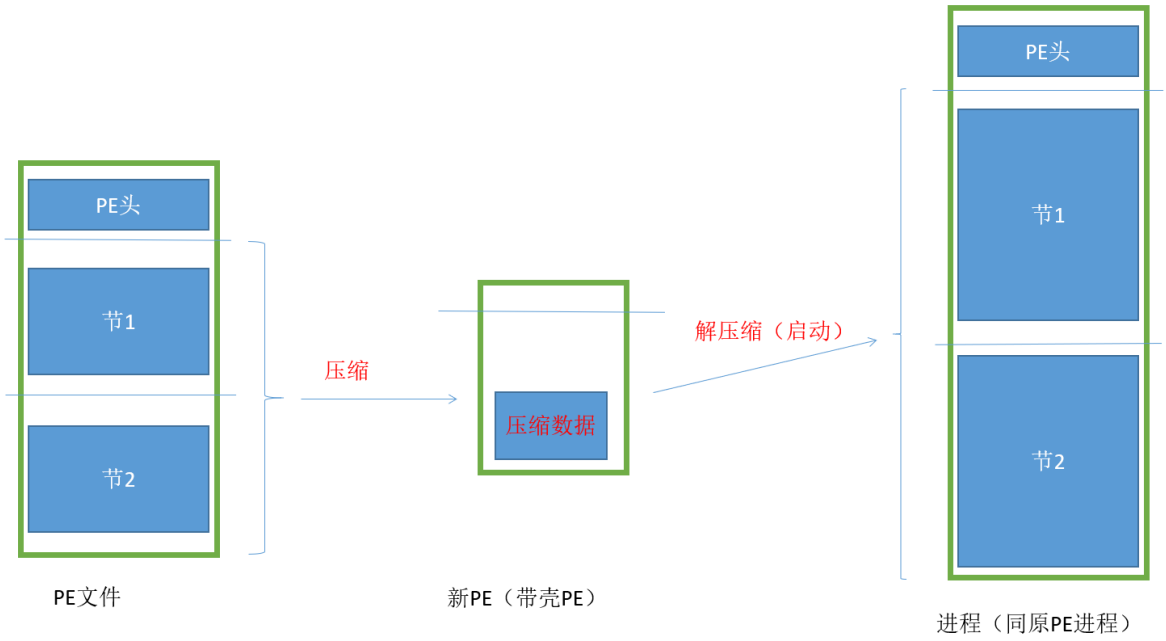
壳的分类

1. 压缩壳
    - 减少PE文件的体积，用的压缩算法
2. 加密壳
    - 防止PE文件被逆向，用的加密算法

压缩壳的原理

两个进程节区数据的分布和内容一样，那么两个进程的功能一样

空节：没有文件数据，只有内存大小



脱壳的步骤

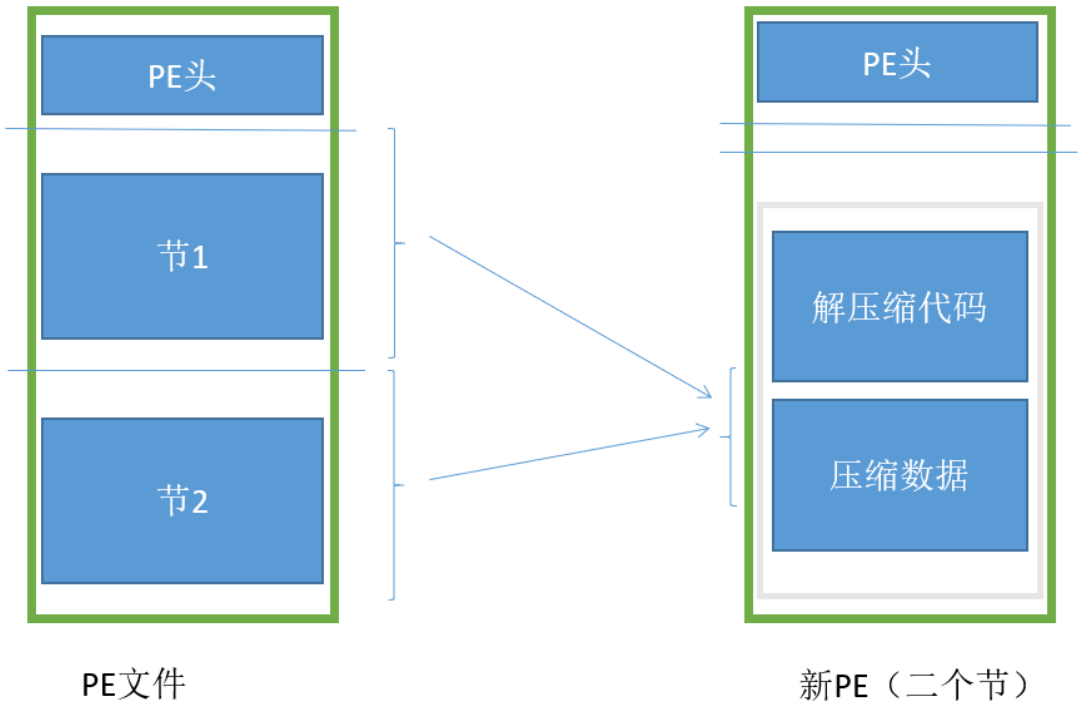
1. 查找OEP
    - 识别OEP：经验
    - 手法
        - ESP定律
        - API（分析每个环境的入口，在关键API上下断点等它来）
        - 单步跟踪（步过循环，只向下跳转）
        - 超长距离的流程转移（大`jmp`、大`call`、大`push + ret`）

        - 特征码（寻找未加壳的程序的特征码，在加壳程序中匹配寻找）
2. Dump
    - 32位程序在32位系统中Dump，64位程序放在64位系统中Dump
3. 修复PE（导入表）

压缩壳的实现

工程分为两个部分：

1. 加壳程序
    - 将原始PE的头部和各个节数据进行压缩
    - 在新PE中添加一个空节用于在执行过程中解压原始PE的数据
    - 另外需要给解压缩的代码额外准备一个节
    - 生成新的PE头部和各节区
    - 注意修改头部字段`NumberOfSections`、`AddressOfEntryPoint`、`SizeOfImage`等



2. 解压缩的代码
    - 地址无关代码（shellcode）：不能包含有绝对地址的指令
        - 第一种方法：汇编`call next, pop reg`
        - 第二种方法：C配合VS编译选项
            - 随机基址开启
            - `Release`版
            - 将入口点的`main`替换掉，链接选项 -> 高级 -> 入口点
            - 关闭GS安全检查，C/C++ -> 所有选项 -> 安全检查
            - 关闭优化
            - 关闭C++异常
            - 注意字符串的使用，通过一个字符一个字符的赋值
            - 注意API的调用，通过自己实现`MyLoadLibrary`拿到`LoadLibrary`