- fishhook
  - PIC位置代码独立
  - 共享缓存
    - 系统函数存放区域
  - MachO
    - 代码段只读
    - 数据段中的内容是可读可写的
  - 防护
    - 只能hook系统c函数，NSLog、getImp、setImp……
      - 原因：自定义为代码段只读，确定了函数地址，不需要动态绑定
      - 系统函数在公共区域，在调用的时候进行懒加载，绑定符号和共享区域的地址，懒加载部分在数据段
    - OC方法hook
      - 使用Runtime提供相关的方法交换接口，接口为C函数
    - 原理：使用fishhook重绑定系统函数，检测系统函数是否被调用
    - 实现：hook系统交换函数，监听该函数是否被调用，收集调用数据进行分析
      - method_exchangeImplementations
      - getImp、setImp
    - 修改machO文件中的字符串“cstring”–不好用
      - 可用于破解，修改原函数名称，再hook
  - 反防护
    - 注入代码保证在最前面，不会被检测到，注入的动态库都在+load前面。动态库注入本身就具有反防护能力
    - 反反防护
      - 自己做一个framework进行插入，工程中使用framework提供的交换方法进行交换或监控
      - 逆向工程的动态库只会插入到所有动态库末尾
      - 继续破解
        - 1、通过MachOView查看修改动态库可执行文件
        - 2、找到hook的函数，修改函数名称即“cstring”中的字段，修改data中的十六进制，修改保存，需要重新打开，value字段才会显示已经被修改了
        - 3、给动态库可执行文件执行权限，将修改过的framework复制到.app中的Frameworks文件下，替换之前的动态库即可
        - 4、重新运行则原防护代码变为无效