猿人学-objection基本使用

纯文本

```
frida
frida-tools
objection

pip install objection
pip install objection==xxxx

objection --help

./frida-server -l 0.0.0.0:6666
adb forward tcp:6666 tcp:6666  端口映射转发
objection -N -h device_ip -p 8888 -g xxxxx explore
objection -g xxxxx explore
objection -g packageName explore --startup-command 'android hooking watch xxx'
frida, env, help frida

cat .objection/objection.log
cat objection.log |grep -i http
objection -g com.android.settings explore -c "2.txt"



Memory 指令
    # 列举加载的modules，也就是so文件
    memory list modules
    memory list modules --json result.txt

    # 列举so文件的导出方法
    memory list exports xxx.so

    # dump所有内存
    memory dump all /tmp/dump

    # dump内存指定地址和大小
    memory dump from_base 0x130b4060 1024 /tmp/dump

    # 在内存搜索
    memory search "64 65 78 0a 30 33 35 00"
    memory search "99999999999" --string
    memory search "66 72 ?? ?? 61"

    # 修改内存
    memory write 0x130b4060 "99999999999" --string


android heap 指令
    //堆内存中搜索指定类的实例, 可以获取该类的实例id
    android heap search instances com.xx.xx.class

    //直接调用指定实例下的方法
    android heap execute [ins_id] [func_name]

    //自定义frida脚本, 执行实例的方法
    android heap execute [ins_id]

android 指令
    android root disable   //尝试关闭app的root检测
    android root simulate  //尝试模拟root环境
```

```
    android ui screenshot [image.png]    //截图
    android ui FLAG_SECURE false         //设置FLAG_SECURE权限

内存漫游
    Objection可以快速便捷地打印出内存
    中的各种类的相关信息，这对App快速定位有着无可比拟的优势，下面
    介绍几个常用命令。
    ①可以使用以下命令列出内存中的所有类。

    # android hooking list classes

    ②可以使用以下命令在内存中所有已加载的类中搜索包含特定关键
    词的类。
    # android hooking search classes
    这里搜索一下包含display关键词的类。
    # android hooking search classes display

    ③可以使用以下命令从内存中搜索所有包含关键词key的方法。
    # android hooking search methods <key>
    从上文中可以发现，内存中已加载的类高达6103个。它们的方法
    是类的个数的数倍，整个过程会相当耗时。这里展示搜索包含display
    # android hooking search methods display

    ④搜索到我们感兴趣的类后，可以使用以下命令查看关心的类的所
    有方法,例如，对android.hardware.display.DisplayManager这个类感
    兴趣
    # android hooking list class_methods android.hardware.display.DisplayManager

    ⑤以上介绍的都是最基础的一些Java类相关的内容。在Android
    中，正如笔者在第2章中介绍的，四大组件的相关内容是非常值得关注
    的，Objection在这方面也提供了支持，可以通过以下命令列出进程所
    有的activity（活动）。
    # android hooking list activities

    ⑥可以通过以下命令列出进程所有的service。
    # android hooking list services
    需要列出其他两个组件的信息时，只要将对应的地方更换为
    receivers和providers即可

hook 方式
    /*
        hook指定方法, 如果有重载会hook所有重载,如果有疑问可以看
        --dump-args : 打印参数
        --dump-backtrace : 打印调用栈
        --dump-return : 打印返回值
    */
    android hooking watch class_method com.xxx.xxx.methodName --dump-args --dump-backtrace --dump-return

    //hook指定类, 会打印该类下的所有调用
    android hooking watch class com.xxx.xxx

    //设置返回值(只支持bool类型)
    android hooking set return_value com.xxx.xxx.methodName false

activity 和 service 操作
    android hooking list activities                   //枚举activity
    android intent launch_activity [activity_class]   //启动activity
    android hooking list services                     //枚举services
    android intent launch_service [services_class]    //启动services

任务管理器
    jobs list            // 查看任务列表
    jobs kill [task_id]  // 关闭任务

关闭 app 的 ssl 校验
    android sslpinning disable
```

```
git clone https://github.com/hluwa/Wallbreaker  ~/.objection/plugins/Wallbreaker
git clone https://github.com/hluwa/Wallbreaker  C:\Users\admin\.objection\plugins\Wallbreaker
objection -g com.android.phone explore -P ~/.objection/plugins
plugin load xxxx
plugin wallbreaker classsearch <pattern>
plugin wallbreaker objectsearch <classname>
plugin wallbreaker classdump <classname> [--fullname]
plugin wallbreaker objectdump <handle> [--fullname]
```

可参考文章：

https://github.com/sensepost/objection/
https://www.52pojie.cn/forum.php?mod=viewthread&tid=1626964
https://saucer-man.com/information_security/911.html
https://www.cnblogs.com/my1127/p/16133663.html
https://www.anquanke.com/post/id/197657
https://www.cnblogs.com/ningskyer/articles/14611822.html

以上基本都是cv的。。。