猿人学-frida基础使用

纯文本

```
开发环境配置：
https://nodejs.org/zh-cn/download/
npm install @types/frida-gum

见过的较好的frida笔记：
https://kevinspider.github.io/frida/frida-hook-java/
https://kevinspider.github.io/frida/frida-hook-so/
https://eternalsakura13.com/2020/07/04/frida/

官方api：
https://frida.re/docs/javascript-api/

课程练习apk源码及下载地址
https://github.com/huaerxiela/LessonTest
https://github.com/huaerxiela/LessonTest/releases

优雅的frida-rpc
https://mp.weixin.qq.com/s/zu3jEd38NxlemBpfN3TnHw

查看手机架构
adb shell getprop ro.product.cpu.abi

Android逆向之分析基操
https://mp.weixin.qq.com/s/IovgsqprLYSnKsH61EP-EQ

细说So动态库的加载流程
https://bbs.pediy.com/thread-255674.htm

.init .initarray和JNIOnload
https://www.cnblogs.com/runope/p/14789784.html

Frida hook Java/Native与init_array 自吐最终方案
https://bbs.pediy.com/thread-267430.htm

dlopen -> CallConstructors -> init，init_array(反调试，检测，ollvm字符串加解密)
dlsym(JNI_ONLOAD)
```

```
对app自身so文件的自定义svc进行hook
1，so加载进内存，完成重定位
2，init，init_array之前

so dump之后修复：SOFIXER

//应用以32位在64位终端环境下运行
//adb install --abi armeabi-v7a <path to apk>



so分析(trace)
  ida-ins trace
  frida-stalker
    目前只支持arm64，why：arm32终将过去式，arm64才是主流
    sktrace：https://github.com/huaerxiela/sktrace.git
      课程环境：
      frida --version: 15.2.2
      frida-server --version: 15.2.2
      example：python sktrace.py -m attach -l libhello-jni.so -i 0x1CFF0 He
  unicorn-unidbg-kingtrace
```

纯文本

```
function searchInterface() {
    Java.perform(function () {
        Java.enumerateLoadedClasses({
            onComplete: function () { },
            onMatch: function (name, handle) {
                if (name.indexOf("com.hexl.lessontest.logic") > -1) { // 使
                    var targetInterface = "com.hexl.lessontest.logic.IAnima
                    if (targetInterface === name) {
                        return;
                    }
                    console.log("find class");
                    var targetClass = Java.use(name);
                    console.log("\t", name);
                    var superClassName;
                    while (1) {
                        var interfaceList = targetClass.class.getInterfaces
                        if (interfaceList.length > 0) {
                            for (var i in interfaceList) {
                                var interString = interfaceList[i].toString
                                if (interString.indexOf(targetInterface) >
                                    console.log("\t\t\t", interString); //
                                    break;
                                }
                            }
                        }
                        superClassName = targetClass.$super.$className;
                        targetClass = targetClass.$super;
                        if ("java.lang.Object" === superClassName) {
                            break;
                        }
                        console.log("\t\t", superClassName) // 打印类名
                    }
                }

            }
        })
    })
}

setImmediate(searchInterface)
```