初识 hook

一.什么是 hook?

Hook 技术又叫做钩子函数，在系统没有调用该函数之前，钩子程序就先捕获该消息，钩子函数先得到控制权，这时钩子函数既可以加工处理（改变）该函数的执行行为，还可以强制结束消息的传递。简单来说，就是把系统的程序拉出来变成我们自己执行代码片段。

Hook 技术本质是函数调用，由于处于 Linux 用户状态，每个进程有自己独立的进程控件，所以必须先注入所要 Hook 的进程空间，修改其内存中进程代码，替换过程表的符号地址，通过 ptrace 函数附加进程，向远程进程注入 so 库，从而达到监控以及远程进程关键函数挂钩；

二.常见的 hook 框架

框架名称	框架特性
Xposed	Java 层的 HOOK 框架，由于要修改 Zgote 进程，需要 Root
CydiaSubstrator	本地层的 HOOK 框架，本质上是一个 inline Hook
dexposed	
AndFix	
Sophix	
AndroidMethodHook	
Legend	在 AndFix 框架的基础上，在方法进行替换前进行了方法的备份
YAHFA	
EPIC 框架；	
VirtualXposed	Virtual APP 与 Xposed 的一个结合
frida	动态代码检测框架,它可以使你的 js 代码或者代码段注入到 Windows,Linux,android,macOS,GNU / Linux，iOS 和 QNX 上的本机

三.使用 frida 进行一个简单的 demon

3.1 为什么选择 frida?

从上面的介绍,我们就可以明显地看出,frida 是一个跨平台,简单实用且强大的 hook 框架!但是 frida 也有本身先天的缺陷,比如基于 Xposed 框架开发的 hook 插件可以直接在客户端运行不需要 PC 的参与,从我们接下来的 demon 我们就可以很清楚地看到,基于 frida 开发的 hook 插件离不开 PC 的参与。

3.1 安装环境

根据官网介绍 frida 的运行环境如下

- Python :建议最新的 3.X
- 操作系统 :Windows,macOS,或者 GNU/Linux
- Android :Genmotion 或者 Android 真机
- Frida-server :Genmotion 下载 x86 版本的,国内的 Android 真机选择下载 arm 版本, 下载地址

本次 demon 所对应的 frida 环境如下:

- Python :3.7.4
- 操作系统 :macOS 10.15.3
- Android :genmotion 模拟器
- Frida-server :frida-server-12.8.11-android-x86.xz

3.2 运行 frida

(1) 使用 adb 将 frida-server-12.8.11-android-x86 移到/data/temp 目录下面,并且改变他的权限。使用到的命令:adb push /Users/tools/安卓/frida/frida-server-12.8.11-android-x86 /data/local/temp,chmod 777 frida-server-12.8.11-android-x86

(2) 转发本地端口流量到 frida-server。使用到的命令 adb forward tcp:27043 tcp:27043

(3) 将 demon.apk 文件安装在 genmotion 模拟器上

(4) PC 上运行 hook.py 进行对 android 运行程序进行 hook

3.3 frida 进行 hook 的效果图

图 1 未进行 hook 时的原始模样

图 2 进行 hook 成功的截图

四打包所有需要的东西

我把编译过的 apk 以及编译前的 java 源码,还有 python 实例程序统一打包,放在我的 github 地址中,只需要在模拟器上运行 apk,frida-server,在本机上运行 python 程序,不需要做任何操作就能进行 hook.
项目地址:https://github.com/ba1ma0/frida-demon