逆向攻防神秘武器，让黑盒世界无所遁形

Frida是什么？

Frida 是一种开源的动态插桩工具，可以理解为“原生应用版的 Greasemonkey”。从专业角度讲，Frida 允许你在程序运行时向原生应用的内存空间注入自定义代码，从而动态监控和修改应用的行为。它支持 Windows、Mac、Linux、Android 和 iOS 等多个平台，这使得它在调试、逆向工程和安全研究等领域非常有用。

很多人可能对 Greasemonkey 的概念并不熟悉，它实际上是 Firefox 浏览器的一套插件体系，允许用户通过脚本动态修改网页的显示和功能。类似地，Frida 通过脚本让你可以对原生应用进行实时控制，无论是调试、监控还是修改应用行为，都能灵活应对。

简单来说，Frida 的主要用途包括：

- **动态调试**
  ：实时分析和修改程序运行状态，帮助开发者调试复杂问题。
- **逆向工程**
  ：通过注入代码观察应用内部的调用和数据流，便于理解程序逻辑。
- **安全研究**
  ：监控和修改程序行为，发现并分析潜在的安全漏洞或恶意行为。

前提环境

- **Python 环境**：确保已安装 Python 3.x（建议 Python 3.6 及以上版本）。
- **pip 工具**：确保 pip 已正确安装并配置。
- **网络连接**：如果你在国内网络环境下，建议使用国内的 pip 镜像源来加快下载速度。
- 工具：frida(目前版本最稳定的为16.1.10，其他版本有可能会报各种错误)

frida安装步骤

首先，安装 frida 时，请依次执行下面两条命令：

```
pip install frida==16.1.10 -i https://pypi.tuna.tsinghua.edu.cn/simple
pip install frida-tools==12.1.3 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

如果出现如下错误：

```
Fatal error in launcher: Unable to create process using '"E:\Tools\Disassembler\IDA_Pro_7.7\python38\
```

则说明 pip 使用的源与当前 Python 环境不匹配。此时，只需通过更新 pip 来解决问题，命令如下：

```
python -m pip install -U pip -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
C:\Windows\system32>pip install frida-tools==12.1.3 -i https://pypi.tuna.tsinghua.edu.cn/simple
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Collecting frida-tools==12.1.3
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/94/1d/5126f28dd9f3f5ee2ef44644c9096b148b1079e3aae047cc46892c44b7dd/frida-tools-12.1.3.tar.gz (177 kB)
     ---------------------------------------- 177.9/177.9 kB 1.8 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting colorama<1.0.0,>=0.2.7 (from frida-tools==12.1.3)
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/d1/d6/3965ed04c63042e047cb6a3e6ed1a63a35087b6a609aa3a15ed8ac56c221/colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Requirement already satisfied: frida<17.0.0,>=16.0.9 in e:\ida_pro_7.7\python38\lib\site-packages (from frida-tools==12.
```

安装完成后确定手机的架构

```
adb shell
getprop ro.product.cpu.abi
```

```
C:\Windows\system32>adb shell
blueline:/ $ su
blueline:/ # getprop ro.product.cpu.abi
arm64-v8a
blueline:/ #
```

arm64架构，去官网找到相对应的frida-server

官网地址https://github.com/frida/frida/releases

下载好之后解压，得到server文件，push进手机里面，修改名称为fs

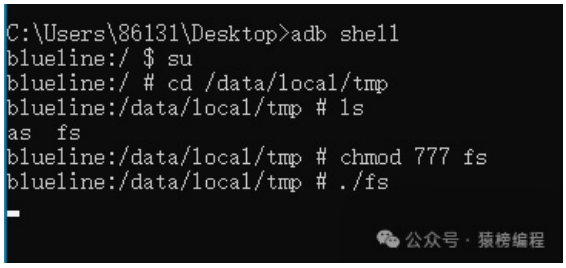
```
adb push frida-server-16.1.0-android-arm64 /data/local/tmp/fs
```

```
C:\Users\86131\Desktop>adb push frida-server-16.1.0-android-arm64 /data/local/tmp/fs
frida-server-16.1.0-android-arm64: 1 file pushed, 0 skipped. 90.3 MB/s (50947424 bytes in 0.538s)

C:\Users\86131\Desktop>
```

接下来进入手机系统，修改文件的权限，运行该文件

```
adb shell
su
cd /data/local/tmp
chmod 777 fs
./fs
```

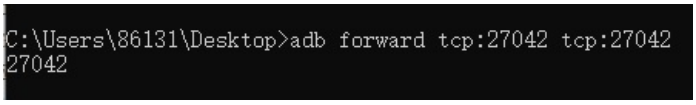


如果没有回显或者有warning，说明运行程序成功

新开一个终端，转发这个端口(./fs默认端口应该是27042)

```
adb forward tcp:27042 tcp:27042
```

转发成功会有对应端口回显，如下图



现在就可以使用frida了，我们可以使用命令查看手机上的端口

```
frida-ps -U
```

Frida 的常见用法和技巧

附加到正在运行的进程：

```
frida -U -n com.example.app
```

其中，`-n` 参数指定应用的包名。

在应用启动时附加并加载脚本：

```
frida -U -f com.example.app -l script.js --no-pause
```

其中，`-f` 参数指定要启动的应用包名，`-l` 参数指定要加载的脚本文件，`--no-pause` 参数表示在加载脚本后立即恢复应用运行。

跟踪特定函数的调用：

```
frida-trace -U -i "open" com.example.app
```

上述命令将跟踪 `com.example.app` 应用中所有名为 `open` 的函数调用。

编写和加载自定义 Hook 脚本：

创建一个 JavaScript 文件（例如 `hook.js`），内容如下：

```
Java.perform(function() {
    var MainActivity = Java.use('com.example.app.MainActivity');
    MainActivity.onResume.implementation = function() {
        console.log('onResume called');
        this.onResume();
    };
});
```