

Frida结合r0capture抓应用层全协议

【学习目标】

- 掌握Frida使用
- 掌握r0capture使用

1. 前言

- 本章内容适用于解决一下两个问题：
 - APP不走HTTP或HTTPS协议，导致burp抓不到包的问题。
 - APP检测证书的问题。
- 通过利用Frida结合r0capture抓取应用层中的全部协议。
- 所用环境和工具：夜神模拟器、安卓9、Frida、r0capture、Wireshark

2. Frida介绍

- Frida是一款轻量级HOOK框架，可用于多平台上，例如android、windows、ios等；它能够对移动应用（包括Android和iOS）进行实时分析和修改，因此可以用来抓取数据包。
- frida分为两部分，服务端运行在目标机上，通过注入进程的方式来实现劫持应用函数，另一部分运行在系统机器上。frida上层接口支持js、python、c等。

3. 什么是Hook?

- HOOK是一种非常强大且灵活的技术，通过拦截和修改程序执行，可以实现多种高级功能，在软件开发和安全研究中有广泛的应用。
- 简单理解hook就是钩子，之所以说是钩子，是因为你可以往钩子上挂任何东西，程序执行到hook的时候，你预先挂上/勾上(hook)的是什么，就执行什么。

4. 本地安装frida

- 本地安装frida

```
pip install frida
pip install frida-tools
pip list      # 查看安装列表
```

默认情况下pip使用的是国外的镜像，如果下载失败，可使用 -i 参数来指定镜像地址

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

```
C:\Users\sunny>pip install frida
Collecting frida
  Downloading frida-16.4.2-cp37-abi3-win_amd64.whl.metadata (2.1 kB)
Requirement already satisfied: typing-extensions in c:\python38\lib\site-packages (from frida) (4.12.2)
Downloading frida-16.4.2-cp37-abi3-win_amd64.whl (33.4 MB)
33.4/33.4 MB 5.6 MB/s eta 0:00:00
Installing collected packages: frida
Successfully installed frida-16.4.2
```

```
C:\Users\sunny>pip install frida-tools
Collecting frida-tools
  Using cached frida_tools-12.4.4-py3-none-any.whl
Requirement already satisfied: colorama<1.0.0,>=0.2.7 in c:\python38\lib\site-packages (from frida-tools) (0.4.6)
Requirement already satisfied: frida<17.0.0,>=16.2.2 in c:\python38\lib\site-packages (from frida-tools) (16.4.2)
Requirement already satisfied: prompt-toolkit<4.0.0,>=2.0.0 in c:\python38\lib\site-packages (from frida-tools) (3.0.47)
Requirement already satisfied: pygments<3.0.0,>=2.0.2 in c:\python38\lib\site-packages (from frida-tools) (2.18.0)
Requirement already satisfied: typing-extensions in c:\python38\lib\site-packages (from frida<17.0.0,>=16.2.2->frida-tools) (4.12.2)
Requirement already satisfied: wcwidth in c:\python38\lib\site-packages (from prompt-toolkit<4.0.0,>=2.0.0->frida-tools) (0.2.13)
Installing collected packages: frida-tools
Successfully installed frida-tools-12.4.4
```

5. 查看CPU位数

- 需要使用adb，ADB（Android Debug Bridge）是一个命令行工具，允许开发人员和用户与Android设备进行通信和交互。
- 添加环境变量

新加卷 (D:) > Install > Nox > bin				
名称	修改日期	类型	大小	
BignoxVMS	2024/7/8 17:59	文件夹		
data	2024/7/8 16:49	文件夹		
language	2024/7/5 10:20	文件夹		
NoxSrv	2024/7/5 10:21	文件夹		
nine	2024/7/5 10:20	文件夹		

- 进入adb

```
adb shell
```

```
C:\Users\sunny>adb shell
beyond1q:/ #
beyond1q:/ # whoami
root
beyond1q:/ # pwd
/
beyond1q:/ # ls
acct      default.prop  init.usb.configfs.rc  oem          root          ueventd.x86.rc
bin       dev           init.usb.rc           plat_file_contexts  sbin          vendor
bugreports  etc          init.x86.rc           plat_hwservice_contexts  sdcard        vendor_file_contexts
cache     fstab.x86    init.zygote32.rc      plat_property_contexts  sepolicy      vendor_hwservice_contexts
charger   init         init.zygote64_32.rc   plat_seapp_contexts    storage       vendor_property_contexts
config    init.envIRON.rc  lib                  plat_service_contexts  sys           vendor_seapp_contexts
d         init.rc       mnt                  proc              system        vendor_service_contexts
data      init.superuser.rc  odm                  product          ueventd.rc   vndservice_contexts
beyond1q:/ #
```

- 获取 CPU 位数

```
getprop ro.product.cpu.abi
```

```
beyond1q:/ # getprop ro.product.cpu.abi
x86_64
beyond1q:/ #
beyond1q:/ #
```

5.1 扩展！

- 获取系统版本

```
getprop ro.build.version.release
```

- 获取设备型号

```
getprop ro.product.model
```

- 获取设备品牌

```
getprop ro.product.brand
```

- 获取设备制造商

```
getprop ro.product.manufacturer
```

- 获取设备名称

```
getprop ro.product.name
```

6. 模拟器安装Frida

- 注意事项
 - 下载的版本要和上面本地pip安装版本保持一致

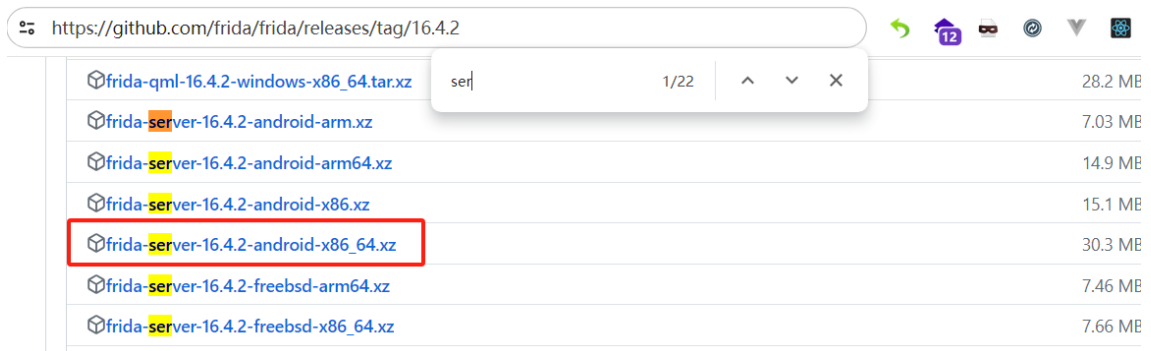
```
C:\Users\sunny>pip install frida
Collecting frida
  Downloading frida-16.4.2-cp37-abi3-win_amd64.whl.metadata (2.1 kB)
Requirement already satisfied: typing-extensions in c:\python38\lib\site-packages (from frida) (4.12.2)
  Downloading frida-16.4.2-cp37-abi3-win_amd64.whl (33.4 MB)
    33.4/33.4 MB 5.6 MB/s eta 0:00:00
Installing collected packages: frida
Successfully installed frida-16.4.2
```

- CPU位数要和模拟器的保持一致

```
beyondiq:/ # getprop ro.product.cpu.abi_
x86_64
beyondiq:/ #
beyondiq:/ #
```

- Frida下载

<https://github.com/frida/frida/releases/tag/16.4.2>



- 下载后进行解压
- 将解压后的frida传到模拟器中

```
adb push C:\Users\sunny\Desktop\frida-server-16.4.2-android-x86_64
/data/local/frida-server-16.4.2
```

```
C:\Users\sunny>adb push C:\Users\sunny\Desktop\frida-server-16.4.2-android-x86_64 /data/local/frida-server-16.4.2
[100%] /data/local/frida-server-16.4.2
C:\Users\sunny>
```

- 修改权限

```
adb shell
cd /data/local/
chmod 777 frida-server-16.4.2
```

也可以直接外面执行:

```
adb shell chmod 777 /data/local/frida-server-16.4.2
```

```
C:\Users\sunny>adb shell
beyondlq:/ #
beyondlq:/ # cd /data/local/
beyondlq:/data/local #
beyondlq:/data/local # ll
total 110720
-rw-rw-rw- 1 root root 113367864 2024-07-10 14:38 frida-server-16.4.2
drwxrwxrwx 2 shell shell 4096 2024-07-08 17:04 tmp
drwxrwxrwx 2 shell shell 4096 2023-10-09 19:33 traces
beyondlq:/data/local #
beyondlq:/data/local # chmod 777 frida-server-16.4.2
beyondlq:/data/local #
beyondlq:/data/local # ll
total 110720
-rwxrwxrwx 1 root root 113367864 2024-07-10 14:38 frida-server-16.4.2
drwxrwxrwx 2 shell shell 4096 2024-07-08 17:04 tmp
drwxrwxrwx 2 shell shell 4096 2023-10-09 19:33 traces
```

- 启动 Frida

```
./frida-server-16.4.2
```

```
beyondlq:/data/local #
beyondlq:/data/local # ./frida-server-16.4.2
```

- 重新打开一个窗口，确定启动成功

```
ps -ef |grep frida
```

```
C:\Users\sunny>adb shell
beyondlq:/ # ps -ef |grep frida
root      3427  3421 0 15:31:18 pts/0 00:00:00 frida-server-16.4.2
root      3445      1 0 15:31:19 ?      00:00:00 ll unix:abstract=/frida-5b4f7b7c-91c3-4d48-b963-25534d40ale7
root      3481  3478 0 15:33:59 pts/1 00:00:00 grep frida
beyondlq:/ #
```

- 查看进程，如果正常返回结果则证明连接成功

```
frida-ps -U
```

```
C:\Users\sunny>frida-ps -U
PID  Name
-----
3215  Amaze
1572  adbd
2088  android.ext.services
1554  android.hardware.audio@2.0-service
1555  android.hardware.bluetooth@1.0-service.btlinux
2330  android.hardware.camera.provider@2.4-service
1557  android.hardware.cas@1.0-service
1558  android.hardware.configstore@1.1-service
1559  android.hardware.dumpstate@1.0-service
1560  android.hardware.light@2.0-service
1561  android.hardware.memtrack@1.0-service
1562  android.hardware.power@1.0-service
1563  android.hardware.usb@1.0-service
1564  android.hardware.wifi@1.0-service
1552  android.hidl.allocator@1.0-service
1568  audioserver
1581  cameraserver
2835  com.android.inputmethod.pinyin
```

- 查看进程详细信息，可显示运行中的APP对应的包名

```
frida-ps -Ua
```

```
C:\Users\sunny>frida-ps -Ua
PID  Name  Identifier
-----
3215  Amaze  com.amaze.filemanager
4869  Soul   cn.soulapp.android
4396  图库   com.android.gallery3d
2015  设置   com.android.settings
```

- 查看frida占用端口

```
netstat -ano
```

```
beyondiq:/ # netstat -antlp
Active Internet connections (established and servers)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program Name
tcp 0 0 0.0.0.0:25000 0.0.0.0:* LISTEN 1565/local_opengl
tcp 0 0 0.0.0.0:8470 0.0.0.0:* LISTEN 1566/local_gps
tcp 0 0 0.0.0.0:18011 0.0.0.0:* LISTEN 1573/noxd
tcp 0 0 0.0.0.0:24800 0.0.0.0:* LISTEN 2330/android.hardware.camera.provider@2.4-service
tcp 0 0 0.0.0.0:27042 0.0.0.0:* LISTEN 3427/frida-server-16.4.2
tcp 0 0 172.16.93.15:18011 172.16.93.2:6998 ESTABLISHED 1573/noxd
tcp 0 0 172.16.93.15:24800 172.16.93.2:6998 ESTABLISHED 2330/android.hardware.camera.provider@2.4-service
tcp6 0 0 :::5555 :::* LISTEN 1572/adbd
tcp6 0 0 :::9527 :::* LISTEN 2835/com.android.inputmethod.pinyin
tcp6 0 0 :::18014 :::* LISTEN 2835/com.android.inputmethod.pinyin
tcp6 0 0 :::18016 :::* LISTEN 1721/system_server
tcp6 32 0 :::ffff:172.16.94.:56050 :::ffff:111.1.37.132:443 CLOSE_WAIT 4869/cn.soulapp.android:pushservice
tcp6 0 0 :::ffff:172.16.94.:35612 :::ffff:118.31.224.:8180 ESTABLISHED 4869/cn.soulapp.android:pushservice
tcp6 0 0 :::ffff:172.16.94.:53908 :::ffff:218.205.76.:5226 ESTABLISHED 4869/cn.soulapp.android:pushservice
tcp6 32 0 :::ffff:172.16.94.:56044 :::ffff:111.1.37.132:443 CLOSE_WAIT 4869/cn.soulapp.android:pushservice
tcp6 32 0 :::ffff:172.16.94.:56048 :::ffff:111.1.37.132:443 CLOSE_WAIT 4869/cn.soulapp.android:pushservice
tcp6 32 0 :::ffff:172.16.94.:56046 :::ffff:111.1.37.132:443 CLOSE_WAIT 4869/cn.soulapp.android:pushservice
tcp6 0 0 :::ffff:172.16.93.1:5555 :::ffff:172.16.93.2:7039 ESTABLISHED 1572/adbd
beyondiq:/ #
```

- 将端口转发出来

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

```
C:\Users\sunny>
C:\Users\sunny>adb forward tcp:27042 tcp:27042
C:\Users\sunny>
```

- 测试是否转发成功

在确保端口转发正常后，可以使用 `frida-ps -R` 命令来列出设备上正在运行的所有进程。

因为 Frida 需要与设备上的进程进行交互，所以必须确保 Frida 客户端能够正确地与设备建立通信。

```
frida-ps -R
```

```
C:\Users\sunny>frida-ps -R
PID  Name
-----
3215  Amaze
4869  Soul
1572  adbd
2088  android.ext.services
1554  android.hardware.audio@2.0-service
1555  android.hardware.bluetooth@1.0-service.btlinux
2330  android.hardware.camera.provider@2.4-service
1557  android.hardware.cas@1.0-service
1558  android.hardware.configstore@1.1-service
1559  android.hardware.dumpstate@1.0-service
1560  android.hardware.health@1.0-service
```

7. r0capture使用

<https://github.com/r0ysue/r0capture>

7.1 安装依赖

```
Python版本>=3.6
pip install loguru
pip install click
```

7.2 使用

- 抓包内容保存成pcap文件供后续分析:

```
python r0capture.py -f -U com.pl.mobile.putong -p tantan01.pcap
```

这里 com.pl.mobile.putong 指的是 frida-ps -ua 查到的app对应的包名

```
C:\Users\sunny>frida-ps -Ua
PID Name Identifier
-----
12253 探索 com.pl.mobile.putong
8967 设置 com.android.settings
```

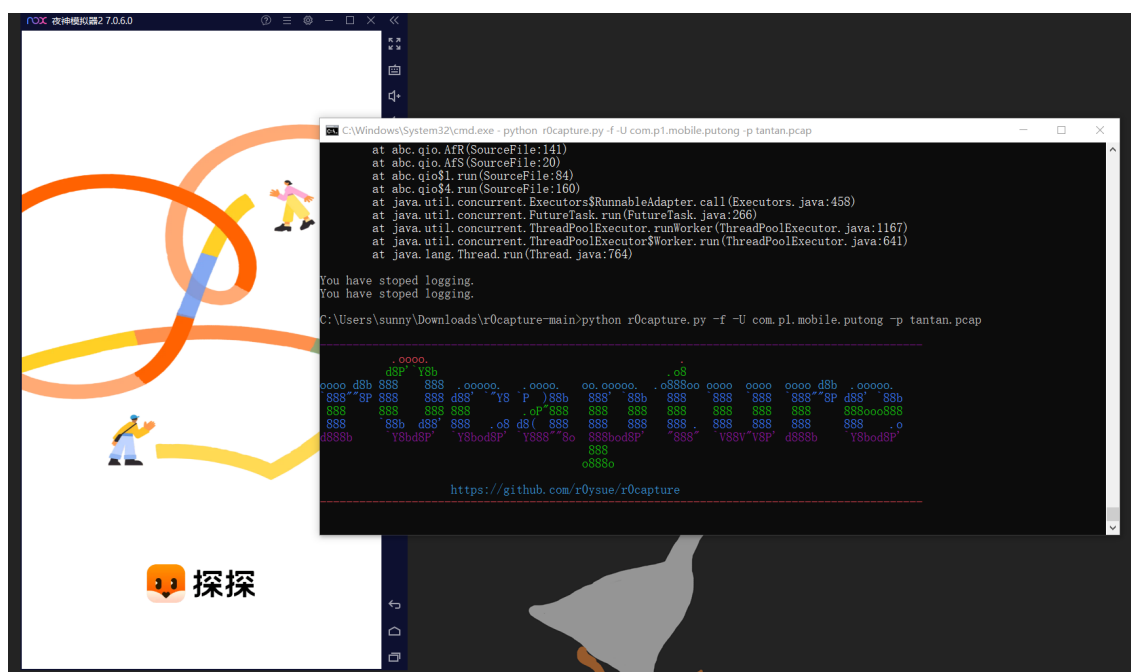
将数据包保存成 pcap 文件, 供后续使用Wireshark进行分析

```
选择 C:\Windows\System32\cmd.exe - python r0capture.py -f com.pl.mobile.putong -p tantan01.pcap

at abc.kdd.Aa(SourceFile:87)
at abc.kde.run(Unknown Source:4)
at abc.kdd$.run(SourceFile:178)

2024-07-10 21:15:09.686 | INFO | __main__:on_message:232 - SSL Session: CCA689B08EBCAB92C9AD09DB074D53C88BE93654301A5
32D2E1C564071E45F519
2024-07-10 21:15:09.687 | INFO | __main__:on_message:233 - [SSL_read] 120.133.43.33:443 -> 172.16.94.15:58804
2024-07-10 21:15:09.689 | INFO | __main__:on_message:241 - ..w....q.v..cU.a.Y>...m...r...p...b...u.b.&=LtA.\.76
4...T!b...z...r...iZ..5...JBb...).c...c...c...Jb).9.../.....L.....T!b...z...*...T!b...z...
V...y...9V!.M...w.K.GX.$..R.....)M.J...4.I.../.aQ.Y>.d*).....f.o.m.lh...H.....d...l...B.O.rWY.W....X
lr...8.....v..L...-.O.p.m...Ih5&J....U7.O...l...B.O.rWY.W....X.lr...8.....q{"meta":{"code":200,"mes
sage":"OK"},"data":{"users":null,"relationships":null,"moments":null,"followships":null,"conversations":null,"messages":
null,"questions":null,"media":null,"shops":null,"stickers":null,"packages":null,"bundles":null,"campaigns":null,"chatGro
ups":null,"interestedGroups":null,"chatGroupMembers":null,"groupApplies":null,"groupNotifications":null,"groupAttributes
":null,"notificationCounters":[{"unread":1,"latestNotificationTime":"1970-01-01T00:00:00.000000+0000"}],giftInfos":null
,"audioTexts":null,"literatures":null,"woodenFishCounters":null,"chatGameInfo":null,"chatAssistantQuestions":null,"chatA
ssistantQuestionSync":null,"aiPictures":null,"aiPictureAuth":null,"pagination":{"total":0,"limit":0,"links":{"previous
":null,"next":null}}}}
2024-07-10 21:15:09.690 | INFO | __main__:on_message:242 - java.lang.Throwable
at com.android.org.conscrypt.ConscryptFileDescriptorSocket$SSLInputStream.read(Native Method)
at abc.wel.read(SourceFile:102)
at abc.wdw$.read(SourceFile:159)
at abc.wes.request(SourceFile:62)
at abc.wes.require(SourceFile:55)
at abc.wcj.Aa(SourceFile:96)
at abc.wch$.execute(SourceFile:668)
at abc.war.run(SourceFile:32)
at java.lang.Thread.run(Thread.java:764)
```

- 运行 r0capture.py 之后, APP会自动打开



- Ctrl+C 结束掉之后，会生成 pcap 后缀的文件，直接用wireshark打开即可

> 下载 > r0capture-main >

名称	修改日期	类型	大小
__pycache__	2024/7/10 18:17	文件夹	
pic	2024/7/10 18:14	文件夹	
LICENSE	2024/7/10 18:14	文件	12 KB
myhexdump.py	2024/7/10 18:14	Python File	16 KB
r0capture.py	2024/7/10 18:14	Python File	14 KB
README.md	2024/7/10 18:14	Markdown File	5 KB
script.js	2024/7/10 18:14	JavaScript 文件	14 KB
tantan.pcap	2024/7/10 20:44	Wireshark captu...	1,650 KB
tantan01.pcap	2024/7/10 21:16	Wireshark captu...	824 KB

7.3 报错解决

- 如果提示下面错误，尝试重启frida

```
C:\Users\sunny\Downloads\r0capture-main>python r0capture.py -f -U com.pl.mobile.putong -p tantan.pcap

.oooo.
d8P' Y8b
oooo d8b 888 888 .ooooo. .oooo. .o888oo ooooo ooooo d8b .ooooo.
888"8P 888 888 d88' "Y8 P )88b 888' 88b 888 888 888 888"8P d88' 88b
888 888 888 888 .oP"888 888 888 888 888 888 888 888ooo888
888 88b d88' 888 .o8 d8( 888 888 888 888 888 888 888 888 .o
d888b Y8bd8P' Y8bod8P' Y888"8o 888bod8P' "888" V88V"V8P' d888b Y8bod8P'
888
o888o

https://github.com/r0ysue/r0capture

Traceback (most recent call last):
  File "r0capture.py", line 365, in <module>
    ssl_log(
  File "r0capture.py", line 260, in ssl_log
    pid = device.spawn([process])
  File "C:\Python38\lib\site-packages\frida\core.py", line 86, in wrapper
    return f(*args, **kwargs)
  File "C:\Python38\lib\site-packages\frida\core.py", line 1029, in spawn
    return self._impl.spawn(program, **kwargs)
frida.TimedOutError: unexpectedly timed out while waiting for app to launch
```

- 先结束frida进程

新打开一个窗口，查看frida的进程id

```
ps -ef | grep frida    #查看进程id
kill -9 2974           #结束进程
```

```
C:\Users\sunny>adb shell
beyondlq:/ #
beyondlq:/ # ps -ef | grep frida
root      2974   2950  0 13:46:34 pts/0 00:00:01 frida-server-16.4.2
root      2990     1  0 13:46:35 ?      00:00:00 11 unix:abstract=/frida-b5e932fd-c57a-4bd8-af3d-5a4b8996d408
root      5875   5825  0 13:52:45 pts/1 00:00:00 grep frida
beyondlq:/ # kill -9 2974
beyondlq:/ # exit
```

- 再重新启动frida

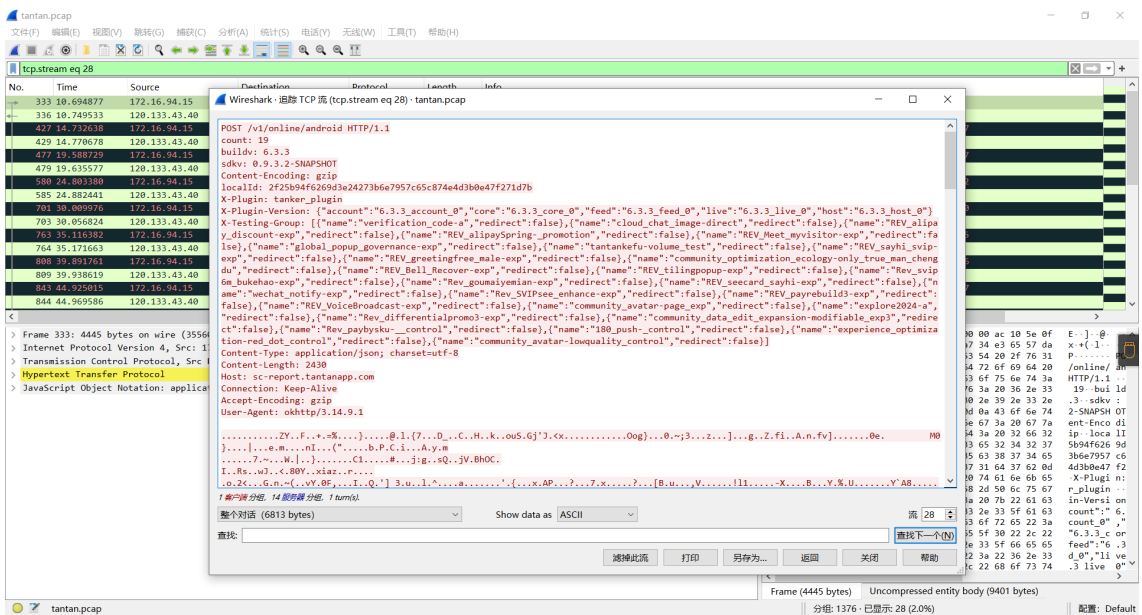
```
./frida-server-16.4.2
```

- 重新进行端口转发

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

8. Wireshark使用

- 打开Wireshark，直接将文件拖进来



8.1 比较操作符

比较操作符用于在 Wireshark 的过滤表达式中进行条件过滤，常见的操作符有：

- `==`: 等于
- `!=`: 不等于
- `>`: 大于
- `<`: 小于
- `>=`: 大于等于
- `<=`: 小于等于

这些操作符可以用于各种字段，如协议、IP 地址、端口等。

8.2 协议过滤

通过在 Wireshark 的过滤框中直接输入协议名，可以过滤出特定协议的数据包列表，例如：

- `tcp`: 只显示 TCP 协议的数据包列表
- `http`: 只显示 HTTP 协议的数据包列表
- `icmp`: 只显示 ICMP 协议的数据包列表
- `arp`: 只显示 ARP 协议的数据包列表

8.3 IP 过滤

针对 IP 地址的过滤可以使用以下表达式：

- `ip.src == 112.53.42.42`: 显示源地址为 112.53.42.42 的数据包列表
- `ip.dst == 112.53.42.42`: 显示目标地址为 112.53.42.42 的数据包列表
- `ip.addr == 112.53.42.42`: 显示源 IP 地址或目标 IP 地址为 112.53.42.42 的数据包列表

8.4 端口过滤

对于端口的过滤可以使用以下表达式：

- `tcp.port == 80` : 显示源主机或目标主机端口为 80 的数据包列表
- `tcp.srcport == 80` : 只显示 TCP 协议的源主机端口为 80 的数据包列表
- `tcp.dstport == 80` : 只显示 TCP 协议的目标主机端口为 80 的数据包列表