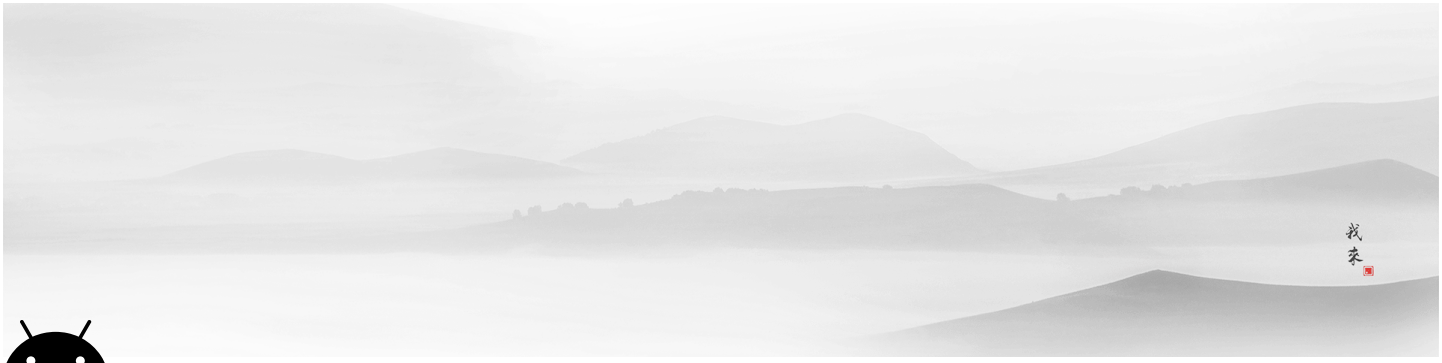
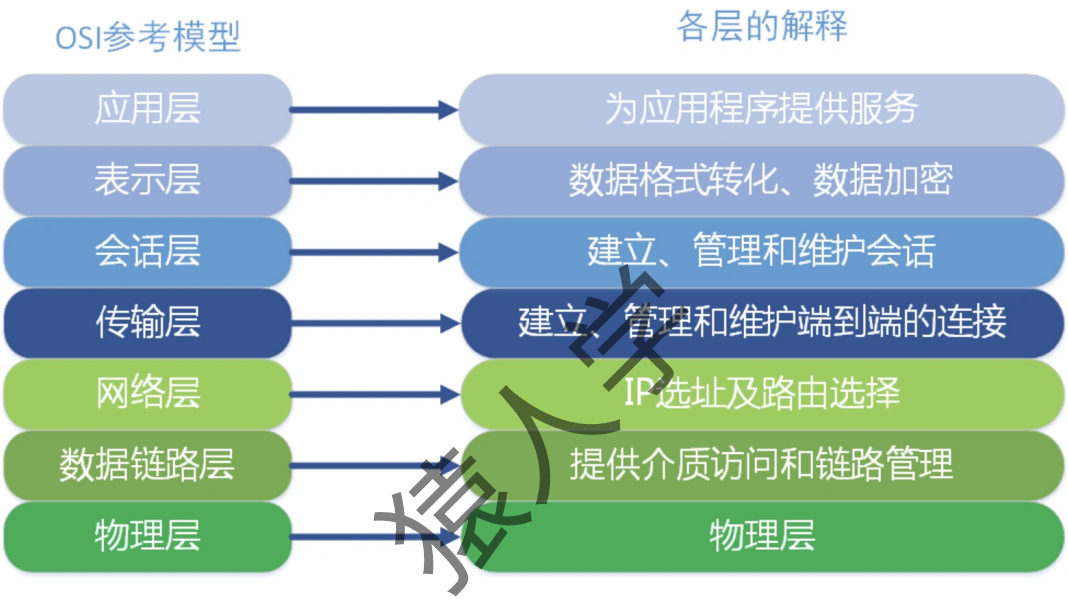


猿人学–Android抓包专题

汇总业内抓包优解，根据个人理解留下实用性较高的方式方法，然后输出。



- http(s) 应用层
- socks 会话层
- tcp、udp 传输层
- ip 网络层

VPN 抓包环境配置

为什么不使用代理方式抓包?

- 易检测
  - System.getProperty("http.proxyHost")

    System.getProperty("http.proxyPort")
- no_proxy
  - new OkHttpClient().newBuilder().proxy(Proxy.NO_PROXY).build();

为什么使用 vpn 方式抓包?

- 运行在网络层，使手机的路由器/路由表改变
- 更加底层，意味着可以捕获更多的上层流量

VPN + 中间人抓包

- postern：https://github.com/postern-overwal/postern-stuff
- charles 下载：https://www.charlesproxy.com/download/
- charles 激活码：https://www.zzzmode.com/mytools/charles/

安装证书

- 同一局域网下，通过系统配置 http 代理方式，浏览器访问 `chls.pro/ssl`，安装证书到设备
- 搞定证书以后，此后就使用 postern 配置 charles 的 socks 模式代理的方式抓包了

信任证书

- /data/misc/user/0/cacerts-added/ --------> /system/etc/security/cacerts
- magisk(v23.0,其他版本未测试) + Move Certificates 插件 或 直接用 re 管理器(可以挂载读写权限)或 mt 管理器搞,尽量别手动 adb 命令,否则可能还要解决一些挂载问题

VPN + 中间人抓包 + 科学上网

- charles 设置 external proxy

手机 VPN App 抓包

- HttpCanary：可以选择特定的 app 进行网络流量分析，下载地址：https://apkpure.com/httpcanary-—-http-sniffer-capture-analysis/com.guoshi.httpcanary
- 至于能不能选择特定的 app 的流量，是由 VPN 软件决定的，postern 这个软件目前不支持

VPN 检测

- java.net.NetworkInterface.getName()

  android.net.ConnectivityManager.getNetworkCapabilities(network)

抓包场景

应用层：http(s)协议抓包

阻碍：证书校验

客户端校验服务端证书

- 网络框架默认校验证书链（这就是为什么中间人 https 抓包第一步就是先把证书弄系统里使其变得可信任）
- SSLPinning
  - HTTPS 在建立 ssl 通道的过程中，当客户端向服务端发送了连接请求后，服务器会发送自己的证书(包括公钥、证书有效期、服务器信息等)给客户端。
  - 客户端在收到服务器的证书后，对该证书进行强校验，验证该证书是不是客户端承认的证书，如果不是，则直接断开连接。
  - APP 是 HTTPS 的服务提供方自己开发的客户端，开发者可以先将自己服务器的证书打包内置到自己的 APP 中，或者将证书签名内置到 APP 中，当客户端在请求服务器建立连接期间收到服务器证书后，先使用内置的证书信息校验一下服务器证书是否合法，如果不合法，直接断开。
  - SSLPinning 开发：https://appmattus.medium.com/android-security-ssl-pinning-1db8acb6621e

服务端校验客户端证书

- 自实现客户端 SSLSocketFactory 逻辑

会话层：socket 端口通信抓包

阻碍：协议底层，二进制流，不够直观

抓包阻碍解决方案

hook 校验位置（生来就是为了解决 sslpinning）

- Xposed 之 JustTruestMe：https://github.com/Fuzion24/JustTrustMe
- Frida 之 SSLUnPinning：https://github.com/WooyunDota/DroidSSLUnpinning

hook 网络框架（无视证书绑定，顺手解决了证书校验）

- hook okhttp3 添加拦截器

hook 系统底层（无视证书绑定，顺手解决了证书校验）

有果必有因，深入系统底层，方能降维打击，才有更多可能

tcp-java

- java.net.SocketOutputStream.socketWrite0('java.io.FileDescriptor', '[B', 'int', 'int')

  java.net.SocketInputStream.socketRead0('java.io.FileDescriptor', '[B', 'int', 'int', 'int')

tcp-native

- sendto(int fd, const void *buf, size_t n, int flags, const struct sockaddr *addr, socklen_t addr_len)

  recvfrom(int fd, void *buf, size_t n, int flags, struct sockaddr *addr, socklen_t *addr_len)

udp-java

- libcore.io.Linux.sendtoBytes(FileDescriptor fd, Object buffer, int byteOffset, int byteCount, int flags, InetAddress inetAddress, int port)

  libcore.io.Linux.sendtoBytes(FileDescriptor fd, Object buffer, int byteOffset, int byteCount, int flags, SocketAddress address)

- libcore.io.Linux.recvfromBytes(FileDescriptor fd, Object buffer, int byteOffset, int byteCount, int flags, InetSocketAddress srcAddress)

udp-native

- sendto(int fd, const void *buf, size_t n, int flags, const struct sockaddr *addr, socklen_t addr_len)
- recvfrom(int fd, void *buf, size_t n, int flags, struct sockaddr *addr, socklen_t *addr_len)

```
udp-native和tcp-native底层都是sendto和recvfrom，但r0capture未选择该hook点
```

ssl-java

- android 版本 > 8
  - com.android.org.conscrypt.ConscryptFileDescriptorSocket$SSLOutputStream.write('[B', 'int', 'int')

    com.android.org.conscrypt.ConscryptFileDescriptorSocket$SSLInputStream.read('[B', 'int', 'int')

```java
// 实际它们才是java层的最底层，但是这个hook点不好获取socket连接的服务器ip地址和端口，故选择了上面的hook点
com.android.org.conscrypt.NativeCrypto.SSL_write(long sslNativePointer, FileDescriptor fd, SSLHandshakeCallbacks s
com.android.org.conscrypt.NativeCrypto.SSL_read(long sslNativePointer, FileDescriptor fd, SSLHandshakeCallbacks sh
```

- android 版本 <= 8
  - com.android.org.conscrypt.OpenSSLSocketImpl$SSLOutputStream.write('[B', 'int', 'int')

    com.android.org.conscrypt.OpenSSLSocketImpl$SSLInputStream.read('[B', 'int', 'int')

ssl-native

- 明文 hook 点：libssl.so 的 SSL_write 和 SSL_read
- 密文 hook 点：libc.so 的 write 和 read

HTTPS 修改为 HTTP

- 解决了 SSLPinning 的问题
- 替换 app 中 https 的 url 为 http
- charles map remote 建立映射规则，将 http 映射为 https 发包

IP 阻断

- 网络框架要存在协议模式切换的机制

```PowerShell
iptables -A INPUT -s ***.**.***.181 -j DROP #屏蔽
iptables -D INPUT -s ***.**.***.181 -j DROP #解除屏蔽
```

自定义 ssl 类库

- 枚举类，枚举 so 符号，hook 验证
- flutter hook

流量转储

- wireshark 捕获某一网卡的网络包
- tcpdump 基于 Unix 系统的命令行式的数据包嗅探工具

本节拓展资料

```纯文本
[原创]如何使用Xposed+JustTrustMe来突破SSL Pinning
https://bbs.pediy.com/thread-226435.htm

利用Frida绕过Android App的SSL Pinning
https://blog.csdn.net/weixin_44677409/article/details/106650473

利用Xposed+JustTrustMe绕过Android App的SSL Pinning
https://blog.csdn.net/weixin_44677409/article/details/106663127

JustTrustMe 原理分析
https://bbs.pediy.com/thread-214012.htm

[原创]关于JustTrustMe对混淆后的App无效的解决方案
https://bbs.pediy.com/thread-267839.htm

FRIDA 使用经验交流分享
https://bbs.pediy.com/thread-265160.htm

Android HTTPS防抓包策略与对抗方法总结
https://curz0n.github.io/2020/08/15/android-ssl-and-intercept/
```

移动安全抓包笔记
https://bbs.pediy.com/thread-266878.htm

android抓包学习的整理和归纳
文中提到的python-server端使用的simpletcp库，用网盘提供的zip包去安装(ps:作者原git好像删了)
https://bbs.pediy.com/thread-267940.htm
跟源码溯源的话最好使用okhttp3.10.0，就能跟文章中保持一致了，好像能更理所当然一点
gradle导入如下：implementation 'com.squareup.okhttp3:okhttp:3.10.0'

自动定位webview中的SLL_read和SSL_write
https://mabin004.github.io/2020/07/24/自动定位webview中的SLL-read和SSL-write/

Hooking Chrome’s SSL functions
https://nytrosecurity.com/2018/02/26/hooking-chromes-ssl-functions/

一种基于frida和drony的针对flutter抓包的方法
https://bbs.pediy.com/thread-261941.htm

Android HTTPS认证的N种方式和对抗方法总结
https://ch3nye.top/Android-HTTPS%E8%AE%A4%E8%AF%81%E7%9A%84N%E7%A7%8D%E6%96%B9%E5%BC%8F%E5%92%8C%E5%AF%B9%E6%8A%97%E6%96%B9
注意证书文件和密钥过期的情况，过期就根据代码注释的命令再生成一遍替换一下

android逆向学习--绕过非标准http框架和非系统ssl库app的sslpinning
https://bbs.pediy.com/thread-268014.htm

意外发现的Android硬核https抓包， 在多个app亲测ok, 自定义ssl也无用哦~
https://www.52pojie.cn/thread-1405917-1-1.html

https://codeshare.frida.re/browse

https://github.com/lasting-yang/frida_bypass_ssl_example

单向，双向，no_proxy，vpn，tcp，hook artmethod，hook网络框架，hook网络底层

```
adb shell "dumpsys window | grep mCurrentFocus"
adb shell su -c "/data/local/tmp/fs_15.1.12_arm64"
```

frida api提示（ps：踩坑记录：我全局安装后没提示，然后本地安装目标frida脚本所在目录还是没提示，后来安装至当前用户目录/Users/hexl下才有提示
npm install --save @types/frida-gum

https://github.com/CreditTone/hooker
https://github.com/siyujie/OkHttpLogger-Frida

拓展了解：
小菜花：https://bbs.pediy.com/user-home-844301.htm
底层直接调用sendto/recvfrom函数
  使用syscall + sendto/recvfrom调用号
    自实现内联汇编sendto/recvfrom
    自实现内联汇编syscall + sendto/recvfrom调用号

抓包阻碍大致解决流程：
vpn检测：
hook_vpn.js
  c校验s：
  DroidSSLUnpinning.js
  multi_unpinning.js
  just_trust_me.js
    okhttp：
    hook_okhttp3.js(推荐度一般)
    okhttp混淆：
    just_trust_me_okhttp_hook_finder.js
    OkHttpLogger-Frida
      s校验c：
      tracer-keystore.js
      keystore_dump.js

```
r0capture中keystore相关
  原生系统底层：
  lesson7_all_in_one.js
    其他情况(拓展了解)：
    hook syscall
    内存扫描+inlinehook
    ptrace(seccomp过滤) + PTRACE_SYSCALL
    源码级内核模块开发
```