# Flutter APP测试记录

## 前置知识

### JNI

javah -jni  生成一个C头文件，然后在对应的C文件中实现功能。Java代码希望使用JNI(Java Native Interface)调用的话，需要加载javah -jni生成的so文件再实例化后调用。

## Flutter 程序抓包

### 介绍

在对某Flutter APP进行测试的过程中，发现即使安卓系统已安装了Zap Proxy的证书，但是仍然无法抓取到Flutter APP的HTTP/S数据包。经过一番googling解决了这个问题。本文记录对Flutter APP抓包使用到的方法。

### Flutter程序无法抓取HTTP/S数据包

### 原因

Flutter无法抓取数据包的原因在于Flutter使用了Dart编程语言开发，Dart使用了BoringSSL进行SSL数据的加密与通信等功能(Dart-lang.github.io, 2019)。且Dart使用了内置的证书而非系统证书，所以即使在系统中加入并信任了自签名的证书，代理软件也无法获取到明文的HTTP流量。

### 解决方案

源代码链接：<https://github.com/google/boringssl/blob/ee0716f386190d2bfb105a0db7400df4915773a1/ssl/handshake.cc>

1. **if** (hs->config->custom\_verify\_callback != nullptr) {
2. ret = hs->config->custom\_verify\_callback(ssl, &alert);
3. **switch** (ret) {
4. **case** ssl\_verify\_ok:
5. hs->new\_session->verify\_result = X509\_V\_OK;
6. **break**;
7. **case** ssl\_verify\_invalid:
8. // If |SSL\_VERIFY\_NONE|, the error is non-fatal, but we keep the result.
9. **if** (hs->config->verify\_mode == SSL\_VERIFY\_NONE) {
10. ERR\_clear\_error();
11. ret = ssl\_verify\_ok;
12. }
13. hs->new\_session->verify\_result = X509\_V\_ERR\_APPLICATION\_VERIFICATION;
14. **break**;
15. **case** ssl\_verify\_retry:
16. **break**;
17. }
18. } **else** {
19. ret = ssl->ctx->x509\_method->session\_verify\_cert\_chain(
20. hs->new\_session.get(), hs, &alert)
21. ? ssl\_verify\_ok
22. : ssl\_verify\_invalid;
23. }

将19行ret置为1即可。

## URL 参数签名

程序对发送至服务器的HTTP请求参数进行了签名，由于程序使用的是flutter进行开发，Flutter将程序逻辑封装为一个so进行加载，称之为AOT shared library。

Flutter通过libflutter JNI\_OnLoad 调用libapp.so

1. 调用System.loadLibrary("flutter");
2. 使用JNI\_OnLoad方法调用libapp.so

FlutterJNI -> nativeRunBundleAndSnapshotFromLibrary

Reference:

Dart-lang.github.io. (2019). *TLS/SSL with Dart*. [online] Available at: https://dart-lang.github.io/server/tls-ssl.html [Accessed 25 Dec. 2019].