Mobile App Security Testing – Quick Reference

Proxy Setup

Make sure that your intercepting proxy is listening on the right interface – not just localhost.

On the mobile device, open WiFi settings and set the proxy host and port to your laptop's IP and port.

iOS: **Settings** -> **WiFi** -> **[Name]**. Tap **Configure Proxy**, select **Manual**, enter details, tap **Save**.
Android: **Settings** -> **Network & Internet** -> **Wi-Fi** -> **[Name]** -> **Edit** -> **Advanced** -> **Proxy** -> **Manual**, enter details, tap **Save**.

Download and trust the proxy's CA certificate – for Burp, navigate to http://burp/ on the device, tap **CA Certificate**

iOS: Open **Safari** on your device and navigate to http://burp. Tap the **CA Certificate** link, and follow the prompts to install the Configuration Profile. Open **Settings**, tap **Profile Downloaded**, then tap **Install** (it will ask a few times). Go to **Settings** -> **General** -> **About** -> **Certificate Trust Settings**, and enable full trust for **PortSwigger CA**.

Android: You may need to convert the certificate from DER (binary) to PEM (Base64) format, openssl can do this. Go to **Settings** -> **Security & Location** -> **Advanced** -> **Encryption & Credentials** -> **Install from SD Card**, and select the downloaded/sideloaded certificate.

Patching the Application

iOS:

```
      security find-identity -p codesigning -v
      objection patchipa -s [your-app].ipa -c hex-value
```

Android:

```
      objection patchapk -N -s [your-app].apk
```

Installing the Application

iOS:

> If you sign apps with a free account, you will need to trust the developer certificate under **Settings** -> **General** -> **Device Management**.
>
> Unzip the patched IPA file. Use ios-deploy to sideload and launch the app: `ios-deploy -b Payload/AppName.app -L`

Android:

> Allow **Install Unknown Apps** for **Files** under **Settings** -> **Apps & Notifications** -> **Special App Access**.
>
> Transfer or download the patched APK file, open **Files** and tap the APK file.

Common Objection Commands

Launch the Objection REPL with `objection explore` (USB devices) or `objection --network --host x.x.x.x explore` (remote TCP)

Show app-accessible filesystem locations: env
Use `cd` and `ls` to browse.
Retrieve a file: `file download [name-on-device] [name-on-laptop]`
Print contents: `file cat [name]`

SQLite:

> Open SQLite client: `sqlite connect [file]`
> Try `.schema`, `.tables`, or run SQL queries.
>
> BLOBs on iOS are often binary plists

Bypass Certificate Pinning: `[ios|android] sslpinning disable`

iOS:

> Decode and pretty-print a plist: `ios plist cat [file].plist`
> Dump KeyChain entries: `ios keychain dump`
> Dump UserDefaults: `ios nsuserdefaults get`
> Bypass weak TouchID/FaceID checks: `ios ui biometrics_bypass`