Setting up an android mobile test lab

BUG BOUNTY COURSE BY UNCLE RAT

Contents

1. Why we need certificate pinning bypass
2. What is frida.
3. Requirements
4. Setup and installation
5. Frida server setup
6. Burp suite configuration
7. Script injection to bypass cert. Pinning

Tips and tricks

We will use android 7.1, but the device needs to be rooted before it will accept the burp cert

Use android 6- if you do not want to root the device

You can use genymotion or android studio virtual device manager

Sometimes the frida script we point to in this course will not work

- This is because some programs use custom certificate pinning

Why we need certificate pinning bypass

Mobile app encrypts traffic for HTTPS

Uses certificate to do this

We can insert Burp certificate into android to capture requests

Developer protection: Certificate pinning

- App checks to make sure only it's own cert can encrypt traffic

Our solution: Certificate pinning bypass

- Overwrite the code in the APK that's responsible for checking cert
- Frida (https://frida.re/docs/hacking/)

What is frida?

" It's Greasemonkey for native apps, or, put in more technical terms, it's a dynamic code instrumentation toolkit. It lets you inject snippets of JavaScript or your own library into native apps on Windows, macOS, GNU/Linux, iOS, Android, and QNX. Frida also provides you with some simple tools built on top of the Frida API. These can be used as-is, tweaked to your needs, or serve as examples of how to use the API."

Requirements

Rooted device/emulator

- Connect real device via USB OR
- Download genymotion and create an emulator (https://www.genymotion.com/fun-zone/)
  - Install a new "Google pixel XL" device
  - At least android 7.1+

Frida packages for python

- python -m pip install Frida
- python -m pip install objection
- python -m pip install frida-tools

OR

- pip install Frida
- pip install objection
- pip install frida-tools

Requirements

Platform tools (ADB = Android device bridge)

- Download from https://dl.google.com/android/repository/platform-tools-latest-windows.zip
- Extract to your local drive

Download the following script

- Will be used to overwrite the certificate pinning mechanism
- https://codeshare.frida.re/@pcipolloni/universal-android-ssl-pinning-bypass-with-frida/

Setup and installation

Enable USB debugging on the emulator/device

- Go to Settings > Developer options
- Tap the option to enable USB debugging

Add the path where you extracted ADB to your path environment variable

- When you type a command, your OS will scan all of these paths and see if it can find an executable file in these folder with the same name as the command
- See next slide

Setup and installation – win path vars

Windows ADB adding to path var

- Copy the location of the ADB path (ex: C:\Users\Gebruiker\Downloads\platform-tools_r30.0.5-windows)
- Press the windows + r key
- Enter 'sysdm.cpl'
- Click the advanced tab
- In the **System variable** window, find the **Path** variable and click **Edit**:
- Click **New**
- Add the path you copied in step 1

Setup and installation – linux path vars

Linux ADB adding to path var

- Copy the location of the ADB path (ex: /users/user/home/downloads/adb)
- Change to your home directory.

```
cd $HOME
```

- Open the `.bashrc` file.
- Add the following line to the file. Replace the JDK directory with the name of your java installation directory.

```
export PATH=<LOCAL ADB DIR>:$PATH
```

- Save the file and exit.
- Use the `source` command to force Linux to reload the `.bashrc` file which normally is read only when you log in each time.

```
source .bashrc
```

Frida server setup

Find out the arch version of the device

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

Download the frida server version from the following url and extract it

https://github.com/frida/frida/releases/

frida-server-xxxx-android-x86.xz
frida-server-xxxx-android-x86_64.xz

Frida server setup

Push frida to the device and give it permissions to execute

```
Windows: adb push C:\ADB\frida-server /data/local/tmp
Linux: adb push ADB/frida-server /data/local/tmp
```

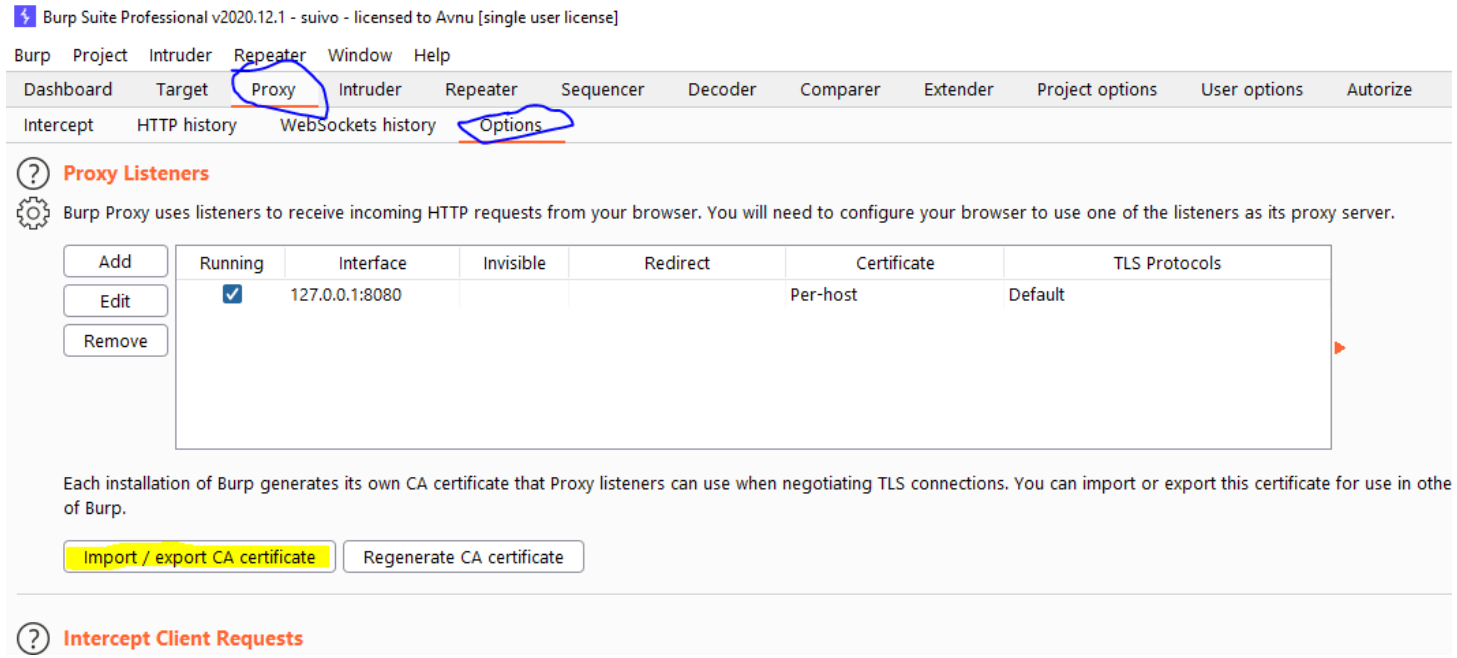
```
Both: adb shell chmod 777 /data/local/tmp/frida-server
```

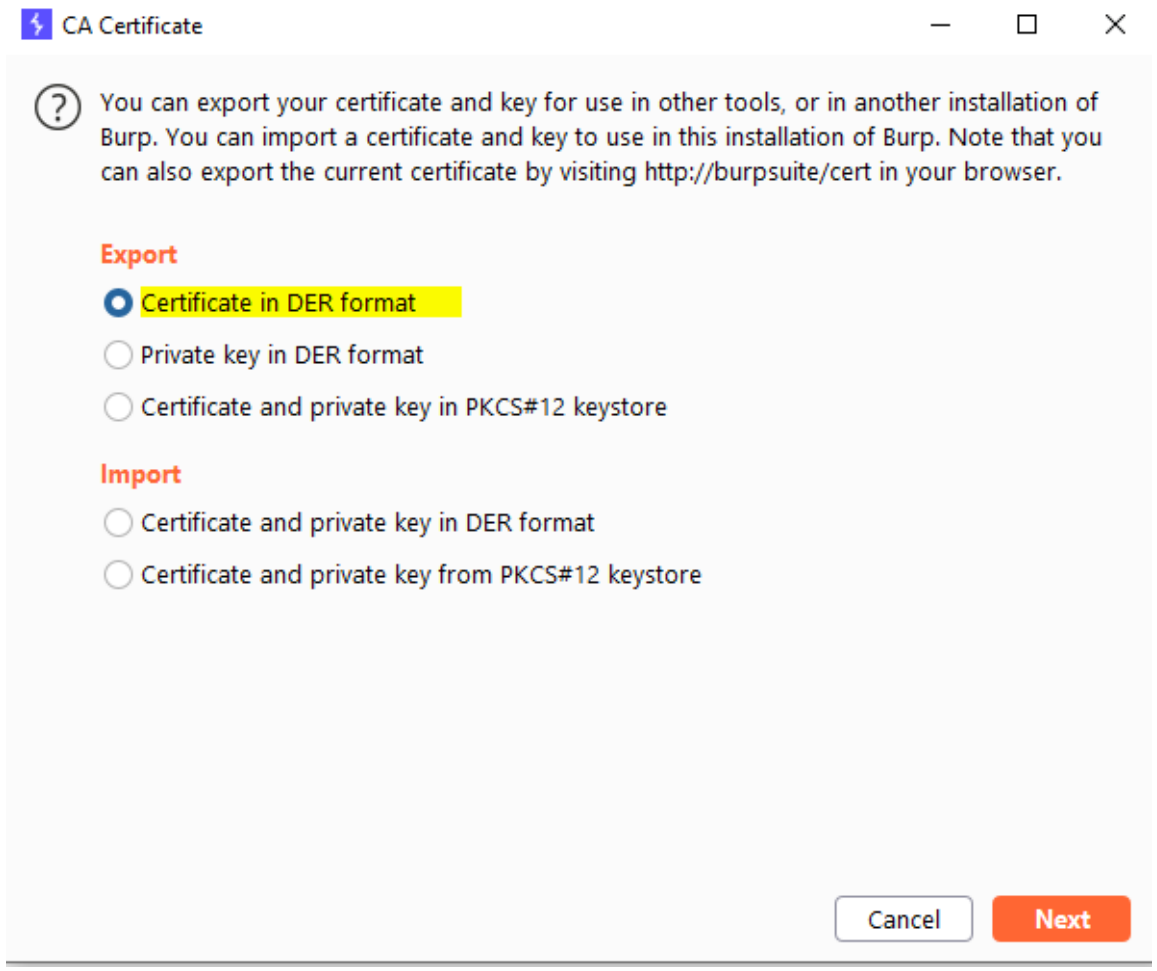
Setting up burp

Open burp suite and go to the proxy tab

Open the options tab

Click the export certificate button





Setting up burp

MAKE SURE YOU EXPORT YOUR CERTIFICATE IN DER FORMAT

Setting up burp

Push the certificate onto the device with ADB, we need this later for frida

// adb push <path to cacert.der> /data/local/tmp/cert-der.crt
adb push cacert.der /data/local/tmp/cert-der.crt

Script injection to bypass cert. Pinning

Push the script onto the device

```
//adb push <path_to_fridascript.js_folder> /data/local/tmp
adb push C:\ADB\fridascript.js /data/local/tmp
```

Check and run frida server on the device

```
adb shell /data/local/tmp/frida-server &
```

List all the running processes on the device

```
frida-ps -U
```

Script injection to bypass cert. Pinning

Locate your applications package name from the list

```
 795  com.android.settings
1247  com.android.smspush
 686  com.android.systemui
1116  com.genymotion.genyd
1111  com.genymotion.systempatcher
1062  com.google.android.ext.services
1275  com.google.android.gms
 770  com.google.android.gms.persistent
1994  com.google.android.gms.unstable
1092  com.google.process.gapps
3672  com.twitter.android
 105  debuggerd
 113  debuggerd:signaller
 260  diskiod
```

Script injection to bypass cert. Pinning

Hook frida script into your application

```
//frida -U -f <your_application_package_name> -l
<path_to_fridascript.js_on_your_computer> --no-paus
```

```
frida -U -f com.twitter.android -l D:\frida\fridascript.js --no-paus
```

BYPASSED

TRY IT ON HOPLR:

HTTPS://APP.INTIGRITI.COM/RESEARCHER/PROGRAMS/HOPLR/HOPLR/DETAIL