SSL Pinning Bypass for Android Application

Rooted device/emulator:

Need Rooted Device / Emulator

Python frida packages installation:

First Install Python on windows.
We need to install some python packages for frida server. For this enter following command in terminal:

```
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
```

Platform-tools (adb):

Download platform-tools for windows from the following the link:

https://dl.google.com/android/repository/platform-tools-latest-windows.zip

Download injection script:

https://codeshare.frida.re/@pcipolloni/universal-android-ssl-pinning-bypass-with-frida/

you can save this code as **fridascript.js** in same folder as adb.

Connect device to adb:

```
//adb connect <ip of device:port>
adb connect 192.168.1.190:5555
```

Download frida server for supported android device's arch version:

We need to download the frida server package for our android device according to our device's arch version.

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

To find out the arch version of the device, run following command.

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

To cut short download following if device configuration is the same as mentioned above:

```
frida-server-12.4.7-android-x86.xz
frida-server-12.4.7-android-x86_64.xz
```

Install the target application in the device.

Install your application whose SSL pinning has to be bypassed in our device. Open the application and keep it running in the background.

Push frida-server into device:

Now we need to push our frida-server file into device. Copy "*frida-server-12.4.7-android-x86.xz*" file in adb folder and run following command.

```
//adb push <path_of_frida_server_folder> space></data/local/tmp>
adb push C:\ADB\frida-server /data/local/tmp
```

Give permissions to frida-server:

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

Setup Burp Suite's:

https://support.portswigger.net/customer/portal/articles/1841101-configuring-an-android-device-to-work-with-burp

Pushing the proxy's CA Certificate:

Push the certificate into the device and into the same location as the frida-server, name it *cert-der.crt* (as this name and path has been already mentioned in fridascript.js to avoid any issues)

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

Push fridascript.js into device:

Copy fridascript.js into adb folder and run following command to push fridascript.js into 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 in device

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

This will run frida-server into device. Maybe you will not get any output of this command in terminal.

List all running processes on device:

Now, we need to find out id of our target application. We will list all running services on devices including your application process.

Open new terminal and type following command.

```
frida-ps -U
```

Locate your application's package name.

```
 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
```

Hook fridascript.js into target application:

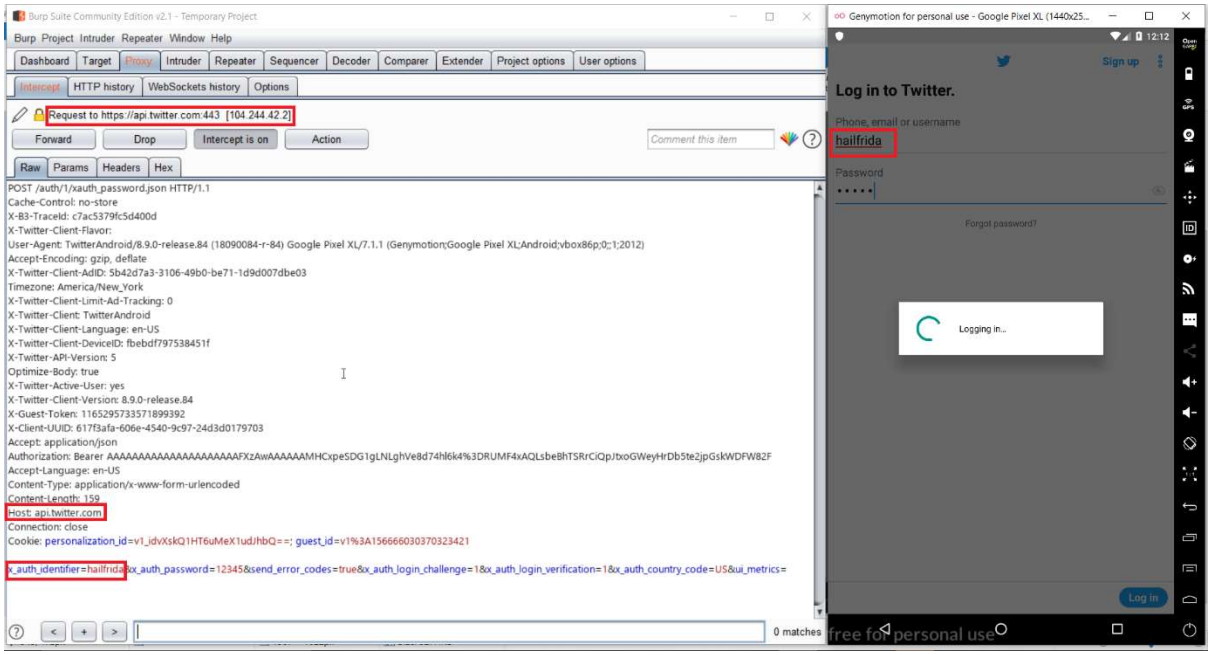
Finally, we will hook fridascript.js into the native application with the following command:

```
//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
```

```
D:\frida>frida -U -f com.twitter.android -l D:\frida\fridascript.js --no-paus
     ____
    / _  |   Frida 12.6.12 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at http://www.frida.re/docs/home/
Spawned `com.twitter.android`. Resuming main thread!
[Genymotion Google Pixel XL::com.twitter.android]->
[.] Cert Pinning Bypass/Re-Pinning
[+] Loading our CA...
[o] Our CA Info: CN=PortSwigger CA, OU=PortSwigger CA, O=PortSwigger, L=PortSwigger, ST=PortSwigger, C=PortSwigger
[+] Creating a KeyStore for our CA...
[+] Creating a TrustManager that trusts the CA in our KeyStore...
[+] Our TrustManager is ready...
[+] Hijacking SSLContext methods now...
[-] Waiting for the app to invoke SSLContext.init()...
[o] App invoked javax.net.ssl.SSLContext.init...
[+] SSLContext initialized with our custom TrustManager!
```

Bypassed!!

Once all things go smooth, all traffic of the target app will get intercepted into Burp Suite. We need to keep frida server on as long as we are intercepting traffic into Burp Suite.



1. ADB Deamon failed to connect

If you are getting error like this:
adb devices
adb server is out of date. killing...
cannot bind 'tcp:5037'
ADB server didn't ACK
failed to start daemon
error:

i. Open environment System properties>>Advanced>>Environment Variables
ii. Click on path and delete entry of C:/Android or path where adb tools are pointed
iii. Copy all platform tools into genymotion>>tools folder
iv. Create new path and add path of genymotion>>tools folder.

2. Frida / pip is not recognized as an internal or external command

i. Open environment System properties>>Advanced>>Environment Variables
ii. Create new path and add path of Python>>script folder

3. Arm translation error while installing application into device.

i. Download arm translation file from here
https://androidfilehost.com/?fid=23252070760974384
ii. Drag and drop file into device emulator or flash this file from recovery if you are using physical device

iii. Restart device and you will be able to drag and drop install target application

4. Failed to spawn: the ‘argv’ option is not supported when spawning Android apps

Check your fridascript.js path on your computer. Path maybe incorrect. You have to give the absolute path of fridascript.js file. Absolute path?

5. Started frida server but not able to list services

Disconnect and re-connect wifi in device.