Easy Way to Bypass SSL Pinning with Objection & Frida [Beginner Friendly]

<u>Aan</u>

After creating an article "Let's Bypass SSL Pinning By Manual Modification", I've thought of making an easy way for beginner in learning how to bypass ssl pinning with **uncomplicated targets** and using automatic method. In this article I will talk about objection for bypassing ssl pinning.

Objection is a runtime mobile exploration toolkit, powered by <u>Frida</u>, built to help you assess the security posture of your mobile applications, without needing a jailbreak.

- Supports both iOS and Android.
- Inspect and interact with container file systems.
- Bypass SSL pinning.
- Dump keychains.
- Perform memory related tasks, such as dumping & patching.
- Explore and manipulate objects on the heap.
- And much, much more...

We need to install and setup Frida because objection uses Frida as the instrumentation. For more information about Frida, you can read at https://frida.re/

Install frida

I'm using virtualenv in MacOS, but you can follow as your environment. Install frida-tools from pip

```
pip -V
pip 22.1.2 from /Users/petruknisme/.venv/lib/python3.9/site
python -V
Python 3.9.13 pip install frida-tools
Collecting frida-tools
   Using cached frida_tools-11.0.0-py3-none-any.whl
Requirement already satisfied: prompt-toolkit<4.0.0,>=2.0.0
Requirement already satisfied: colorama<1.0.0,>=0.2.7 in /I
Collecting frida<16.0.0,>=15.2.0
   Using cached frida-15.2.2-cp39-cp39-macosx_10_12_x86_64.\[
Requirement already satisfied: pygments<3.0.0,>=2.0.2 in /I
Requirement already satisfied: setuptools in /Users/petruknism
```

We've successfully installed frida-15.2.2 and frida-tools-11.0.0. For frida to work in our android/ios, we need to setup the frida-server.

Successfully installed frida-15.2.2 frida-tools-11.0.0

Installing collected packages: frida, frida-tools

Setup frida-server in Android

Go to https://github.com/frida/frida/frida/releases and choose the frida-server-version-android-arm.xz, change the version with your installed frida version like I bold above. Frida client and server must have the same-version.

Download the frida to host machine

> wget https://github.com/frida/frida/releases/download/15

Length: 6824388 (6.5M) [application/octet-stream] Saving to: 'frida-server-15.2.2-android-arm64.xz'frida-server-

Extract xz file

```
> xz -d frida-server-15.2.2-android-arm64.xz
> ls | grep frida
frida-server-15.2.2-android-arm64
> mv frida-server-15.2.2-android-arm64 frida-server
```

After downloading, rename the file to frida-server, the next step is to copy the file to android. In my device, I copy to /data/local/tmp/

```
> adb push frida-server /data/local/tmp
frida-server: 1 file pushed, 0 skipped. 23.1 MB/s (1997298)
```

Run the frida-server

```
> adb shell
jasmine_sprout:/ $ su
jasmine_sprout:/ # cd /data/local/tmp/
jasmine_sprout:/data/local/tmp # chmod +x frida-server
jasmine_sprout:/data/local/tmp # ./frida-server &
```

If running sucessfully, we can check with ps and grep

```
1|jasmine_sprout:/data/local/tmp # ps | grep frida
root 23299 23280 39280 19640 poll_schedule_time
```

We've done with setuping frida in android. Now, we will setup the objection.

Install Objection

```
> pip install -U objection
Collecting objection
Downloading objection-1.11.0.tar.gz (327 kB)
```

Successfully built objection
Installing collected packages: objection
Successfully installed objection-1.11.0

Testing if Objection is working properly with frida-server from our android

> objection --gadget "com.android.settings" device-type
Using USB device `Mi A2`

Agent injected and responds ok!

Connection: USB

Name: com.android.settings

System: jasmine_sprout

Model: xiaomi Version: 10

Asking jobs to stop...

Unloading objection agent...

From the above message, it tells us that it's successfully injected the agent and showing device information.

In this article, we will use a sample application to learn how objection bypass the ssl pinning. Download and install the application from https://github.com/aancw/android-ssl-pinning-signed-demo

> wget https://github.com/aancw/android-ssl-pinning-signed-Performing Streamed Install Success

When we open the application, the button color will be purple. But when we click the button, it will change the color to red(fail)/green(success) when doing https request to the pinned

server.

SSL Pinning Demo app

Proxy Setup

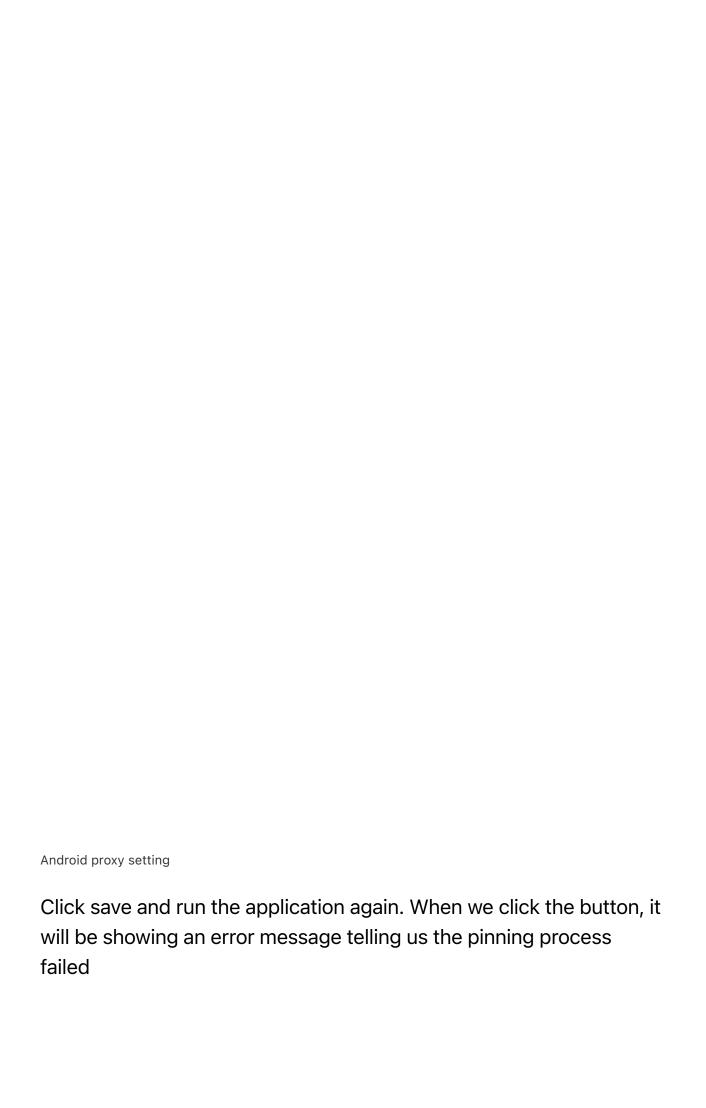
We need to test the app using a proxy to know the app is fail or success when doing http request. Run burp suite app and set listen address

Setting	Burp	Proxy
---------	------	-------

Explanation:

- Setting bind port to 8090, this port will be used in android setting
- Setting listen address to our LAN IP

Setting the android wifi with manual proxy setting



The main topic for this article is bypassing ssl pinning with different pinning method/library like okhttp, trustkit, volley and much more.

Bypass SSL Pinning

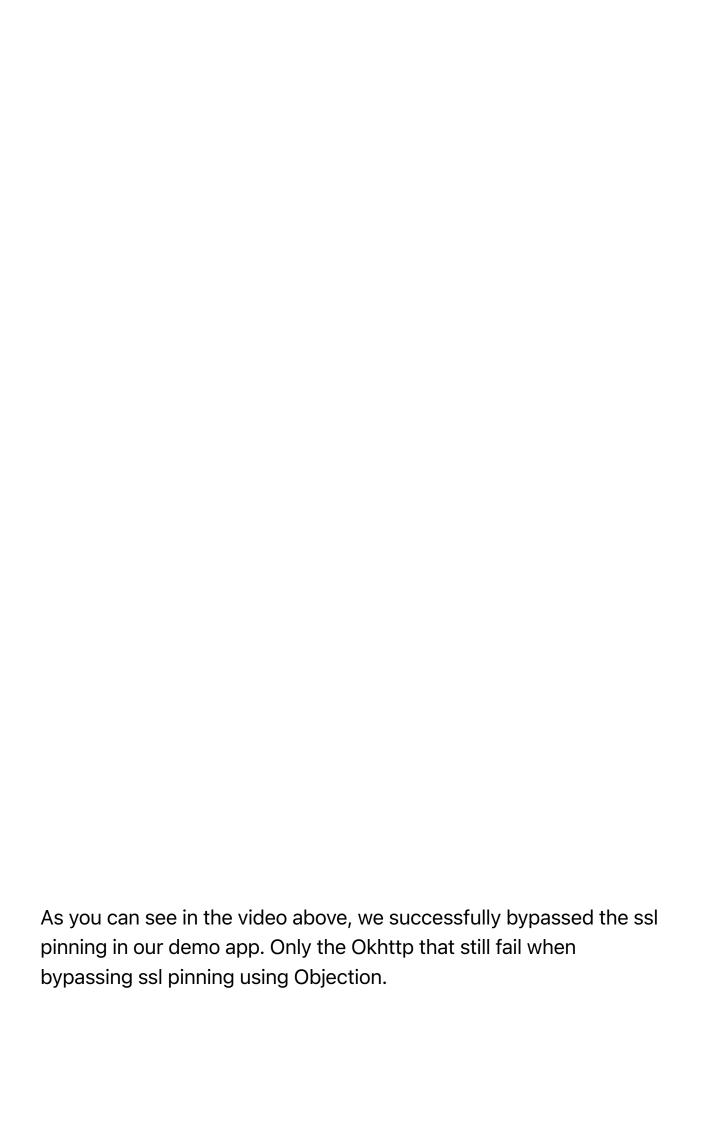
Check the target running app package name

```
frida-ps -Uia | grep pinning
21332 SSL Pinning Demo tech.httptoolkit.pinning_demo
```

Running objection and injecting the agent. Our target application will be running again with the new session. When we click the button, it will still be showing us an error message when doing pinning request.

For doing ssl pinning bypass, we can use the command android sslpinning disable

```
tech.httptoolkit.pinning_demo on (xiaomi: 10) [usb] # andro (agent) Custom TrustManager ready, overriding SSLContext.in (agent) Found okhttp3.CertificatePinner, overriding Certificagent) Found okhttp3.CertificatePinner, overriding Certificagent) Found com.android.org.conscrypt.TrustManagerImpl, (agent) Found com.android.org.conscrypt.TrustManagerImpl, (agent) Registering job 751373. Type: android-sslpinning-di
```



As far as I know, the objection method can bypass ssl pinning for Okhttp. But for this target, it seems missing some function to hook. Let's start to dig into the problem.

Debugging with Frida

We will use Frida to debug our demo application to know what is causing objection so can't bypass the ssl pinning. From our last experiment above, it tells us that application throwing an error at javax.net.ssl.SSLPeerUnverifiedException.

Result from pidcat.py

Frida supports <u>Javascript API</u> for instrumenting the binary. So we can patch and modify the binary at runtime with javascript. In this case, we will create a javascript file to debug our application and try to bypass the ssl pinning that was missed by objection.

Because we are already spawned the application before, so we will use attach mode for frida to launch the script. More information about spawning vs attaching for frida: https://summit-labs.frida.ninja/frida-tool-reference/frida

Get PID of application that running with objection in previous section

> frida-ps -Uia | grep pinning_demo

Our application PID is 8988. Note this number for later.

Creating our Frida Javascript

Create an instance of javax.net.ssl.SSLPeerUnverifiedException and initialize it. This instance is from our previous error.

```
Java.perform(function () {const UnverifiedCertError = Java
});
```

We will hook the constructor of a class using \$init as the method name and hook Method in a class and set the implementation to the custom code.

```
UnverifiedCertError.$init.implementation = function (str) .
};
```

In the above function, we will add java stack trace. So we know which function/method causing the previous error

```
UnverifiedCertError.$init.implementation = function (str)
};
```

After that we will filter the stack trace to only index the javax.net.ssl.SSLPeerUnverifiedException

```
const callingFunctionStack = stackTrace[exceptionS<sup>-</sup>
};
```

Then, we will map the className and methodName that causing the problem

We are done for debugging the stack trace. The final code for debugging the stack trace:

Save the file as ssl-error.js or whatever you name it. Run frida with our javascript

```
> frida -U -l <js_file> --no-pause -p <PID_Number>
```

Frida will attach the script to our running process

```
Frida 15.1.17 - A world-class dynamic instrum(
| (_| |
> _ | Commands:
/_/ |_| help -> Displays the help system
object? -> Display information about 'ol
exit/quit -> Exit

More info at https://frida.re/docs/home/[Mi A/
```

Switch	to androi	d ssl	pinning	demo	application	and t	then	click	Okhttp
Pinning	g Request	. This	s is what	: I get f	rom frida				

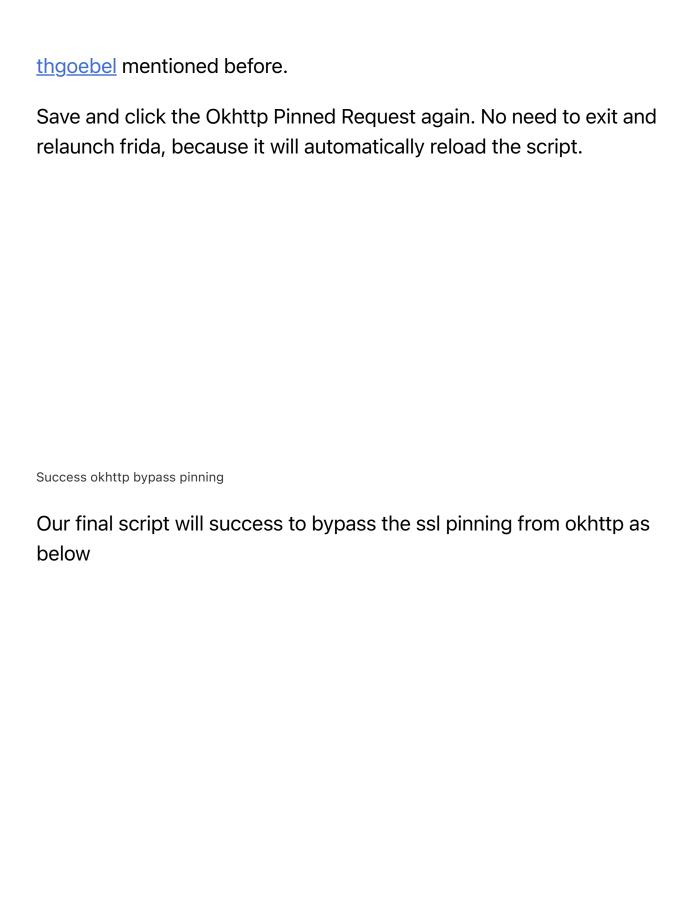
Frida attach

From the message above, it is clear that error message is throwing by okhttp3.CertificatePinner->check\$okhttp.

After hours googling, I found that someone experience this issue too when using objection

https://github.com/sensepost/objection/issues/475

From the above message, he suspect non-obfuscated Kotlin apps need little change for function hook in the script. Our app is non- obfuscated kotlin app too, so it's relevant issue.
Creating Bypass Script for non-obfuscated Kotlin
We will add the script from objection but change the function as





know.

Feedback are always welcome because it's valuable thing and there will be always room for improvement. Thank you!