SSL-Pinning Bypass

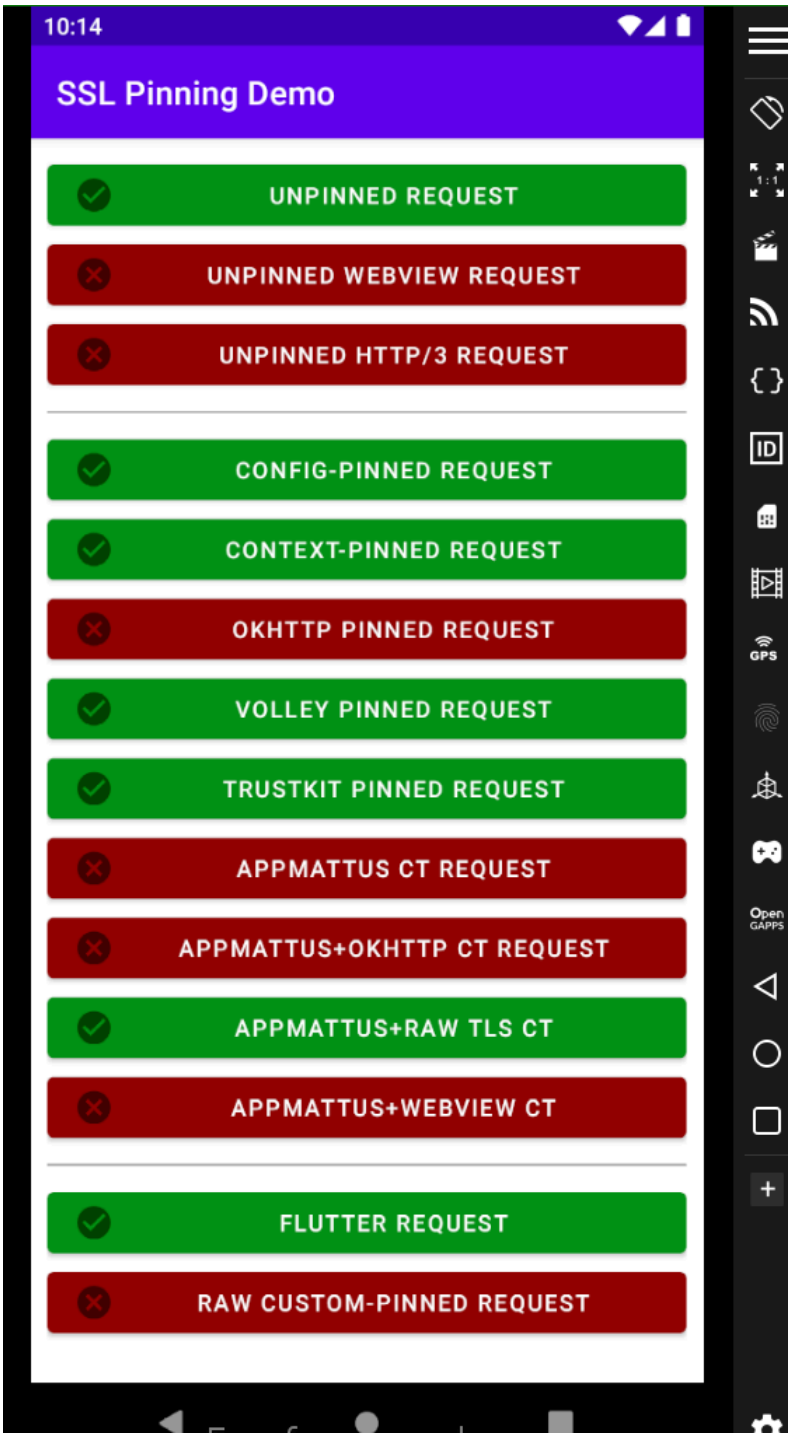
```
adb shell "settings put global http_proxy 192.168.194.97:8080"
```

Method 1

Using Burp Cert

```
adb push burpca-cert-der.crt /data/local/tmp/cert-der.crt
```

```
frida -U \
  --codeshare "pcipolloni/universal-android-ssl-pinning-bypass-with-frida" \
  -f "com.example.pinned"
```



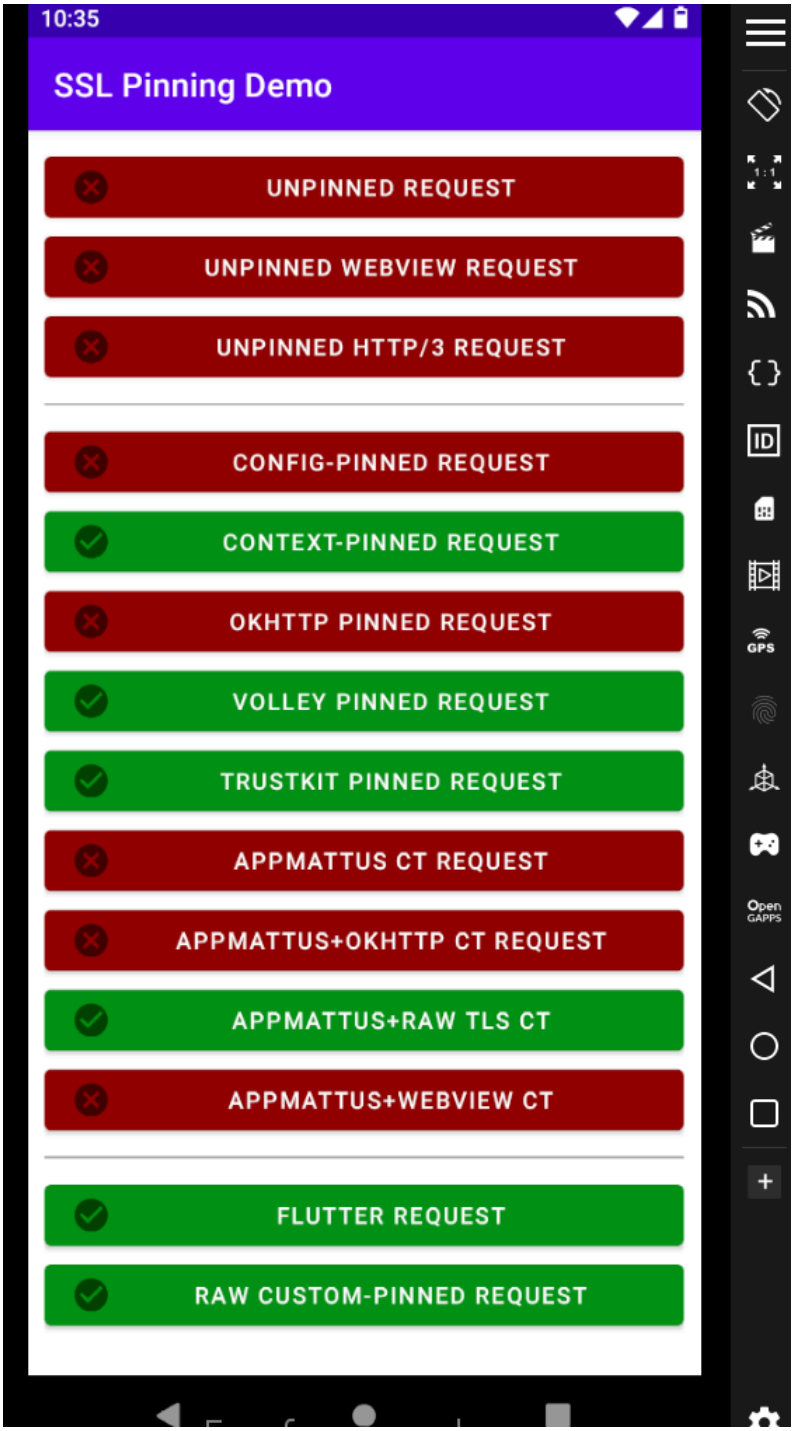
Method2

Generic Script to bypass SSL context and okhttp

```
Java.perform(function () {

// Helper function to bypass SSL pinning by returning a custom TrustManager

function bypassSSL() {

    var X509TrustManager = Java.use('javax.net.ssl.X509TrustManager');

    var SSLContext = Java.use('javax.net.ssl.SSLContext');


    var TrustManager = Java.registerClass({
```

```
        name: 'org.frida.TrustManager',
        implements: [X509TrustManager],
        methods: {
            checkClientTrusted: function (chain, authType) {},
            checkServerTrusted: function (chain, authType) {},
            getAcceptedIssuers: function () { return []; }
        }
    });

    var TrustManagers = [TrustManager.$new()];
    var SSLContextInit = SSLContext.init.overload('[Ljavax.net.ssl.KeyManager;', '[Ljavax.net.ssl.TrustManager;',
'java.security.SecureRandom');
    SSLContextInit.implementation = function (keyManager, trustManager, secureRandom) {
        SSLContextInit.call(this, keyManager, TrustManagers, secureRandom);
    };

    console.log('SSL pinning bypass active');
}

// Bypass okhttp3
try {
    var OkHttpClient = Java.use('okhttp3.OkHttpClient');
    var Builder = OkHttpClient.Builder;
    Builder.sslSocketFactory.overload('javax.net.ssl.SSLSocketFactory',
'javax.net.ssl.X509TrustManager').implementation = function (sslSocketFactory, trustManager) {
        var newTrustManager = TrustManager.$new();
        return this.sslSocketFactory.call(this, sslSocketFactory, newTrustManager);
    };
    console.log('Bypassed OkHttpClient SSL pinning');
} catch (e) {
    console.log('Failed to bypass OkHttpClient: ' + e);
}

// Bypass TrustManager
```

```
try {
    bypassSSL();
} catch (e) {
    console.log('Failed to bypass TrustManager: ' + e);
}

});
```



Method3

convert .cer to .crt id necessary

```
openssl x509 -inform DER -in ca.cer -out ca.crt
```

Get hash of the certificate

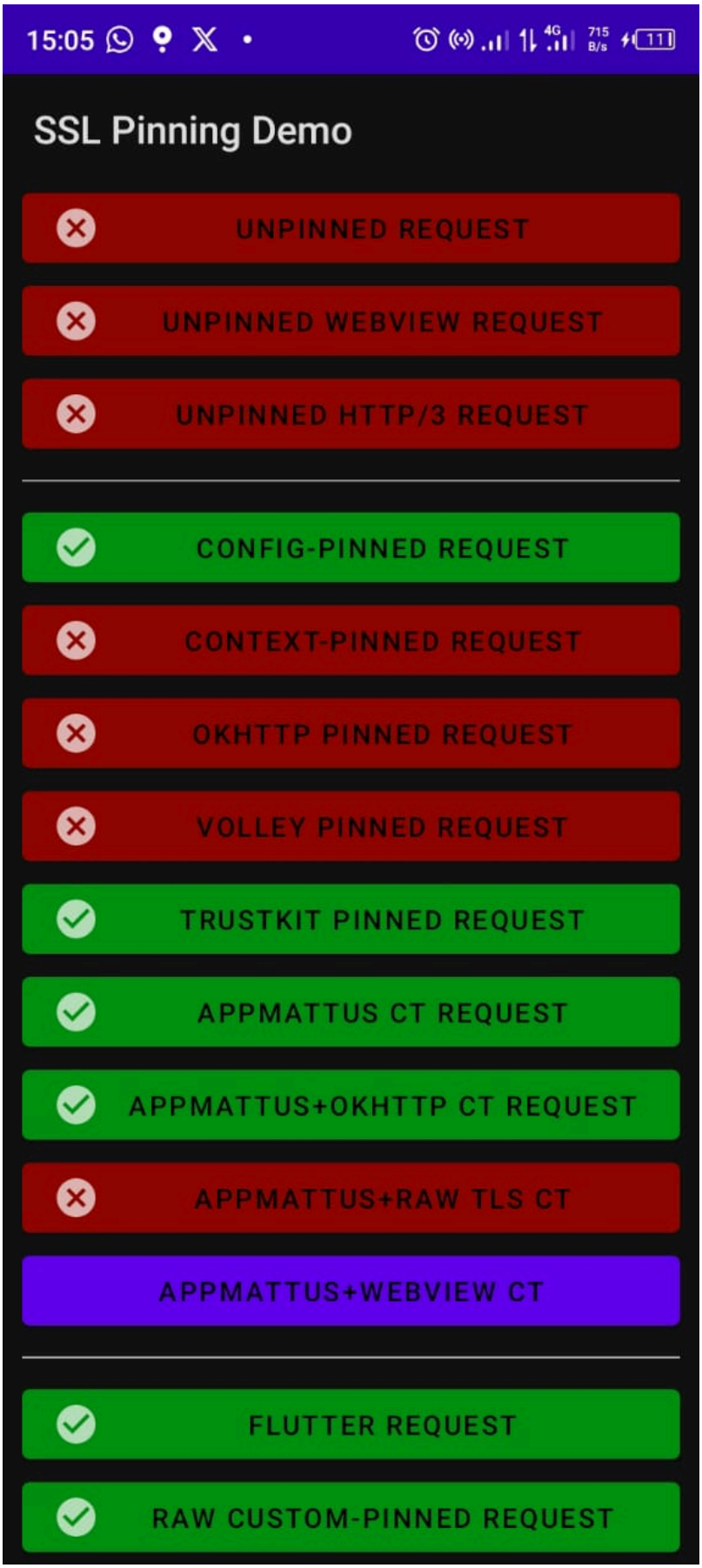
```
openssl x509 -in ca.crt -pubkey -noout | openssl pkey -pubin -outform der | openssl dgst -sha256 -binary | openssl enc -base64
```

Replace the hash in the app with this custom one, then recompile the app and sign it

we also have to add the cert to the app and edit the xml file

```
<?xml version="1.0" encoding="utf-8"?>

<network-security-config>

    <domain-config>

        <domain includeSubdomains="false">sha256.badssl.com</domain>

        <pin-set>

             <pin digest="SHA-256">C5+lpZ7tcVwmwQIMcRtPbsQtWLABXhQzejna0wHFr8M=</pin>

            <pin digest="SHA-256">ABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFABCDEFA</pin>

        </pin-set>

        <trust-anchors>

            <certificates src="@raw/ca" />

        </trust-anchors>

        <trustkit-config enforcePinning="true">

            <report-uri>http://trustkit-report-url.test</report-uri>

        </trustkit-config>

    </domain-config>

    <domain-config>

        <domain includeSubdomains="false">rsa4096.badssl.com</domain>

        <pin-set>

            <pin digest="SHA-256">1jH9aw6Fmyc/T2ixHmOi46tPeefDdW7ZPzO78arW1Kc=</pin>

        </pin-set>

        <trust-anchors>

            <certificates src="@raw/ca" />

        </trust-anchors>

    </domain-config>

</network-security-config>
```

```
java -jar C:\tools\apktool\uber-apk-signer-1.3.0.jar -a .\hash-patched.apk
```



Method 4

Using reflutter to patch flutter apps

Method 5

Using Objection