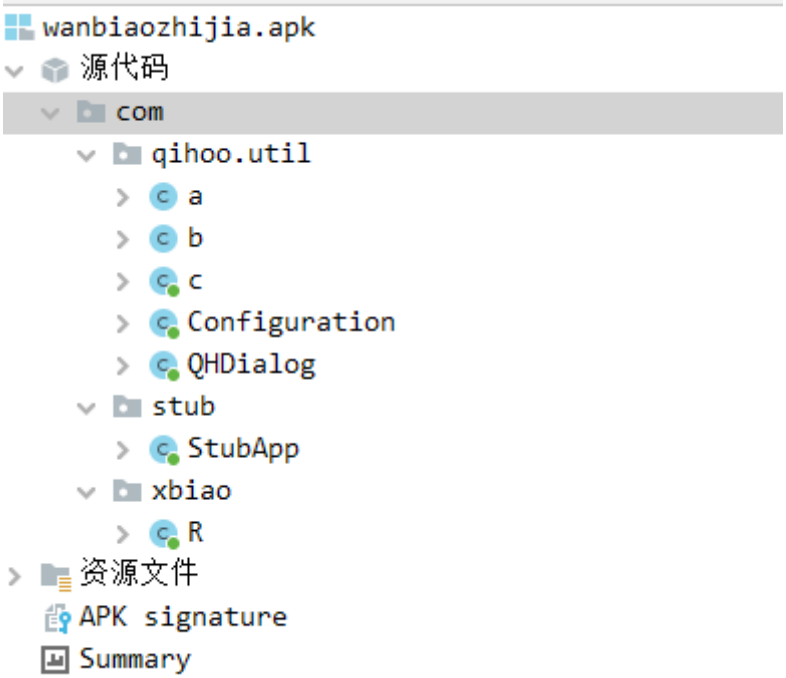
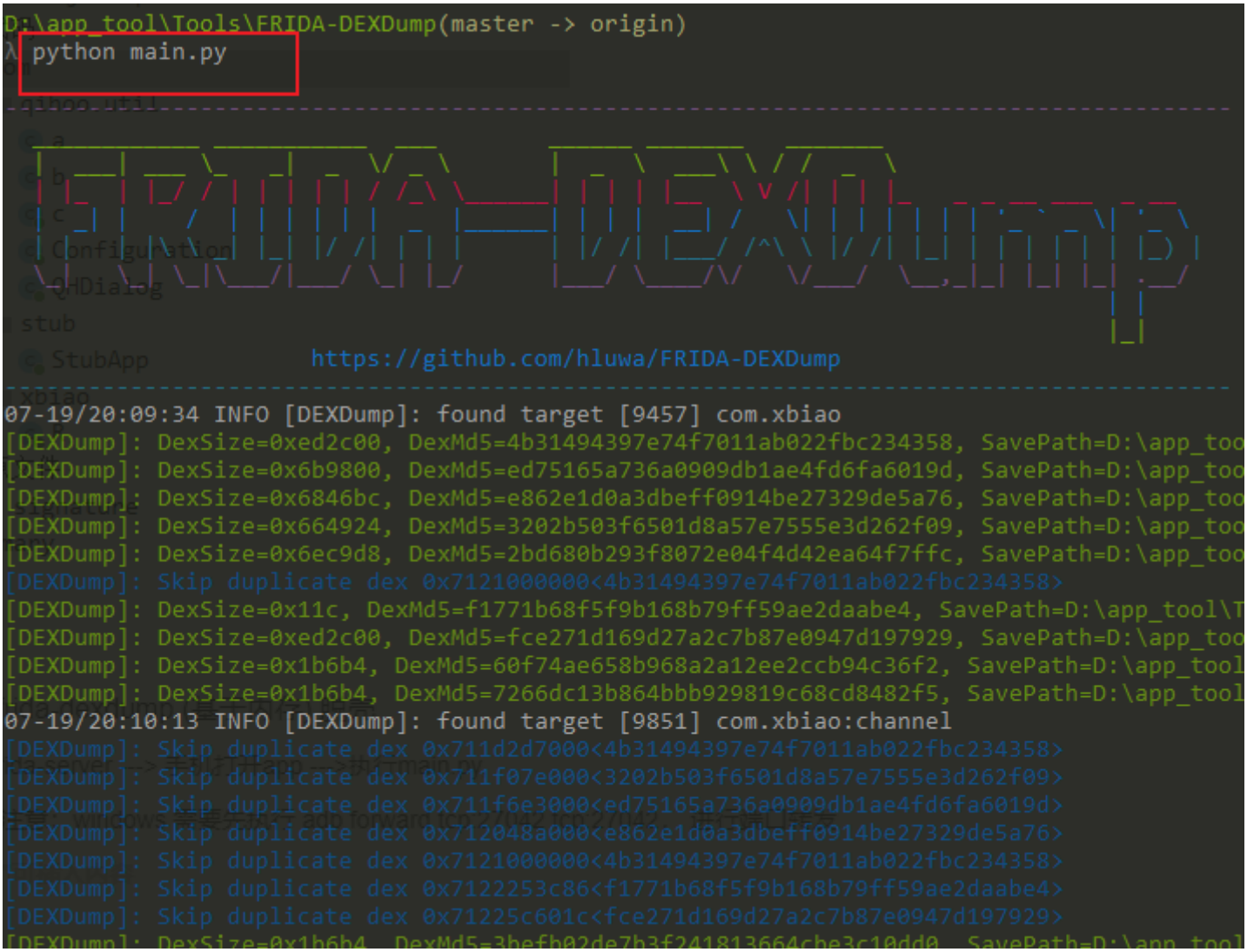
APP分析

通过jadx-gui 打开app 看到如下图所示，很明显该APP通过 “ **奇虎360** ” 加固



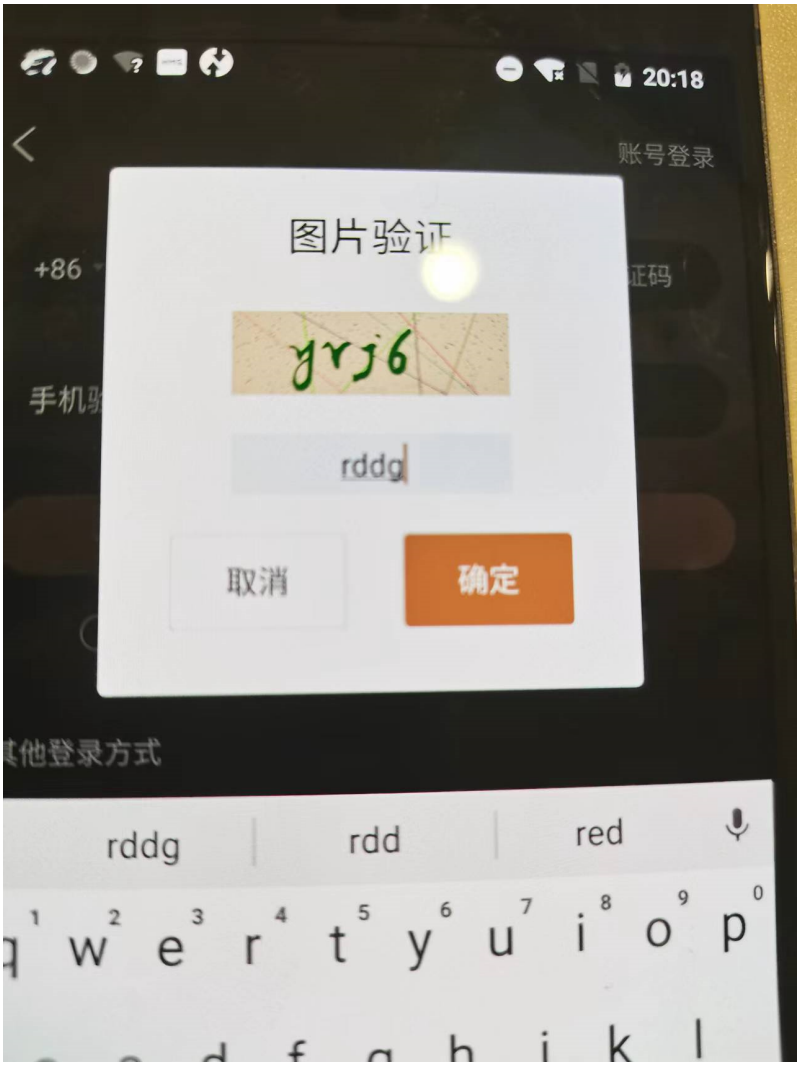
脱壳

使用frida-dexdump (基于内存) 脱壳， 启动frida-server ---> 手机打开app --->执行main.py, 注意：windows 需要先执行 adb forward tcp:27042 tcp:27042， 进行端口转发



后续把 脱壳出来的dex 一起拖入jadx-gui 即可看到APP源代码

抓包



200	POST	android.xbiao.com	/apps/AndroidXbiao/android-Xbiao-5_3-GOOGLE-1080_1794/user/sendCode/	167 ms
200	POST	android.xbiao.com	/apps/AndroidXbiao/android-Xbiao-5_3-GOOGLE-1080_1794/user/sendCode/	159 ms

Filter: sendCode

Overview | Contents | Summary | Chart | Notes

Name	Value
PHPSESSID	58r4c5e3lilvg4cbgtikrid9q6
fontSize	15
publicKey	u9J7A00rHJO69ZF4kCjD4XG3NQ4FENcrkjaVrUM0Dkw%3D

Headers | Cookies | Text | Hex | Form | Raw

1 MowTw8j1dq7WO5q3ids+jNyoO//FxDNAuaEMwW+U4d50OziRjXkCy/8+KOVI6zrtdy45t/8U6ASLDH+XytHNVIZ98/ZbjaD/P+rQyuXviD0=

目标是对 提交验证码的请求 返回结果进行解密

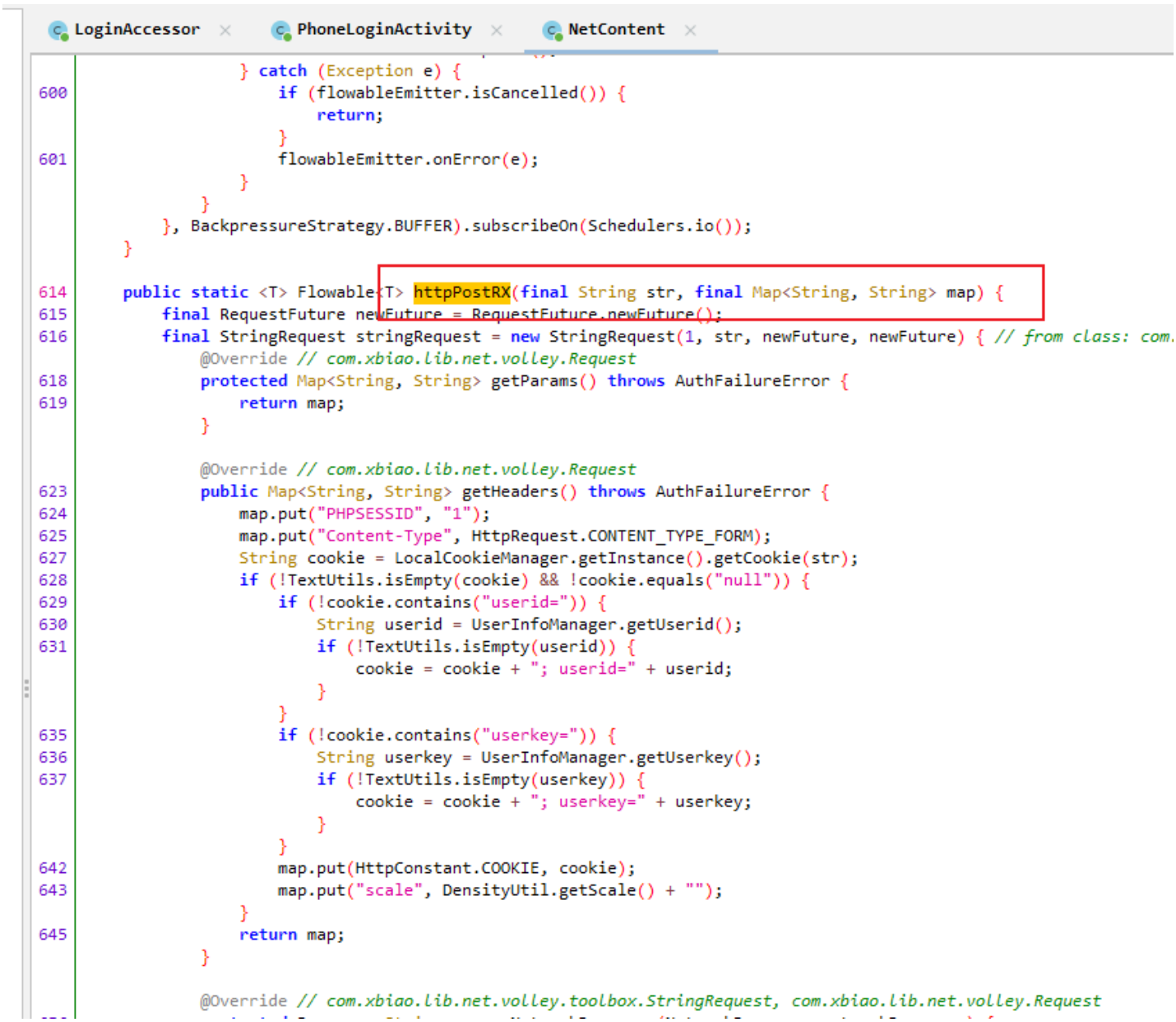
MowTw8j1dq7WO5q3ids+jNyoO//FxDNAuaEMwW+U4d50OziRjXkCy/8+KOVI6zrtdy45t/8U6ASLDH+XytHNVIZ98/ZbjaD/P+rQyuXviD0=

看起来像base64 加密的结果，其实不是

具体分析

全局搜索 sendCode

以查找用例的方式深入代码查看， 最终找到发送请求的位置



通过HOOK 该类下所有方法 查找关键点

frida -UF com.xbiao -l frida_wanbiaozhijia.js --no-pause

frida_wanbiaozhijia.js 参考代码

搜索返回内容 定位所在方法位置

```
Terminal:  Local  +
        at java.util.concurrent.FutureTask.run(FutureTask.java:266)
        at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:301)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1162)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:636)
        at java.lang.Thread.run(Thread.java:764)

com.xbiao.utils.net.NetContent.decryptResponse(java.lang.String,java.lang.String,)
argument:"MowTw8j1dq7WO5q3ids+jNyoO//FxDNAuaEMwW+U4d50OziRjXkCy/8+KOVI6zrtdy45t/8U6ASLDH+XytHNVIZ98/ZbjaD/P+rQyuXviD0="
argument:"https://android.xbiao.com/apps/AndroidXbiao/android-Xbiao-5_3-GOOGLE-1080_1794/user/sendCode/"
Error: can't decode byte 0xc0 in position 68
    at frida/runtime/core.js:144
    at frida/node_modules/frida-java-bridge/lib/env.js:607
    at frida/node_modules/frida-java-bridge/lib/class-factory.js:1239
    at input:1
```

再HOOK 对应的方法查看返回内容

```
var NetContent = Java.use("com.xbiao.utils.net.NetContent");
NetContent["decryptResponse"].implementation = function (str, str2) {
    console.log('decryptResponse is called' + ', ' + 'str: ' + str + ', ' + 'str2: ' + str2);
    var ret = this.decryptResponse(str, str2);
    console.log('decryptResponse ret value is ' + ret);
    return ret;
};
```

```
enter
decryptResponse is called, str: MowTw8j1dq7WO5q3ids+jNyoO//FxDNAuaEMwW+U4d50OziRjXkCy/8+KOVI6zrtdy45t/8U6ASLDH+XytHNVIZ98/ZbjaD/P+rQyuXviD0=, str2: https://android.xbiao.com
dXbiao/android-Xbiao-5_3-GOOGLE-1080_1794/user/sendCode/
decryptResponse ret value is {"status":-1,"message":"\u56fe\u5f62\u9a8c\u8bc1\u7801\u9519\u8bef"}
[Google Pixel::腕表之家]->
```

```
/* JADX INFO: Access modifiers changed from: private */
public static String decryptResponse(String str, String str2) {
    String secretKey = LocalCookieManager.getInstance().getSecretKey();
    String decrypt = (TextUtils.isEmpty(str) || TextUtils.isEmpty(secretKey)) ? "" : AESedeUtil.decrypt(secretKey, str);
    try {
        decrypt = changeCharset(decrypt, "utf-8");
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    String trim = decrypt.trim();
    if (TextUtils.isEmpty(trim)) {
        trim = "";
    }
    Log.i("liuguangyou", "url:" + str2.replace(NewsAccessor.BASE_HEAD_URL, "") + "  最终结果response**************:" + trim);
    return trim;
}
```

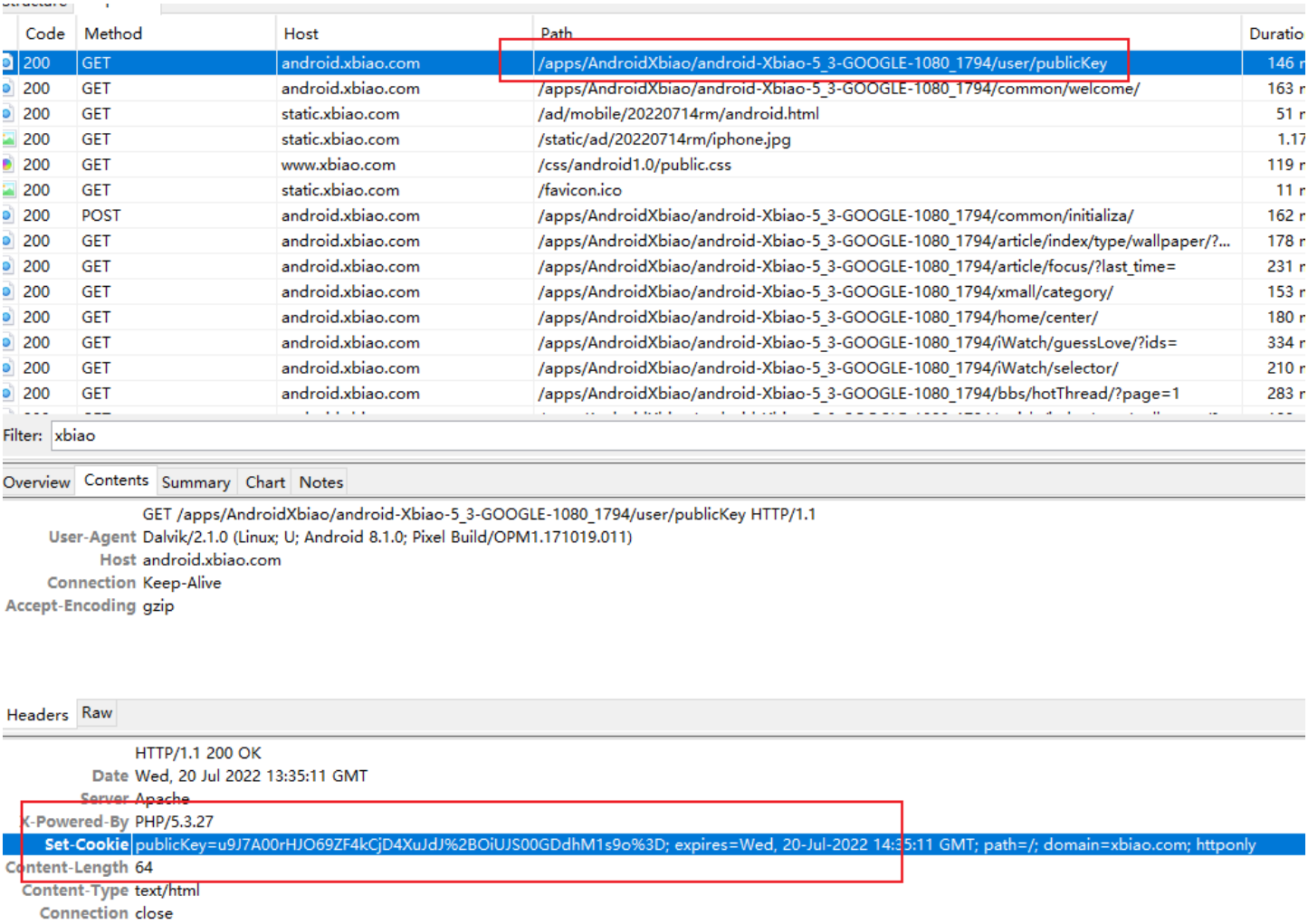
分析解密方式

```
/* JADX INFO: Access modifiers changed from: private */
public static String decryptResponse(String str, String str2) {
    String secretKey = LocalCookieManager.getInstance().getSecretKey();
    String decrypt = (TextUtils.isEmpty(str) || TextUtils.isEmpty(secretKey)) ? "" : AESedeUtil.decrypt(secretKey, str);
    try {
        decrypt = changeCharset(decrypt, "utf-8");
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    String trim = decrypt.trim();
    if (TextUtils.isEmpty(trim)) {
        trim = "";
    }
    Log.i("liuguangyou", "url:" + str2.replace(NewsAccessor.BASE_HEAD_URL, "") + "  最终结果response**************:" + trim);
    return trim;
}
```

secretKey 是请求Cookie publicKey 的前5后11 组成

```
public void subCookieSecretKey(String str) {
    String decode;
    int indexOf;
    if (TextUtils.isEmpty(str) || (indexOf = (decode = URLDecoder.decode(str.replaceAll(":eq:", "; "))).indexOf(this.searchCookiePubli
        return;
    }
    int indexOf2 = decode.indexOf(f.b, indexOf);
    if (indexOf2 == -1) {
        indexOf2 = decode.length();
    }
    String replace = decode.substring(indexOf, indexOf2).trim().replace(this.searchCookiePublicKey, "");
    String substring = replace.substring(0, 5);
    String substring2 = replace.substring(replace.length() - 11, replace.length());
    this.secretKey = substring + substring2;
    if (TextUtils.isEmpty(this.secretKey)) {
        this.secretKey = SharedPreferencesPublicKeyUtil.getStringValueFromSP(SharedPreferencesPublicKeyUtil.SECRET_KEY);
    }
    SharedPreferencesPublicKeyUtil.setStringDataIntoSP(SharedPreferencesPublicKeyUtil.SECRET_KEY, this.secretKey);
}
```

Cookie publicKey 是在如下请求的set-cookie 中



Code	Method	Host	Path	Duratio
200	GET	android.xbiao.com	/apps/AndroidXbiao/android-Xbiao-5_3-GOOGLE-1080_1794/user/publicKey	146 r
200	GET	android.xbiao.com	/apps/AndroidXbiao/android-Xbiao-5_3-GOOGLE-1080_1794/common/welcome/	163 r
200	GET	static.xbiao.com	/ad/mobile/20220714rm/android.html	51 r
200	GET	static.xbiao.com	/static/ad/20220714rm/iphone.jpg	1.17
200	GET	www.xbiao.com	/css/android1.0/public.css	119 r
200	GET	static.xbiao.com	/favicon.ico	11 r
200	POST	android.xbiao.com	/apps/AndroidXbiao/android-Xbiao-5_3-GOOGLE-1080_1794/common/initializa/	162 r
200	GET	android.xbiao.com	/apps/AndroidXbiao/android-Xbiao-5_3-GOOGLE-1080_1794/article/index/type/wallpaper/?...	178 r
200	GET	android.xbiao.com	/apps/AndroidXbiao/android-Xbiao-5_3-GOOGLE-1080_1794/article/focus/?last_time=	231 r
200	GET	android.xbiao.com	/apps/AndroidXbiao/android-Xbiao-5_3-GOOGLE-1080_1794/xmall/category/	153 r
200	GET	android.xbiao.com	/apps/AndroidXbiao/android-Xbiao-5_3-GOOGLE-1080_1794/home/center/	180 r
200	GET	android.xbiao.com	/apps/AndroidXbiao/android-Xbiao-5_3-GOOGLE-1080_1794/iWatch/guessLove/?ids=	334 r
200	GET	android.xbiao.com	/apps/AndroidXbiao/android-Xbiao-5_3-GOOGLE-1080_1794/iWatch/selector/	210 r
200	GET	android.xbiao.com	/apps/AndroidXbiao/android-Xbiao-5_3-GOOGLE-1080_1794/bbs/hotThread/?page=1	283 r

Filter: xbiao

Overview | Contents | Summary | Chart | Notes

GET /apps/AndroidXbiao/android-Xbiao-5_3-GOOGLE-1080_1794/user/publicKey HTTP/1.1
User-Agent: Dalvik/2.1.0 (Linux; U; Android 8.1.0; Pixel Build/OPM1.171019.011)
Host: android.xbiao.com
Connection: Keep-Alive
Accept-Encoding: gzip

Headers | Raw

HTTP/1.1 200 OK
Date: Wed, 20 Jul 2022 13:35:11 GMT
Server: Apache
X-Powered-By: PHP/5.3.27
Set-Cookie: publicKey=u9J7A00rHJO69ZF4kCjD4XuJdJ%2BOiUJS00GDdhM1s9o%3D; expires=Wed, 20-Jul-2022 14:35:11 GMT; path=/; domain=xbiao.com; httponly
Content-Length: 64
Content-Type: text/html
Connection: close

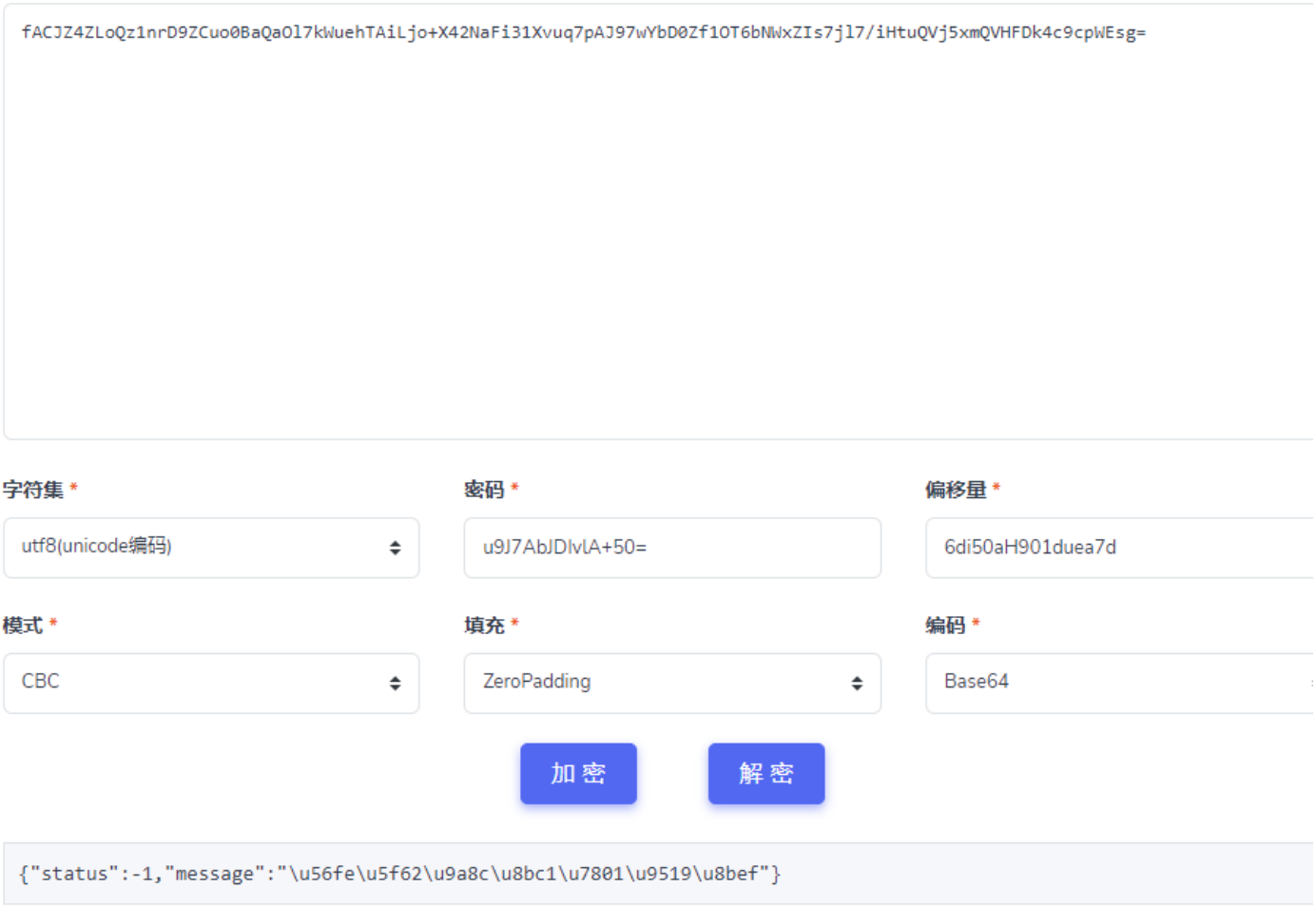
AES 偏移量， 加密模式，填充方式等

```java
public class AESedeUtil {
    private static final String INIT_VECTOR = "6di50aH901duea7d";

    @SuppressLint({"NewApi"})
    public static String encrypt(String str, String str2) {
        try {
            IvParameterSpec ivParameterSpec = new IvParameterSpec(INIT_VECTOR.getBytes("UTF-8"));
            SecretKeySpec secretKeySpec = new SecretKeySpec(str.getBytes("UTF-8"), "AES");
            Cipher cipher = Cipher.getInstance("AES/CBC/NOPADDING");
            cipher.init(1, secretKeySpec, ivParameterSpec);
            return java.util.Base64.getEncoder().encodeToString(cipher.doFinal(str2.getBytes()));
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }

    @SuppressLint({"NewApi"})
    public static String decrypt(String str, String str2) {
        try {
            IvParameterSpec ivParameterSpec = new IvParameterSpec(INIT_VECTOR.getBytes("UTF-8"));
            SecretKeySpec secretKeySpec = new SecretKeySpec(str.getBytes("UTF-8"), "AES");
            Cipher cipher = Cipher.getInstance("AES/CBC/NOPADDING");
            cipher.init(2, secretKeySpec, ivParameterSpec);
            return new String(cipher.doFinal(android.util.Base64.decode(str2, 0)));
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

结果



扩展Frida RPC 方式

直接调用APP 方法拿到返回结果

参考代码 frida_wanbiao_rpc.py frida_wanbiao_rpc.js

```python
import codecs

session = frida.get_remote_device().attach("com.xbiao")

time.sleep(1)

with codecs.open("frida_wanbiao_rpc.js", 'r', encoding="utf-8") as f:
    script = session.create_script(f.read())


def my_message_handler(message, payload):
    print(message)
    print(payload)


script.on("message", my_message_handler)
script.load()

print(script.exports.callfun())
```

注意 frida_wanbiao_rpc.js 里面的返回内容和当前手机的Cookie publicKey 一致才可以执行成功

```javascript
function callFun() {
    var result = "";
    Java.perform(function fn() {
        result = Java.use("com.xbiao.utils.net.NetContent").decryptResponse('fACJZ4ZLoQz1nrD9ZCuo0BaQaOl7kWuehTAi_jo+X4
            "dXbiao/android-Xbiao-5_3-GOOGLE-1080_1794/user/sendCode/")
        // console.log(result)
    })
    return result
}
rpc.exports = {
    callfun: callFun
}
```

本文中的代码 github地址 https://github.com/sunxhap/app_crack/tree/master/wanbiaozhijia

APP 自行下载 链接：https://pan.baidu.com/s/1U2m50_TassfnZdac4U6rxA 提取码：pv8l

参考

frida 官方api https://frida.re/docs/javascript-api/#stalker