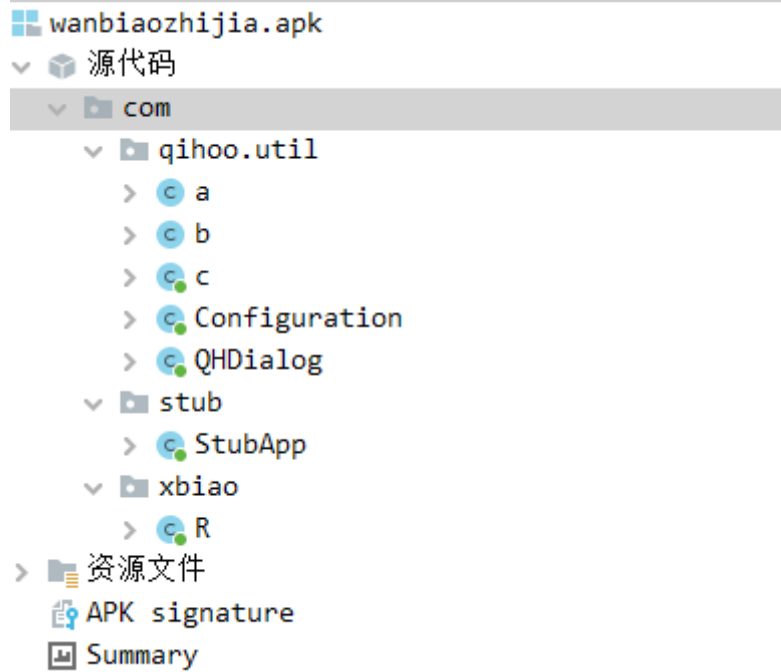


APP分析

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

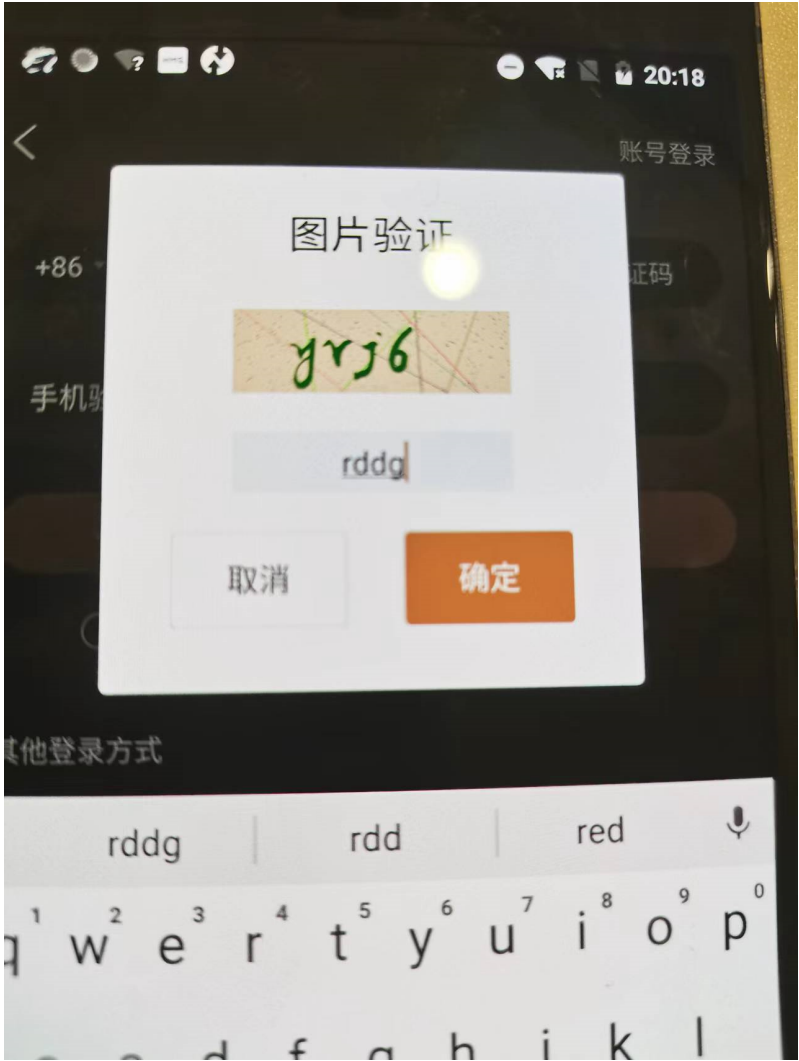
```
D:\app_tool\Tools\FRIDA-DEXDump(master -> origin)
```

```
python main.py
```

```
-----
a
b
c
d Configuration
e Dialog
f stub
g StubApp
https://github.com/hluwa/FRIDA-DEXDump
-----
xbiao
07-19/20:09:34 INFO [DEXDump]: found target [9457] com.xbiao
[DEXDump]: DexSize=0xed2c00, DexMd5=4b31494397e74f7011ab022fbc234358, SavePath=D:\app_tool\T
[DEXDump]: DexSize=0x6b9800, DexMd5=ed75165a736a0909db1ae4fd6fa6019d, SavePath=D:\app_tool\T
[DEXDump]: DexSize=0x6846bc, DexMd5=e862e1d0a3dbeff0914be27329de5a76, SavePath=D:\app_tool\T
[DEXDump]: DexSize=0x664924, DexMd5=3202b503f6501d8a57e7555e3d262f09, SavePath=D:\app_tool\T
[DEXDump]: DexSize=0x6ec9d8, DexMd5=2bd680b293f8072e04f4d42ea64f7ffc, SavePath=D:\app_tool\T
[DEXDump]: Skip duplicate dex 0x7121000000<4b31494397e74f7011ab022fbc234358>
[DEXDump]: DexSize=0x11c, DexMd5=f1771b68f5f9b168b79ff59ae2daabe4, SavePath=D:\app_tool\T
[DEXDump]: DexSize=0xed2c00, DexMd5=fce271d169d27a2c7b87e0947d197929, SavePath=D:\app_tool\T
[DEXDump]: DexSize=0x1b6b4, DexMd5=60f74ae658b968a2a12ee2ccb94c36f2, SavePath=D:\app_tool\T
[DEXDump]: DexSize=0x1b6b4, DexMd5=7266dc13b864bbb929819c68cd8482f5, SavePath=D:\app_tool\T
07-19/20:10:13 INFO [DEXDump]: found target [9851] com.xbiao:channel
[DEXDump]: Skip duplicate dex 0x711d2d7000<4b31494397e74f7011ab022fbc234358>
[DEXDump]: Skip duplicate dex 0x711f07e000<3202b503f6501d8a57e7555e3d262f09>
[DEXDump]: Skip duplicate dex 0x711f6e3000<ed75165a736a0909db1ae4fd6fa6019d>
[DEXDump]: Skip duplicate dex 0x712048a000<e862e1d0a3dbeff0914be27329de5a76>
[DEXDump]: Skip duplicate dex 0x7121000000<4b31494397e74f7011ab022fbc234358>
[DEXDump]: Skip duplicate dex 0x7122253c86<f1771b68f5f9b168b79ff59ae2daabe4>
[DEXDump]: Skip duplicate dex 0x71225c601c<fce271d169d27a2c7b87e0947d197929>
[DEXDump]: DexSize=0x1b6b4, DexMd5=3befb02de7b3f241813664cbe3c10dd0, SavePath=D:\app_tool\T
```

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/	166 ms
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+KOVl6zrtdy45t/8U6
ASLDH+XytHNVIZ98/ZbjaD/P+rQyuXviD0=

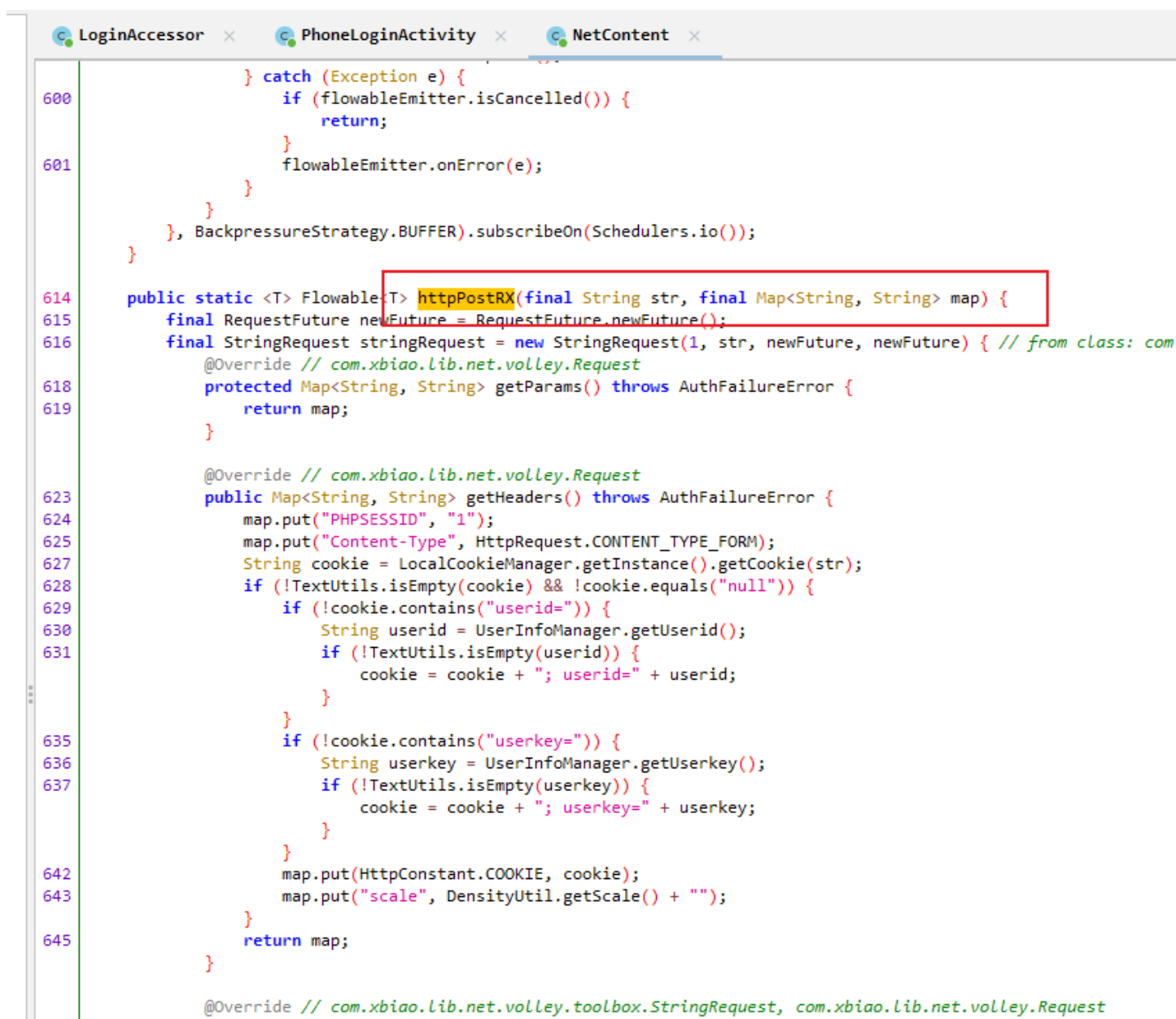
看起来像base64 加密的结果，其实不是

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 x +
  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:"MowTw8j1dq7W05q3ids+jNyo0//FxDNAuaEMwW+U4d500ziRjXkCy/8+K0VI6zrtdy45t/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: MowTw8j1dq7W05q3ids+jNyo0//FxDNAuaEMwW+U4d500ziRjXkCy/8+K0VI6zrtdy45t/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\u9519\u8bef"}
[Google Pixel::脚本之家]->
```

```
/* 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 index0f;
    if (TextUtils.isEmpty(str) || (index0f = (decode = URLDecoder.decode(str.replaceAll(":" + ":", "; "))).indexOf(this.searchCookiePubli
        return;
    }
    int index0f2 = decode.indexOf(f.b, index0f);
    if (index0f2 == -1) {
        index0f2 = decode.length();
    }
    String replace = decode.substring(index0f, index0f2).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	Duration
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

```

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

    @SuppressWarnings("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;
        }
    }

    @SuppressWarnings("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

fACJZ4ZLoQz1nrD9ZCuo0BaQa017kwuehTAiLjo+X42NaFi31Xvuq7pAJ97wYbD0Zf10T6bNWxZIs7j17/iHtuQVj5xmQVHFDk4c9cpWEsg=

字符集 *

utf8(unicode编码)

密码 *

u9J7AbJDlvIA+50=

偏移量 *

6di50aH901duea7d

模式 *

CBC

填充 *

ZeroPadding

编码 *

Base64

加密

解密

{"status":-1,"message":"\u56fe\u5f62\u9a8c\u8bc1\u7801\u9519\u8bef"}

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

```

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

```

function callFun() {
    var result = "";
    Java.perform(function fn() {
        result = Java.use("com.xbiao.utils.net.NetContent").decryptResponse(
            "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

参考

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

