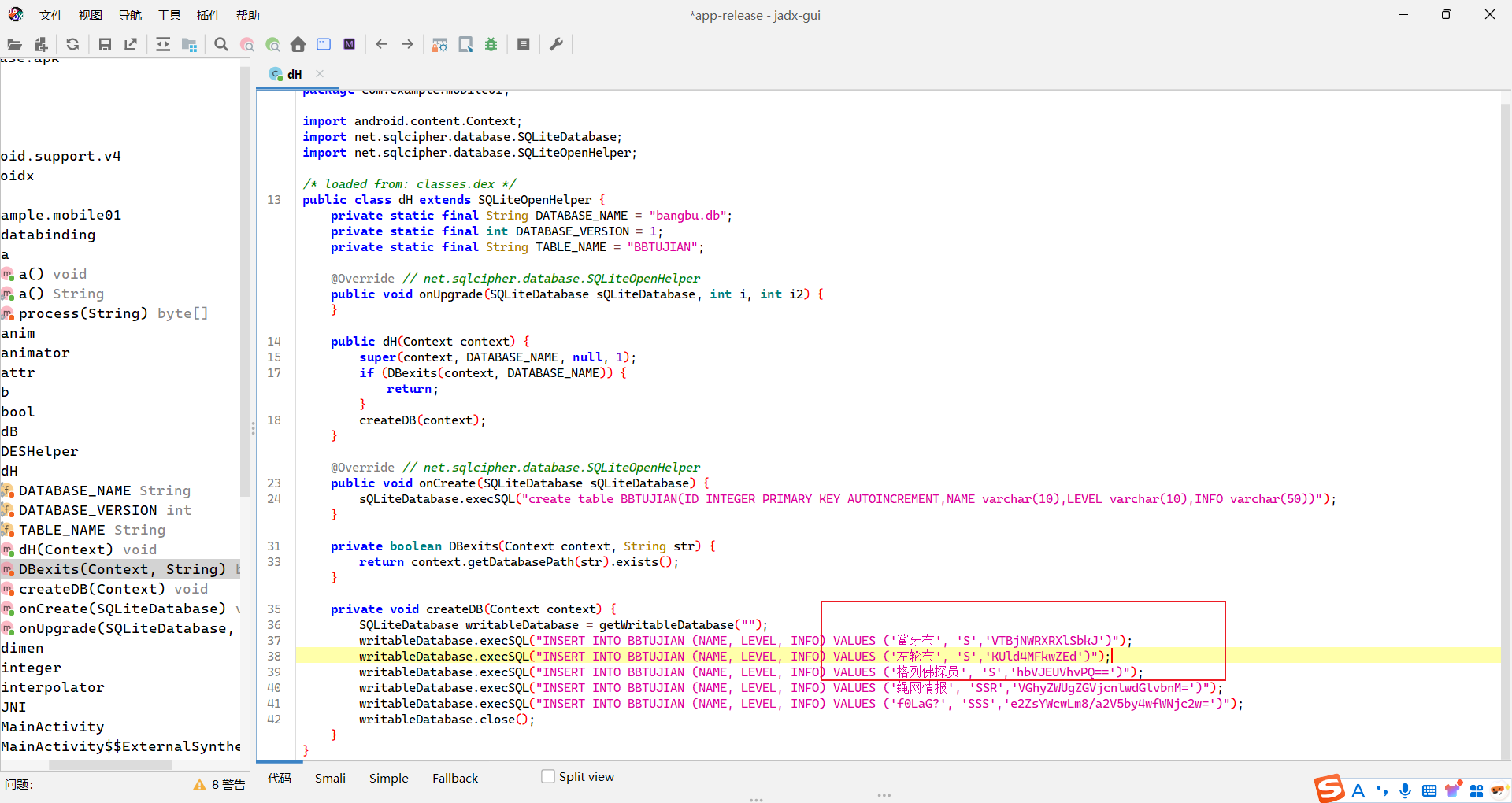
## QLNU-SEC-2403+于光泽+191897071@qq.com

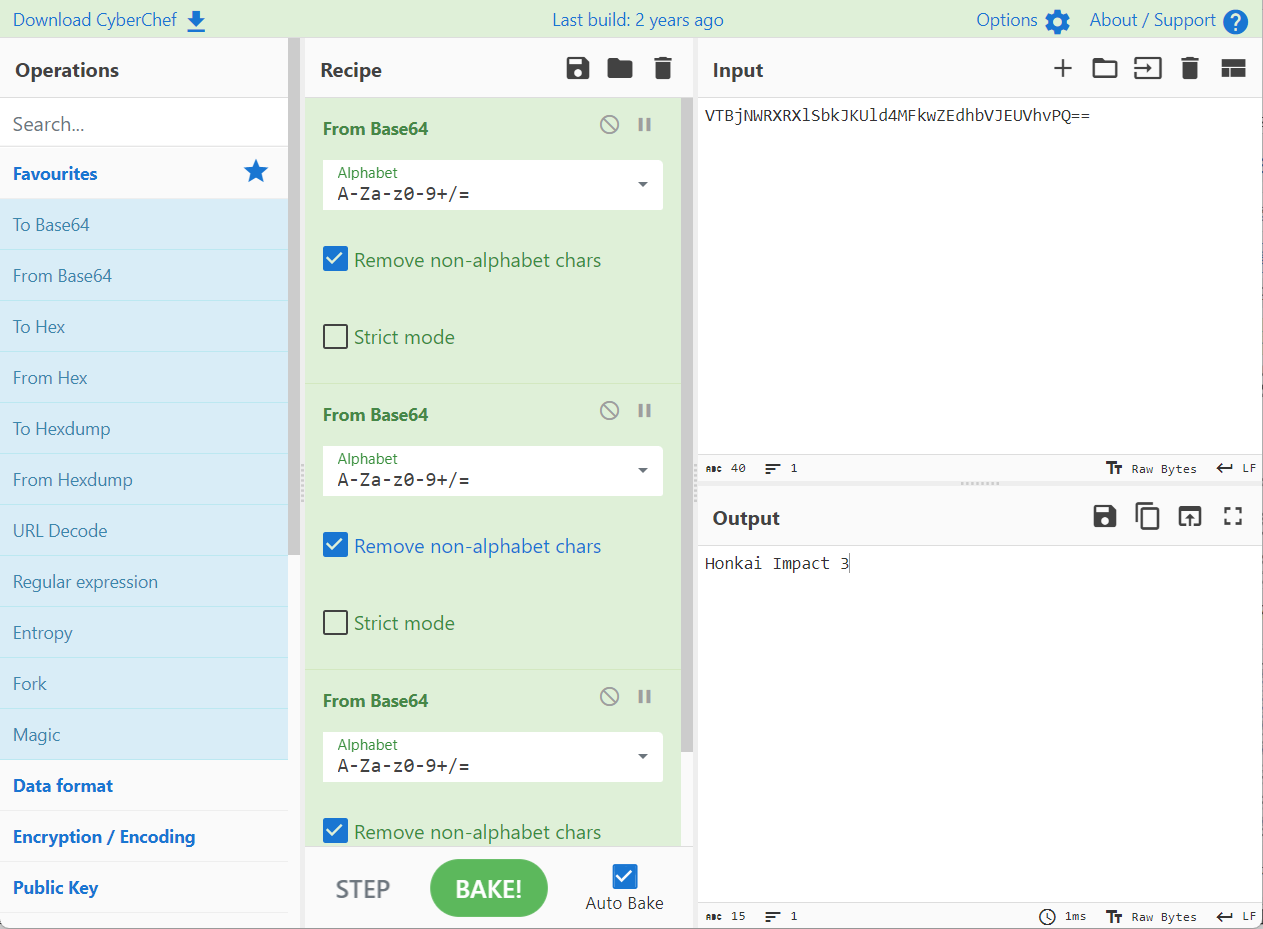
## MOBILE+邦布出击

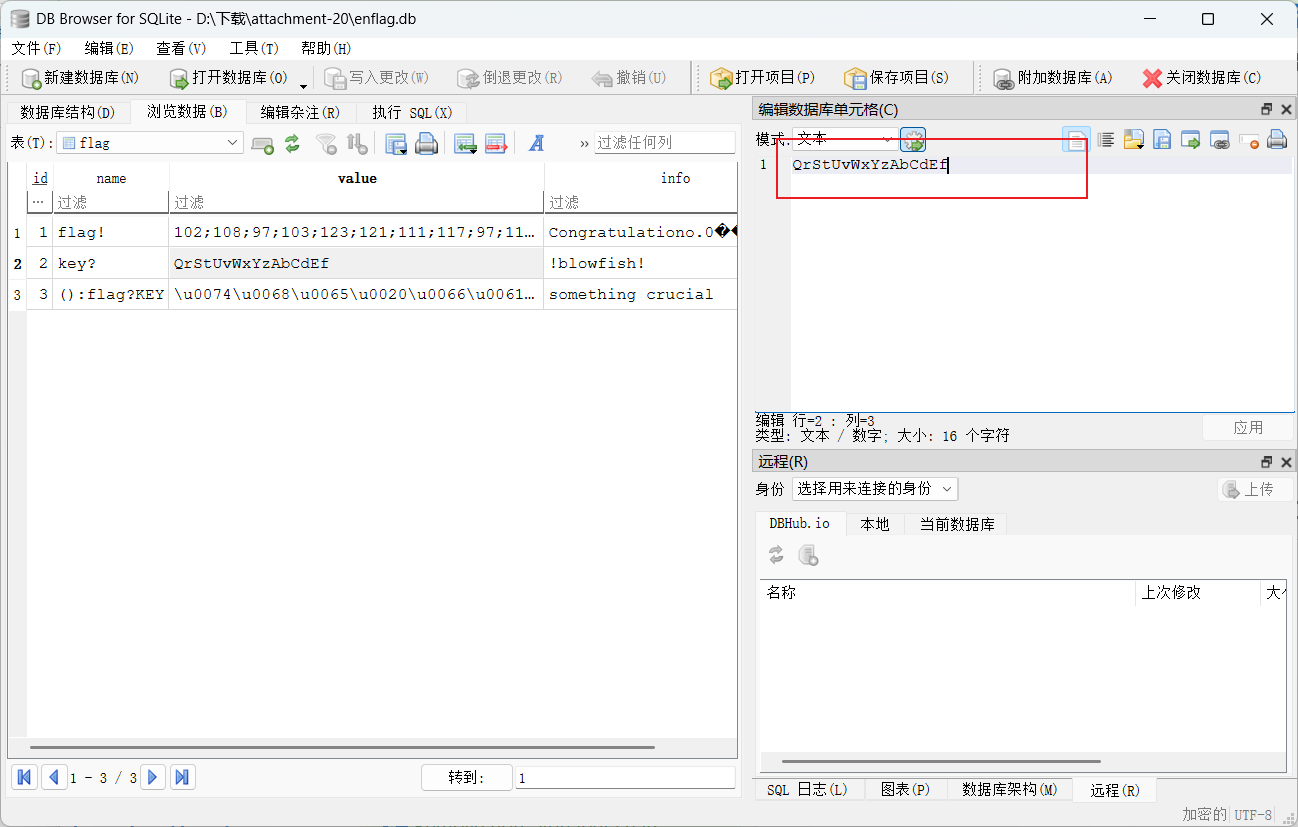
## 解题思路

1.放入jadx中分析

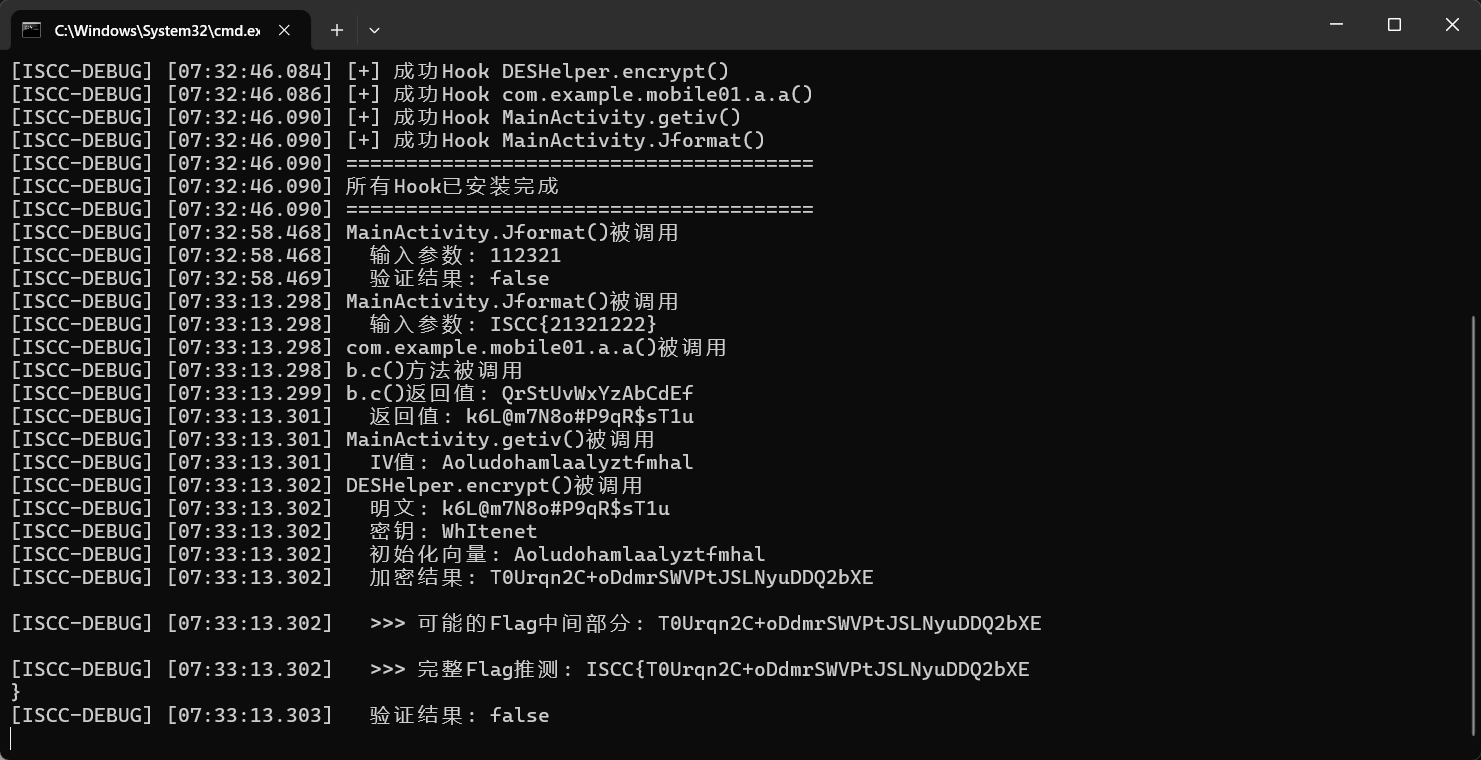


2.将前三个拼接在一起，解3次base64得到密码

3.用获得的密码将enflag.db放到DB Browser (SQLCipher)解密得到key



4.模拟器中安装运行软件，基于Frida编写js脚本进行hook得到flag



Flag: ISCC{T0Urqn2C+oDdmrSWVPtJSLNyuDDQ2bXE}

## Exp:

//脚本

Java.perform(function() {

// 全局配置

const DEBUG = true;

const LOG\_PREFIX = "[ISCC-DEBUG] ";

// 辅助函数：添加时间戳

function logWithTimestamp(message) {

const now = new Date();

const timeStr = now.toISOString().split('T')[1].slice(0, 12);

console.log(`${LOG\_PREFIX}[${timeStr}] ${message}`);

}

// Hook com.example.mobile01.b类的c方法

try {

let b = Java.use("com.example.mobile01.b");

b["c"].implementation = function() {

logWithTimestamp("b.c()方法被调用");

const result = "QrStUvWxYzAbCdEf";

logWithTimestamp(`b.c()返回值: ${result}`);

return result;

};

logWithTimestamp("[+] 成功Hook com.example.mobile01.b.c()");

} catch (err) {

console.error(`${LOG\_PREFIX}[-] 无法Hook b.c(): ${err}`);

}

// Hook DES加密方法

try {

var desHelper = Java.use("com.example.mobile01.DESHelper");

desHelper.encrypt.implementation = function(data, key, iv) {

logWithTimestamp("DESHelper.encrypt()被调用");

logWithTimestamp(` 明文: ${data}`);

logWithTimestamp(` 密钥: ${key}`);

logWithTimestamp(` 初始化向量: ${iv}`);

// 调用原始方法

const originalResult = this.encrypt(data, key, iv);

logWithTimestamp(` 加密结果: ${originalResult}`);

logWithTimestamp(` >>> 可能的Flag中间部分: ${originalResult}`);

logWithTimestamp(` >>> 完整Flag推测: ISCC{${originalResult}}`);

// 可以在这里修改加密结果

return originalResult;

};

logWithTimestamp("[+] 成功Hook DESHelper.encrypt()");

} catch (err) {

console.error(`${LOG\_PREFIX}[-] 无法Hook DESHelper.encrypt(): ${err}`);

}

// Hook com.example.mobile01.a类的a方法

try {

var classA = Java.use("com.example.mobile01.a");

classA.a.implementation = function() {

logWithTimestamp("com.example.mobile01.a.a()被调用");

const result = this.a();

logWithTimestamp(` 返回值: ${result}`);

// 可以在这里修改返回值

return result;

};

logWithTimestamp("[+] 成功Hook com.example.mobile01.a.a()");

} catch (err) {

console.error(`${LOG\_PREFIX}[-] 无法Hook com.example.mobile01.a.a(): ${err}`);

}

// Hook IV获取方法

try {

var mainActivity = Java.use("com.example.mobile01.MainActivity");

mainActivity.getiv.implementation = function() {

logWithTimestamp("MainActivity.getiv()被调用");

const iv = this.getiv();

logWithTimestamp(` IV值: ${iv}`);

// 可以在这里修改IV值

return iv;

};

logWithTimestamp("[+] 成功Hook MainActivity.getiv()");

} catch (err) {

console.error(`${LOG\_PREFIX}[-] 无法Hook MainActivity.getiv(): ${err}`);

}

// Hook格式验证方法

try {

mainActivity.Jformat.implementation = function(input) {

logWithTimestamp("MainActivity.Jformat()被调用");

logWithTimestamp(` 输入参数: ${input}`);

const isValid = this.Jformat(input);

logWithTimestamp(` 验证结果: ${isValid}`);

// 可以在这里修改验证结果

return isValid;

};

logWithTimestamp("[+] 成功Hook MainActivity.Jformat()");

} catch (err) {

console.error(`${LOG\_PREFIX}[-] 无法Hook MainActivity.Jformat(): ${err}`);

}

// 打印Hook完成信息

logWithTimestamp("=======================================");

logWithTimestamp("所有Hook已安装完成");

logWithTimestamp("=======================================");

});