

从XCTF的app3题目简单了解安卓备份文件以及sqliteCipher加密数据库

原创

windy_ll 于 2020-03-04 15:17:31 发布 361 收藏 1

文章标签： 安卓 安全

版权声明：本文为博主原创文章，遵循[CC 4.0 BY-SA](#)版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/qq_41374107/article/details/104654250

版权

从XCTF的app3题目简单了解安卓备份文件以及sqliteCipher加密数据库

[一、题目来源](#)

[二、解题过程](#)

[三、总结](#)

[四、附件](#)

一、题目来源

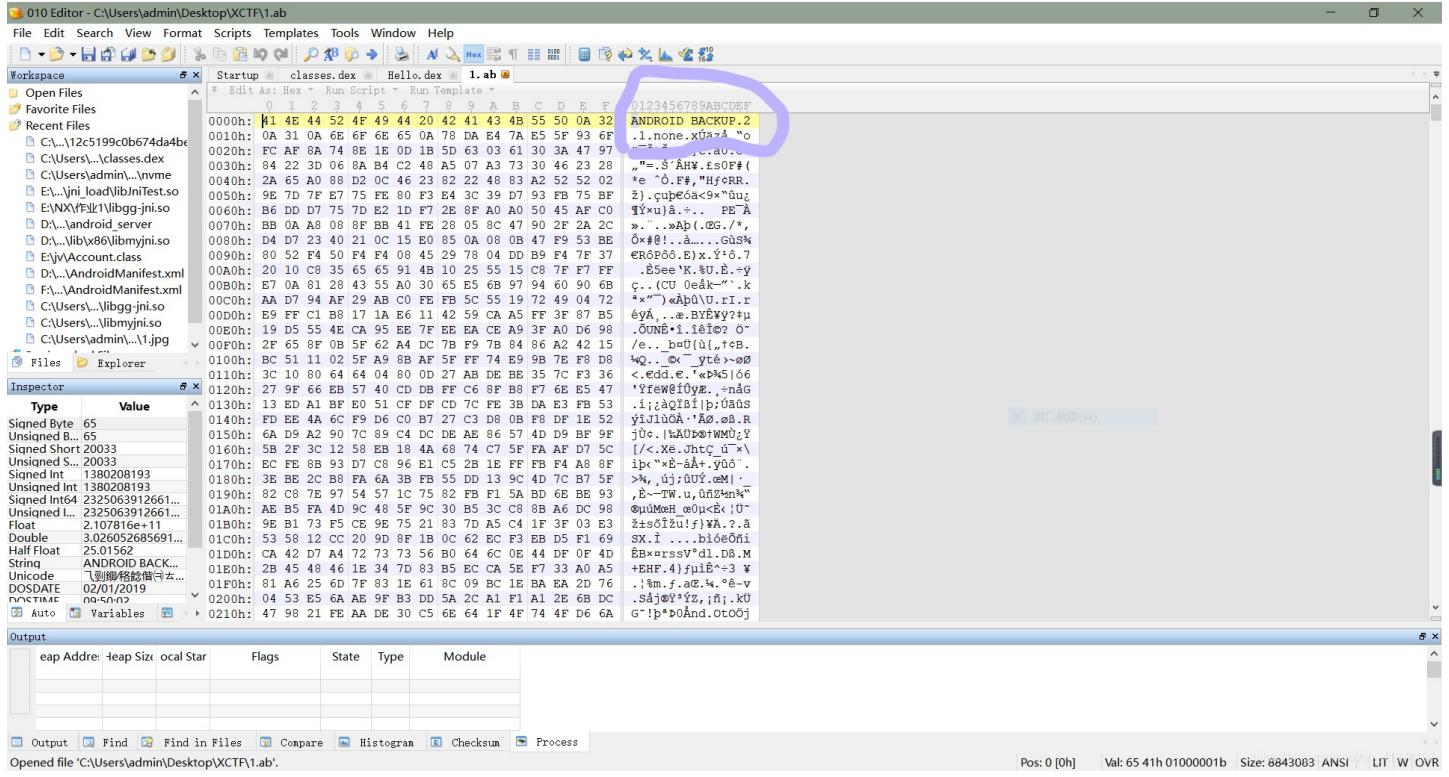
题目来源：XCTF app3题目

二、解题过程

1、下载好题目，下载完后发现是.ab后缀名的文件，如下图所示：



2、什么是.ab文件？.ab后缀名的文件是Android系统的备份文件格式，它分为加密和未加密两种类型，.ab文件的前24个字节是类似文件头的东西，如果是加密的，在前24个字节中会有AES-256的标志，如果未加密，则在前24个字节中会有none的标志，如下图所示：



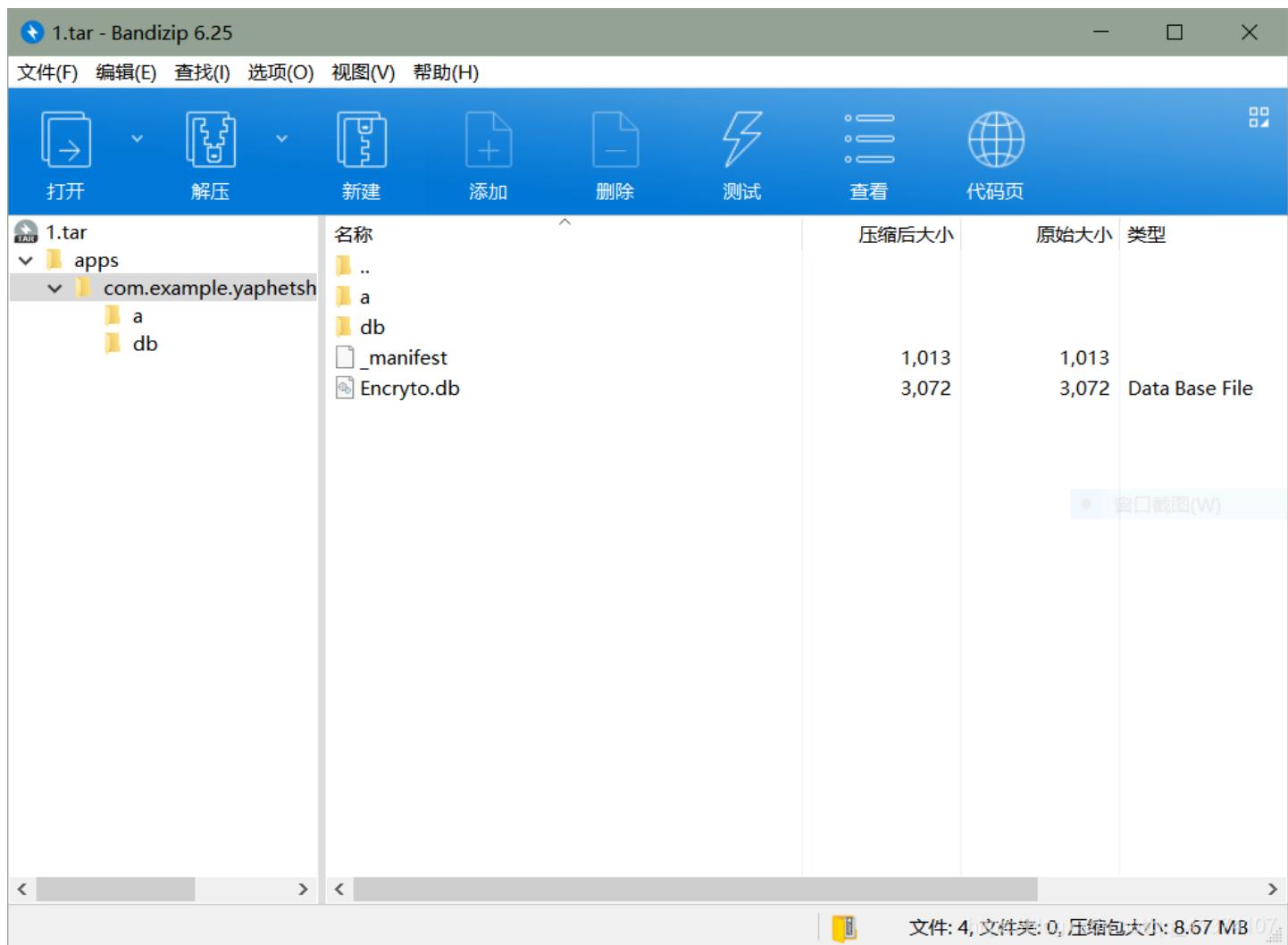
3、怎么获取.ab文件中的数据？在github上有个开源项目 [Android backup extractor](#) 可以将.ab文件转换为.tar文件，然后用解压软件打开即可！！！项目地址：<https://github.com/nelenkov/android-backup-extractor>

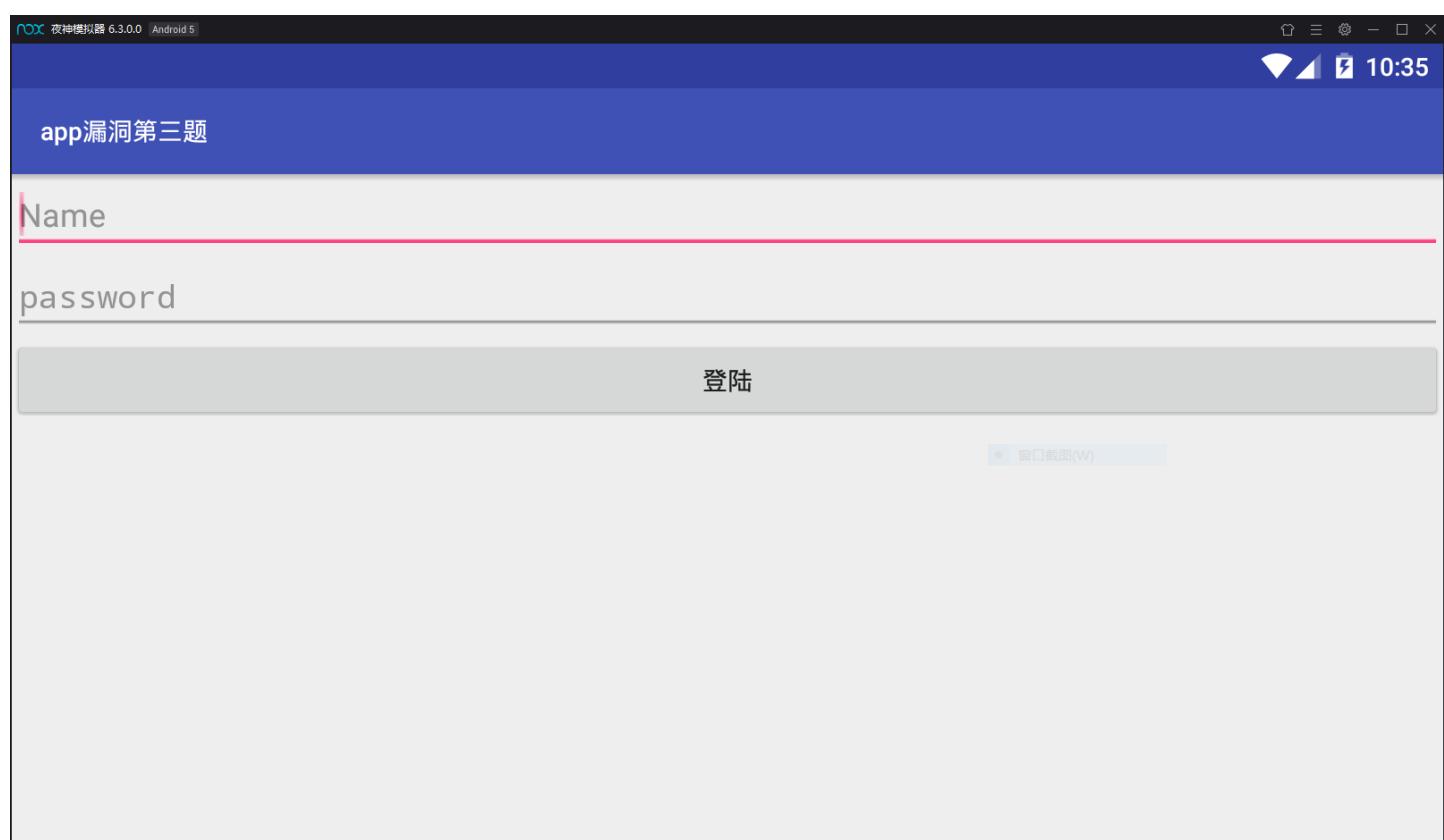
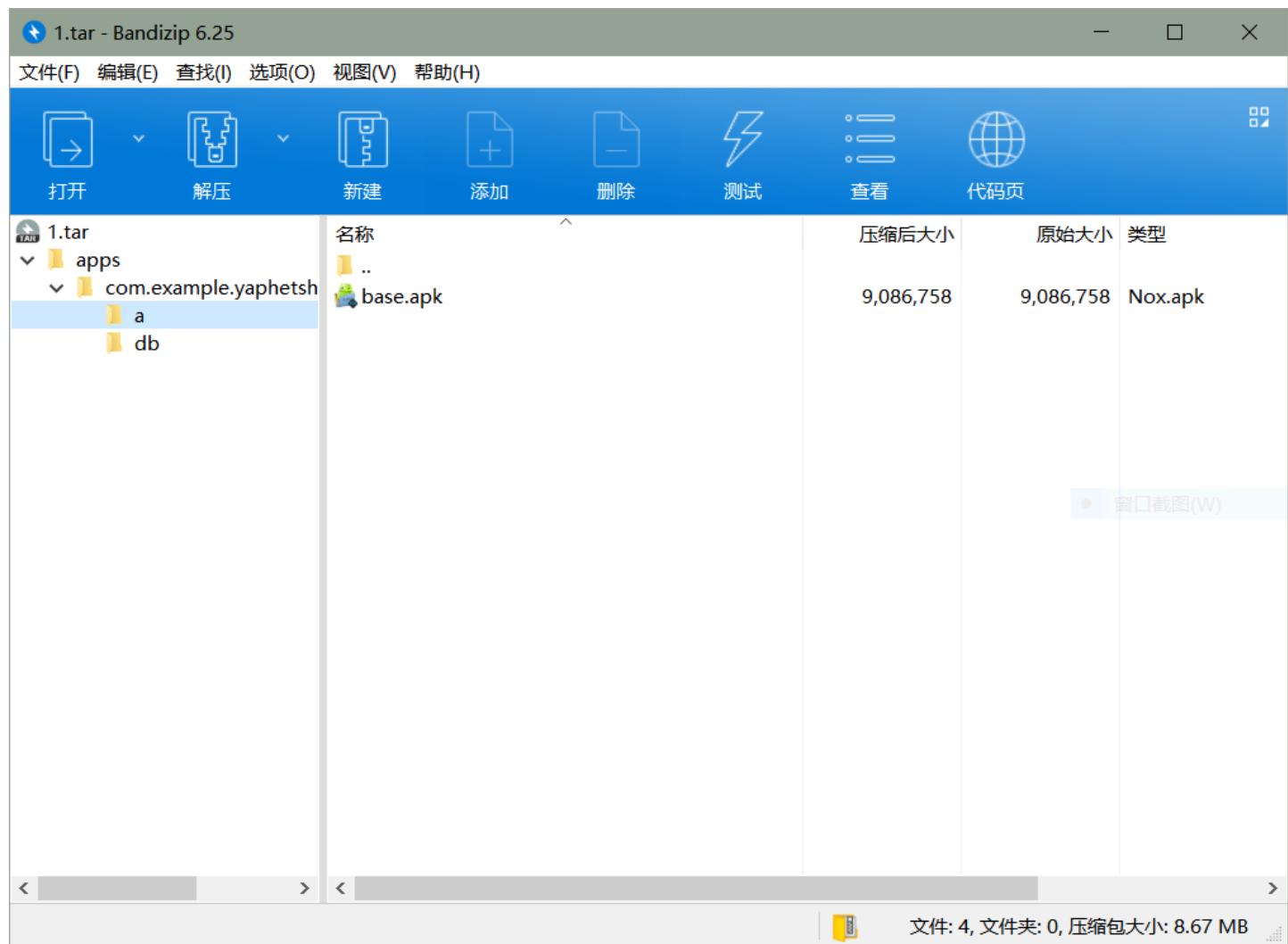
```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.18362.535]
(c) 2019 Microsoft Corporation。保留所有权利。

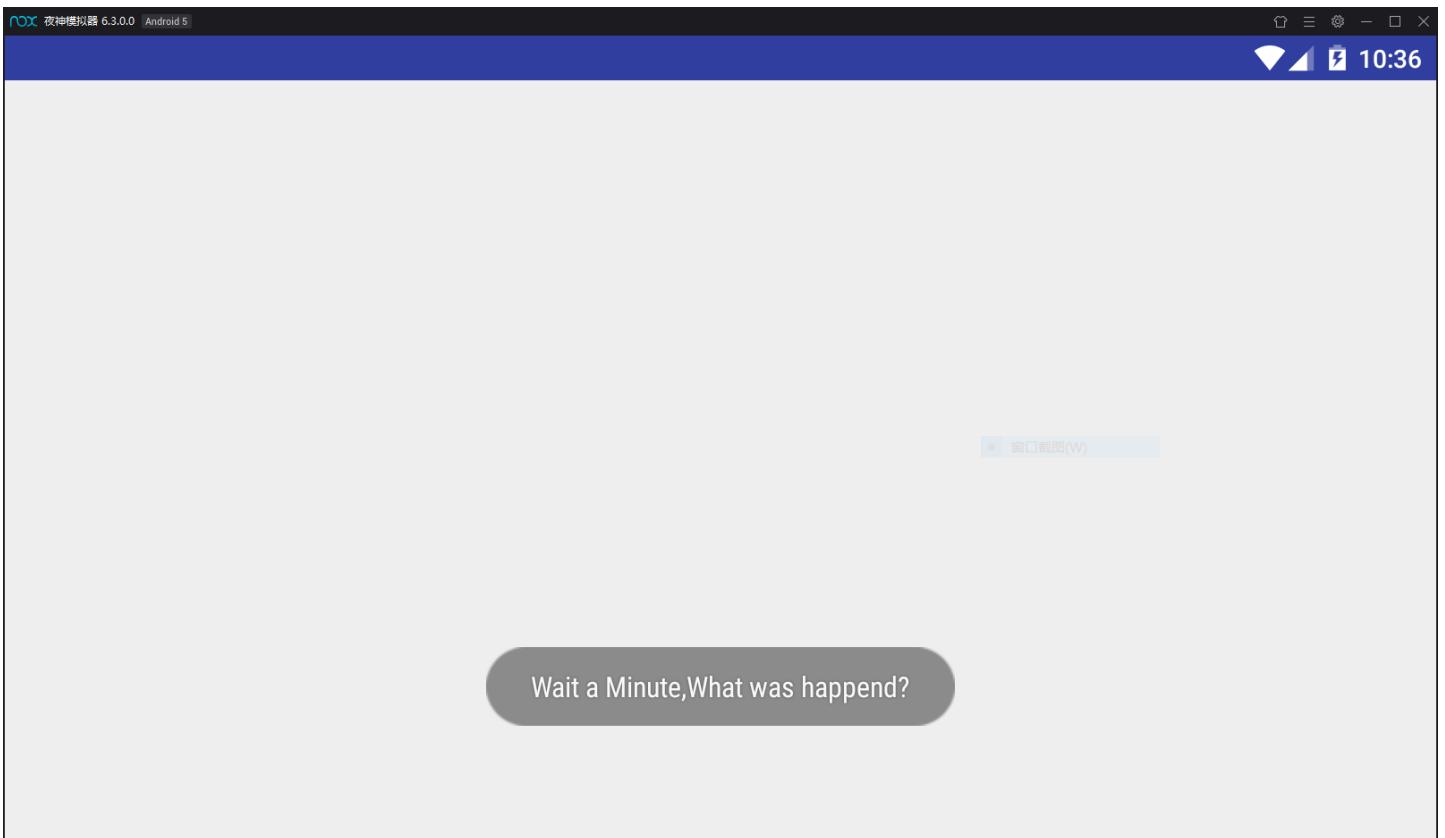
D:\androidtools\ab文件提取>java -jar ade.jar unpack C:\Users\admin\Desktop\XCTF\1.ab C:\Users\admin\Desktop\XCTF\1.tar
0% 1% 2% 3% 4% 5% 6% 7% 8% 9% 10% 11% 12% 13% 14% 15% 16% 17% 18% 19% 20% 21% 22% 23% 24% 25% 26% 27% 28% 29% 30% 31% 32%
% 33% 34% 35% 36% 37% 38% 39% 40% 41% 42% 43% 44% 45% 46% 47% 48% 49% 50% 51% 52% 53% 54% 55% 56% 57% 58% 59% 60% 61% 62%
% 63% 64% 65% 66% 67% 68% 69% 70% 71% 72% 73% 74% 75% 76% 77% 78% 79% 80% 81% 82% 83% 84% 85% 86% 87% 88% 89% 90% 91% 92%
% 93% 94% 95% 96% 97% 98% 99% 100%
9097216 bytes written to C:\Users\admin\Desktop\XCTF\1.tar.

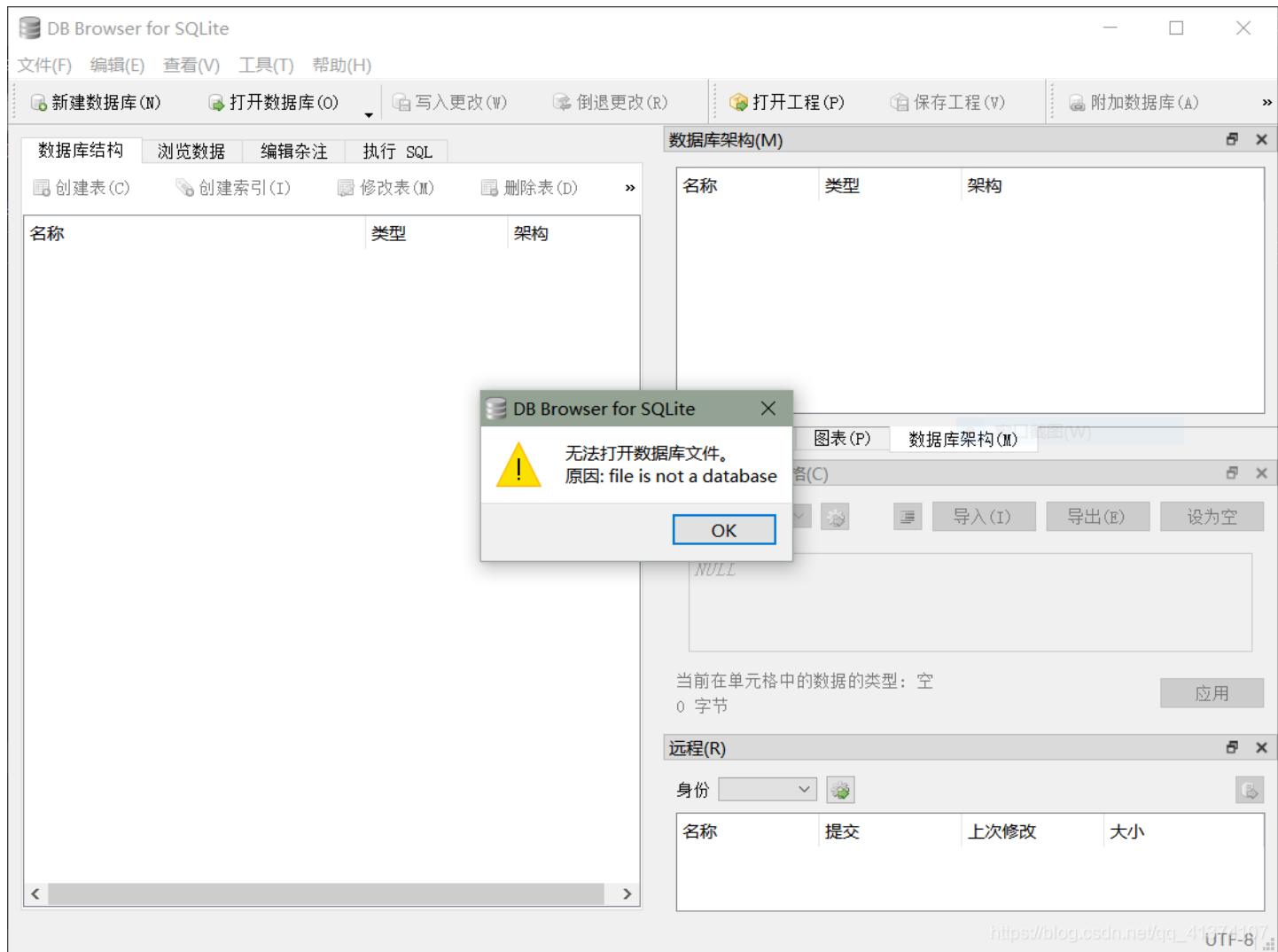
D:\androidtools\ab文件提取>
```

4、使用ade.jar将1.ab文件转为tar文件解压发现有一个apk文件和两个sqlite数据库文件，将apk安装到夜神中，发现没什么有用的东西，去查看数据库，直接使用sqlitebrowser打开，提示需要密码，看来数据库多半被加密了，如下图所示：

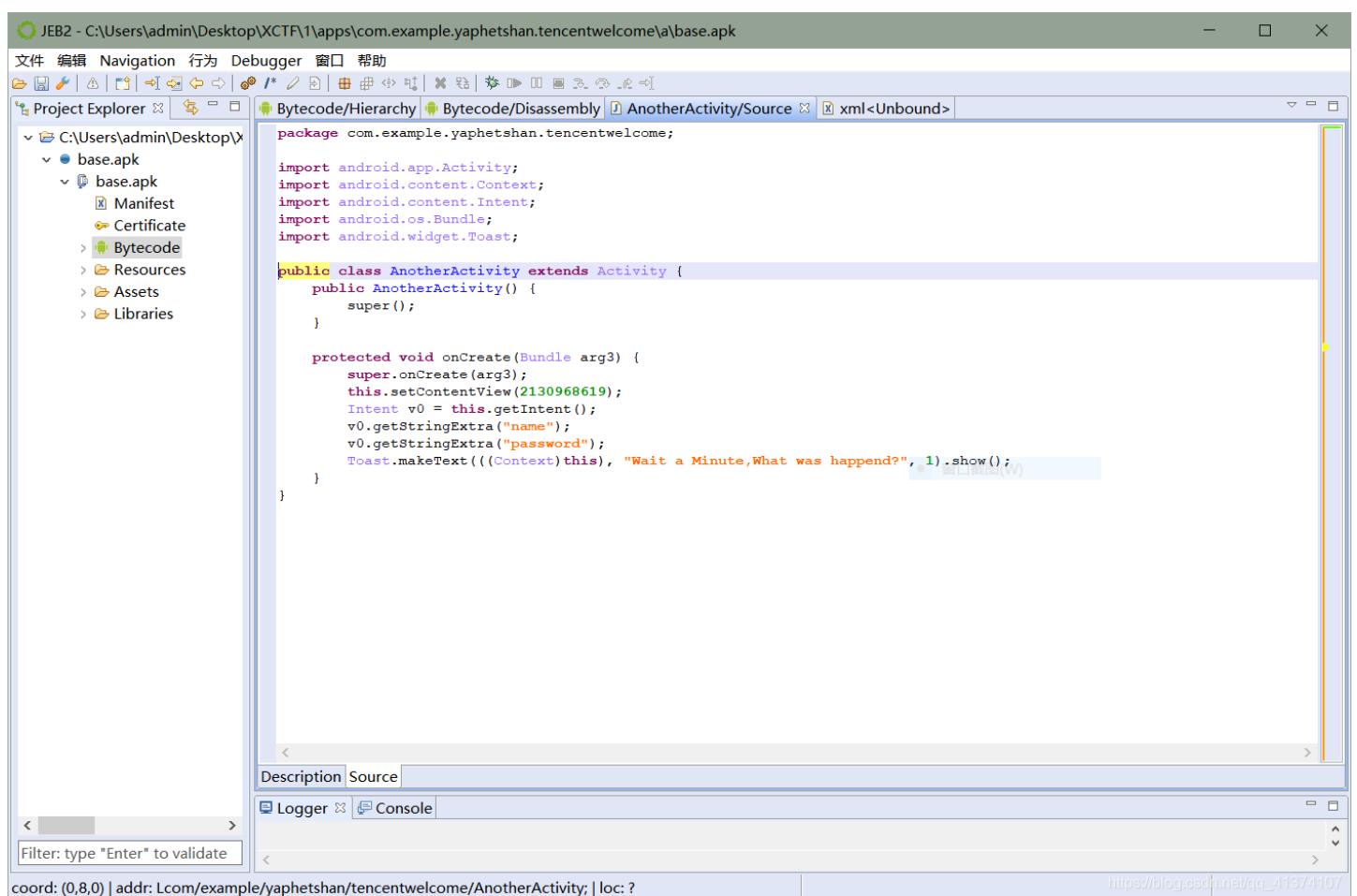
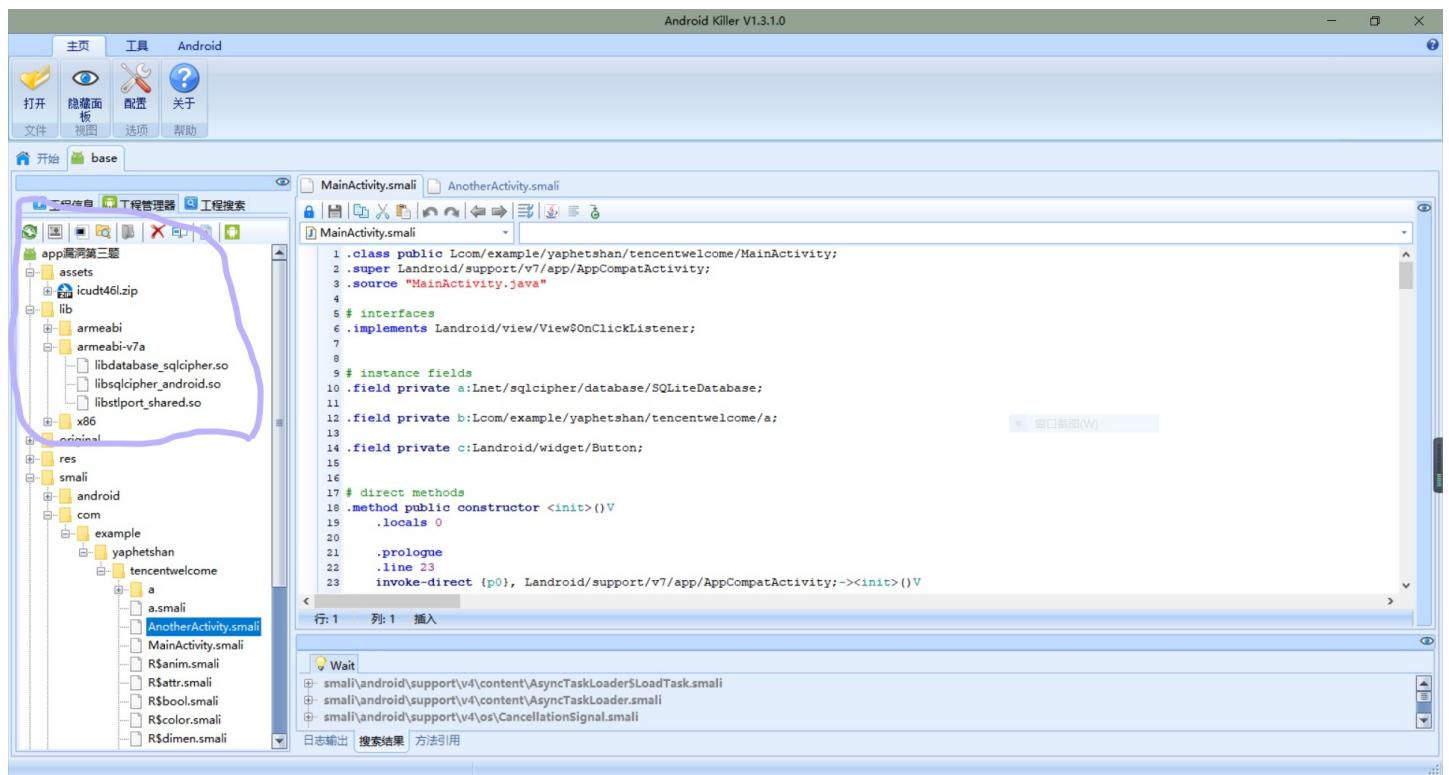








5、直接使用AndroidKiller和jeb将该APK反编译，发现了存在`asset`目录和`libs`目录，并且这两个目录下存放了和`sqlitecipher`相关的文件，可以推断数据库被`sqlitecipher`加密了，再搜索一下在夜神里点击登陆后弹出的信息`Wait ...`，发现该信息在`AnotherActivity.java`文件中，转为`java`后，发现没什么有用的信息。。。



6、于是打开>MainActivity.java文件，果然发现了一个函数 a()，代码如下：

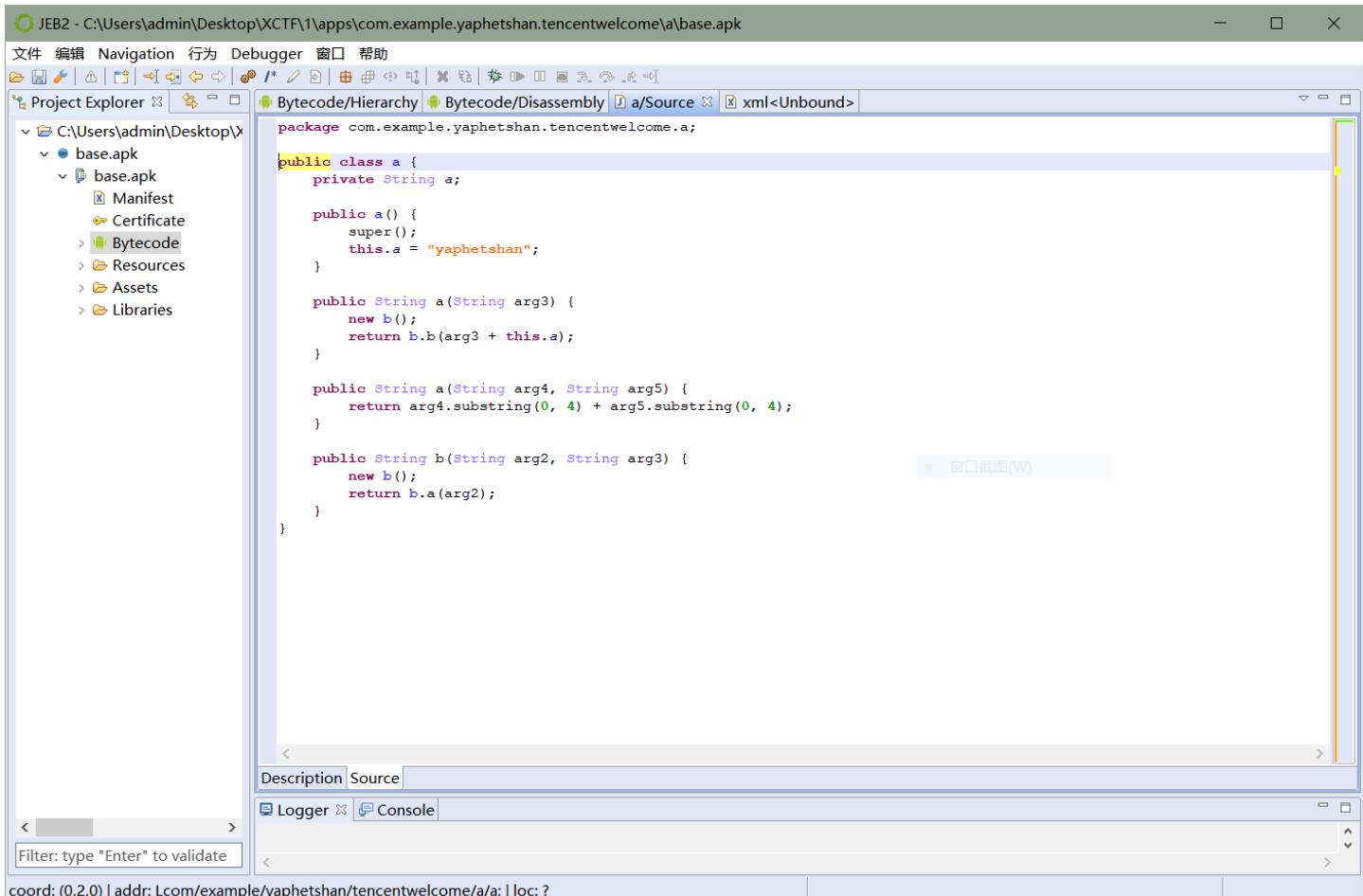
```

private void a() {
    SQLiteDatabase.loadLibs((Context)this);
    this.b = new a((Context)this), "Demo.db", null, 1);
    ContentValues v0 = new ContentValues();
    v0.put("name", "Stranger");
    v0.put("password", Integer.valueOf(123456));
    com.example.yaphetshan.tencentwelcome.a.a v1 = new com.example.yaphetshan.tencentwelcome.a.a();
    String v2 = v1.a(v0.getAsString("name"), v0.getAsString("password"));
    this.a = this.b.getWritableDatabase(v1.a(v2 + v1.b(v2, v0.getAsString("password"))).substring(0, 7));
    this.a.insert("TencentMicrMsg", null, v0);
}

```

- 第一行 `SQLiteDatabase.loadLibs((Context)this);` 将所需要的sqlitecipher库文件加载进来。
- 第二行实例化一个sqlitehelper类。
- 第三、五行实例化了一个ContentValues类并将键值对 `name:Stranger`、`password:123456` 放入其中。
- 第六行实例化了一个 `com.example.yaphetshan.tencentwelcome.a.a` 类。
- 第七行获取了v2变量的值。
- 第八行调用了 `getWritableDatabase` 函数，传进去的字符串参数即是数据库解密的密钥。

7、现在目标已经很明确了，就是获取数据库解密密钥（猜一下flag就藏在加密的sqlite数据库中），而该密钥由 `com.example.yaphetshan.tencentwelcome.a.a` 里面的方法生成，而这个类又调用了 `b.java` 里面的方法，如图所示：



JEB2 - C:\Users\admin\Desktop\XCTF\1\apps\com.example.yaphetshan.tencentwelcome\apk\base.apk

文件 编辑 Navigation 行为 Debugger 窗口 帮助

Project Explorer Bytecode/Hierarchy Bytecode/Disassembly b/Source xml<Unbound>

```
package com.example.yaphetshan.tencentwelcome.a;
import java.security.MessageDigest;

public class b {
    public b() {
        super();
    }

    public static final String a(String arg9) {
        String v0_2;
        int v0 = 0;
        char[] v2 = new char[]{'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f'};
        try {
            byte[] v1 = arg9.getBytes();
            MessageDigest v3 = MessageDigest.getInstance("MD5");
            v3.update(v1);
            byte[] v3_1 = v3.digest();
            int v4 = v3_1.length;
            char[] v5 = new char[v4 * 2];
            int v1_1 = 0;
            while(v0 < v4) {
                int v6 = v3_1[v0];
                int v7 = v1_1 + 1;
                v5[v1_1] = v2[v6 >>> 4 & 15];
                v1_1 = v7 + 1;
                v5[v7] = v2[v6 & 15];
                ++v0;
            }
            v0_2 = new String(v5);
        } catch(Exception v0_1) {
            v0_2 = null;
        }
        return v0_2;
    }

    public static final String b(String arg9) {
        String v0_2;
        int v0 = 0;
        char[] v2 = new char[]{'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f'};
        try {
            byte[] v1 = arg9.getBytes();
            MessageDigest v3 = MessageDigest.getInstance("SHA-1");
            v3.update(v1);
            byte[] v3_1 = v3.digest();
            int v4 = v3_1.length;
            char[] v5 = new char[v4 * 2];
            int v1_1 = 0;
            while(v0 < v4) {
                int v6 = v3_1[v0];
                int v7 = v1_1 + 1;
                v5[v1_1] = v2[v6 >>> 4 & 15];
                v1_1 = v7 + 1;
                v5[v7] = v2[v6 & 15];
                ++v0;
            }
            v0_2 = new String(v5);
        } catch(Exception v0_1) {
            v0_2 = null;
        }
        return v0_2;
    }
}
```

Description Source

Logger Console Unit "b" (java) was created

Filter: type "Enter" to validate

coord: (0,4,0) | addr: Lcom/example/yaphetshan/tencentwelcome/a/b; | loc: ?

JEB2 - C:\Users\admin\Desktop\XCTF\1\apps\com.example.yaphetshan.tencentwelcome\apk\base.apk

文件 编辑 Navigation 行为 Debugger 窗口 帮助

Project Explorer Bytecode/Hierarchy Bytecode/Disassembly b/Source xml<Unbound>

```
    v0_2 = new String(v5);
}
catch(Exception v0_1) {
    v0_2 = null;
}

return v0_2;
}

public static final String b(String arg9) {
String v0_2;
int v0 = 0;
char[] v2 = new char[]{'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f'};
try {
byte[] v1 = arg9.getBytes();
MessageDigest v3 = MessageDigest.getInstance("SHA-1");
v3.update(v1);
byte[] v3_1 = v3.digest();
int v4 = v3_1.length;
char[] v5 = new char[v4 * 2];
int v1_1 = 0;
while(v0 < v4) {
int v6 = v3_1[v0];
int v7 = v1_1 + 1;
v5[v1_1] = v2[v6 >>> 4 & 15];
v1_1 = v7 + 1;
v5[v7] = v2[v6 & 15];
++v0;
}

v0_2 = new String(v5);
} catch(Exception v0_1) {
    v0_2 = null;
}

return v0_2;
}
}
```

Description Source

Logger Console Unit "b" (java) was created

Filter: type "Enter" to validate

coord: (0,4,0) | addr: Lcom/example/yaphetshan/tencentwelcome/a/b; | loc: ?

8、a、b类里面生成密钥的算法涉及到了sha-1、md5等算法，没必要去重新写一编，搞清楚密钥生成逻辑然后把b类里面的两个函数复制出来调用即可生成密钥，简单分析一下密钥生成逻辑：首先得到变量v2,v2调用了a类中的 `a(String, String)` 方法获取,该方法返回第一个参数前四个字符加第二个参数的前四个字符，而调用该方法传进去的参数为 `(Stranger,123456)`，所以 `v2 = Stra1234`，密钥为 `v1.a(String).sunstring()`（调用v1.a()方法然后将返回值截取前7位作为密钥），关键就在传进去的这个字符串，可以看到这个字符串是 `v2 + v1.b(v2, '123456')`，而 `v1.b(String, String)` 这个函数将调用了b类的 `a(String)` 函数，传进去的参数是变量 `v2`，获取到返回值后，我们就可以得到这个字符串，然后调用 `v1.a(String)` 函数得到密钥，这个函数将传进去的字符串加上 `yaphetshan` 字符串作为参数调用b类的b方法，其返回值取前7位即是密钥，写了一个java获取密钥的代码，运行结果如下（ps:代码粘贴在文末中）：

```

JEB2 - C:\Users\admin\Desktop\XCTF\1\apps\com.example.yaphetshan.tencentwelcome\apk\base.apk
文件 编辑 Navigation 行为 Debugger 窗口 帮助
Project Explorer [ ] Bytecode/Hierarchy [ ] Bytecode/Disassembly [ ] MainActivity/Source [ ] xml<Unbound>
C:\Users\admin\Desktop\XCTF\1\apps\apk\apk\base.apk
base.apk
Manifest
Certificate
Bytecode
Resources
Assets
Libraries

import android.content.Intent;
import android.content.SharedPreferences$Editor;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View$OnClickListener;
import android.widget.Button;
import net.sqlcipher.database.SQLiteDatabase;

public class MainActivity extends AppCompatActivity implements View$OnClickListener {
    private SQLiteDatabase a;
    private b;
    private Button c;
    public MainActivity() {
        super();
    }

    private void a() {
        SQLiteDatabase.loadLibs((Context)this);
        this.b = new a((Context)this), "Demo.db", null, 1);
        ContentValues v0 = new ContentValues();
        v0.put("name", "Stranger");
        v0.put("password", Integer.valueOf(123456));
        com.example.yaphetshan.tencentwelcome.a.a v1 = new com.example.yaphetshan.tencentwelcome.a.a();
        String v2 = v1.a(v0.getAsString("name"), v0.getAsString("password"));
        this.a = this.b.getWritableDatabase(v1.a(v2 + v1.b(v2, v0.getAsString("password"))).substring(0, 7));
        this.a.insert("TencentMicrMsg", null, v0);
    }

    public void onClick(View arg4) {
        if(arg4 == this.c) {
            Intent v0 = new Intent();
            v0.putExtra("name", "name");
            v0.putExtra("password", "pass");
            v0.setClass((Context)this, AnotherActivity.class);
            this.startActivity(v0);
        }
    }

    protected void onCreate(Bundle arg4) {
        super.onCreate(arg4);
    }
}

Logger [ ] Console [ ]
Unit "AnotherActivity" (java) was created

```

```
C:\Windows\System32\cmd.exe
E:\jv>javac b.java
E:\jv>java b
KEY = ae56f99
E:\jv>
```

9、获取到密钥后，使用sqlitebrowser打开加密数据库，发现了一串Base64的字符串，解码得到了flag

The screenshot shows the DB Browser for SQLite interface. The main window displays a table named 'TencentMicrMsg' with three columns: name, password, and F_1_a_g. A single row is present with values: Stranger, 123456, and VGN0ZntIM2x... respectively. The 'F_1_a_g' column contains a long string of characters, which is highlighted with a blue selection bar. To the right of the table, the 'Database Structure' pane shows the table's definition: CREATE TABLE TencentMicrMsg(). Below the table, the status bar indicates the string is currently in 'Text' mode and is 44 characters long. The bottom right corner of the status bar shows '加密的 UTF-8' (Encrypted UTF-8).

三、总结

刚下载下来题目发现一看后缀名就慌了，重来没见过的文件了，百度了n久，终于弄懂了android备份文件和ssqlitecipher这两个东西。

给大家分享一下有关这两个东西的知识点我觉得写的比较好的博客！！！

- Android备份文件: https://blog.csdn.net/qq_33356474/article/details/92188491
- SqliteCipher: <https://www.cnblogs.com/android100/p/Android-SQLCipher.html>

四、附件

题目以及所用到的工具：百度网盘链接https://pan.baidu.com/s/1Wam_Hjg8rNlpqywVqqASpQ, 密码 0y89

获取密钥java代码如下：

```
import java.security.MessageDigest;
import java.util.*;

public class b {
    public b() {
        super();
    }

    public static void main(String[] args)
    {
        String varV2 = "Stra1234";
        String varV1B = a(varV2);
        String varKey = varV2 + varV1B + "yaphetshan";
        System.out.print("KEY = ");
        System.out.print(b(varKey).substring(0,7));
    }

    public static final String a(String arg9) {
        String v0_2;
        int v0 = 0;
        char[] v2 = new char[]{'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f'};
        try {
            byte[] v1 = arg9.getBytes();
            MessageDigest v3 = MessageDigest.getInstance("MD5");
            v3.update(v1);
            byte[] v3_1 = v3.digest();
            int v4 = v3_1.length;
            char[] v5 = new char[v4 * 2];
            int v1_1 = 0;
            while(v0 < v4) {
                int v6 = v3_1[v0];
                int v7 = v1_1 + 1;
                v5[v1_1] = v2[v6 >>> 4 & 15];
                v1_1 = v7 + 1;
                v5[v7] = v2[v6 & 15];
                ++v0;
            }
            v0_2 = new String(v5);
        }
        catch(Exception va_1) {
```

```
catch(Exception v0_1) {
    v0_2 = null;
}

return v0_2;
}

public static final String b(String arg9) {
    String v0_2;
    int v0 = 0;
    char[] v2 = new char[]{'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f'};
    try {
        byte[] v1 = arg9.getBytes();
        MessageDigest v3 = MessageDigest.getInstance("SHA-1");
        v3.update(v1);
        byte[] v3_1 = v3.digest();
        int v4 = v3_1.length;
        char[] v5 = new char[v4 * 2];
        int v1_1 = 0;
        while(v0 < v4) {
            int v6 = v3_1[v0];
            int v7 = v1_1 + 1;
            v5[v1_1] = v2[v6 >>> 4 & 15];
            v1_1 = v7 + 1;
            v5[v7] = v2[v6 & 15];
            ++v0;
        }

        v0_2 = new String(v5);
    }
    catch(Exception v0_1) {
        v0_2 = null;
    }

    return v0_2;
}
}
```