

攻防世界-app3

原创

bufsnake 于 2019-07-24 18:33:02 发布 1430 收藏 2

分类专栏: [android打怪升级](#) 文章标签: [攻防世界](#) [app3](#) [ctf](#) [Android逆向](#) [writeup](#)

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_40640243/article/details/97142658

版权



[android打怪升级](#) 专栏收录该内容

1 篇文章 0 订阅

订阅专栏

所需工具

```
jadx  
android-backup-extractor  
DB Browser for SQLite
```

考察考点

如何提取或解压缩.ab文件 (Android备份文件)
sqlite3数据库加密以及解密

艰辛的过程

拿到程序, 发现是ab结尾的文件, 就上网搜了一下, 发现是android应用的备份文件, 需要用android-backup-extractor里的abe.jar将数据提取出来

```
java -jar abe.jar unpack app3.ab app3.tar
```

接下来解压app3.tar, 一番搜寻后, 发现了base.apk, 也就是我们需要分析的程序

```
Desktop cd apps/com.example.yaphetshan.tencentwelcome  
com.example.yaphetshan.tencentwelcome ls  
Demo.db Encryto.db _manifest a db  
com.example.yaphetshan.tencentwelcome tree  
.  
├── Demo.db  
├── Encryto.db  
├── _manifest  
├── a  
│   └── base.apk  
└── db
```

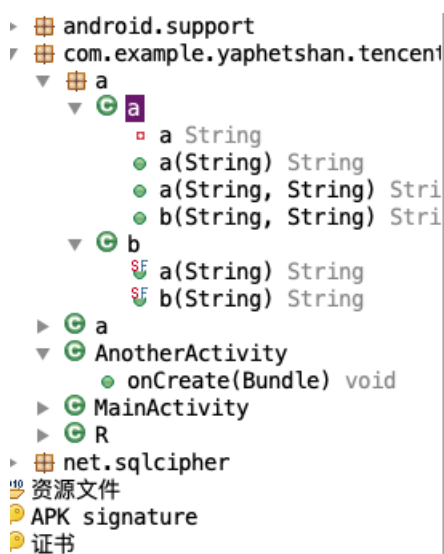
用jadx打开分析程序的逻辑结构，找到主要代码

```

12 import net.sqlcipher.database.SQLiteDatabase;
13
14 public class MainActivity extends AppCompatActivity implements OnClickListener {
15     private SQLiteDatabase a;
16     private a b;
17     private Button c;
18
19     /* access modifiers changed from: protected */
20     public void onCreate(Bundle bundle) {
21         super.onCreate(bundle);
22         setContentView((int) R.layout.activity_main);
23         this.c = (Button) findViewById(R.id.add_data);
24         this.c.setOnClickListener(this);
25         Editor edit = getSharedPreferences("test", 0).edit();
26         edit.putString("Is_Encroty", "1");
27         edit.putString("Encryto", "SqlCipher");
28         edit.putString("ver_sion", "3_4_0");
29         edit.apply();
30         a();
31     }
32
33     private void a() {
34         SQLiteDatabase.loadLibs(this);
35         this.b = new a(this, "Demo.db", null, 1);
36         ContentValues contentValues = new ContentValues();
37         contentValues.put("name", "Stranger");
38         contentValues.put("password", Integer.valueOf(123456));
39         a aVar = new a();
40         String a2 = aVar.a(contentValues.getAsString("name"), contentValues.getAsString("password"));
41         this.a = this.b.getWritableDatabase(aVar.a(a2 + aVar.b(a2, contentValues.getAsString("password"))).substring(0, 7));
42         this.a.insert("TencentMicMsg", null, contentValues);
43     }
44
45     public void onClick(View view) {
46         if (view == this.c) {
47             Intent intent = new Intent();
48             intent.putExtra("name", "name");
49             intent.putExtra("password", "pass");
50             intent.setClass(this, AnotherActivity.class);
51             startActivity(intent);
52         }
53     }
54 }

```

https://blog.csdn.net/qq_40640243

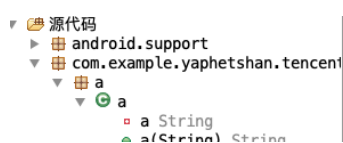


```

1 package com.example.yaphetshan.tencentwelcome.a;
2
3 /* compiled from: Cipher */
4 public class a {
5     private String a = "yaphetshan";
6
7     public String a(String str, String str2) {
8         String substring = str.substring(0, 4);
9         return substring + str2.substring(0, 4);
10    }
11
12    public String b(String str, String str2) {
13        new b();
14        return b.a(str);
15    }
16
17    public String a(String str) {
18        new b();
19        return b.b(str + this.a);
20    }
21 }

```

https://blog.csdn.net/qq_40640243



```

1 package com.example.yaphetshan.tencentwelcome.a;
2
3 import java.security.MessageDigest;
4
5 /* compiled from: SHA1Manager */

```

```

6 public class D {
7     public static final String a(String str) {
8         byte[] digest;
9         char[] cArr = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f'};
10        try {
11            byte[] bytes = str.getBytes();
12            MessageDigest instance = MessageDigest.getInstance("MD5");
13            instance.update(bytes);
14            char[] cArr2 = new char[(r4 * 2)];
15            int i = 0;
16            for (byte b : instance.digest()) {
17                int i2 = i + 1;
18                cArr2[i] = cArr[(b >>> 4) & 15];
19                i = i2 + 1;
20                cArr2[i2] = cArr[b & 15];
21            }
22            return new String(cArr2);
23        } catch (Exception e) {
24            return null;
25        }
26    }
27
28    public static final String b(String str) {
29        byte[] digest;
30        char[] cArr = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f'};
31        try {
32            byte[] bytes = str.getBytes();
33            MessageDigest instance = MessageDigest.getInstance("SHA-1");
34            instance.update(bytes);
35            char[] cArr2 = new char[(r4 * 2)];
36            int i = 0;
37            for (byte b : instance.digest()) {
38                int i2 = i + 1;
39                cArr2[i] = cArr[(b >>> 4) & 15];
40                i = i2 + 1;
41                cArr2[i2] = cArr[b & 15];
42            }
43            return new String(cArr2);
44        } catch (Exception e) {
45            return null;
46        }
47    }
48 }

```

https://blog.csdn.net/qq_40640243

主要逻辑如下：

将Stranger和123456两个字符串取前四个，并拼接到一起，得到Stra1234将Stra1234进行md5加密后用base16加密，得到的字符串前面加上Stra1234，在拼接上字符串yaphetshan后用SHA-1加密，在用base16加密，最后取得到字符串的前七位，即是最后需要解数据库密码的密码

解密代码如下：

```

import java.security.MessageDigest;

public class b {

    public static String a(String str) {
        byte[] digest;
        char[] cArr = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f'};
        try {
            byte[] bytes = str.getBytes();
            MessageDigest instance = MessageDigest.getInstance("MD5");
            instance.update(bytes);

            char[] cArr2 = new char[(16 * 2)];
            int i = 0;
            for (byte b : instance.digest()) {
                int i2 = i + 1;
                cArr2[i] = cArr[(b >>> 4) & 15];
                i = i2 + 1;
                cArr2[i2] = cArr[b & 15];
            }
            return new String(cArr2);
        } catch (Exception e) {
            return null;
        }
    }

    public static final String b(String str) {
        byte[] digest;
        char[] cArr = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f'};
        try {
            byte[] bytes = str.getBytes();
            MessageDigest instance = MessageDigest.getInstance("SHA-1");
            instance.update(bytes);
            char[] cArr2 = new char[(32 * 2)];
            int i = 0;
            for (byte b : instance.digest()) {
                int i2 = i + 1;
                cArr2[i] = cArr[(b >>> 4) & 15];
                i = i2 + 1;
                cArr2[i2] = cArr[b & 15];
            }
            return new String(cArr2);
        } catch (Exception e) {
            return null;
        }
    }

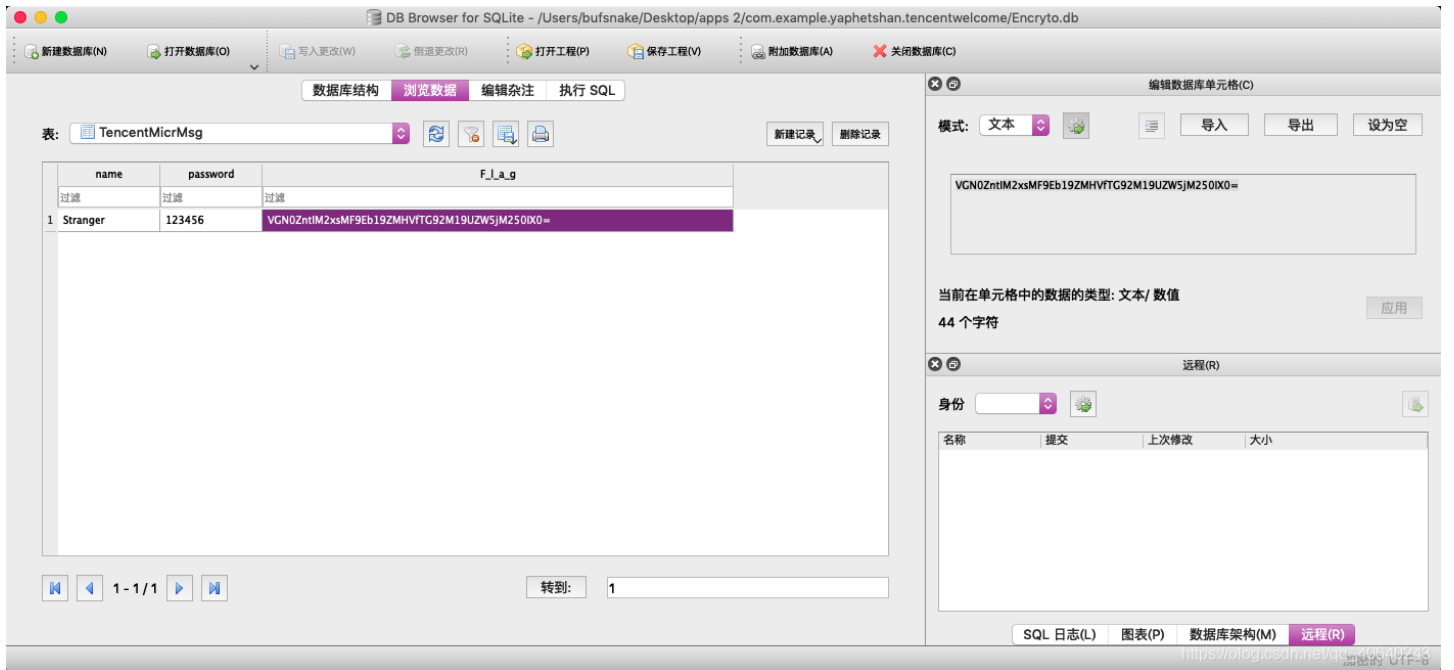
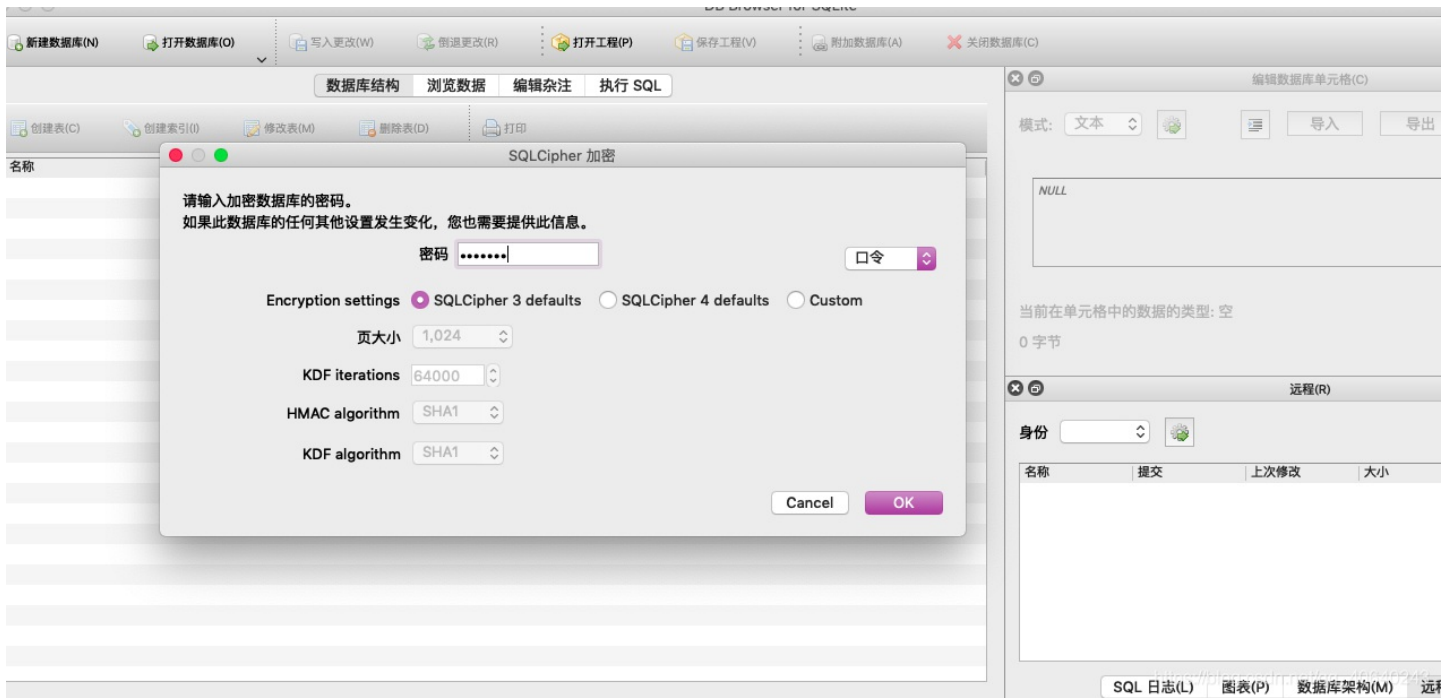
    public static void main(String[] args){
        String two = ("Stra1234"+a("Stra1234")+ "yaphetshan");
        System.out.println(b(two).substring(0,7));
    }
}

```

得到密码

ae56f99

接下来使用DB Browser for SQLite打开需要解密的数据库



VGN0ZntIM2xsMF9Eb19ZMHVfTG92M19UZW5jM250IX0=

base64解密得到flag

Tctf{H3!l0_Do_Y0u_Lov3_Tenc3nt!}

总结

一道题做下来，发现，稳如菜狗