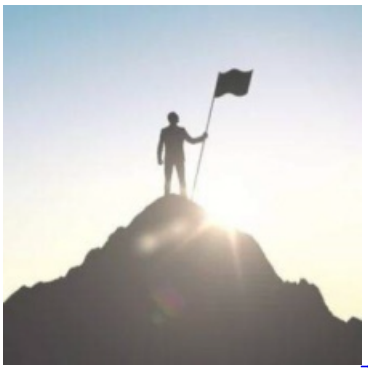
XCTF MOBILE 新手 app3

原创

A_dmins 于 2019-10-03 01:15:50 发布 931 收藏 1

分类专栏： CTF题 XCTF

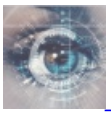
本文链接：https://blog.csdn.net/qq_42967398/article/details/101949942



CTF题 同时被 2 个专栏收录

115 篇文章 11 订阅

订阅专栏



XCTF

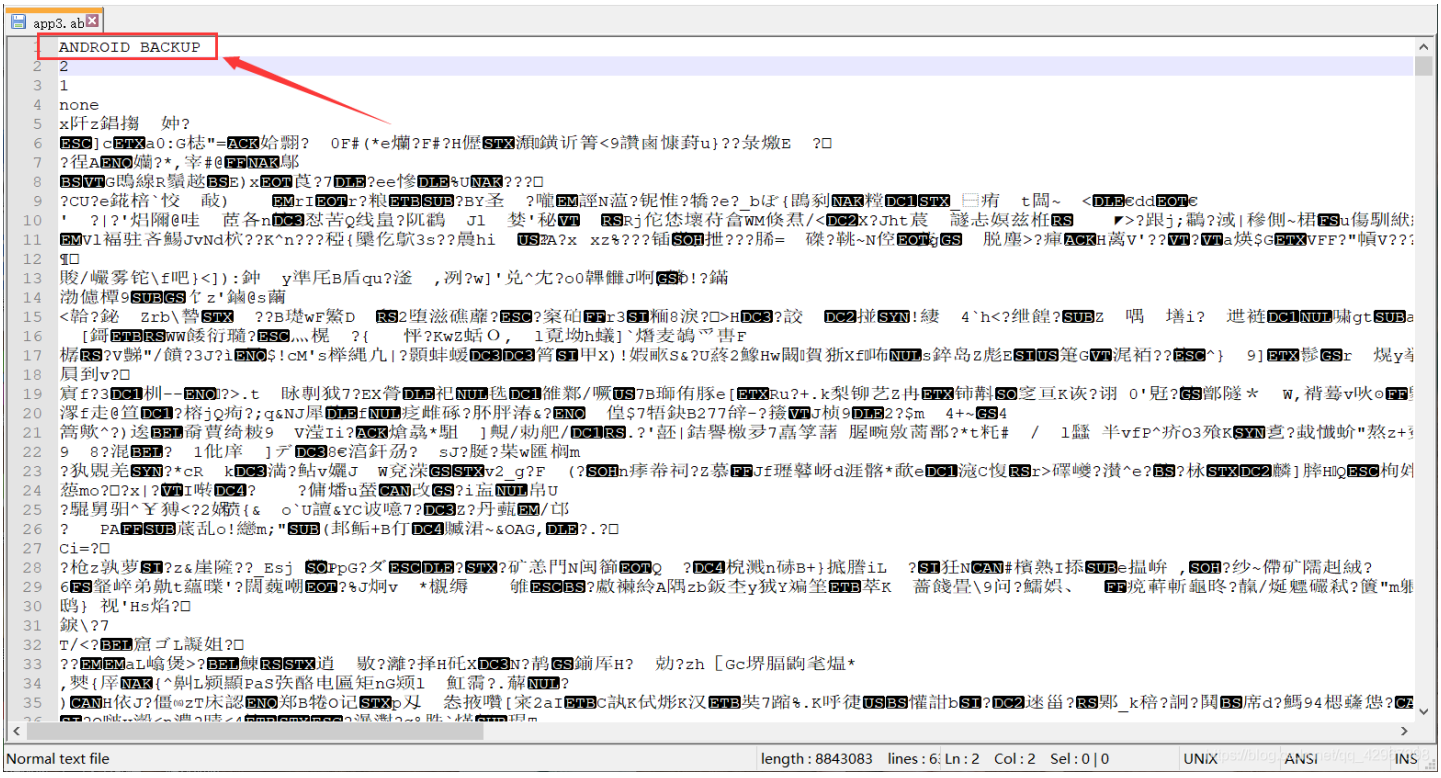
24 篇文章 0 订阅

订阅专栏

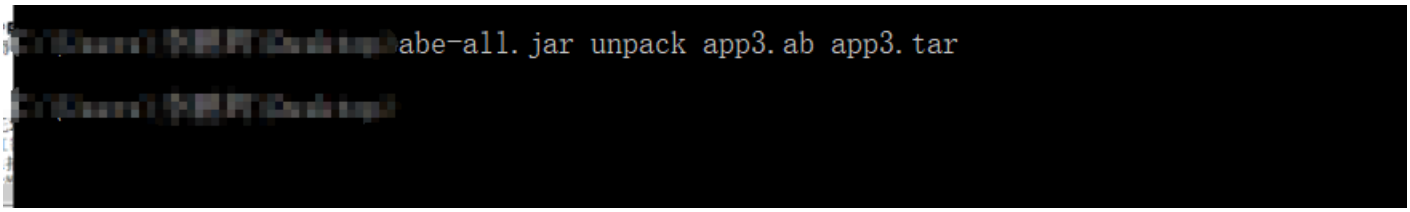
XCTF MOBILE 新手 app3

没事做，随便玩玩，谁知道一玩就是一晚上

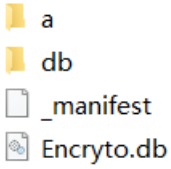
下载下来是个.ab文件，emmm，直接使用notepad++查看一下：



emmm，不知道是啥文件，经过百度查找得知是apk的备份文件
可以使用abe.jar进行反编译，直接使用工具：



解压最后可以得到这些文件：



在a文件夹中找到了apk文件，直接使用APKIDE打开，直接使用java反编译得到主函数：

```
{
  private SQLiteDatabase a;
  private a b;
  private Button c;

  private void a()
  {
    SQLiteDatabase.loadLibs(this);
    this.b = new a(this, "Demo.db", null, 1);
    ContentValues localContentValues = new ContentValues();
    localContentValues.put("name", "Stranger");
    localContentValues.put("password", Integer.valueOf(123456));
    Object localObject = new com.example.yaphetshan.tencentwelcome.a.a();
    String str1 = ((com.example.yaphetshan.tencentwelcome.a.a)localObject).a(localContentValues.getAsString("name"), localContentValues.getAsString("password
    String str2 = ((com.example.yaphetshan.tencentwelcome.a.a)localObject).b(str1, localContentValues.getAsString("password"));
    localObject = ((com.example.yaphetshan.tencentwelcome.a.a)localObject).a(str1 + str2);
    this.a = this.b.getWritableDatabase(((String)localObject).substring(0, 7));
    this.a.insert("TencentMicrMsg", null, localContentValues);
  }

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

  protected void onCreate(Bundle paramBundle)
  {
    super.onCreate(paramBundle);
    setContentView(2130968603);
    this.c = ((Button)findViewById(2131427417));
    this.c.setOnClickListener(this);
    paramBundle = getSharedPreferences("test", 0).edit();
    paramBundle.putString("Is_Encroty", "1");
    paramBundle.putString("Encryto", "SqlCipher");
    paramBundle.putString("ver_sion", "3_4_0");
    paramBundle.apply();
    a();
  }
```


好像是连接数据库啥的，，，，emmmm看样子还调用了a类和b类
直接去看看：

```
package com.example.yaphetshan.tencentwelcome.a;

public class a
{
  private String a = "yaphetshan";

  public String a(String paramString)
  {
```

```
    new b();
    return b.b(paramString + this.a);
  }

  public String a(String paramString1, String paramString2)
  {
    paramString1 = paramString1.substring(0, 4);
    paramString2 = paramString2.substring(0, 4);
    return paramString1 + paramString2;
  }

  public String b(String paramString1, String paramString2)
  {
    new b();
    return b.a(paramString1);
  }
}
```


b类太长了就不截图了，粗略地看了下里面好像有两个加密的函数

```
    for (;;)
    {
      Object localObject;
      int k;
      int j;
      try
      {
        paramString = paramString.getBytes();
        localObject = MessageDigest.getInstance("MD5");
        ((MessageDigest)localObject).update(paramString);
        paramString = ((MessageDigest)localObject).digest();
        k = paramString.length;
        localObject = new char[k * 2];
        j = 0;
      }
      catch (Exception paramString)
      {
        return null;
      }
      paramString = new String((char[])localObject);
      return paramString;
      while (i < k)
      {
        int m = paramString[i];
        int n = j + 1;
        localObject[j] = arrayOfChar[(m >>> 4 & 0xF)];
        j = n + 1;
        localObject[n] = arrayOfChar[(m & 0xF)];
        i += 1;
      }
    }
  }
```



```
  for (;;)
  {
    Object localObject;
    int k;
    int j;
    try
    {
      paramString = paramString.getBytes();
      localObject = MessageDigest.getInstance("SHA-1");
      ((MessageDigest)localObject).update(paramString);
      paramString = ((MessageDigest)localObject).digest();
      k = paramString.length;
      localObject = new char[k * 2];
      j = 0;
    }
    catch (Exception paramString)
    {
      return null;
    }
```

```
    paramString = new String((char[])localObject);
    return paramString;
    while (i < k)
    {
      int m = paramString[i];
      int n = j + 1;
      localObject[j] = arrayOfChar[(m >>> 4 & 0xF)];
      j = n + 1;
      localObject[n] = arrayOfChar[(m & 0xF)];
      i += 1;
    }
  }
}
```


到这里思绪还是乱的，，，，，

稍微整理一下，，，要确定我们接下来该干嘛！！！

首先我们是能看见有数据库的，也就是.db文件

我们使用DB Browser for SQLite打开数据库文件发现需要密码，，，ok，现在目标应该明确了

应该是要我们将密码给找出来，然后读取数据库中的内容，因为这里有提示：

```
public void onCreate(SQLiteDatabase paramSQLiteDatabase)
{
  paramSQLiteDatabase.execSQL("create table TencentMicrMsg(name text,password integer,F_l_a_g text)");
}
```

我们接下来又返回到主函数，，，，

```
private void a()
{
  SQLiteDatabase.loadLibs(this);
  this.b = new a(this, "Demo.db", null, 1);
  ContentValues localContentValues = new ContentValues();
  localContentValues.put("name", "Stranger");
  localContentValues.put("password", Integer.valueOf(123456));
  Object localObject = new com.example.yaphetshan.tencentwelcome.a.a();
  String str1 = ((com.example.yaphetshan.tencentwelcome.a.a)localObject).a(localContentValues.getAsString("name"), localContentValues.getAsString("password"));
  String str2 = ((com.example.yaphetshan.tencentwelcome.a.a)localObject).b(str1, localContentValues.getAsString("password"));
  localObject = ((com.example.yaphetshan.tencentwelcome.a.a)localObject).a(str1 + str2);
  this.a = this.b.getWritableDatabase(((String)localObject).substring(0, 7));
  this.a.insert("TencentMicrMsg", null, localContentValues);
}
```


首先调用了a文件夹中的a类，三个构造函数都调用了

两个字符串进行一系列操作，然后传入getWritableDatabase（）函数

觉得getWritableDatabase这个函数很可疑！！！截取7个字符！！会不会就是数据库的密码呢

首先是给了我们明文的！！name和password，分别是Stranger和123456

进到a类中去查看一下构造函数：

```
public class a
{
  private String a = "yaphetshan";

  public String a(String paramString)
  {
    new b();
    return b.b(paramString + this.a);
  }

  public String a(String paramString1, String paramString2)
  {
    paramString1 = paramString1.substring(0, 4);
    paramString2 = paramString2.substring(0, 4);
    return paramString1 + paramString2;
  }

  public String b(String paramString1, String paramString2)
  {
```

```
        new b();
        return b.a(paramString1);
    }
}
```

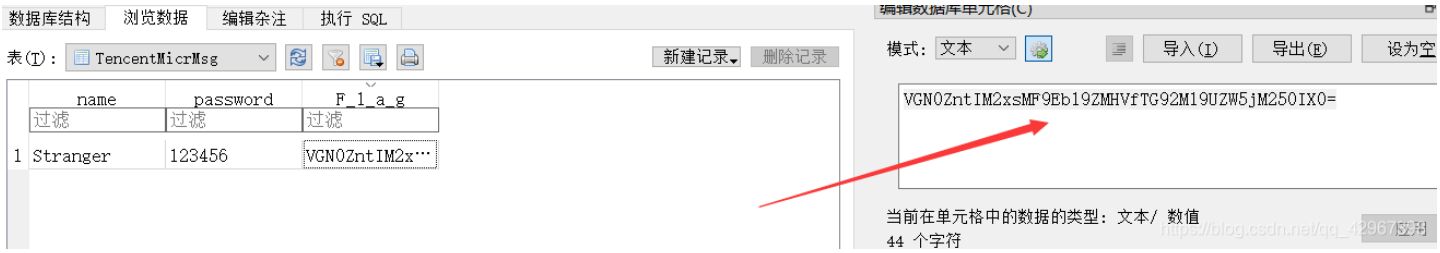

可以很明显的看到三个构造函数根据传入的字符串进行了不同的操作
刚刚b类中的两个函数也用到了，感觉那7个字符的应该就是密码了~~
我们直接将密码生成两个函数模拟一下，重现一下算法生成密码，利用python：

```python
import hashlib

def md5a(string):
 tmp = "0123456789abcdef"
 md5value = hashlib.md5(string.encode('utf-8')).digest()
 k = len(md5value)
 i = 0
 j = 0
 str = [0]*k*2
 while i < k:
  m = md5value[i]
  n = j + 1
  str[j] = tmp[(m >> 4 & 0xf)]
  j = n + 1
  str[n] = tmp[(m & 0xf)]
  i += 1
 return "".join(str)

def shab(string):
 tmp = "0123456789abcdef"
 shavalue = hashlib.sha1(string.encode('utf-8')).digest()
 k = len(shavalue)
 i = 0
 j = 0
 str = [0]*k*2
 while i < k:
  m = shavalue[i]
  n = j + 1
  str[j] = tmp[(m >> 4 & 0xf)]
  j = n + 1
  str[n] = tmp[(m & 0xf)]
  i += 1
 return "".join(str)

a = "yaphetshan"
str1 = "Stra1234"
str2 = md5a(str1)
print(shab(str1+str2+a)[0:7])
```

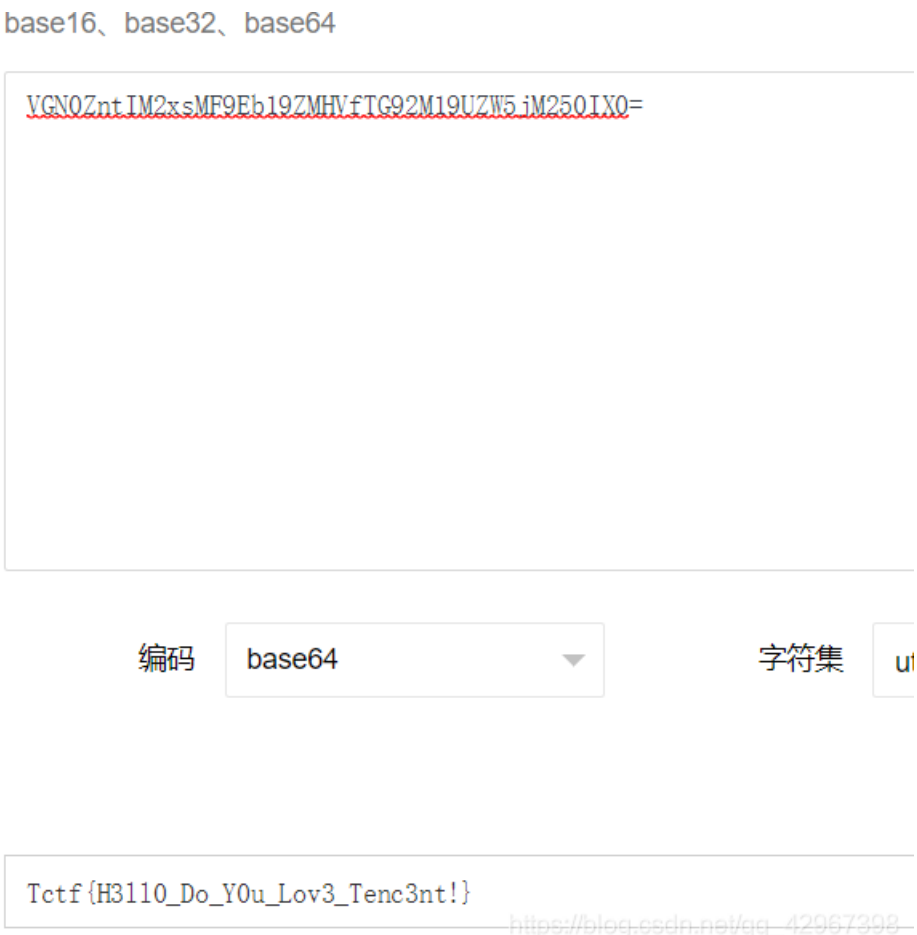
运行得到密码：

```
python 1.py
ae56f99
```

再次利用DB Browser for SQLite打开数据库，输入密码ae56f99成功进入，找到数据：



经过base64解密：



得到flag：Tctf{H3ll0_Do_Y0u_Lov3_Tenc3nt!}

我是真的服了，这种题目是给新手做的吗？？日了狗了，看看评论都，，，太难了啊