XCTF-mobile app3

原创

夏了茶糜 于 2020-03-22 21:52:26 发布 381 收藏

分类专栏： CTF-REVERSE 文章标签： 安全 java

app3-安卓逆向
题目下载地址

app3-安卓逆向

1.提取文件

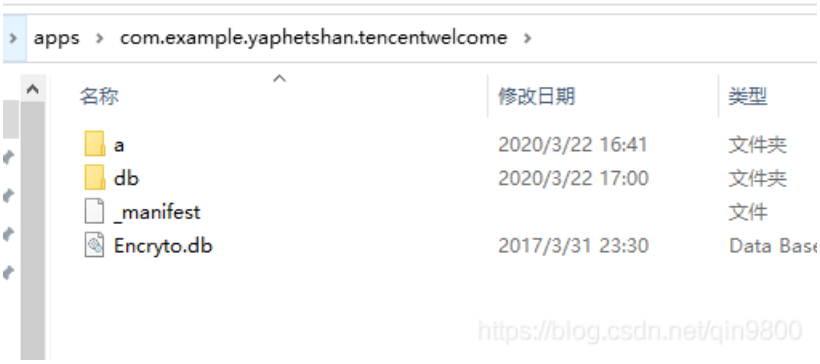
下载题目后得到一个是.ab 后缀名的文件，查下资料

> 什么是一 .AB 文件？
> Android的SDK（软件开发工具包）程序创建并访问附加的文件.ab扩展。该软件有一个版本的基于Microsoft Windows的系统，用户可以安装其他版本的基于Mac平台。而Android调试桥应用程序是一个小程序集成到Android的SDK软件，这是一个命令行程序。文件中的AB格式的内容包括参考使用Android开发项目的输出文件的开发库和其他文件。这些AB文件是用来还原关联使用Android SDK软件创建一个Android应用程序开发项目的数据备份文件。谷歌开发了Android SDK软件和AB文件格式。一般，AB文件不包括APK文件，它们是Android移动应用程序本身，而只是快照和引用其他文件和元素。 “亚行备份-all”是一个命令的用户可以键入创建一个Android SDK项目的AB备份文件，并将其存储在该项目中的目录。在另一方面“亚行备份-apk -all”创建与目录相关的APK文件，其中Android的SDK项目发现了一个AB的备份文件。在“亚行恢复backup.ab”命令将恢复应用程序

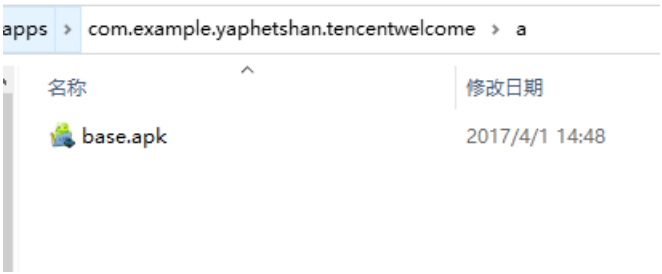
使用abe-all.jar文件吧.ab文件转换为tar文件

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

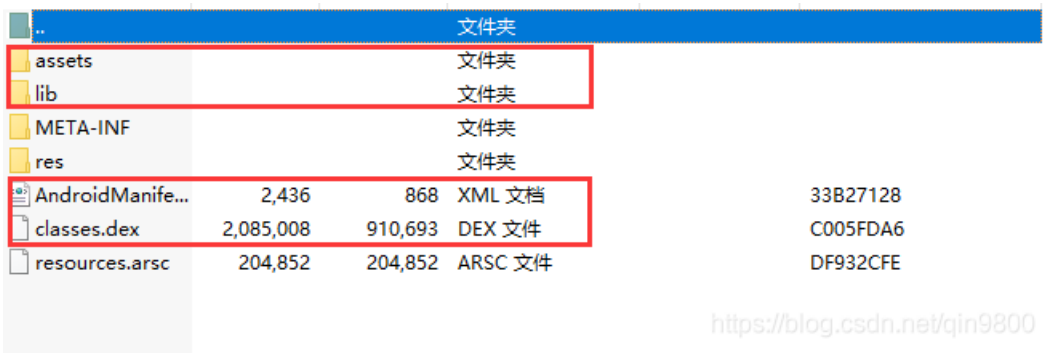
解压后得到



文件中有两个db文件，这是数据库文件，尝试使用DB Browser for SQLite打开，发现数据库被加密了，尝试逆序APP找到密码



a文件夹下有一个APK文件，更改app安装包后缀为zip，



把这四个文件提取出来。
使用dex2jar将dex文件转换为jar文件，得到一个jar文件，这个文件中包含了大部分程序逻辑

```
d2j-dex2jar.bat classes.dex
```

2.分析文件

□先查看MainActivity，可以看到onCreate后，调□了a函数。

```
private void a() {
    SQLiteDatabase.loadLibs(this);
    this.b = new a(this, "Demo.db", null, 1);
    ContentValues contentValues = new ContentValues();
    contentValues.put("name", "Stranger");
    contentValues.put("password", Integer.valueOf(123456));
    a aVar = new a();
    String a = aVar.a(contentValues.getAsString("name"), contentValues.getAsString("password"));
    this.a = this.b.getWritableDatabase(aVar.a(a + aVar.b(a, contentValues.getAsString("password"))).substring(0, 7));
    this.a.insert("TencentMicrMsg", null, contentValues);
}
```

重点分析这里可以的到密码

aVar.a(contentValues.getAsString(“name”), contentValues.getAsString(“password”));
得到Stra1234

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

/* compiled from: Cipher */
public class a {
    private String a = "yaphetshan";

    public String a(String str, String str2) {
        String substring = str.substring(0, 4);
        return substring + str2.substring(0, 4);
    }

    public String b(String str, String str2) {
        b bVar = new b();
        return b.a(str);
    }

    public String a(String str) {
        b bVar = new b();
        return b.b(str + this.a);
    }
}
```


把这个分解下

```java
a aVar = new a();
String a = aVar.a(contentValues.getAsString("name"), contentValues.getAsString("password"));
this.a = this.b.getWritableDatabase(aVar.a(a + aVar.b(a, contentValues.getAsString("password"))).substring(0, 7));
this.a.insert("TencentMicrMsg", null, contentValues);
```

tmp = aVar.b(a, contentValues.getAsString(“password”))

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

/* compiled from: Cipher */
public class a {
    private String a = "yaphetshan";

    public String a(String str, String str2) {
        String substring = str.substring(0, 4);
        return substring + str2.substring(0, 4);
    }

    public String b(String str, String str2) {
        b bVar = new b();
        return b.a(str);
    }

    public String a(String str) {
        b bVar = new b();
        return b.b(str + this.a);
    }
}
```


这里调用了b类中的a方法只传递了第一个参数（“Stra1234”）。
再分析b类中的a方法

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

import java.security.MessageDigest;

/* compiled from: SHA1Manager */
public class b {
    public static final String a(String str) {
        int i = 0;
        char[] cArr = new char[]{'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);
            byte[] digest = instance.digest();
            int length = digest.length;
            char[] cArr2 = new char[(length * 2)];
            int i2 = 0;
            while (i < length) {
                byte b = digest[i];
                int i3 = i2 + 1;
                cArr2[i2] = cArr[(b >>> 4) & 15];
                i2 = i3 + 1;
                cArr2[i3] = cArr[b & 15];
                i++;
            }
            return new String(cArr2);
        } catch (Exception e) {
            return null;
        }
    }
```


经过b类中的a方法加密后返回一个值，假设为tmp。这里的a是“Stra1234”。

aVar.a(a + tmp).substring(0, 7)

这里接着调用a类中的a方法，因为这里只有一个参数所以调用的是这个。

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

/* compiled from: Cipher */
public class a {
    private String a = "yaphetshan";

    public String a(String str, String str2) {
        String substring = str.substring(0, 4);
        return substring + str2.substring(0, 4);
    }

    public String b(String str, String str2) {
        b bVar = new b();
        return b.a(str);
    }

    public String a(String str) {
        b bVar = new b();
        return b.b(str + this.a);
    }
}
```


这个方法中调用了b类的b方法，参数是 `“Stra1234”+ tmp+"yaphetshan"`

```java
    }

    public static final String b(String str) {
        int i = 0;
        char[] cArr = new char[]{'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);
            byte[] digest = instance.digest();
            int length = digest.length;
            char[] cArr2 = new char[(length * 2)];
            int i2 = 0;
            while (i < length) {
```

```
            while (i < length) {
                byte b = digest[i];
                int i3 = i2 + 1;
                cArr2[i2] = cArr[(b >>> 4) & 15];
                i2 = i3 + 1;
                cArr2[i3] = cArr[b & 15];
                i++;
            }
            return new String(cArr2);
        } catch (Exception e) {
            return null;
        }
    }
}
```


最后的返回值要取前7位，大概算法分析完成
a方法和b方法我们可以直接复制出来用，就不用使用别的语言再单独实现了，其实这个算法并不复杂

3.解密脚本

```
/*
* @Author: 夏了茶糜
* @Date:   2020-03-22 21:16:41
* @Last Modified by:   夏了茶糜
* @Last Modified time: 2020-03-22 21:26:55
*/
import java.security.MessageDigest;

public class main {
    public static void main(String[] args)
    {
        String str2 = "Stra1234";
        String str3 = a(str2);
        System.out.print("密钥 = ");
        System.out.print(b(str2 + str3 + "yaphetshan").substring(0,7));
    }

    public static final String a(String str) {
        int i = 0;
        char[] cArr = new char[]{'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);
            byte[] digest = instance.digest();
            int length = digest.length;
            char[] cArr2 = new char[(length * 2)];
            int i2 = 0;
            while (i < length) {
                byte b = digest[i];
                int i3 = i2 + 1;
                cArr2[i2] = cArr[(b >>> 4) & 15];
                i2 = i3 + 1;
                cArr2[i3] = cArr[b & 15];
                i++;
            }
            return new String(cArr2);
        } catch (Exception e) {
            return null;
        }
    }
```

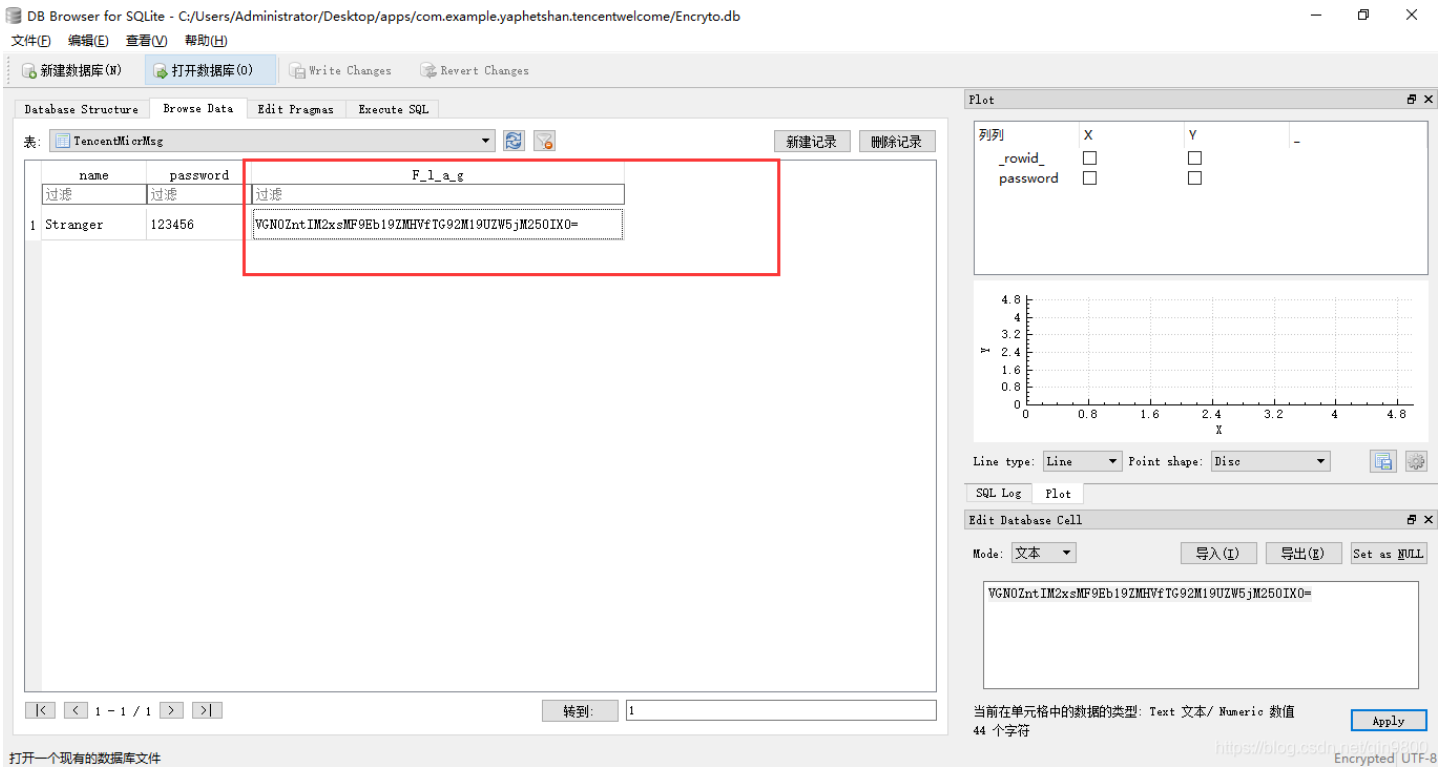
```java
public static final String b(String str) {
        int i = 0;
        char[] cArr = new char[]{'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f'}
;
        try {
            byte[] bytes = str.getBytes();
            MessageDigest instance = MessageDigest.getInstance("SHA-1");
            instance.update(bytes);
            byte[] digest = instance.digest();
            int length = digest.length;
            char[] cArr2 = new char[(length * 2)];
            int i2 = 0;
            while (i < length) {
                byte b = digest[i];
                int i3 = i2 + 1;
                cArr2[i2] = cArr[(b >>> 4) & 15];
                i2 = i3 + 1;
                cArr2[i3] = cArr[b & 15];
                i++;
            }
            return new String(cArr2);
        } catch (Exception e) {
            return null;
        }
    }

}
```

```
C:\Users\Administrator\Desktop>java main
KEY = ae56f99
```

得到密钥ae56f99

使用DB Browser for SQLite打开数据库



得到一个加密的数据

```
VGN0ZntIM2xsMF9Eb19ZMHVfTG92M19UZW5jM250IX0=
```

看着像base64

```
In [8]: import base64

In [1]: base64.b64decode("VGN0ZntIM2xsMF9Eb19ZMHVfTG92M19UZW5jM250IX0=".encode("UTF-8")).decode("UTF-8")
Out[1]: 'Tctf{H3ll0_Do_Y0u_Lov3_Tenc3nt!}'
```

得到flag

```
Tctf{H3ll0_Do_Y0u_Lov3_Tenc3nt!}
```