xctf---easyjava的WriteUp

原创

windy_ll 于 2020-03-04 16:03:54 发布 363 收藏

文章标签： 安卓 安全

本文链接：https://blog.csdn.net/qq_41374107/article/details/104655458

xctf---easyjava的WriteUp

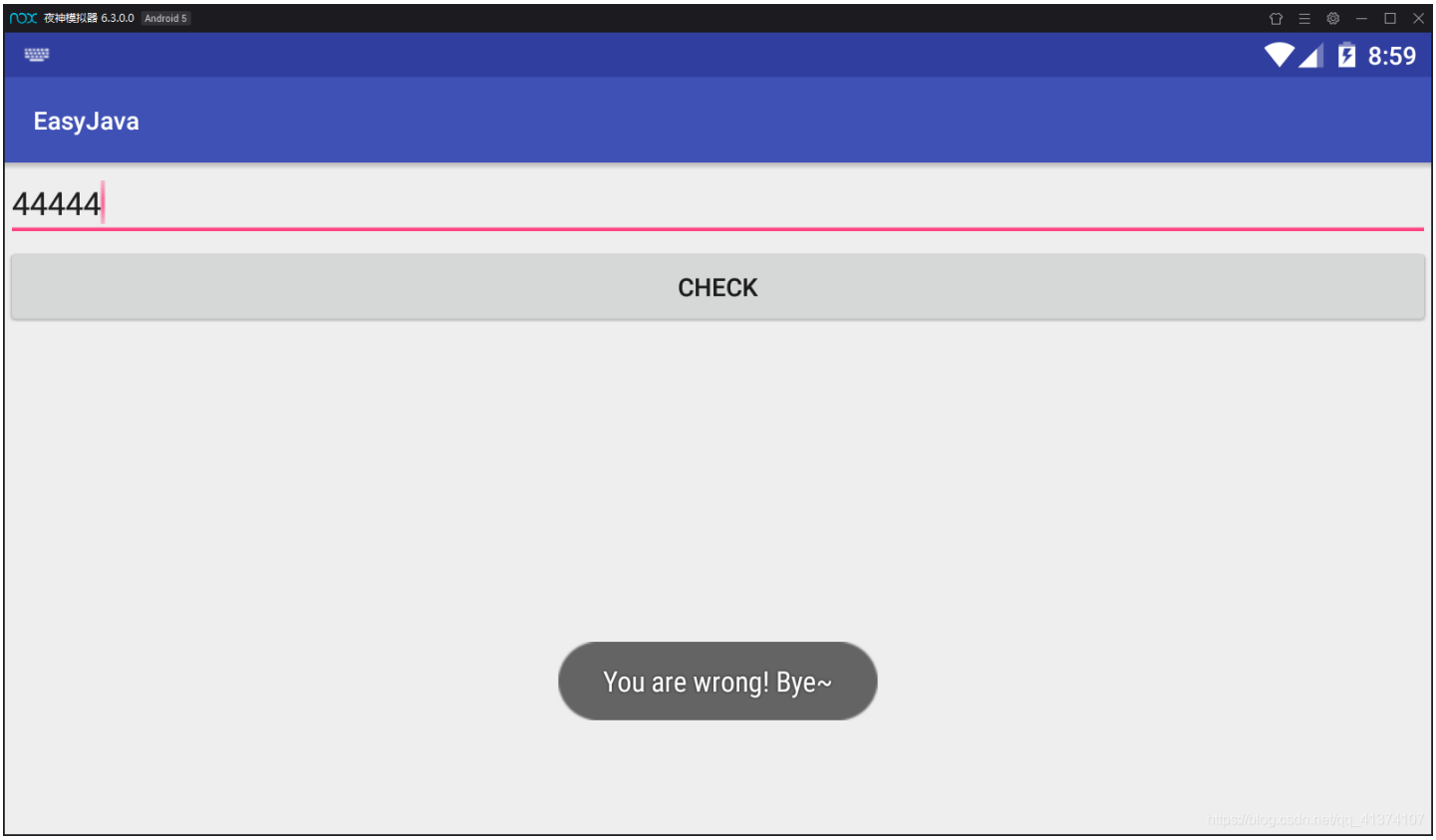
一、题目来源

题目来源：XCTF题库安卓区easyjava

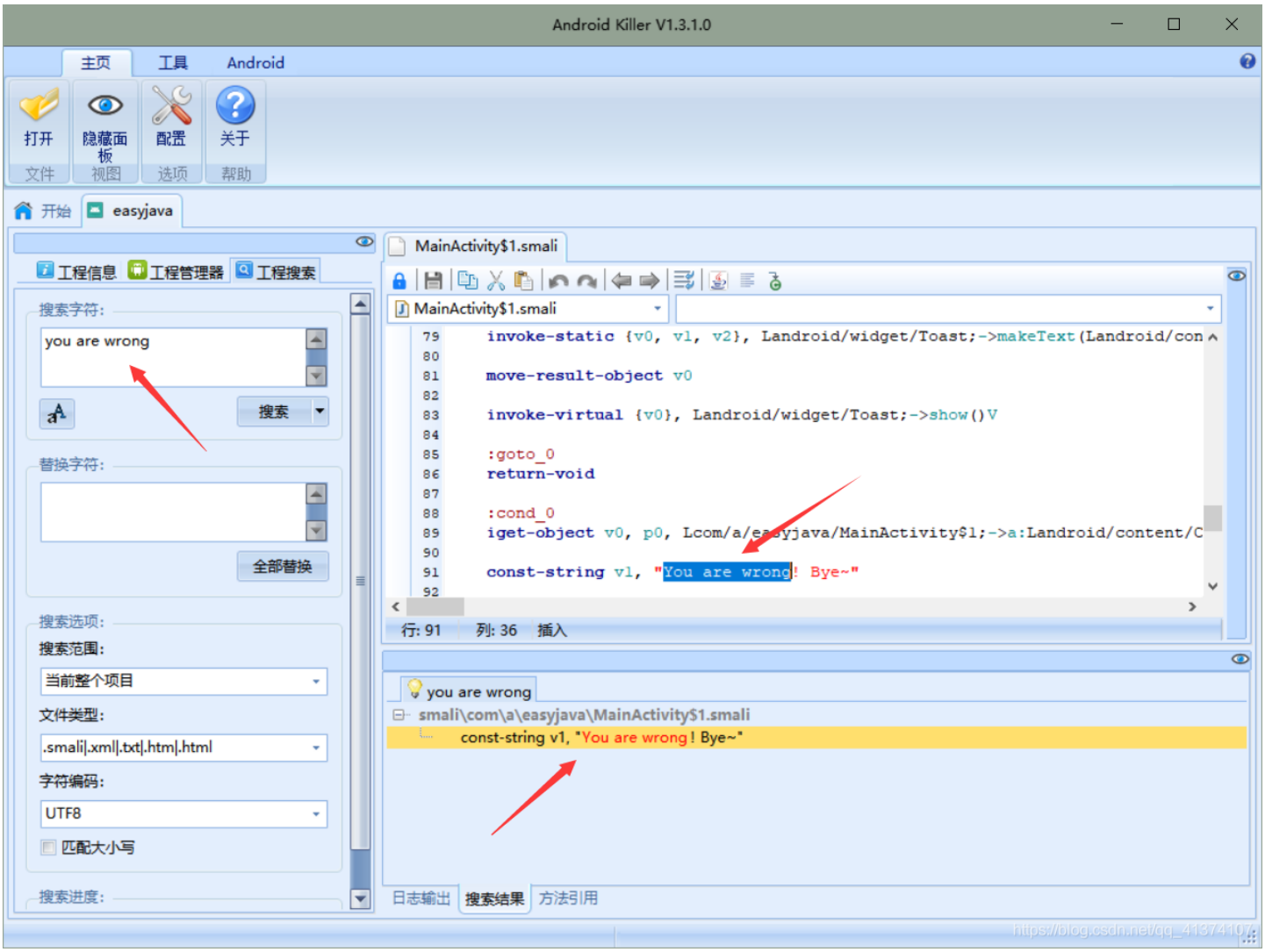
题目下载链接：下载地址

二、解题过程

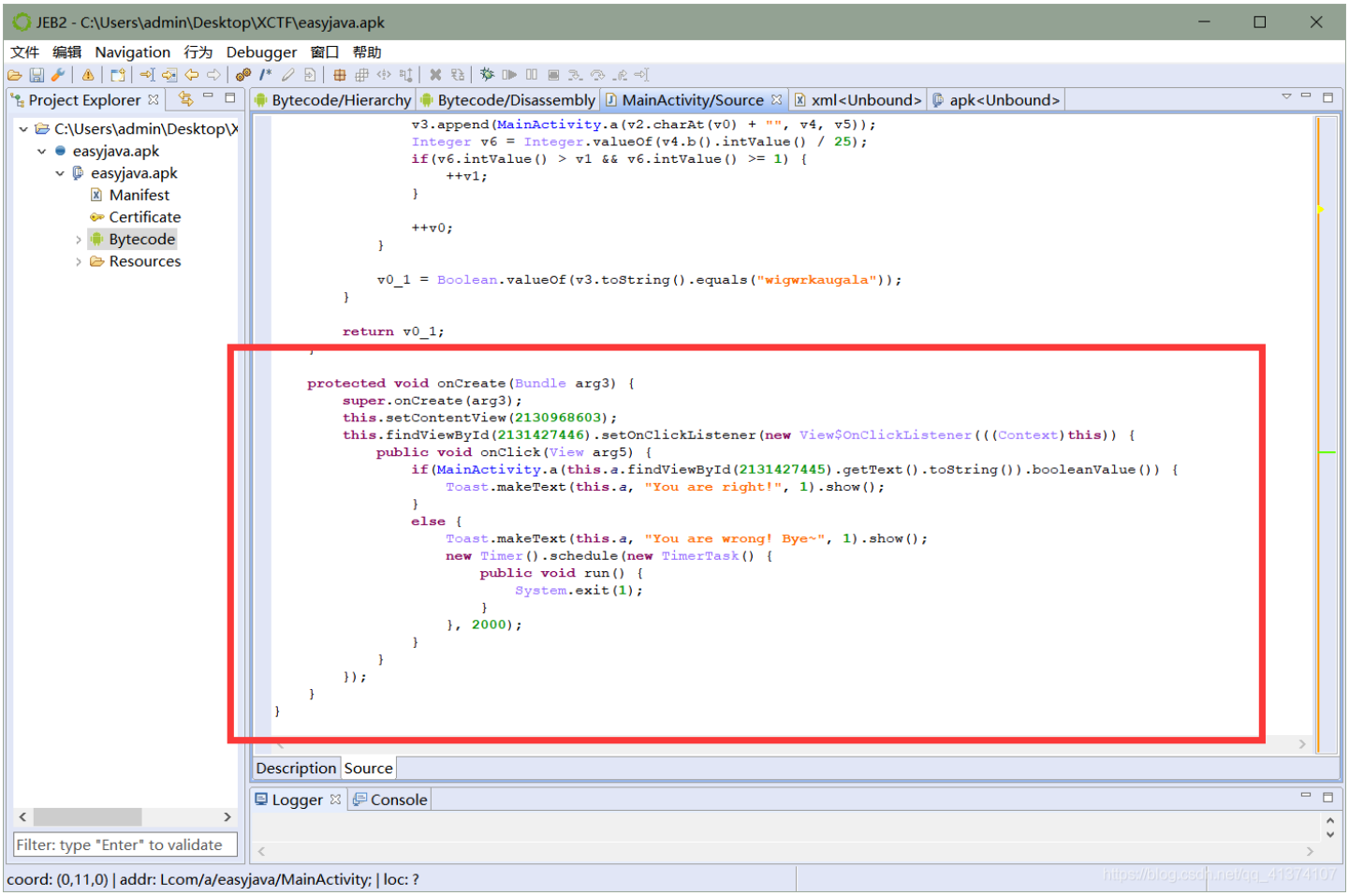
1、将该apk安装进夜神模拟器中，发现有一个输入框和一个按钮，随便输入信息，点击按钮，发现弹出信息`You are wrong!Bye~`。


夜神模拟器 6.3.0.0
Android 5
8:59
EasyJava
44444
CHECK
You are wrong! Bye~

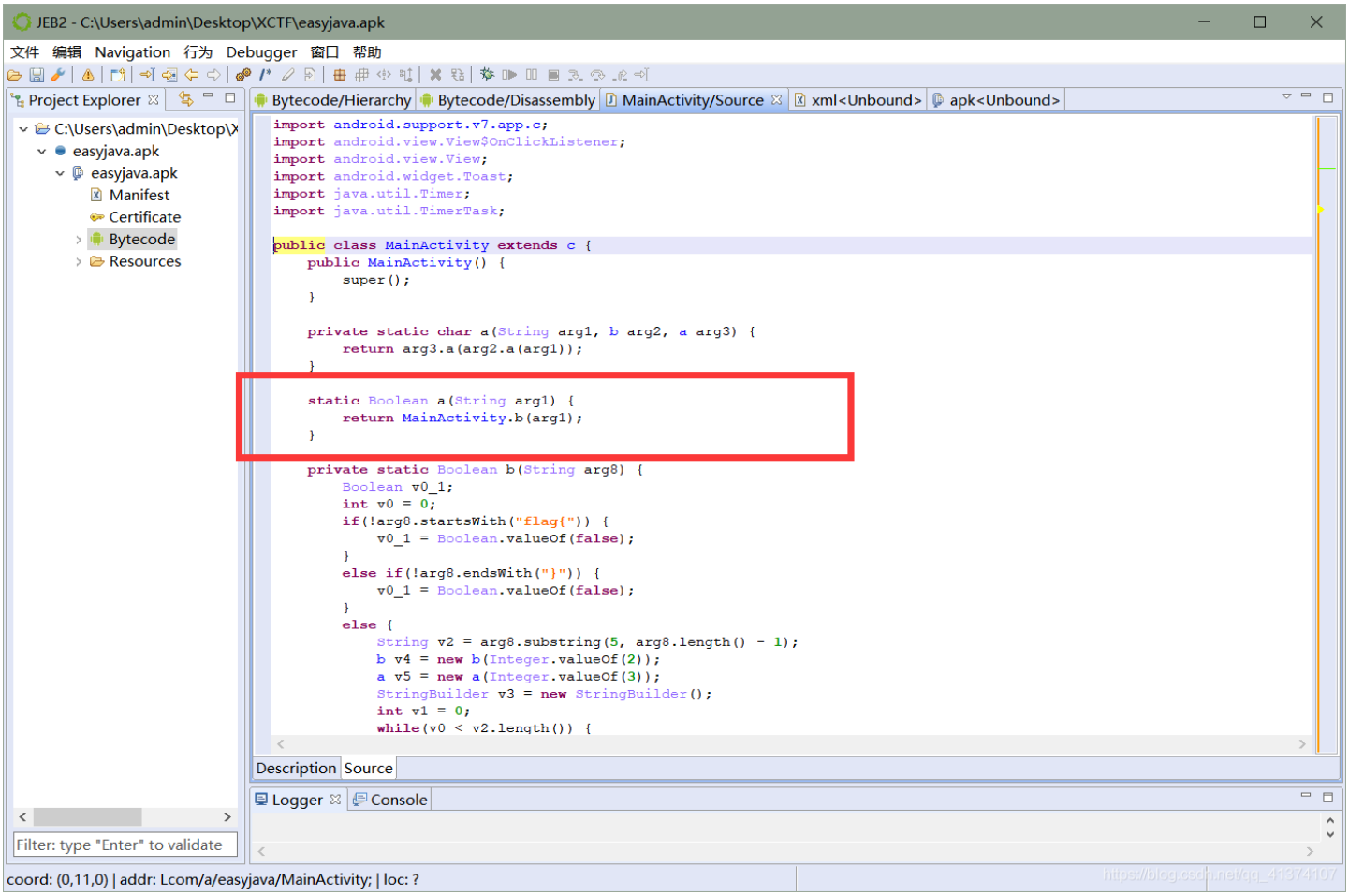
2、将该APK拖进AndroidKiller中反编译，反编译完成后，搜索字符串`You are wrong`，发现该字符串位于MainActivity.java中，如下图所示：

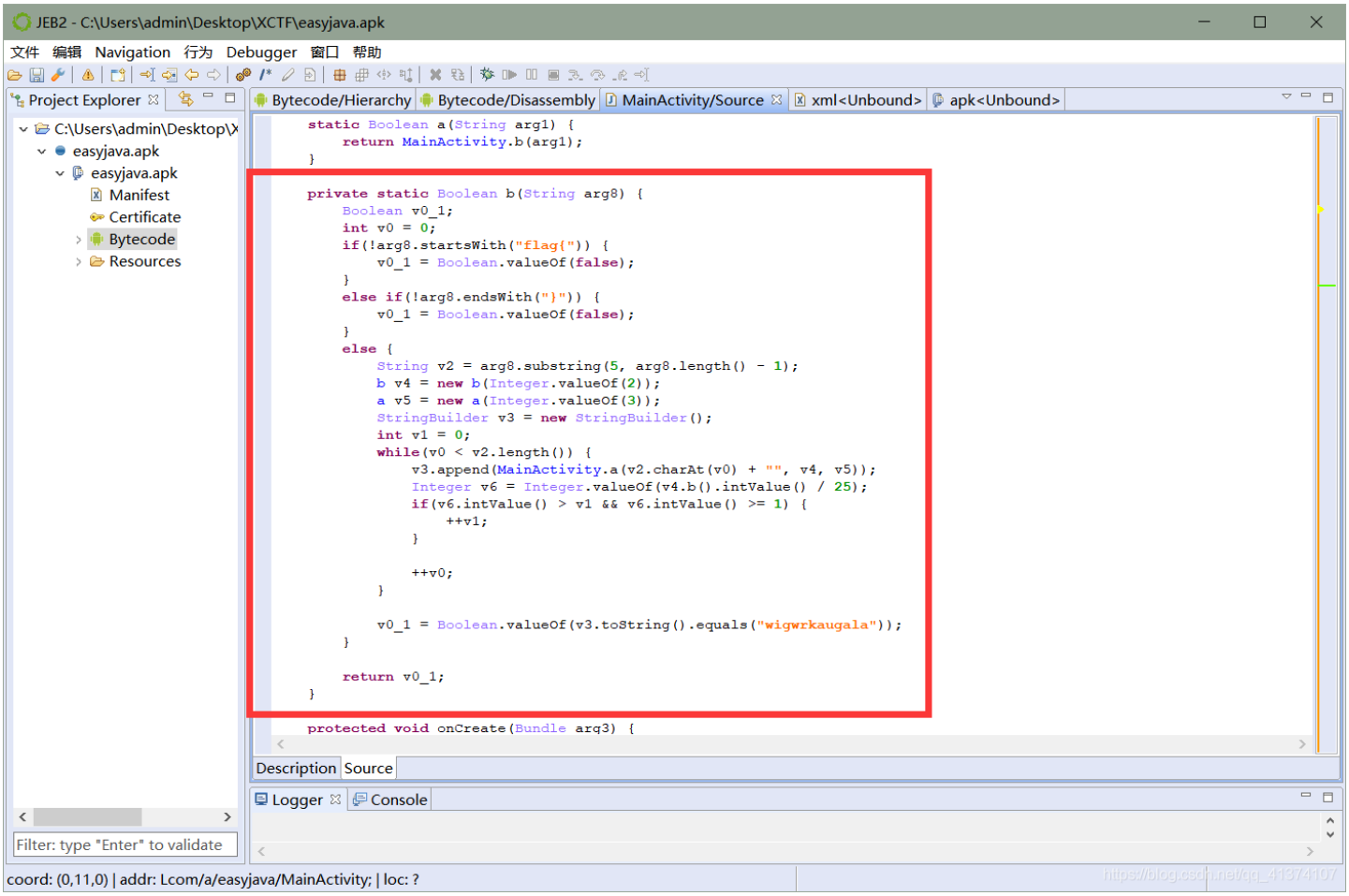


3、用jeb反编译该apk，反编译完成后直接将smali层代码转为java代码，发现在`onCreate`函数出现该字符串，发现将输入框中的字符串作为参数传入MainActivity的`a(string)`函数中，返回的布尔值确认flag是否正确。

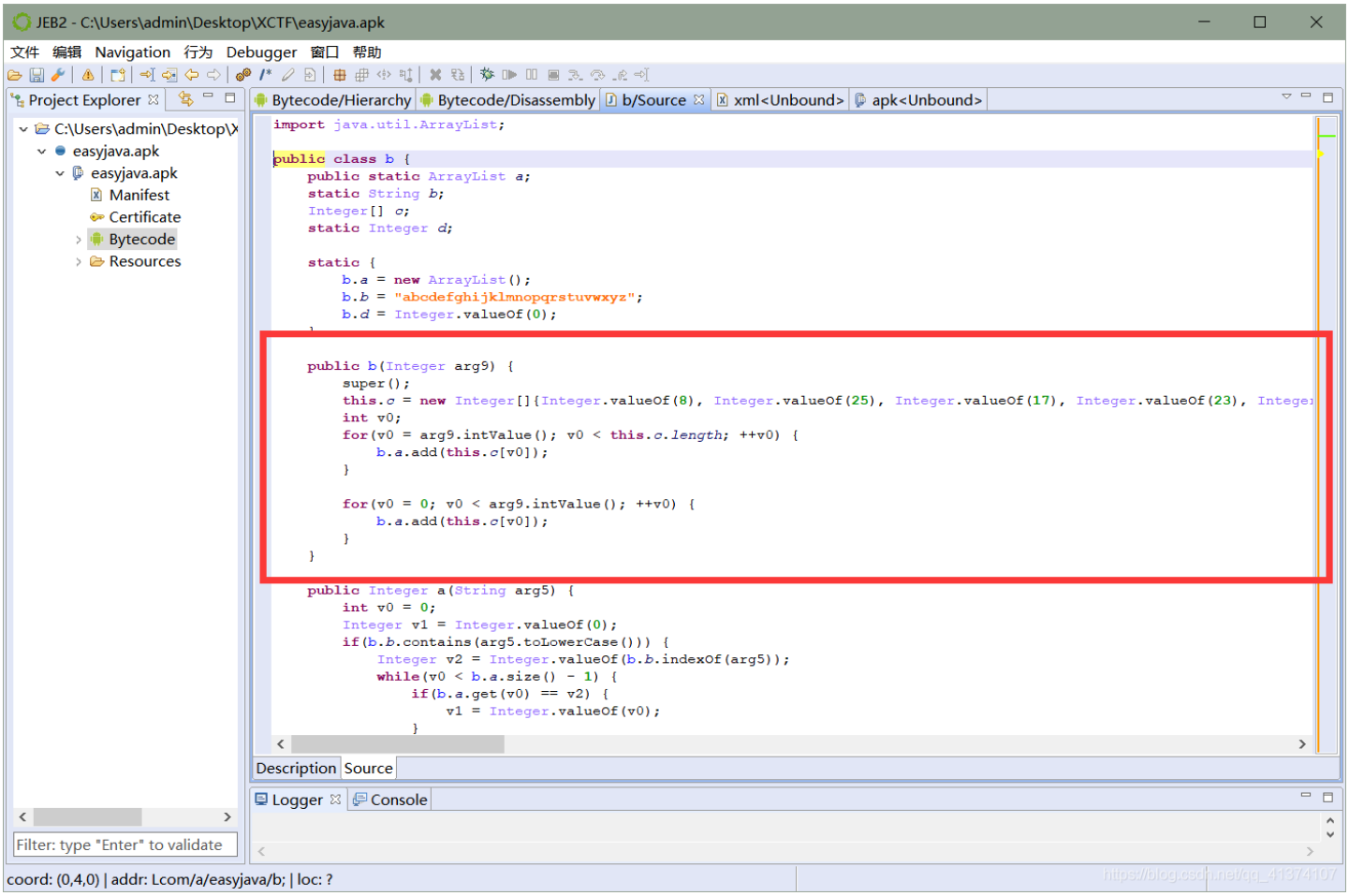


4、而该a函数调用了MainActivity的b函数，b函数返回一个布尔值，b函数作用为首先获取输入框的字符串截取前5个符`flag{`与最后一个字符`}`之间的字符串。

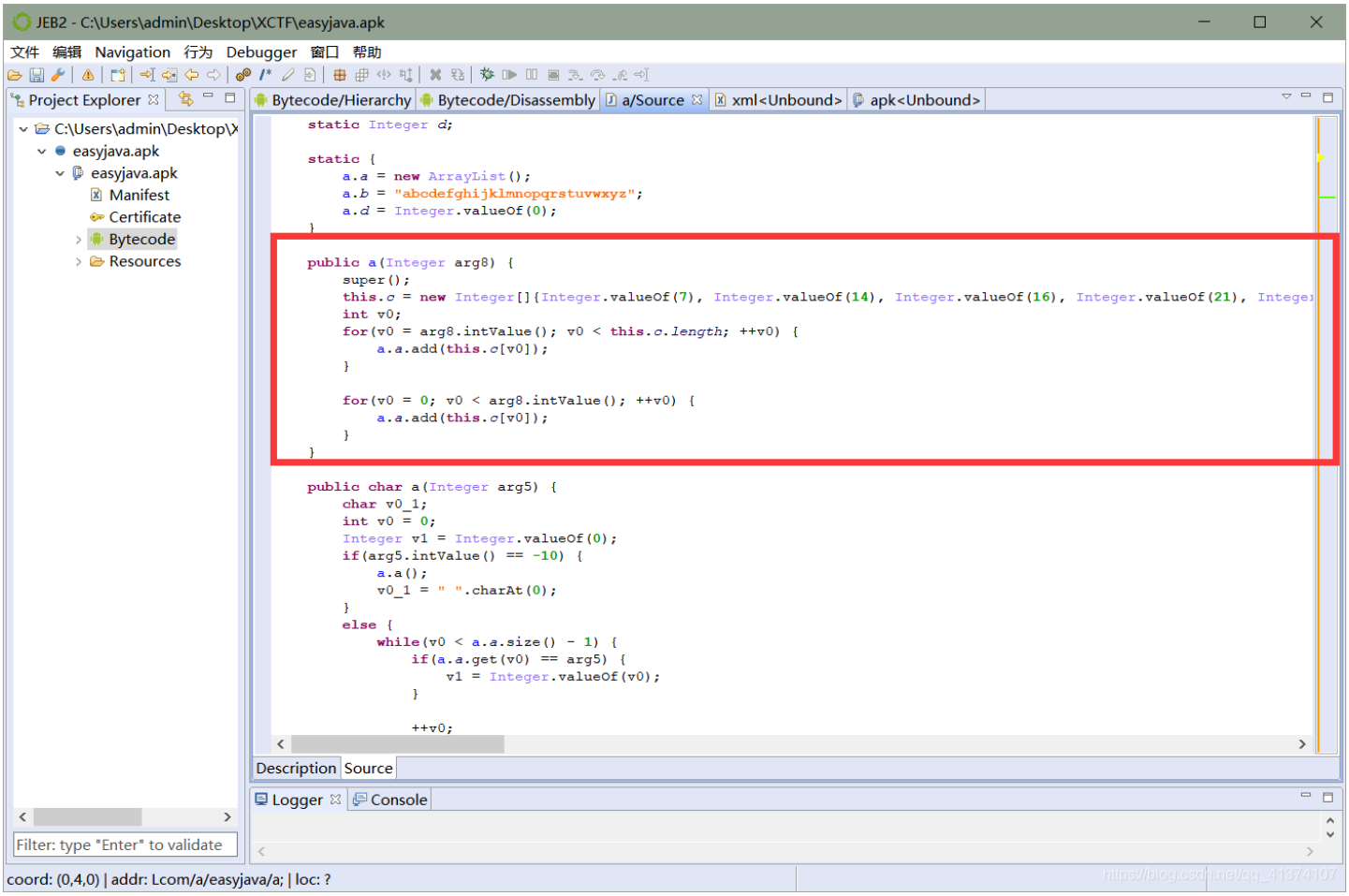

JEB2 - C:\Users\admin\Desktop\XCTF\easyjava.apk
文件 编辑 Navigation 行为 Debugger 窗口 帮助
Project Explorer
C:\Users\admin\Desktop\X
easyjava.apk
easyjava.apk
Manifest
Certificate
Bytecode
Resources
Filter: type "Enter" to validate
Bytecode/Hierarchy
Bytecode/Disassembly
MainActivity/Source
xml<Unbound>
apk<Unbound>
import android.support.v7.app.c;
import android.view.View$OnClickListener;
import android.view.View;
import android.widget.Toast;
import java.util.Timer;
import java.util.TimerTask;
public class MainActivity extends c {
public MainActivity() {
super();
}
private static char a(String arg1, b arg2, a arg3) {
return arg3.a(arg2.a(arg1));
}
static Boolean a(String arg1) {
return MainActivity.b(arg1);
}
private static Boolean b(String arg8) {
Boolean v0_1;
int v0 = 0;
if(!arg8.startsWith("flag(")) {
v0_1 = Boolean.valueOf(false);
}
else if(!arg8.endsWith("}")) {
v0_1 = Boolean.valueOf(false);
}
else {
String v2 = arg8.substring(5, arg8.length() - 1);
b v4 = new b(Integer.valueOf(2));
a v5 = new a(Integer.valueOf(3));
StringBuilder v3 = new StringBuilder();
int v1 = 0;
while(v0 < v2.length()) {
Description
Source
Logger
Console
coord: (0,11,0) | addr: Lcom/a/easyjava/MainActivity; | loc: ?



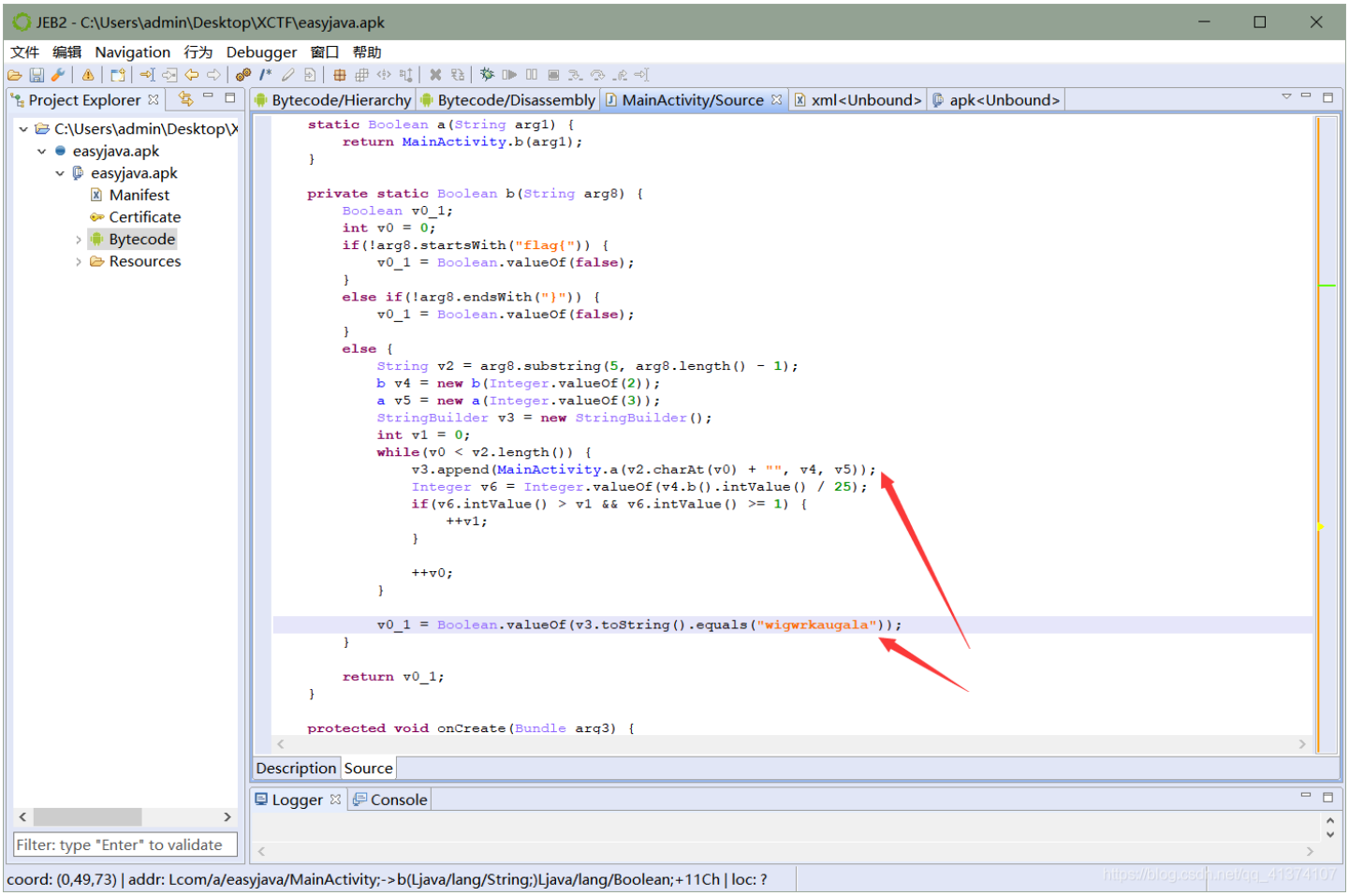
然后实例化了一个b类，并且传入了一个参数2，而b类的构造函数首先为一个整形数组复制，然后左移该整形数组，左移的次数就是传进来的整数，这里是2。



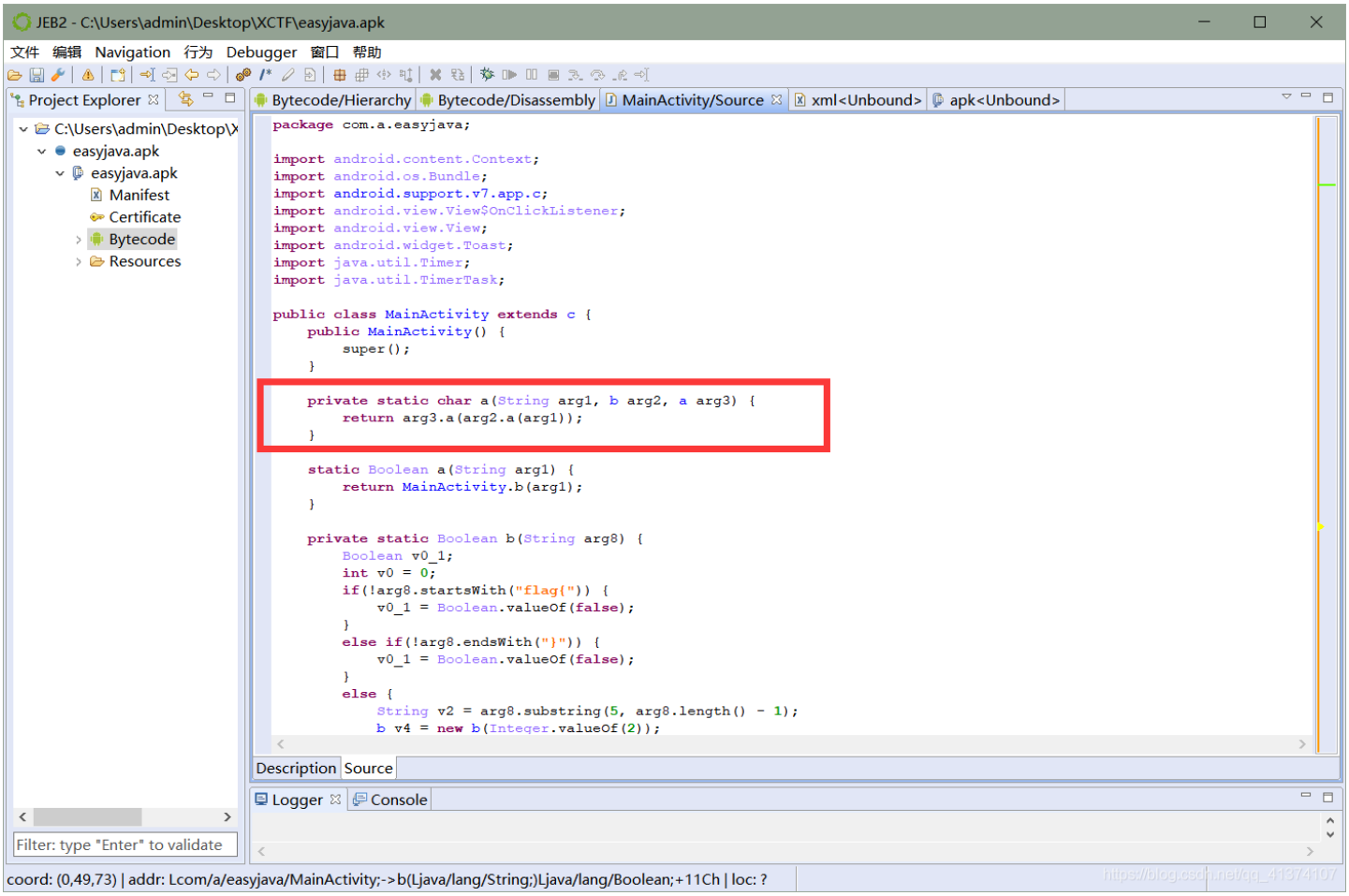
在然后实例化了一个a类，传入了一个参数3，而a类的构造函数与b类的构造函数基本相同，就不再说了。



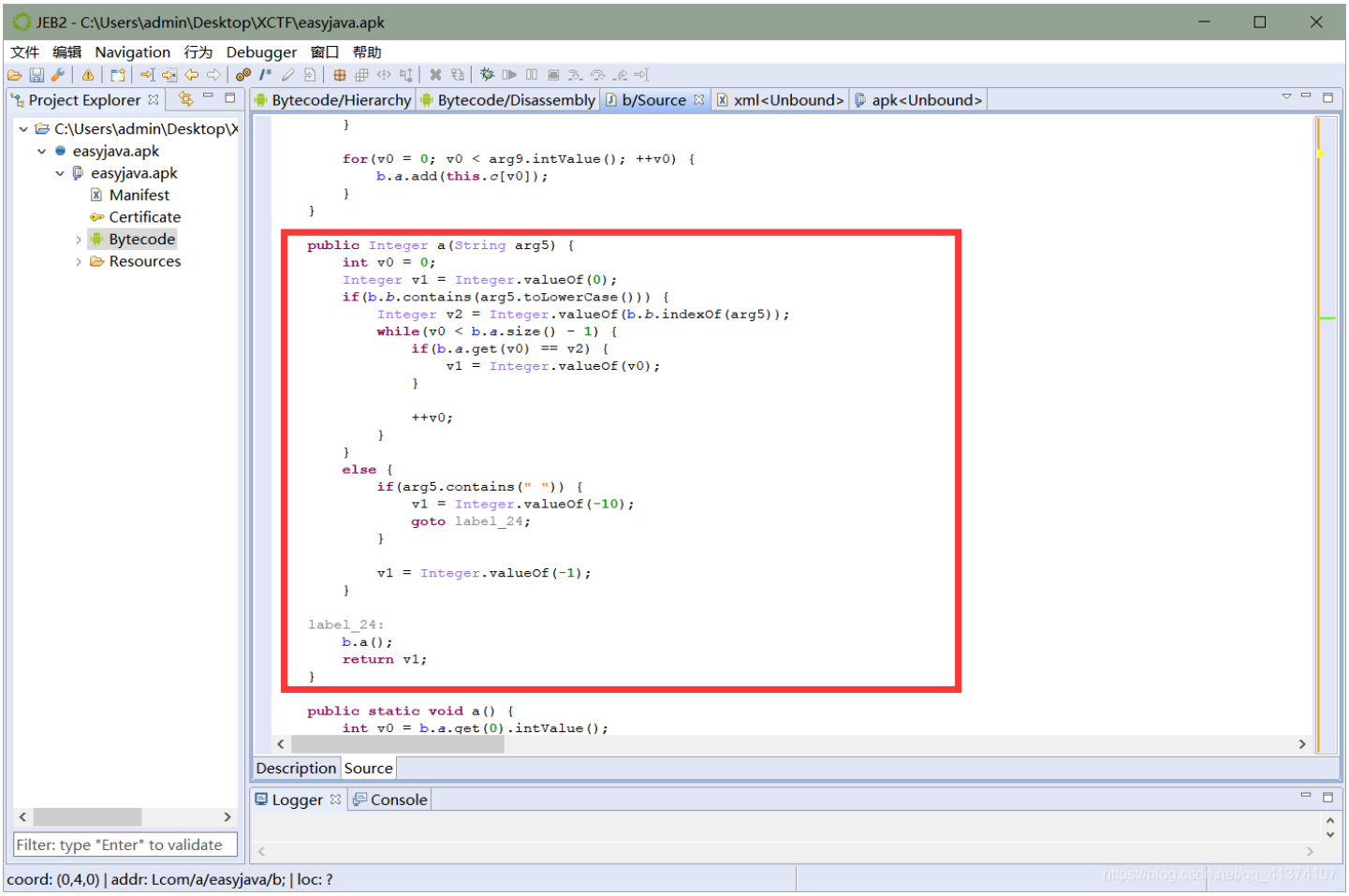
之后遍历截取下来的字符串的每个字符，然后将调用MainActivity中的`a(string,b,a)`函数返回回来的字符串添加到变量v3的尾部，遍历完后，将得到的字符串与字符串`wigwrkaugala`比较，若结果为真，返回`True`，否则返回`False`。

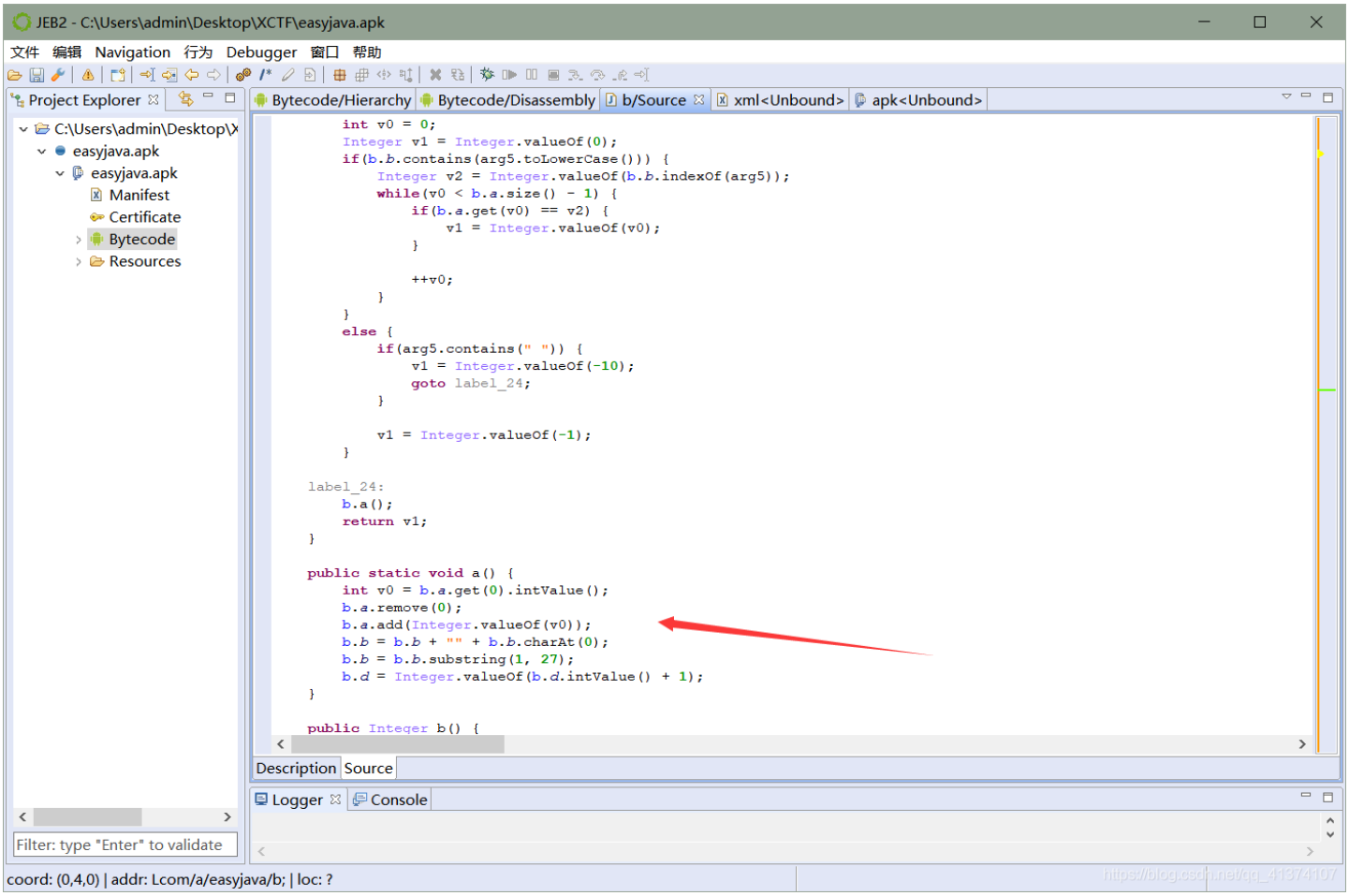


5、接着来看MainActivity的`a(string,b,a)`函数，该函数首先将传进来的第一个字符串作为调用b类的a函数的参数传入，然后将返回回来的结果作为调用a类的a函数的参数传入，最后返回一个字符。

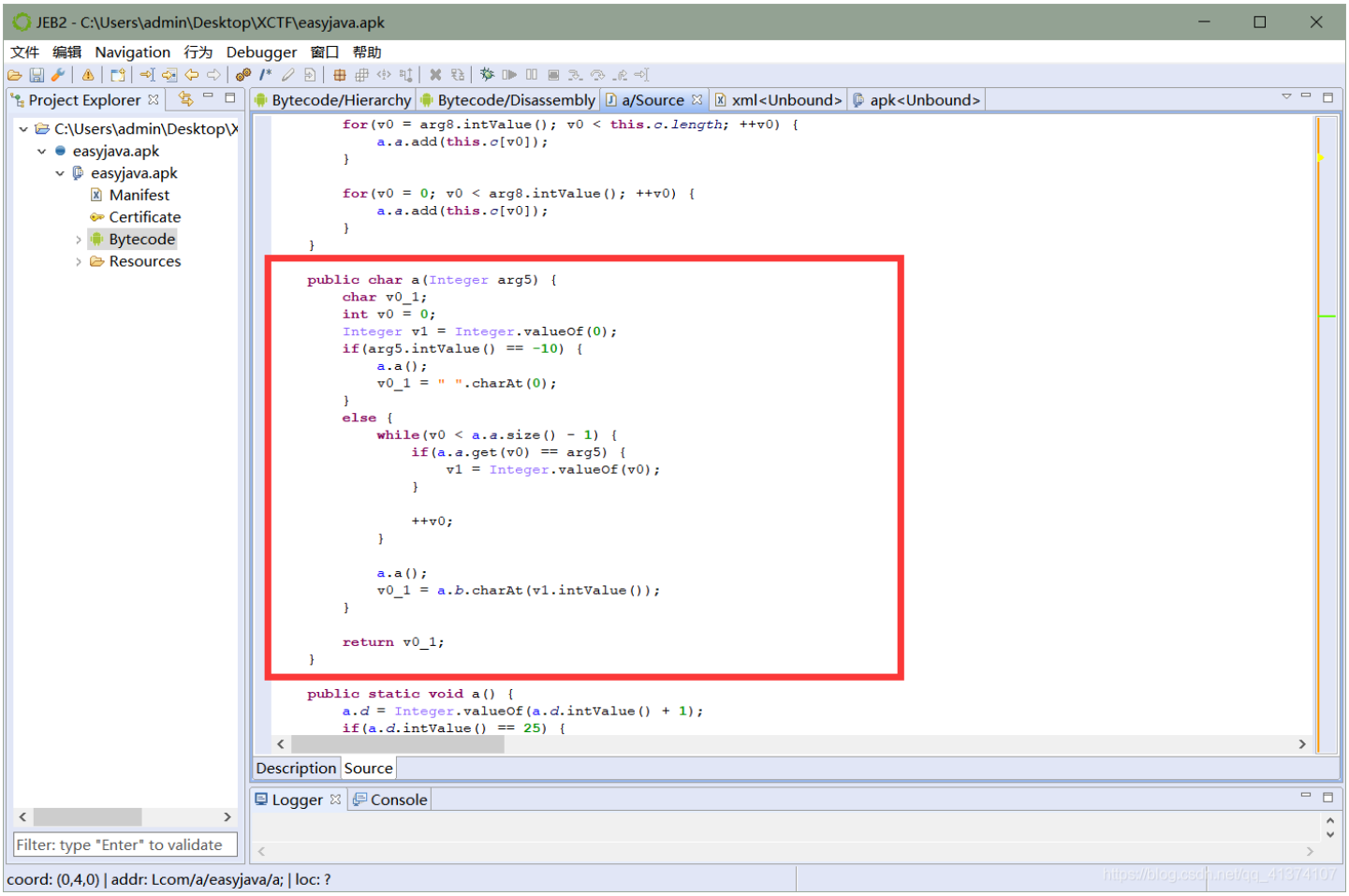


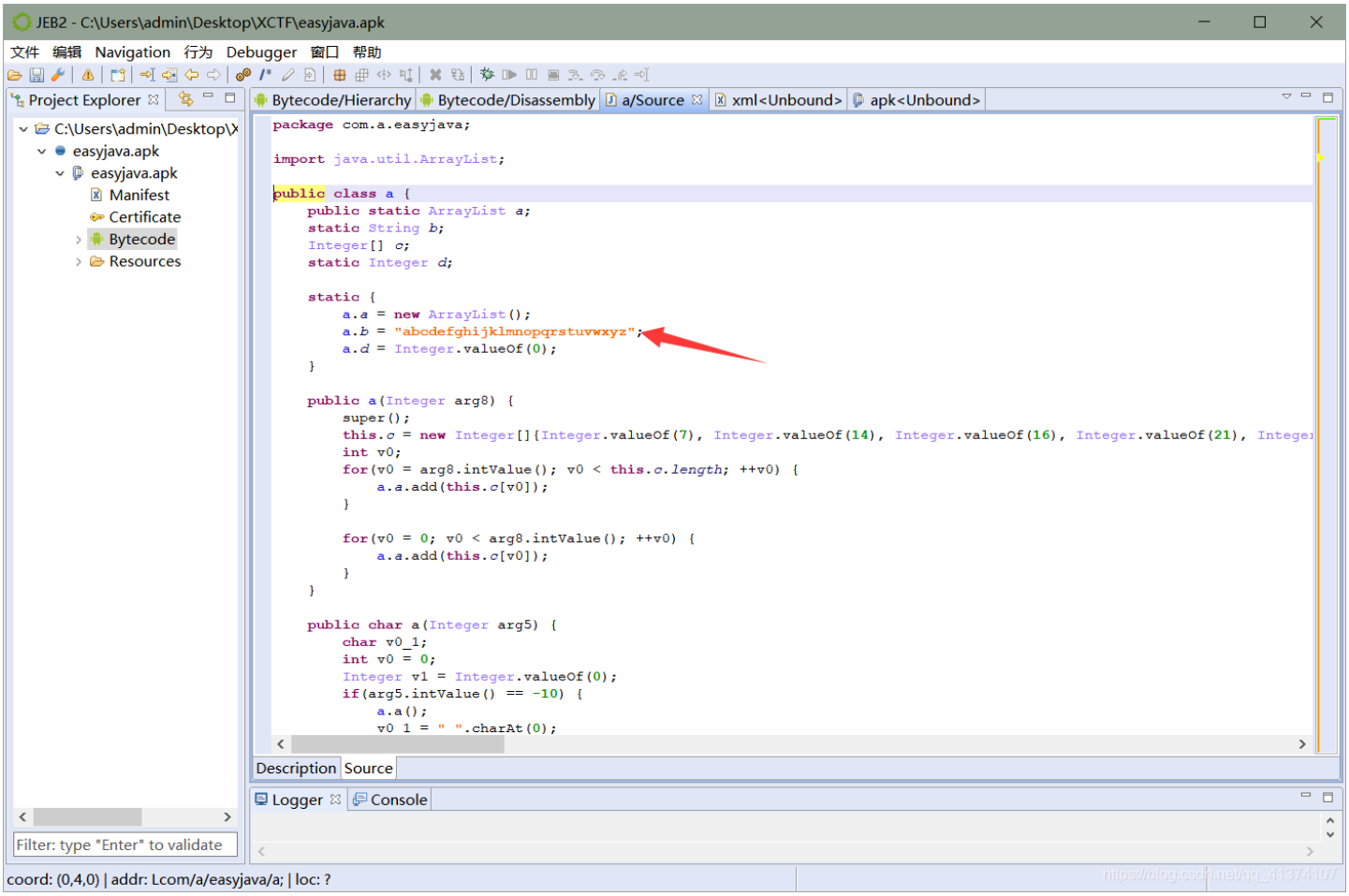
再来看b类的a函数，该函数首先获取传进来的字符在字符串b.b中的索引，然后得到在b类中定义的整形数组中与该该索引相等的在数组中的索引，然后调用b类的`a()`函数，该函数作用为将b类中数组与字符串左移一位，然后返回该数组索引。


JEB2 - C:\Users\admin\Desktop\XCTF\easyjava.apk
文件 编辑 Navigation 行为 Debugger 窗口 帮助
Project Explorer
C:\Users\admin\Desktop\
easyjava.apk
easyjava.apk
Manifest
Certificate
Bytecode
Resources
Filter: type "Enter" to validate
Bytecode/Hierarchy
Bytecode/Disassembly
b/Source
xml<Unbound>
apk<Unbound>
}
for(v0 = 0; v0 < arg9.intValue(); ++v0) {
b.a.add(this.c[v0]);
}
}
public Integer a(String arg5) {
int v0 = 0;
Integer v1 = Integer.valueOf(0);
if(b.b.contains(arg5.toLowerCase())) {
Integer v2 = Integer.valueOf(b.b.indexOf(arg5));
while(v0 < b.a.size() - 1) {
if(b.a.get(v0) == v2) {
v1 = Integer.valueOf(v0);
}
++v0;
}
}
else {
if(arg5.contains(" ")) {
v1 = Integer.valueOf(-10);
goto label_24;
}
v1 = Integer.valueOf(-1);
}
label_24:
b.a();
return v1;
}
public static void a() {
int v0 = b.a.get(0).intValue();
Description
Source
Logger
Console
coord: (0,4,0) | addr: Lcom/a/easyjava/b; | loc: ?

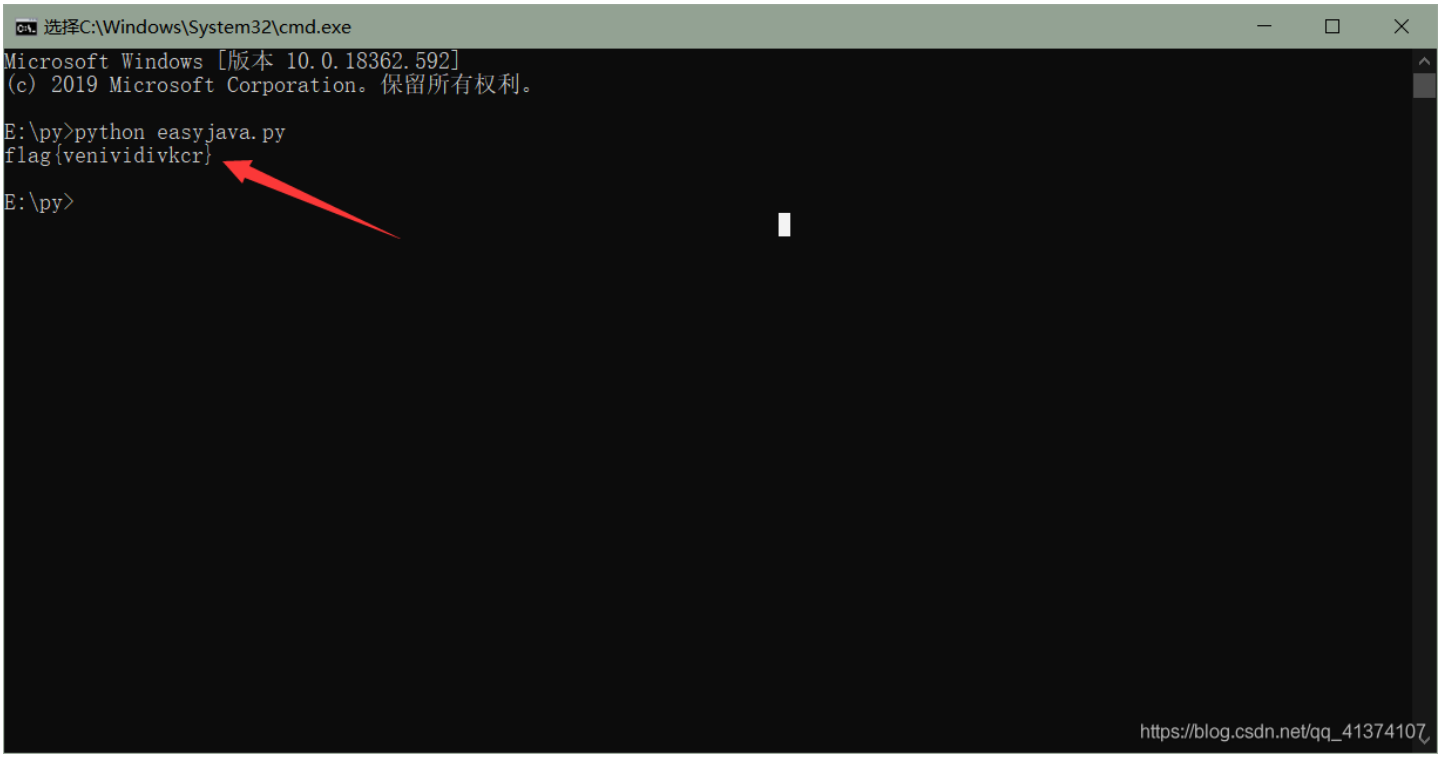


接着再来看a类中的`a(int)`函数，该函数首先获取与传进来的参数相等的数组中的值的索引，然后获取在字符串中索引为该数组索引的字符，最后返回该字符，当然，其中也调用`a()`函数，但是该函数要求等于25，所以该函数木有任何作用。


JEB2 - C:\Users\admin\Desktop\XCTF\easyjava.apk
文件 编辑 Navigation 行为 Debugger 窗口 帮助
Project Explorer
C:\Users\admin\Desktop\X
easyjava.apk
easyjava.apk
Manifest
Certificate
Bytecode
Resources
Filter: type "Enter" to validate
Bytecode/Hierarchy
Bytecode/Disassembly
a/Source
xml<Unbound>
apk<Unbound>
for(v0 = arg8.intValue(); v0 < this.c.length; ++v0) {
a.a.add(this.c[v0]);
}
for(v0 = 0; v0 < arg8.intValue(); ++v0) {
a.a.add(this.c[v0]);
}
}
public char a(Integer arg5) {
char v0_1;
int v0 = 0;
Integer v1 = Integer.valueOf(0);
if(arg5.intValue() == -10) {
a.a();
v0_1 = " ".charAt(0);
}
else {
while(v0 < a.a.size() - 1) {
if(a.a.get(v0) == arg5) {
v1 = Integer.valueOf(v0);
}
++v0;
}
a.a();
v0_1 = a.b.charAt(v1.intValue());
}
return v0_1;
}
public static void a() {
a.d = Integer.valueOf(a.d.intValue() + 1);
if(a.d.intValue() == 25) {
Description
Source
Logger
Console
coord: (0,4,0) | addr: Lcom/a/easyjava/a; | loc: ?



6、上面几步以及详细讲述了该apk的flag加密流程，要获取flag，逆向解密即可，解密脚本跑出结果如下：



三、附件

解密脚本：（要求python版本3.6或者与上）

```
cipherText = 'wigwrkaugala'

aArray = [21,4,24,25,20,5,15,9,17,6,13,3,18,12,10,19,0,22,2,11,23,1,8,7,14,16]
aString = 'abcdefghijklmnopqrstuvwxyz'

bArray = [17,23,7,22,1,16,6,9,21,0,15,5,10,18,2,24,4,11,3,14,19,12,20,13,8,25]
bString = 'abcdefghijklmnopqrstuvwxyz'

def changeBArrayandString():
 global bString
 global bArray
 chArray = bArray[0]
 chString = bString[0:1]
 for i in range(len(bArray) - 1):
  bArray[i] = bArray[i + 1]
 bArray[len(bArray) - 1] = chArray
 bString = bString[1:]
 bString += chString

def getBchar(ch):
 v2 = bArray[ch]
 arg = bString[v2]
 changeBArrayandString()
 return arg

def getAint(ch):
 global aString
 global aArray
 v1 = aString.index(ch)
 arg5 = aArray[v1]
 return arg5

print('flag{',end='')
for k in cipherText:
 v0 = getAint(k)
 print(getBchar(v0),end='')
print('}')
```