**[高危] 敏感数据泄露: 获取视频文件**

**1.缺陷位置:**

**YunChengAndroid.zip/app/src/main/java/Utils/utils.java: 157**

**解决方法:**

**可以尝试使用while(cur.moveToNext()){}替换。**

**[高危] 空指针解引用**

**1.缺陷位置:**

**YunChengAndroid.zip/app/src/main/java/Utils/clutter.java: 231**

**漏洞分析：**

当传入的参数tempFiles为空时，RandomAccessFile的实例ra将一直为null，在后面的ra.close中将产生null.close。

**解决方法:**

关闭时进行判断 。参考代码：

if(ra != null){

ra.close;

}

**2.缺陷位置:**

**YunChengAndroid.zip/app/src/main/java/Utils/clutter.java: 260**

**漏洞分析：**

当传入的参数partsPaths为空时，RandomAccessFile的实例ra将一直为null,在后面的ra.close中将产生null.close。

**解决方法:**

关闭时进行判断。参考代码：

if(ra != null){

ra.close;

}

**3.缺陷位置:**

**YunChengAndroid.zip/app/src/main/java/Utils/getGps.java: 24**

**漏洞分析：**

findActivity(context)的返回值可能为空。

**解决方法:**

使用前进行判断。参考代码：

if(utils.findActivity(context) != null){

locationManager = (LocationManager)

utils.findActivity(context).getSystemService(LOCATION\_SERVICE);

}

**[严重] 推测的空指针解引用问题**

**1.缺陷位置:**

**YunChengAndroid.zip/app/src/main/java/Utils/utils.java: 379**

**漏洞分析：**

使用If分解问题，导致str=null时运行代码null.equals(null).

**解决方法:**

方案一：依然使用if进行问题分解,增加判断条件，使用短路跳过equals方法。参考代码:

if ( is && str.equals(null)){

is= false;

}

方案二：取消使用if分解问题，使用if-else结构。参考代码：

if (str==null){

is= false;

}else if(str.equals(null))){

is= false;

}

**[严重] 文件描述符泄漏**

**1.缺陷位置:**

**YunChengAndroid.zip/app/src/main/java/Utils/CrashHandler.java: 201**

**漏洞分析：**

对fos调用flush()方法时，可能会出现异常，导致后面的关闭方法无法执行。

**解决方法:**

无法看到方法的全部内容，不知道是否对整个方法有异常处理或者进行了什么样的处理。一般的解决方法为对该行代码进行异常处理，将紧跟其后的fos.close();放在finally语句块中。参考代码：

Try{

fos.flush();

Catch(IOException e){

//你需要对数据丢失所进行的操作

}finally{

fos.close();

}

**2.缺陷位置:**

**YunChengAndroid.zip/app/src/main/java/Utils/clutter.java: 228**

**漏洞分析：**

写入文件时出现异常，导致紧跟其后的关闭流无法进行。

**解决方法:**

方案一：重构

问题出现在try的代码块太大。在这里需要缩小try的代码块，使得可以对异常精确处理。使用try-catch-finally代码块进行控制。将异常在catch中进行处理。将ra和fos的close()方法放到finally代码块中。

方案二：简单解决

如果不需要对异常进行处理，仅仅是需要对异常进行转换抛出，那么就把ra和fos的close()方法放在该文件中216行的try所对应的finally代码块中。

**3.缺陷位置:**

**YunChengAndroid.zip/app/src/main/java/Utils/clutter.java: 257**

**漏洞分析：**

问题同上。

**解决方法:**

解决方法同上。

**4.缺陷位置:**

**YunChengAndroid.zip/app/src/main/java/Utils/clutter.java: 302**

**漏洞分析：**

代码文件不全，无法进行分析。

**解决方法:**

无法得出解决方案。

**[建议改进] 忽略函数返回值**

**此处问题多出现在条件语句中直接使用了方法返回值，而没有进行有效性检查。一般的解决方法是对方法的返回值进行有效检查。**

**1.缺陷位置:**

**YunChengAndroid.zip/app/src/main/java/Utils/clutter.java: 277**

**漏洞分析：**

available()方法在网络编程中当网络不稳定时，可能会读取到0值。

**解决方法:**

使用一个局部变量读取该方法的返回值，进行有效性检查。然后进行操作。参考代码：

Int available = 0;

//轮询可能出现死循环，可以设置一个变量操作停止，比如轮询次数，然后设置读取不到需要进行的操作。因看不到整体代码，无法写出失败后需要的操作。

While(available == 0){

available = fis.available();

}

byte[] tmpBytes = new byte[fis.available()];

**2.缺陷位置:**

**YunChengAndroid.zip/app/src/main/java/Utils/utils.java: 147**

**解决方法:**

方案一：

if (cur.moveToFirst() == false){

方案二：

可以使用while()避开使用cur.moveToFirst();

while(cur.moveToNext()){

}

**3.缺陷位置:**

**YunChengAndroid.zip/app/src/main/java/Utils/utils.java: 250**

**解决方法**

在使用newStr = newStr + str.charAt(i);之前，将 str.charAt(i)赋值给一个char类型变量，进行有效性检查。

**4.缺陷位置:**

**YunChengAndroid.zip/app/src/main/java/Utils/utils.java: 382**

**解决方法:**

使用boolean类型获取，然后在进行使用。