某宝系 tb tm sgmain x-sign 分析 - unidbg

2021/9/7 6:09:19

编程Tag: import new com TM VM Sign public sgmain unidbg

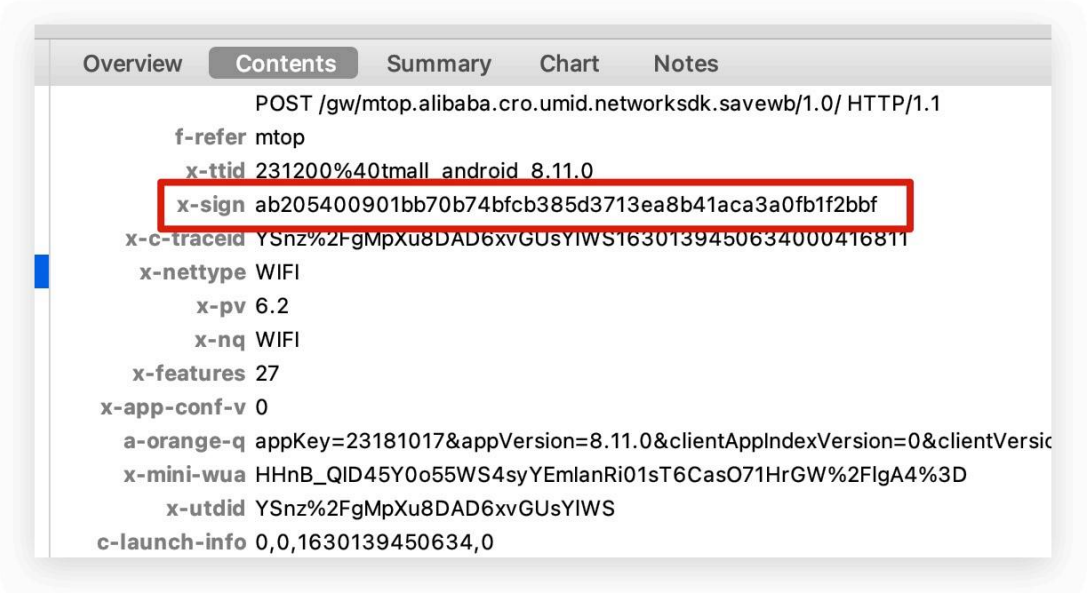
本文主要是介绍某宝系 tb tm sgmain x-sign 分析 - unidbg，对大家解决编程问题具有一定的参考价值，需要的程序猿们随着小编来一起学习吧!

仅供学习研究 。请勿用于非法用途，本人将不承担任何法律责任。

前言

apk 版本, 天猫 8.11.0, 本次主要说分析下如何使用 unidbg 跑通 x-sign，博主也是初学者，有啥问题可以加博主一起讨论哈，非常欢迎

charles 数据包分析



主要分析这个 x-sign，知道目标后，直接开始使用 unidbg

unidbg

具体的函数定位就不说了，各位看官应该都知道 JNICLibrary.doCommandNative 这个是 so 入口

```
1  package com.xiayu;
2
3  import com.github.unidbg.AndroidEmulator;
4  import com.github.unidbg.Emulator;
5  import com.github.unidbg.LibraryResolver;
6  import com.github.unidbg.file.FileResult;
7  import com.github.unidbg.file.IOResolver;
8  import com.github.unidbg.file.linux.AndroidFileIO;
9  import com.github.unidbg.linux.android.AndroidEmulatorBuilder;
10 import com.github.unidbg.linux.android.AndroidResolver;
11 import com.github.unidbg.linux.android.dvm.AbstractJni;
12 import com.github.unidbg.linux.android.dvm.DvmClass;
13 import com.github.unidbg.linux.android.dvm.DvmObject;
14 import com.github.unidbg.linux.android.dvm.VM;
```

```java
import com.github.unidbg.memory.Memory;
import com.github.unidbg.spi.SyscallHandler;
import com.github.unidbg.virtualmodule.android.AndroidModule;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.File;
import java.io.IOException;
import java.util.LinkedHashMap;
import java.util.Map;

public class TianMaoXSign1 extends AbstractJni implements IOResolver<AndroidFileIO>
{
    private final AndroidEmulator emulator;
    private final VM vm;
    private long slot;

    public String sgMain = "unidbg-android/src/test/resources/test_so/tianmao8110/libsgmainso-6.4.156.so";
    public String sgSecurityBody = "unidbg-android/src/test/resources/test_so/tianmao8110/libsgsecuritybodyso-6.4.90.so";
    public String sgAvMp = "unidbg-android/src/test/resources/test_so/tianmao8110/libsgavmpso-6.4.34.so";
    public String sgMisc = "unidbg-android/src/test/resources/test_so/tianmao8110/libsgmiscso-6.4.44.so";

    public File sgMainFile = new File(sgMain);
    public File sgSecurityBodyFile = new File(sgSecurityBody);
    public File sgAvMpFile = new File(sgAvMp);
    public File sgMiscFile = new File(sgMisc);

    public String dataAppPath = "/data/app/com.tmall.wireless-NsaOVgz2fomXJNoPTrbOwg==";
    public String packageName = "com.tmall.wireless";
    public String methodSign = "doCommandNative(I[Ljava/lang/Object;)Ljava/lang/Object;";
    public DvmClass JNICLibrary;
    public DvmObject<?> context;
    public DvmObject<?> ret;

    public String APK_INSTALL_PATH = dataAppPath + "/base.apk";
    public File APK_FILE = new File("/Users/admin/Desktop/android/file/tianmao-8.11.0.apk");
    private static LibraryResolver createLibraryResolver() {
        return new AndroidResolver(23);
    }

    private static AndroidEmulator createARMEmulator() {
        return AndroidEmulatorBuilder
                .for32Bit()
                .setRootDir(new File("appFile/tianmao-xsign1"))
                .setProcessName("com.tmall.wireless")
                .build();
    }

    public TianMaoXSign1() {
        emulator = createARMEmulator();

        Map<String, Integer> iNode = new LinkedHashMap<>();
        iNode.put("/data/system", 671745);
        iNode.put("/data/app", 327681);
        iNode.put("/sdcard/android", 294915);
        iNode.put("/data/user/0/com.tmall.wireless", 655781);
        iNode.put("/data/user/0/com.tmall.wireless/files", 655864);
        emulator.set("inode", iNode);
        emulator.set("uid", 10074);

        Memory memory = emulator.getMemory();
        memory.setLibraryResolver(createLibraryResolver());
        SyscallHandler<AndroidFileIO> handler = emulator.getSyscallHandler();
        handler.setVerbose(false);
        handler.addIOResolver(this);

        vm = emulator.createDalvikVM(APK_FILE);
        vm.setJni(this);
        vm.setVerbose(true);

        new AndroidModule(emulator, vm).register(memory);

        JNICLibrary = vm.resolveClass("com/taobao/wireless/security/adapter/JNICLibrary");
        context = vm.resolveClass("android/content/Context").newObject(null);
    }

    public static void main(String[] args) throws IOException {
        TianMaoXSign1 tm2 = new TianMaoXSign1();
        tm2.destroy();
    }

```

```
104         public void destroy() throws IOException {
                emulator.close();
            }

            @Override
            public FileResult<AndroidFileIO> resolve(Emulator<AndroidFileIO> emulator,
        String pathname, int oflags) {
                System.out.println("resolve.pathname: " + pathname);
                return null;
            }
        }
```

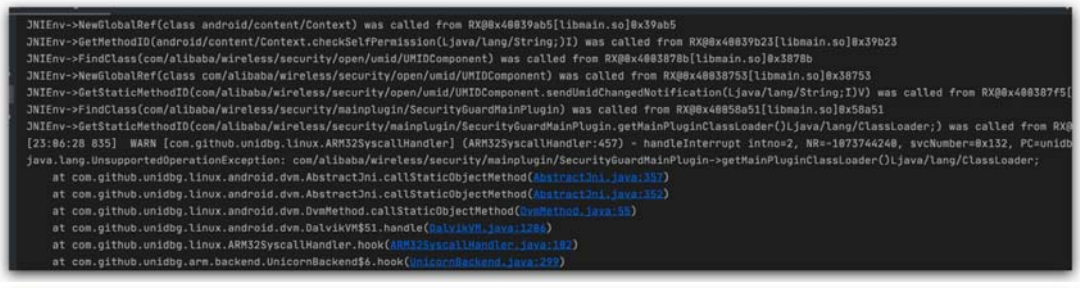
这里先把框架搭起来，`IOResolver<AndroidFileIO>` 这个接口类可以直接使用 `unidbg` 的虚拟文件系统，方便补文件。运行，如果没啥错误就说明一切正常，继续下一步

下面开始对各个 `so` 进行初始化，具体的初始化流程，可以使用 `frida hook` 查看整体流程，使用 `jnitrace` 不全，比较容易出问题

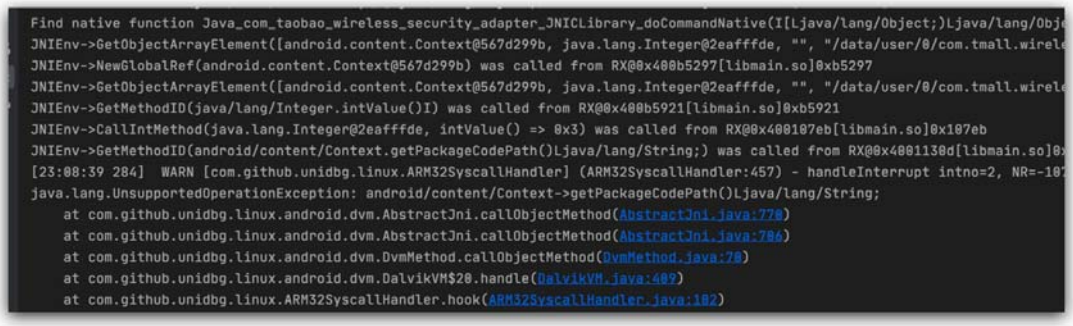
libsgmainso

```
1   public void initMain() {
2       DalvikModule dm = vm.loadLibrary(sgMainFile, true);
3       dm.callJNI_OnLoad(emulator);
4
5       ret = JNICLibrary.callStaticJniMethodObject(
6               emulator, methodSign, 10101,
7               new ArrayObject(
8                       context,
9                       DvmInteger.valueOf(vm, 3),
10                      new StringObject(vm, ""),
11                      new StringObject(vm, "/data/user/0/" + packageName +
12  "/app_SGLib"),
13                      new StringObject(vm, "")
14              ));
15      System.out.println("xiayu, initMain.ret-10101: " + ret.getValue().toString());
16
17      ret = JNICLibrary.callStaticJniMethodObject(
18              emulator, methodSign, 10102,
19              new ArrayObject(
20                      new StringObject(vm, "main"),
21                      new StringObject(vm, "6.5.156"),
22                      new StringObject(vm,
23  "/data/user/0/com.tmall.wireless/app_SGLib/app_1627957761/main/libsgmainso-
24  6.5.156.so")
                ));
        System.out.println("xiayu, initMain.ret-10102: " + ret.getValue().toString());
    }
```

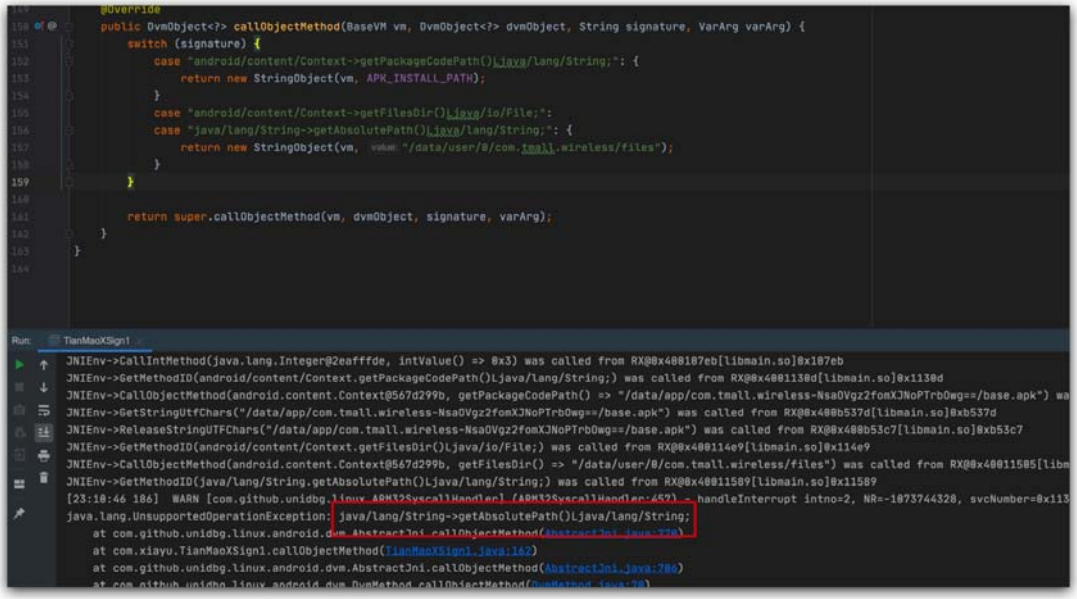
这里是初始化 `sgmain` 具体流程是 `frida hook` 的结果，写好后在 `main` 函数里调用一下，开始运行



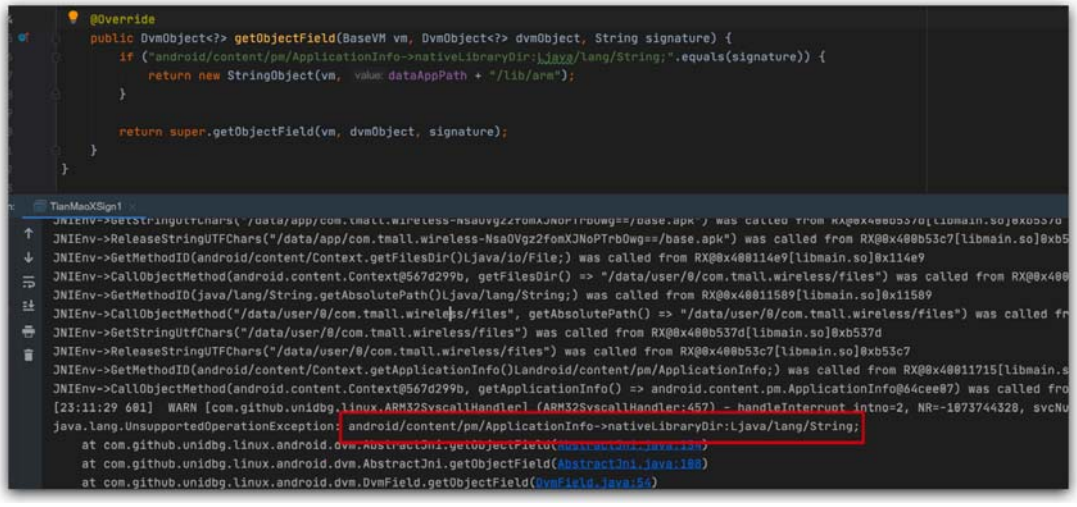
这里就开始报常见的环境错误了，我们给他加上



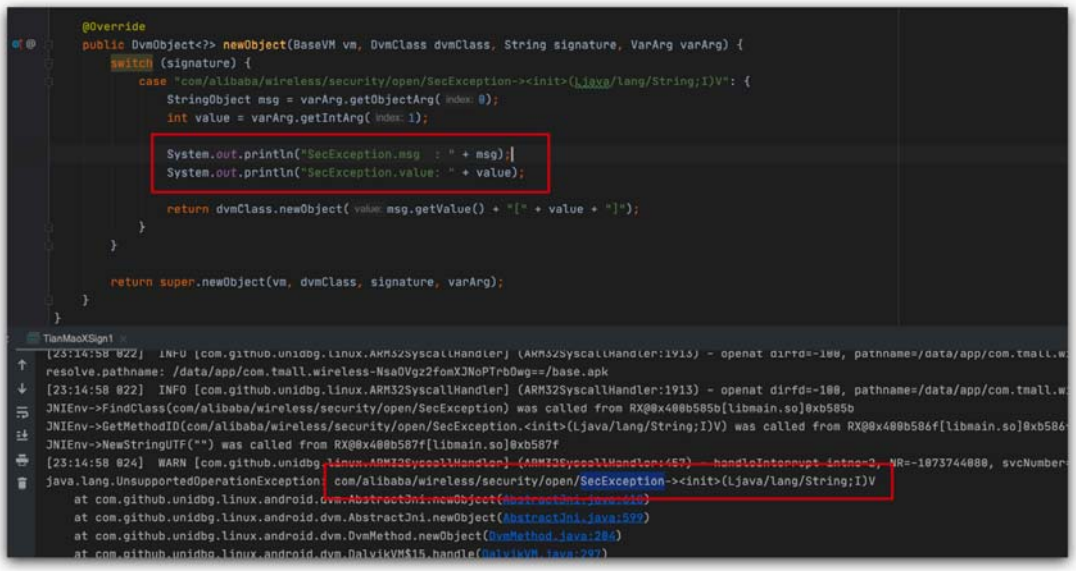
这里继续，返回 apk 的 base.apk 路径



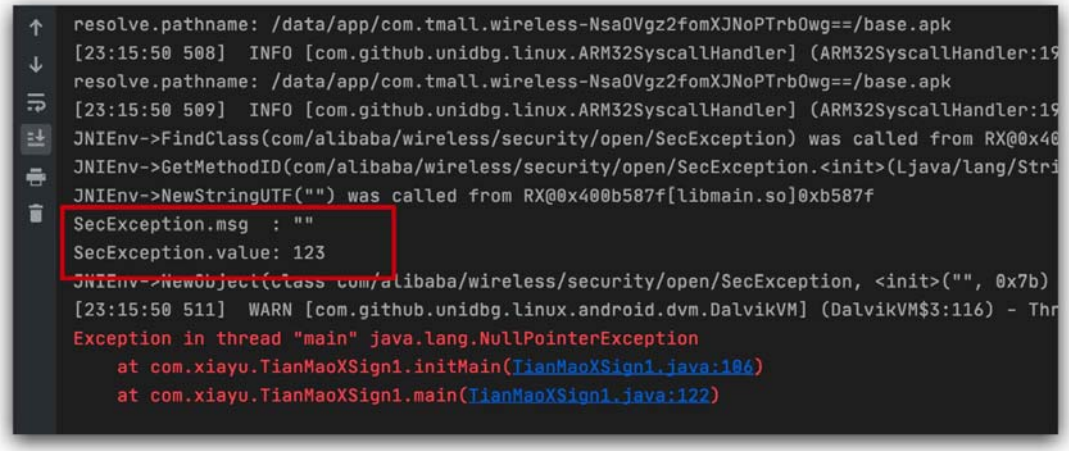
这里连续报的两个错误，可以统一处理，返回 files 文件夹路径



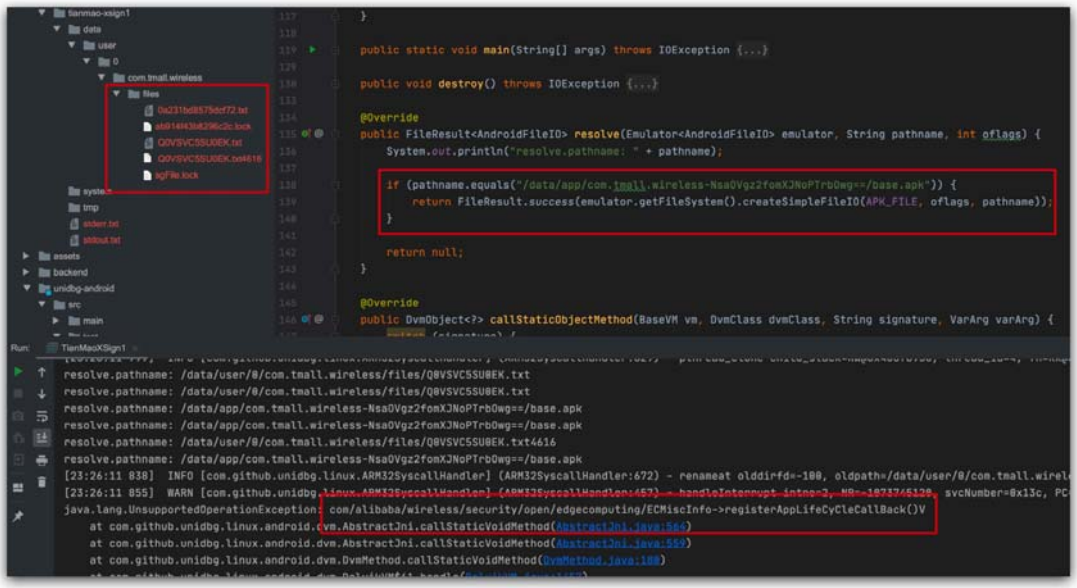
这里需要返回 apk 的 native lib/arm 路径



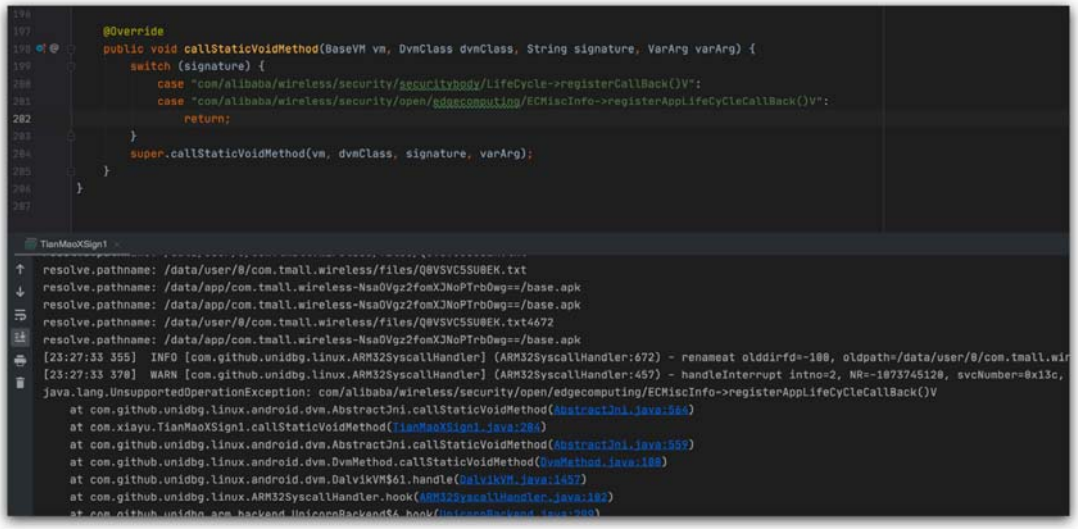
这个是 sgmain 抛异常的类，我们需要把 msg code 打印下，看看报了啥错误



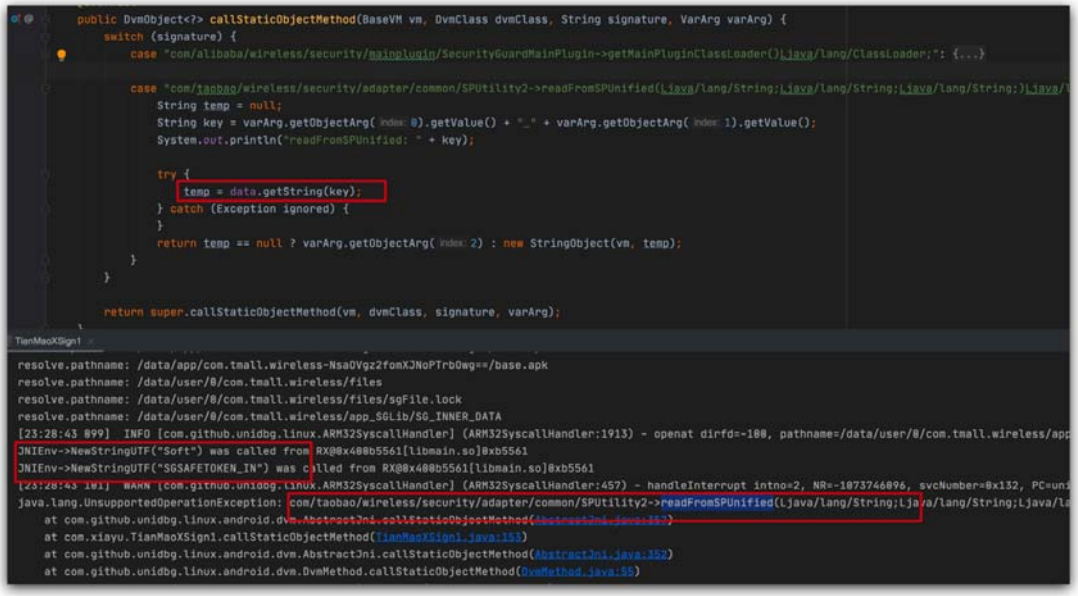
继续执行，果然报错了 code = 123。这里查找错误原因有两种，第一种是 百度搜索 阿里聚安全 123 是啥错误，第二种是自己查看 log 日志，猜测试错。过程还是比较坑的，我这里踩了很多坑，就直接说答案吧是缺少了文件。具体缺少啥文件，搜索 resolve.pathname: 这个字眼，不知道补哪些就能补的都补下

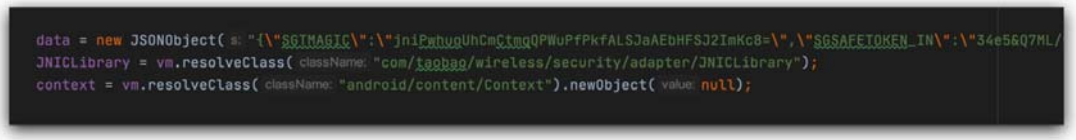


我这里是补了左侧的几个文件，跟 base.apk，文件就在手机里对应的目录下，pull 下来就行，运行后发现不再报 123 错误了



这里报了两个错误，可以一起补





中间又补了几个错误之后，这里是读取文件里的 `json` 内容，因为是固定的我就直接取出来了，这个补完之后再补一个常见的就 `OK` 了

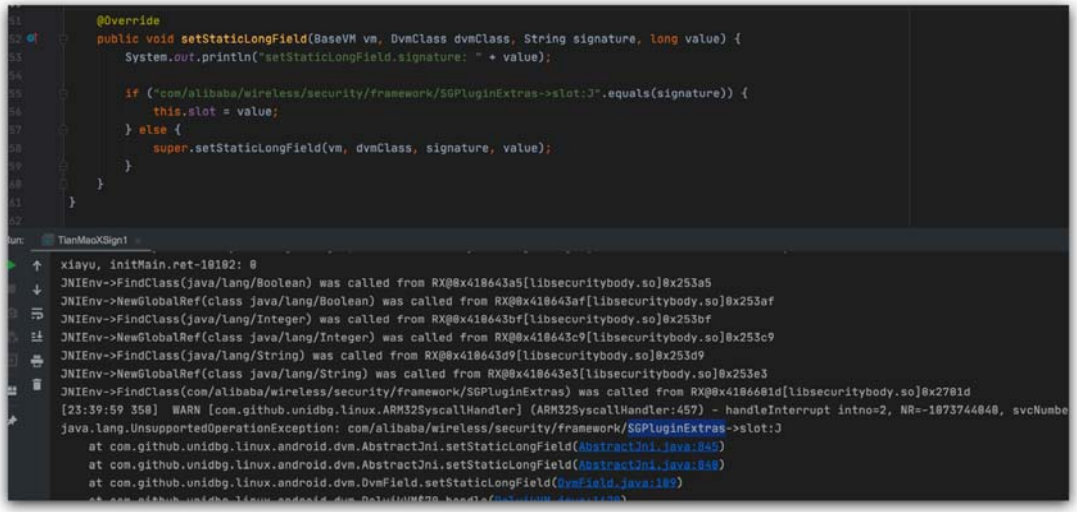
这里算是 `sgmain` 就补完了，全都正常返回了 `0`，这个结果我是 `frida call` 出来的结果

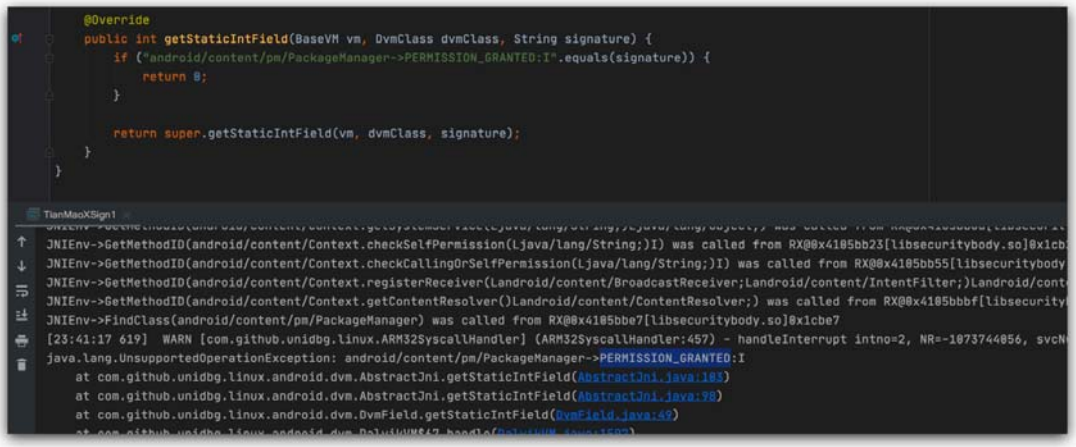
libsgsecuritybodyso

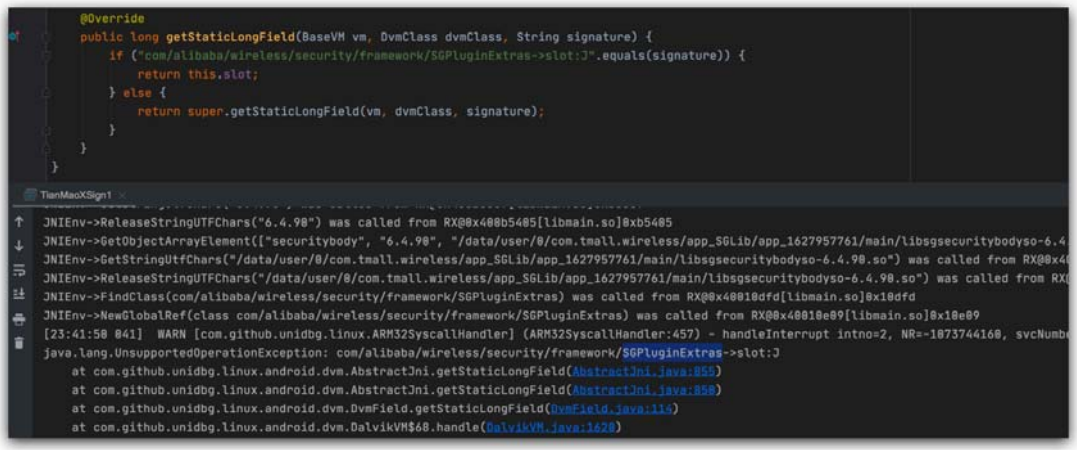
```
public void initSecurityBody() {
    DalvikModule securityBody = vm.loadLibrary(sgSecurityBodyFile, true);
    securityBody.callJNI_OnLoad(emulator);

    ret = JNICLibrary.callStaticJniMethodObject(
            emulator, methodSign, 10102,
            new ArrayObject(
                    new StringObject(vm, "securitybody"),
                    new StringObject(vm, "6.4.90"),
                    new StringObject(vm,
"/data/user/0/com.tmall.wireless/app_SGLib/app_1627957761/main/libsgsecuritybodyso-
6.4.90.so")
            ));
    System.out.println("xiayu, initSecurityBody.ret-10102: " +
ret.getValue().toString());
}
```

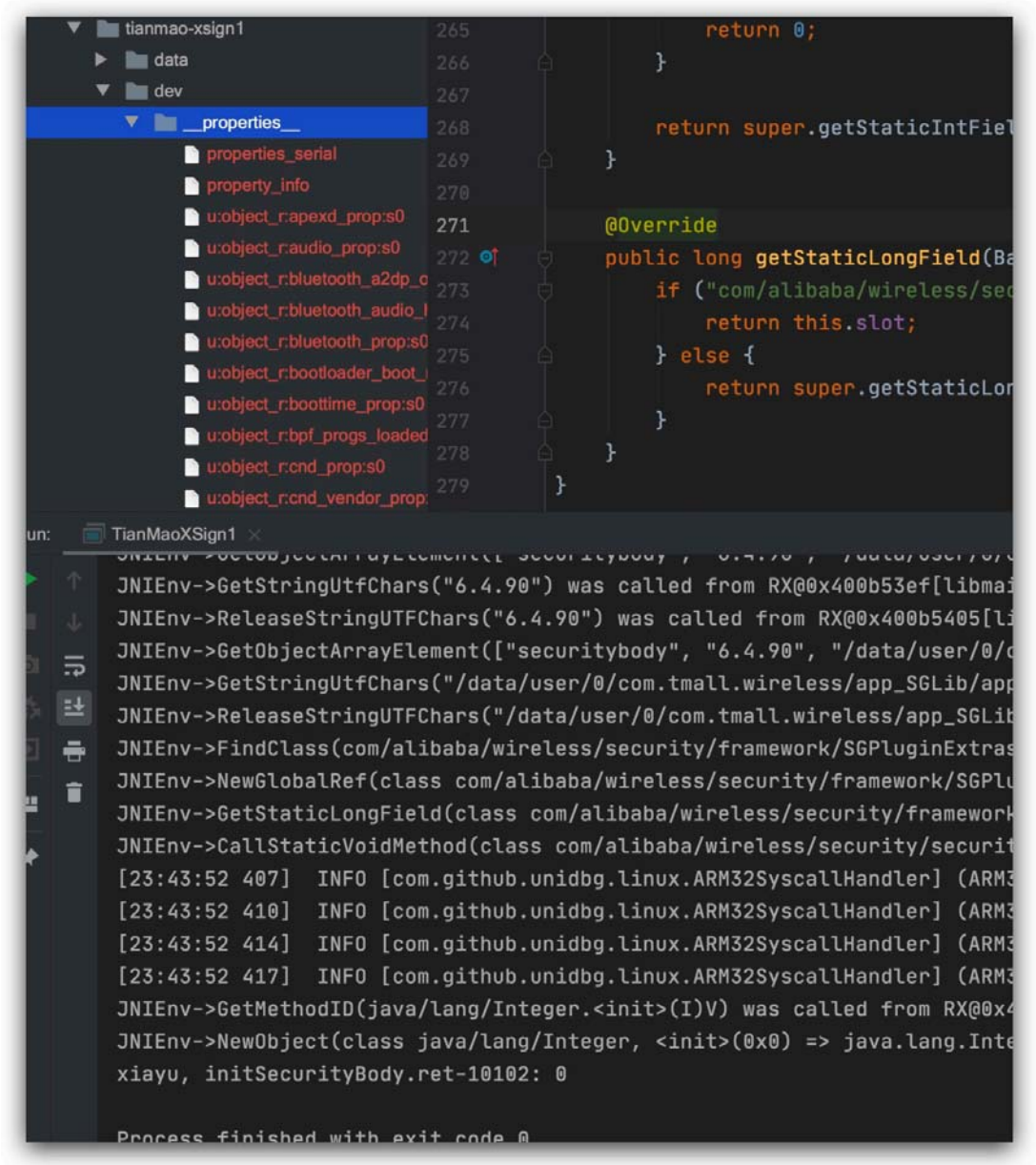
新增个函数，`main` 函数里调用







上图都是执行结果却啥补啥，就行



补到最后报了个指针错误，因为啥也不知道，还是使用相同的方法补文件，经过测试补 `dev/__properties__` 即可，继续运行，`OK` 了

libsgavmpso

```
public void initAvMp() {
    DalvikModule avMp = vm.loadLibrary(sgAvMpFile, true);
    avMp.callJNI_OnLoad(emulator);

    ret = JNICLibrary.callStaticJniMethodObject(
            emulator, methodSign, 10102,
            new ArrayObject(
                    new StringObject(vm, "avmp"),
                    new StringObject(vm, "6.4.34"),
                    new StringObject(vm,
"/data/user/0/com.tmall.wireless/app_SGLib/app_1627957761/main/libsgavmpso-
6.4.34.so")
            ));
    System.out.println("xiayu, initAvMp.ret-10102: " + ret.getValue().toString());
}
```

这个比较简单，跟上诉逻辑差不多，就不说了，缺啥补啥

get x-sign

```
public void getXSign() {
    Map<String, String> map = new HashMap<>();
    map.put("INPUT",
"&&&231817&1c9d79ea8dd4bc56fb7a7727a30366&1629886&mtop.tmall.inshopsearch.searchitems&1.
    DvmObject<?> ret = JNICLibrary.callStaticJniMethodObject(
            emulator, methodSign, 10401,
            new ArrayObject(
```

```
                    vm.resolveClass("java/util/HashMap").newObject(map),
                    new StringObject(vm, "23181017"),
                    DvmInteger.valueOf(vm, 7),
                    null,
                    DvmBoolean.valueOf(vm, true)
            ));

    System.out.println("xiayu, getXSign.ret-10401: " + ret.getValue().toString());
}
```

```
        case "java/util/HashMap->keySet()Ljava/util/Set;": {
            HashMap<?, ?> map = (HashMap<?, ?>) dvmObject.getValue();
            return vm.resolveClass( className: "java/util/Set").newObject(map.keySet());
        }
    }
```

```
JNIEnv->GetStaticLongField(class com/alibaba/wireless/security/framework/SGPluginExtras, slot => 0x4146e0ee) was called from RX@
[23:48:35 270]  INFO [com.github.unidbg.linux.ARM32SyscallHandler] (ARM32SyscallHandler:829) - pthread_clone child_stack=RW@0x41
JNIEnv->GetMethodID(java/lang/Integer.<init>(I)V) was called from RX@0x400b5aa7[libmain.so]0xb5aa7
JNIEnv->NewObject(class java/lang/Integer, <init>(0x0) => java.lang.Integer@5c0369c4) was called from RX@0x40011007[libmain.so]0x
xiayu, initAvMp.ret-10102: 0
Find native function Java_com_taobao_wireless_security_adapter_JNICLibrary_doCommandNative(I[Ljava/lang/Object;)Ljava/lang/Objec
JNIEnv->GetObjectArrayElement([java.util.HashMap@d70c109, "23181017", java.lang.Integer@17ed40e0, null, true], 0) was called fro
[23:48:35 271]  WARN [com.github.unidbg.linux.ARM32SyscallHandler] (ARM32SyscallHandler:457) - handleInterrupt intno=2, NR=-1073
java.lang.UnsupportedOperationException: java/util/HashMap->keySet()Ljava/util/Set;
    at com.github.unidbg.linux.android.dvm.AbstractJni.callObjectMethod(AbstractJni.java:770)
    at com.xiayu.TianMaoXSign1.callObjectMethod(TianMaoXSign1.java:229)
    at com.github.unidbg.linux.android.dvm.AbstractJni.callObjectMethod(AbstractJni.java:786)
```

```
        case "java/util/Set->toArray()[Ljava/lang/Object;": {
            Set<?> set = (Set<?>) dvmObject.getValue();
            Object[] array = set.toArray();
            DvmObject<?>[] objects = new DvmObject[array.length];
            for (int i = 0; i < array.length; i++) {
                if (array[i] instanceof String) {
                    objects[i] = new StringObject(vm, (String) array[i]);
                } else {
                    throw new IllegalStateException("array=" + array[i]);
                }
            }
            return new ArrayObject(objects);
```

```
TianMaoXSign1
JNIEnv->GetStringUtfChars("6.4.34") was called from RX@0x400b53ef[libmain.so]0xb53ef
JNIEnv->ReleaseStringUTFChars("6.4.34") was called from RX@0x400b5405[libmain.so]0xb5405
JNIEnv->GetObjectArrayElement(["avmp", "6.4.34", "/data/user/0/com.tmall.wireless/app_SGLib/app_1627957761/main/libsgavmpso-6.4.34.so"],
JNIEnv->GetStringUtfChars("/data/user/0/com.tmall.wireless/app_SGLib/app_1627957761/main/libsgavmpso-6.4.34.so") was called from RX@0x400
JNIEnv->ReleaseStringUTFChars("/data/user/0/com.tmall.wireless/app_SGLib/app_1627957761/main/libsgavmpso-6.4.34.so") was called from RX@0x
JNIEnv->GetStaticLongField(class com/alibaba/wireless/security/framework/SGPluginExtras, slot => 0x4146e0ee) was called from RX@0x40010e7
[23:49:28 303]  INFO [com.github.unidbg.linux.ARM32SyscallHandler] (ARM32SyscallHandler:829) - pthread_clone child_stack=RW@0x41599930, th
JNIEnv->GetMethodID(java/lang/Integer.<init>(I)V) was called from RX@0x400b5aa7[libmain.so]0xb5aa7
JNIEnv->NewObject(class java/lang/Integer, <init>(0x0) => java.lang.Integer@5c0369c4) was called from RX@0x40011007[libmain.so]0x11007
xiayu, initAvMp.ret-10102: 0
Find native function Java_com_taobao_wireless_security_adapter_JNICLibrary_doCommandNative(I[Ljava/lang/Object;)Ljava/lang/Object; => RX@
JNIEnv->GetObjectArrayElement([java.util.HashMap@d70c109, "23181017", java.lang.Integer@17ed40e0, null, true], 0) was called from RX@0x400
JNIEnv->CallObjectMethod(java.util.HashMap@d70c109, keySet() => java.util.Set@50675690) was called from RX@0x400b6e43[libmain.so]0xb6e43
[23:49:28 305]  WARN [com.github.unidbg.linux.ARM32SyscallHandler] (ARM32SyscallHandler:457) - handleInterrupt intno=2, NR=-1073744232, s
java.lang.UnsupportedOperationException: java/util/Set->toArray()[Ljava/lang/Object;
    at com.github.unidbg.linux.android.dvm.AbstractJni.callObjectMethod(AbstractJni.java:770)
    at com.xiayu.TianMaoXSign1.callObjectMethod(TianMaoXSign1.java:233)
    at com.github.unidbg.linux.android.dvm.AbstractJni.callObjectMethod(AbstractJni.java:786)
```

```
        case "java/util/HashMap->get(Ljava/lang/Object;)Ljava/lang/Object;": {
            HashMap<?, ?> map = (HashMap<?, ?>) dvmObject.getValue();
            Object key = varArg.getObjectArg( index: 0).getValue();
            Object obj = map.get(key);
            if (obj instanceof String) {
                return new StringObject(vm, (String) obj);
            } else {
                throw new IllegalStateException("array=" + obj);
            }
        }
    }
```

```
TianMaoXSign1
JNIEnv->CallObjectMethod(java.util.HashMap@d70c109, keySet() => java.util.Set@50675690) was called from RX@0x400b6e43[libmain.s
JNIEnv->CallObjectMethod(java.util.Set@50675690, toArray() => ["INPUT"]) was called from RX@0x400b6e5f[libmain.so]0xb6e5f
JNIEnv->GetArrayLength(["INPUT"] => 1) was called from RX@0x400b6e6f[libmain.so]0xb6e6f
JNIEnv->GetObjectArrayElement(["INPUT"], 0) was called from RX@0x400b6f0d[libmain.so]0xb6f0d
JNIEnv->GetStringUtfChars("INPUT") was called from RX@0x400b537d[libmain.so]0xb537d
JNIEnv->ReleaseStringUTFChars("INPUT") was called from RX@0x400b53c7[libmain.so]0xb53c7
[23:50:08 682]  WARN [com.github.unidbg.linux.ARM32SyscallHandler] (ARM32SyscallHandler:457) - handleInterrupt intno=2, NR=-107
java.lang.UnsupportedOperationException: java/util/HashMap->get(Ljava/lang/Object;)Ljava/lang/Object;
    at com.github.unidbg.linux.android.dvm.AbstractJni.callObjectMethod(AbstractJni.java:770)
    at com.xiayu.TianMaoXSign1.callObjectMethod(TianMaoXSign1.java:246)
    at com.github.unidbg.linux.android.dvm.AbstractJni.callObjectMethod(AbstractJni.java:786)
    at com.github.unidbg.linux.android.dvm.DvmMethod.callObjectMethod(DvmMethod.java:70)
    at com.github.unidbg.linux.android.dvm.DalvikVM$20.handle(DalvikVM.java:489)
```

```
resolve.pathname: /data/user/0/com.tmall.wireless/files
resolve.pathname: /data/user/0/com.tmall.wireless/files
JNIEnv->NewStringUTF("ab20540890c7675f4cbd71d4b0f2791f34ceb601500fe63966") was called from RX@0x400b5561[libmain.so]0xb5
xiayu, getXSign.ret-10401: ab20540890c7675f4cbd71d4b0f2791f34ceb601500fe63966
```

上图环境报错全部补完之后，就可以出结果了，经过测试，这个结果是可以使用的

这篇关于某宝系 tb tm sgmain x-sign 分析 - unidbg的文章就介绍到这儿，希望我们推荐的文章对大家有所帮助，也希望大家多多支持为之网！

原文链接: https://blog.csdn.net/qq_40000081/article/details/120147710

相关编程文章 更多>

- 别再手动处理数据了！FastGPT 这个新功能让你提前下班 2024-11-08
- 大厂算法入门详解 2024-11-05
- 朴素贪心算法入门详解 2024-11-05
- 深度优先遍历算法详解与实践教程 2024-11-05
- 初学者指南：轻松理解树形模型 2024-11-05
- 搜索算法入门指南：轻松掌握基础概念与实现方法 2024-11-05
- 算法入门指南：轻松掌握编程基础 2024-11-05
- 算法高级入门教程：从基础到进阶 2024-11-05
- 随机贪心算法入门教程 2024-11-05
- 贪心算法入门详解 2024-11-05