OpenHarmony

开放原子开源基金会 OpenHarmony开发者大会 2023

OpenHarmony系统开发常用权限问题分析及解决分享

陆道

诚迈科技高级技术专家

OpenHarmony

目录 Contents

01 简介

OpenHarmony

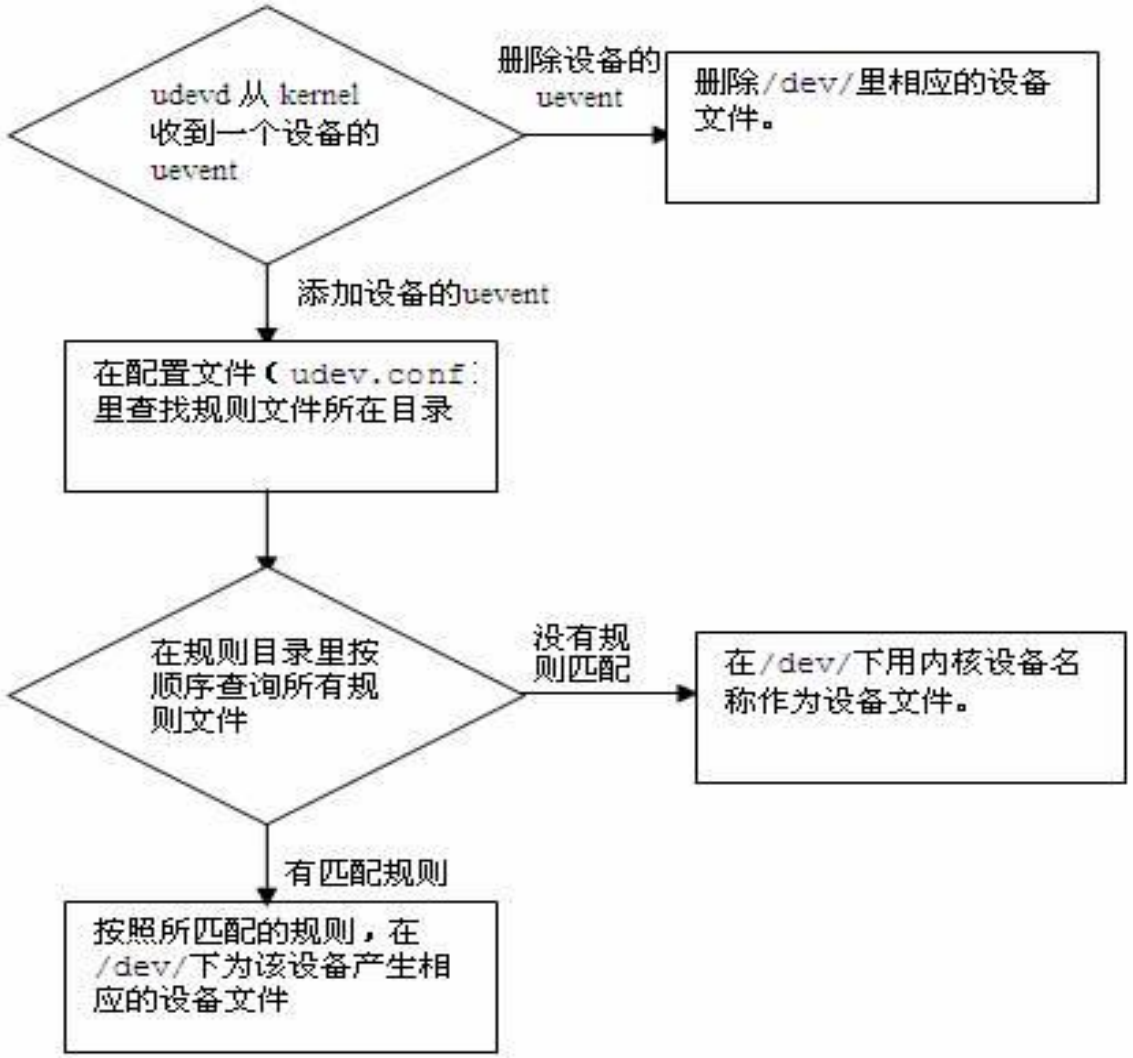
权限问题是我们开发者在开发过程中必须要面对的一个问题，也是作为新加入的开发者感觉到十分头疼的问题，其中包含的问题也非常多，我就我们项目开发过程中经常遇到的问题为例，为大家讲解我们分析问题的思路以及解决办法。

02 驱动节点访问权限--udev简介

1、udev为用户空间和设备驱动之间架设了一个桥梁，它主要的功能是管理/dev目录底下的设备节点。它同时也用来接替devfs及热插拔的功能，这意味着它要在添加/删除硬件时处理/dev目录以及所有用户空间的行为。

2、udev工作流程。

3、OpenHarmony的规则源文件是放在foundation/multimodalinput/input/patch/prebuild_eudev/rules.d/目录中。



02 驱动节点访问权限--USB转串口设备权限问题

OpenHarmony

串口问题	我们在开发工业控制设备驱动时，发现设备节点/dev/ttyUSB0的权限只有运行root权限和对应的group用户才能访问到，而我们自己写的NAPI服务却无法访问到这个节点，这时候我们就需要借助udev配置文件来生成对应权限的设备节点。首先第一步，我们需要找到设备节点对于生成的位置，然后添加对应的规则来改变设备节点的权限。
解决方案	foundation/multimodalinput/input/patch/prebuild_eudev/rules.d/40-usb_modeswitch.rules 在KERNEL==“ttyUSB*”， ATTRS{bNumConfigurations}==“*”，后面添加 KERNEL=="ttyUSB*", MODE="0777"

02 驱动节点访问权限——USB指纹模块权限问题

OpenHarmony

指纹问题

调试USB指纹模块时，我们发现上层通过NAPI接口打开USB设备出错，没有权限。

由于该设备是热插拔设备，所以我们分析此问题时分为两种情况：

1、先开机再插入USB指纹模块；

2、先插入USB指纹模块再开机；

02 驱动节点访问权限--USB指纹模块权限问题

OpenHarmony

解决方法

情况1：先开机再插入设备

1、通过Log文件，获取纹宁指纹模块的idVendor和idProduct。

2、开机以后，插入USB指纹模块，上层APP无法打开设备，应用层无法获取打开USB设备权限。

在foundation\multimodalinput\input\patch\prebuild_eudev\rules.d\60-libfprint-2.rules中添加：

```
SUBSYSTEM=="usb", ATTRS{idVendor}=="22bc", ATTRS{idProduct}=="2009", MODE="0777"
```

02 驱动节点访问权限--USB指纹模块权限问题

OpenHarmony

情况2：先插入USB指纹模块再开机

解决方法

插着USB指纹模块再开机，应用层无法获取打开USB设备的权限。

通过在系统里预置一个shell脚本，在脚本中修改USB设备的权限，开机以后由USB sevice去执行一次该脚本，解决USB设备权限的问题。

1）在/base/usb/usb_manager/bundle.json中添加//base/usb/usb_manager/services:usbchmod

2）在/base/usb/usb_manager/services/BUILD.gn中添加

```
ohos_prebuilt_executable("usbchmod")
{source = "usbchmod"
install_images = [ "system" ]
install_enable = true
part_name = "usb_manager “
}
```

3）在/base/usb/usb_manager/services/usb_service.cfg中添加usbchmod启动服务

4）在/base/usb/usb_manager/services目录下添加文件usbchmod

usbchmod文件内容如下：

```
#!/bin/bash
chmod -R 0777 /dev/bus/usb/*
```

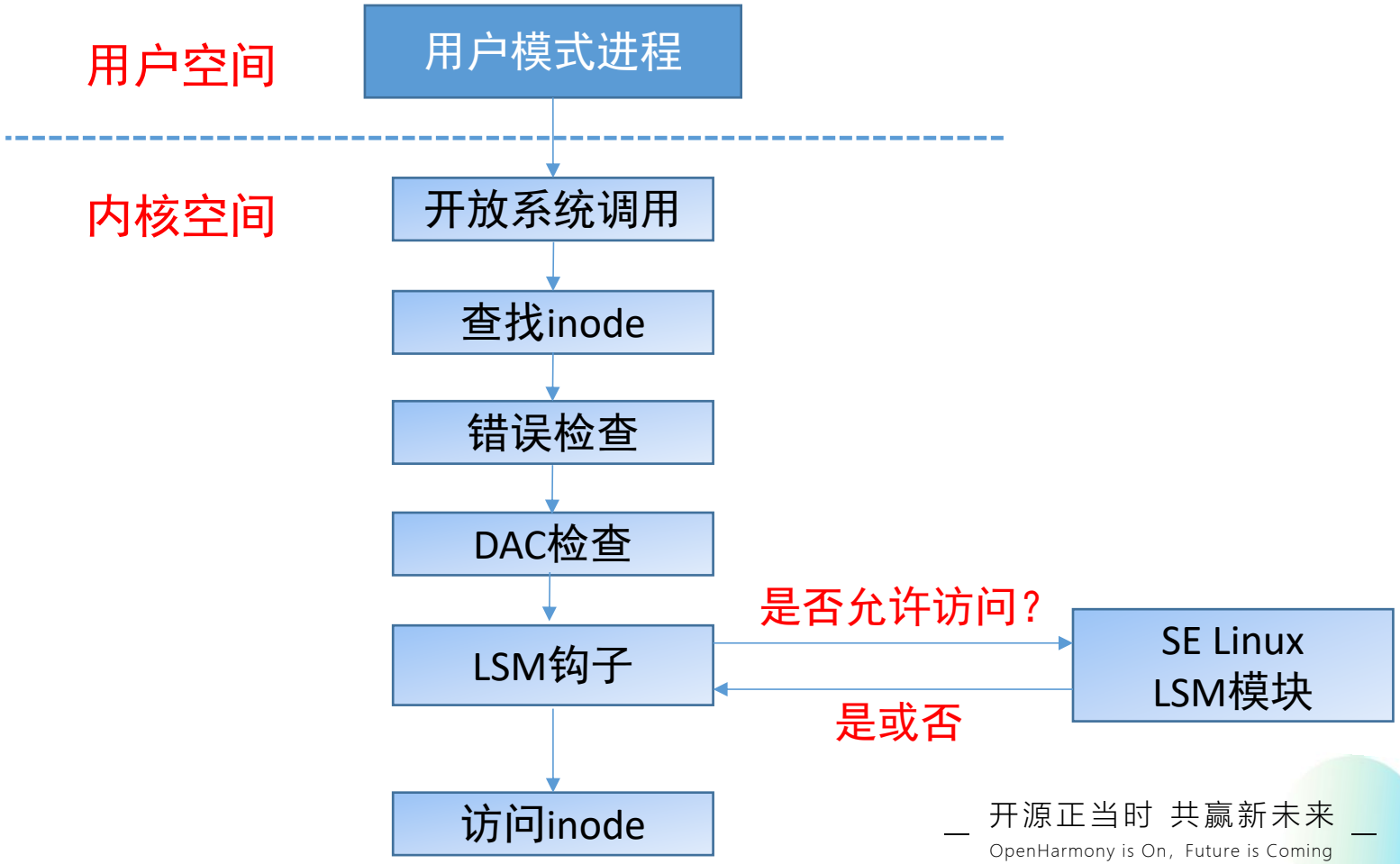
03 SELinux权限配置

OpenHarmony

1）SELinux介绍

安全增强型 Linux（Security-Enhanced Linux）简称 SELinux，它是一个 Linux 内核模块，也是 Linux 的一个安全子系统。

前面两部分介绍了MAC机制和Flask架构，最终SELinux的实现是依赖于Linux提供的Linux Security Module框架简称为LSM。其实LSM的名字并不是特别准确，因为他并不是Linux模块，而是一些列的hook，同样也不提供任何的安全机制。LSM的的重要目标是提供对Linux接入控制模块的支持。



03 SELinux权限配置

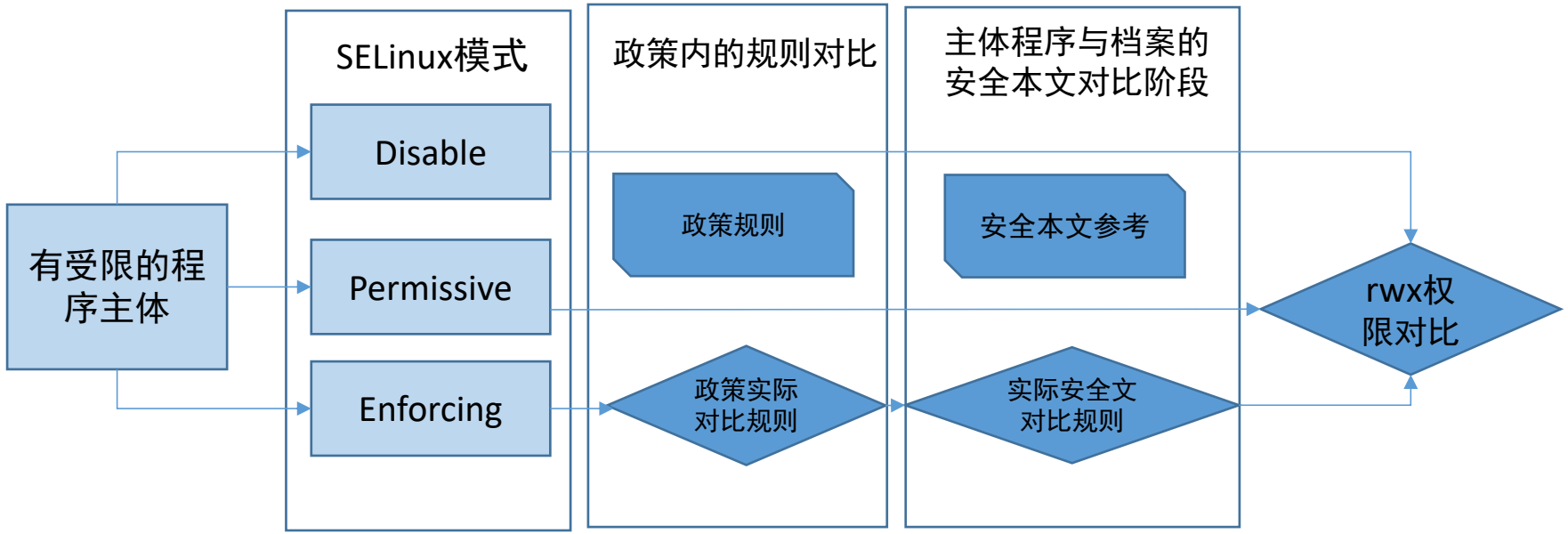
OpenHarmony

2）SELinux工作模式以及关闭方法

将模式设置成Permissive模式，这样我们可以根据kernel日志中的报错日志来添加对应的规则，直至最后将模式设置成Enforcing模式

base/security/selinux/selinux.gni

```
declare_args() {
  selinux_enforce = false
}
```



03 SELinux权限配置

3）自建SA服务添加SELiunx权限步骤

1、新建SA服务

```
"services" : [{
        ...
        "permission" :
[
"ohos.permission.DISTRIBUTED_DATA
SYNC",
"ohos.permission.DISTRIBUTED_SOFT
BUS_CENTER",
"ohos.permission.GET_BUNDLE_INFO_
PRIVILEGED"
        ],...
    }
```

2、添加用户支持

```
#base/startup/init/services/etc/g
roup
industrialbus:x:3050:
#base/startup/init/services/etc/p
asswd
industrialbus:x:3050:3050:::/bin/
false
```

03 SELinux权限配置

3）自建SA服务添加SELinux权限步骤

3、SA添加SELinux权限

```
#base/startup/init/services/etc/group
industrialbus:x:3050:
#base/startup/init/services/etc/passwd
industrialbus:x:3050:3050:::/bin/false
```

4、添加industrialbus进程的SA访问权限

```
base/security/security_selinux/sepolicy/base/te
# 添加文件industrialbus.te文件
allow industrialbus
sa_param_watcher:samgr_class { get };
```

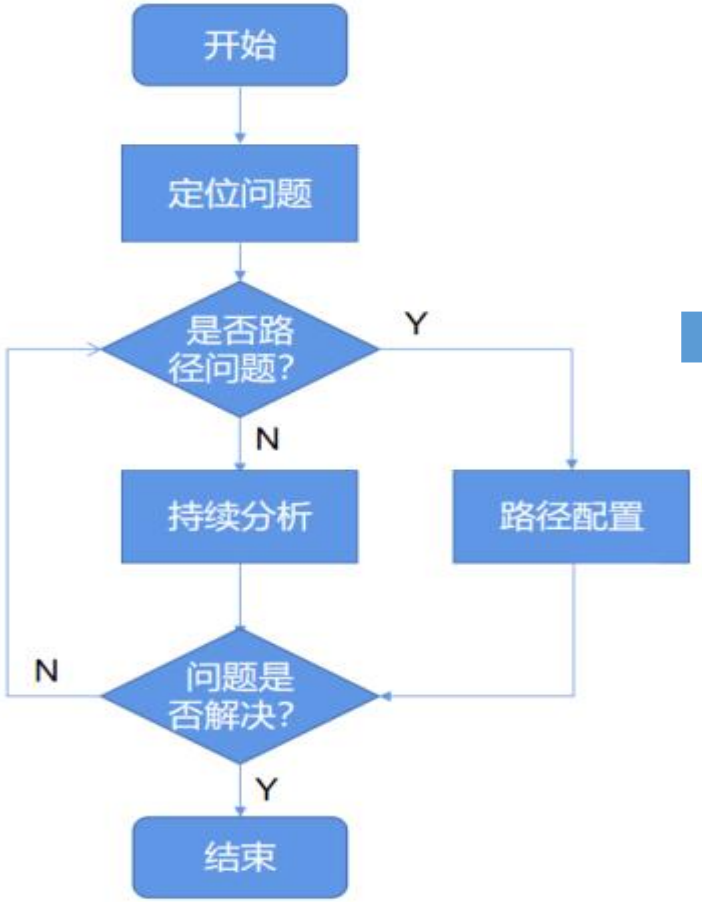
5、在shell里访问industrialbus的SA

```
base/security/security_selinux/sepolicy/base/te/sh.te
allow sh
sa_industrialbus_service:samgr_class
{ get };
llow sh
sa_industrialbus_service:samgr_class
{ get };
allow industrialbus
sa_param_watcher:samgr_class { get };
```

03 沙箱机制权限配置

1）沙箱机制介绍

沙箱机制，增加目录可见性数据访问防线，减少了应用数据和用户隐私信息泄露，建立了更加严格安全的应用沙盒隔离。启用应用沙箱之后，应用命名空间内无法再访问物理路径下数据目录的访问方式，而是只能通过context接口来访问应用的数据目录。




```
],
"com.ohos.camera" : [{
    "sandbox-switch": "OFF",
    "sandbox-root" : "/mnt/sandbox/<PackageName>",
    "mount-paths" : [],
    "symbol-links" : []
}],
"com.example.Browser" : [{
    "sandbox-switch": "OFF",
    "sandbox-root" : "/mnt/sandbox/<PackageName>",
    "mount-paths" : [],
    "symbol-links" : []
}],
```

03 沙箱机制权限配置

2）人脸识别文件无法生成问题

问题一：

调试人脸识别模块，需要将数据模型AntiColor.pkg预置在系统中，将AntiColor.pkg预置在/vendor/etc/model下面，调用时，无法读取文件，打开失败。

解决办法：

配置沙箱base/startup/appspawn/appdata-sandbox.json

"src-path" : "/vendor/etc/model",

"sandbox-path" : "/vendor/etc/model",

```
        }, {
            "src-path" : "/vendor/etc/model",
            "sandbox-path" : "/vendor/etc/model",
            "sandbox-flags" : [ "bind", "rec" ],
            "check-action-status": "false"
```

03 沙箱机制权限配置

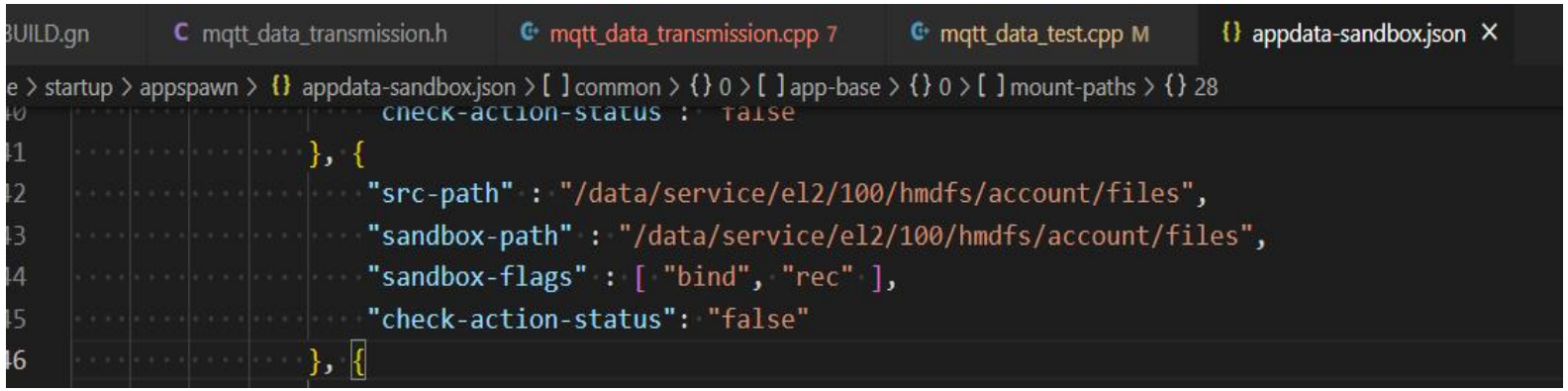
2）人脸识别文件无法生成问题

问题二：

三方应用通过NAPI接口调用人脸识别算法对两种图片进行比较，无法访问/data/service/el2/100/hmdfs/account/files/Pictures/Screenshots中的图片

解决办法：

配置沙箱base/startup/appspawn/appdata-sandbox.json



03 沙箱机制权限配置

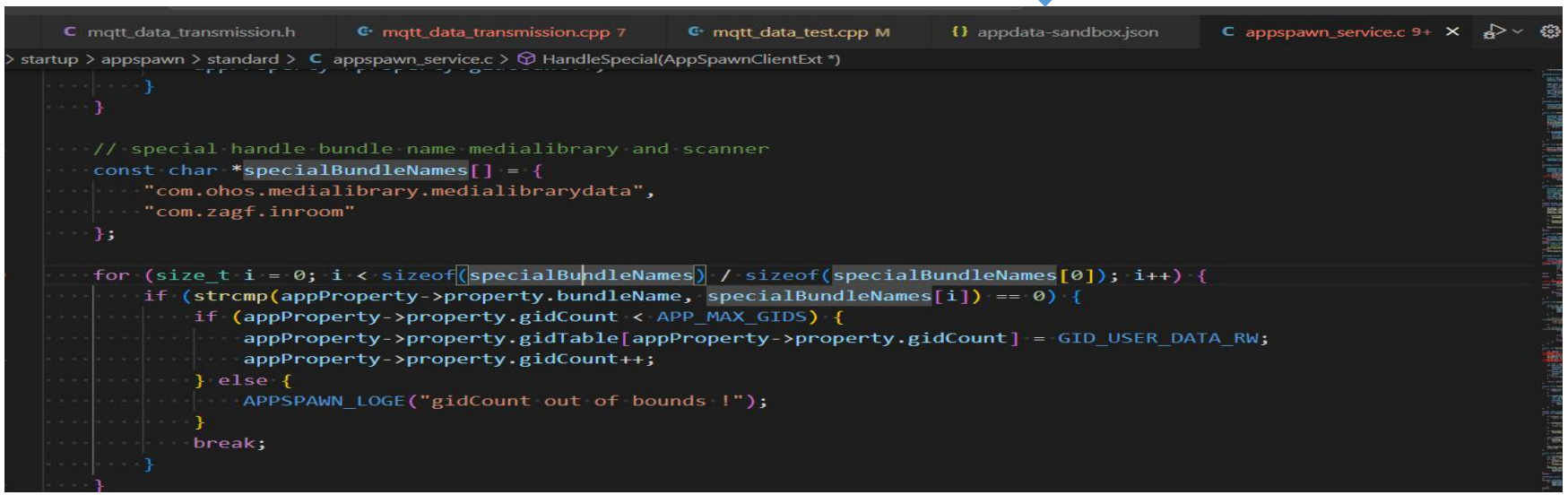
2）人脸识别文件无法生成问题

问题三：

三方应用无法访问多目录文件

解决办法：

将三方应用的包名加入medialibrary权限组，与medialibrary权限保持一致，保证三方应用可以访问/data/service/el2/100/hmdfs/account/files/Pictures子目录：配置base/startup/appspawn/standard/appspawn_service.c添加三方应用白名单



OpenHarmony

THANK YOU

长按识别二维码 关注官方公众号

【官网网址】www.openharmony.cn