# **autojs开机自启**

牙叔教程 简单易懂

第一步: 编写用于测试的开机运行脚本

main.js

"ui";

ui.layout(

   <vertical>

       <button textSize="30sp">牙叔教程 开机自启</button>

   </vertical>

);

第二步: 打包这个脚本, 打包单文件或者项目都可以,

第三步: 添加启动类

这里我们使用的是mt管理器,

首先我们看看test.dex里面写的什么, 查看dex的工具这里用的是 jadx-gui

package com;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.Intent;

import com.stardust.autojs.inrt.SplashActivity;

/\* loaded from: C:\Users\ADMINI~1\AppData\Local\Temp\jadx-4849598729232909477.dex \*/

public class AutoBootActivity extends BroadcastReceiver {

   @Override // android.content.BroadcastReceiver

   public void onReceive(Context context, Intent intent) {

       if (intent.getAction().equals("android.intent.action.BOOT\_COMPLETED")) {

           Intent intent2 = new Intent(context, SplashActivity.class);

           intent2.setFlags(268435456);

           context.startActivity(intent2);

       }

   }

}

这个代码是继承了广播接收器, 如果是开机广播, 就启动intent2, intent2是SplashActivity, 也就是一般app的启动页

接下来开始用mt管理器合并这个开机类;

安卓用的是smali文件, 上面的类对应的smali是

.class public Lcom/AutoBootActivity;

.super Landroid/content/BroadcastReceiver;

.source "AutoBootActivity.java"

# direct methods

.method public constructor <init>()V

   .registers 1

   .prologue

   .line 7

   invoke-direct {p0}, Landroid/content/BroadcastReceiver;-><init>()V

   return-void

.end method

# virtual methods

.method public onReceive(Landroid/content/Context;Landroid/content/Intent;)V

   .registers 6

   .param p1, "context"  # Landroid/content/Context;

   .param p2, "intent"  # Landroid/content/Intent;

   .prologue

   .line 10

   invoke-virtual {p2}, Landroid/content/Intent;->getAction()Ljava/lang/String;

   move-result-object v1

   const-string v2, "android.intent.action.BOOT\_COMPLETED"

   invoke-virtual {v1, v2}, Ljava/lang/String;->equals(Ljava/lang/Object;)Z

   move-result v1

   if-eqz v1, :cond\_1b

   .line 11

   new-instance v0, Landroid/content/Intent;

   const-class v1, Lcom/stardust/autojs/inrt/SplashActivity;

   invoke-direct {v0, p1, v1}, Landroid/content/Intent;-><init>(Landroid/content/Context;Ljava/lang/Class;)V

   .line 12

   .local v0, "intent2":Landroid/content/Intent;

   const/high16 v1, 0x10000000

   invoke-virtual {v0, v1}, Landroid/content/Intent;->setFlags(I)Landroid/content/Intent;

   .line 13

   invoke-virtual {p1, v0}, Landroid/content/Context;->startActivity(Landroid/content/Intent;)V

   .line 15

   .end local v0  # "intent2":Landroid/content/Intent;

   :cond\_1b

   return-void

.end method

# **用mt管理器添加smali的步骤**

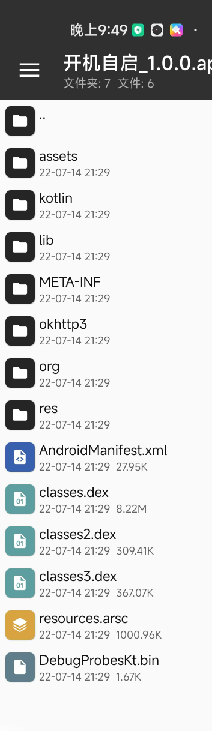
第一步: 用mt管理器的安装包提取功能, 提取我们刚才打包好的app

mt的安卓包提取选项: 打开mt, 点击左上角更多菜单, 就可以看到安装包提取选项

提取的apk所在路径:

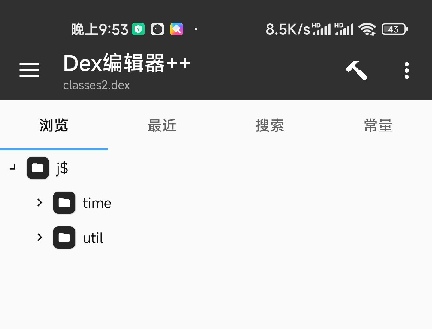
文件已保存到 /storage/emulated/0/MT2/apks/开机自启\_1.0.0.apk

查看app的文件



发现他有3个dex文件, 我们挑一个文件比较小的dex, classes2.dex文件是最小的, 309k, 就它了;

点他, 然后选择 dex编辑器++



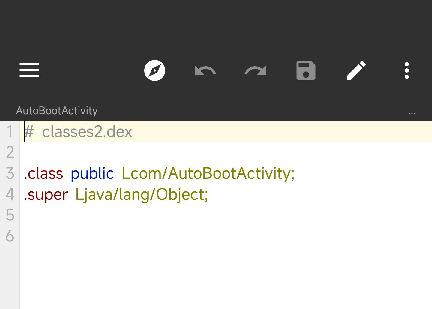
长按那个j$, 点击添加,

* 包名修改为 com
* 类名修改为 AutoBootActivity

点击确定



再点开autobootactivity文件



把里面的全删了, 然后把上面的smali文件内容黏贴进去



点击保存, 再按返回键, 保存并退出, 确定.

然后就修改完啦, 接下来测试一下是不是真的能开机自启

卸载原来的app, 然后安装我们修改后的app,

我先把开机密码删掉, 然后打开一次app, 给app自启动权限, 再重启手机;

额, 开机后等了一分钟, 没有成功,

我再把后台弹出界面也给他, 再次重启手机,

额, 开机后等了一分钟, 还是没有成功,

省电策略, 改成无限制, 悬浮窗也给他,

突然想起来, 清单文件还没加, 这个是

静态注册广播, 监听开机广播的

<receiver android:name="com.AutoBootActivity">

   <intent-filter>

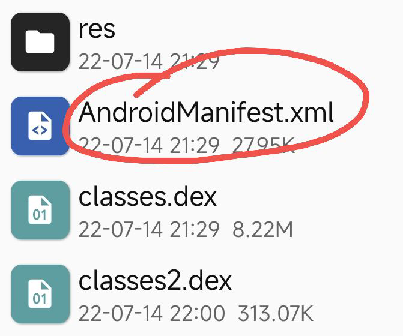
       <action android:name="android.intent.action.BOOT\_COMPLETED" />

       <category android:name="android.intent.category.LAUNCHER" />

   </intent-filter>

</receiver>

mt管理器查看修改后的app, 然后点他, 反编译



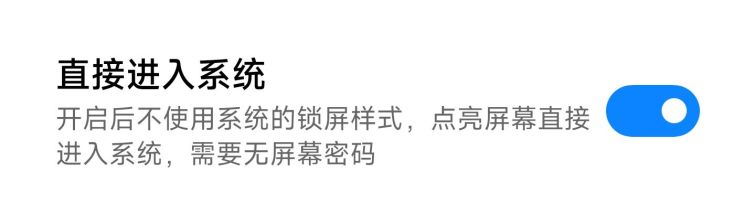
搜receiver标签, 把上面的静态注册广播代码,

放到<receiver>标签同级别即可, 然后保存, 确定;

卸载原来的app, 安装修改后的app

打开一次app, 授予他自启动, 和 后台弹出界面权限,

去开发者选项, 勾选 直接进入系统,



再次重启手机, 开机后等了一分钟, 还是没反应;

这次再给一些权限,

* 悬浮窗
* 省电策略无限制

等了48秒, 启动成功