**android安全问题之程序锁**

原理：

启动服务扫描运行中的task栈，取出**task**中的topActivity，然后进行判断，如果是你想锁定的应用，然后弹出一个需要输入密码的activity对原程序进行覆盖，当验证通过才让用户使用应用。

代码如下：

启动类MainActivity（也可使用广播作为入口）：

package com.home.testlock;

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

public class MainActivity extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

startService(new Intent(this, LockService.class));

}

}

程序锁的后台服务类LockService：

package com.home.testlock;

import java.util.Timer;

import android.app.Notification;

import android.app.Service;

import android.content.Intent;

import android.os.IBinder;

/\*\*

\* 程序锁的后台服务类

\*

\*/

public class LockService extends Service {

private Timer timer;// 定时器

public static final int FOREGROUND\_ID = 0;

@Override

public IBinder onBind(Intent arg0) {

return null;

}

@Override

public void onCreate() {

super.onCreate();

startForeground(FOREGROUND\_ID, new Notification());

}

@Override

public int onStartCommand(Intent intent, int flags, int startId) {

startTimer();

return super.onStartCommand(intent, flags, startId);

}

@Override

public void onDestroy() {

stopForeground(true);

timer.cancel();

timer.purge();

timer = null;

super.onDestroy();

}

/\*\*

\* 启动定时器任务

\*/

private void startTimer() {

if (timer == null) {

timer = new Timer();

LockTask lockTask = new LockTask(this);

timer.schedule(lockTask, 0L, 1000L);

}

}

}

程序锁的任务类LockTask:

package com.home.testlock;

import java.util.TimerTask;

import android.app.ActivityManager;

import android.content.ComponentName;

import android.content.Context;

import android.content.Intent;

/\*\*

\* 程序锁的任务类

\*

\*

\*/

public class LockTask extends TimerTask {

private Context mContext;

// 这是笔者自己手机上的测试应用

private String testPackageName = "com.lovo.sport";

private String testClassName = "com.lovo.ui.RegistActivity";

private ActivityManager am;

public LockTask(Context context) {

mContext = context;

am = (ActivityManager) mContext.getSystemService("activity");

}

@Override

public void run() {

ComponentName cn = am.getRunningTasks(1).get(0).topActivity;

String packageName = cn.getPackageName();

String className = cn.getClassName();

if (SharedUtil.getIntence().getIsFinished(mContext)) {

return;

}

// 如果想对整个程序锁定判断包名即可

if (testPackageName.equals(packageName)

&& testClassName.equals(className)) {

Intent intent = new Intent();

intent.setClassName("com.home.testlock",

"com.home.testlock.PWDActivity");

intent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);

mContext.startActivity(intent);

}

}

}

PWDActivity：

package com.home.testlock;

import android.app.Activity;

import android.os.Bundle;

import android.view.KeyEvent;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

public class PWDActivity extends Activity implements OnClickListener {

private Button sureBtn;

private EditText pwdText;

private static final String PASSWORD = "1234";

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.pwd);

initWidget();

}

private void initWidget() {

sureBtn = (Button) findViewById(R.id.pwd\_btn\_sure);

pwdText = (EditText) findViewById(R.id.pwd\_et);

sureBtn.setOnClickListener(this);

}

@Override

public void onClick(View v) {

if (v == sureBtn) {

String pwd = pwdText.getText().toString();

if (PASSWORD.equals(pwd)) {

SharedUtil.getIntence().saveIsFinished(this, true);

finish();

}

}

}

@Override

public boolean onKeyDown(int keyCode, KeyEvent event) {

if (KeyEvent.KEYCODE\_BACK == keyCode) {

return true;// 屏蔽返回键

}

return super.onKeyDown(keyCode, event);

}

}

SharedUtil：

package com.home.testlock;

import android.content.Context;

import android.content.SharedPreferences;

/\*\*

\* 处理SharedPreferences工具类

\*

\* @author 孤云

\*

\*/

public class SharedUtil {

private static SharedUtil shared;

/\*\*

\* 使用单例模式获取SharedUtil实例

\*

\* @return

\*/

public static synchronized SharedUtil getIntence() {

if (shared == null) {

shared = new SharedUtil();

}

return shared;

}

/\*\*

\* 获取是否验证过密码

\*

\* @param context

\* @return

\*/

public boolean getIsFinished(Context context) {

SharedPreferences sp = context.getSharedPreferences("data",

Context.MODE\_PRIVATE);

boolean isFinished = sp.getBoolean("isFinished", false);

return isFinished;

}

/\*\*

\* 保存验证逻辑

\*

\* @param context

\* @param isFinished

\*/

public void saveIsFinished(Context context, boolean isFinished) {

SharedPreferences sp = context.getSharedPreferences("data",

Context.MODE\_PRIVATE);

sp.edit().putBoolean("isFinished", isFinished).commit();

}

}

布局文件省略。

权限：

另外：如果不想某个本程序的某个activity显示在task中，对其activity加如下属性：

android:excludeFromRecents="true"