

Android权限控制培训问题解析

Last edited by caoquanli 1 month ago

Android权限控制培训问题解析

1.自定义权限可以申请为危险权限吗？ 申请后能成功实现并弹窗吗？

自定义权限可以申请危险权限，并实现弹窗。

例子： 首先我们在APP A中定义了危险权限

```
<activity android:name=".TesdActivity"
    android:permission="com.demo.sunchao.openactivity">
    <intent-filter>
        <category android:name="android.intent.category.DEFAULT"/>
        <action android:name="com.demo.sunchao.openactivity"/>
    </intent-filter>
</activity>
```

如果想要打开这个Activity，则需要APP B中的AndroidManifest.xml去使用这个权限。

```
<uses-permission android:name="com.demo.sunchao.openactivity"/>
```

然后需要在代码中去请求获取这个权限，如果不请求，应用直接crash，并出现安全异常

```
public class MainActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button button = (Button) findViewById(R.id.bt);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                intent1 = new Intent("com.demo.sunchao.openactivity");
                if (ContextCompat.checkSelfPermission(MainActivity.this,
                    "com.demo.sunchao.openactivity") != PackageManager.PERMISSION_GRANTED)
                {
                    requestPermissions(new String[]{"com.demo.sunchao.openactivity"},1);
                } else {
                    startActivity(intent1);
                }
            }
        });
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNu
        switch (requestCode){
            case 1 :
                if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED
                    startActivity(intent1);
                } else {
                    Toast.makeText(this,"You denied the permission",Toast.LENGTH_SHORT).show();
                }
                break;
            default:
        }
    }
}
```

这样当我们启动这个需要权限的Activcity的时候，会弹出如下窗口：





要允许**DemoTest**
执行未知操作吗？

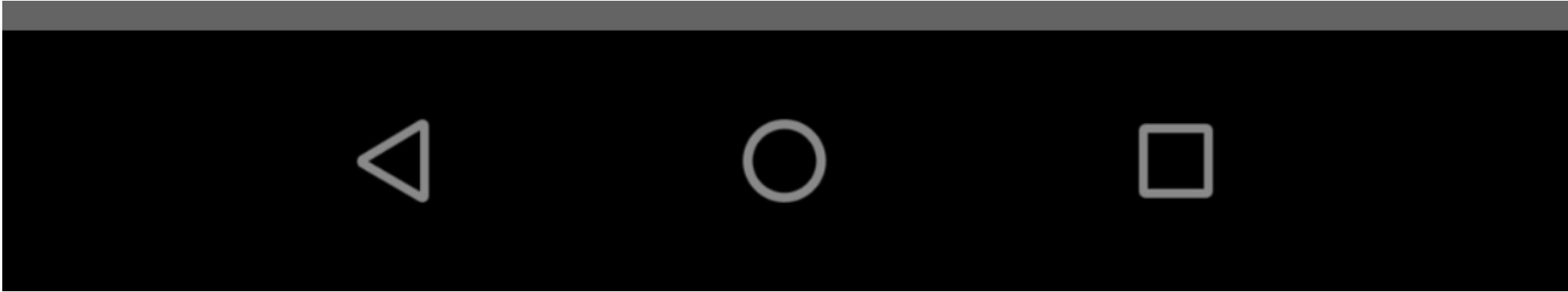


不再询问

拒绝

允许





2.如何区分 Runtime Permission ， Request Permission，install permissions类型的权限?

举例说明： 在应用中的AndroidManifest.xml使用了如下权限：

```
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.BROADCAST_STICKY" />
```

我们安装完成后， dumpsys 命令去dumpsys这个包的信息的时候会出现下面的信息，

```
requested permissions:
  android.permission.WRITE_SETTINGS
  android.permission.SYSTEM_ALERT_WINDOW
  android.permission.WAKE_LOCK
  android.permission.INTERNET
  android.permission.READ_PHONE_STATE
  android.permission.READ_EXTERNAL_STORAGE
  android.permission.WRITE_EXTERNAL_STORAGE
  android.permission.BROADCAST_STICKY
install permissions:
  android.permission.WRITE_SETTINGS: granted=true
  android.permission.SYSTEM_ALERT_WINDOW: granted=true
  android.permission.INTERNET: granted=true
  android.permission.BROADCAST_STICKY: granted=true
  android.permission.WAKE_LOCK: granted=true
User 0: ceDataInode=523677 installed=true hidden=false suspended=false stopped=false notLaunched=false enabled=0 instant=false virtual=false
  gids=[3003]
  runtime permissions:
    android.permission.READ_EXTERNAL_STORAGE: granted=false, flags=[ USER_SET GRANTED_BY_DEFAULT ]
    android.permission.WRITE_EXTERNAL_STORAGE: granted=false, flags=[ USER_SET GRANTED_BY_DEFAULT ]
```

对上面这些信息解析：

requested permissions:(请求权限)： 这个是我们需要去请求的权限，比如一些危险权限，特殊权限，普通权限，可以理解为在AndroidMainfest.xml中使用 <uses-permission>标签权限。

install permissions（安装时权限）：是指在安装app的时候，赋予app的权限。normal和signature级别(包括SignatureOrSystem)的权限都是安装时权限。不会给用户提示界面，系统自动决定权限的赋予或拒绝， 权限信息会被写入到data/system/packages.xml。

runtime permissions（运行时权限）：是指在app运行过程中，赋予app的权限，这个过程中,会显示明显的权限授予界面，让用户决定是否授予权限。权限信息会被写入到/data/system/user/0/runtime-permissions.xml。

3.系统APP和特权APP需要危险权限时会弹窗吗？

因为系统APP分为特权APP和普通系统APP。所以特权APP不会弹窗，直接使用。普通系统APP会弹窗。