Instituto Infnet

TESTE DE PERFORMANCE - TP1

MAGNO VALDETARO DE OLIVEIRA
E-MAIL: mvaldetaro@gmail.com
MATRÍCULA: 10403782775

RIO DE JANEIRO
JULHO - 2017

MAGNO VALDETARO DE OLIVEIRA

TESTE DE PERFORMANCE - TP1

Trabalho apresentado ao Professor
Flávio Almada França
da disciplina Segurança, Monetização e
Publicação de Aplicativos Android
da turma ADS-LV Turma 1 ,
Turno Noite
do curso de Análise e Desenvolvimento
de Sistemas

Instituto Infnet
Rio de Janeiro - 29 de Julho de 2017

SUMÁRIO

1 - INTRODUÇÃO

Neste teste de performance respondo questoes sobre segurança no ambiente android.

2 - DESENVOLVIMENTO

1 - Quais são os três principais elementos da Arquitetura Android? Descreva cada um deles.

Os principais elementos da arquiterura Android são, Hardware do dispositivo, Sistema operacional Android e Android Application Runtime.

Hardware do dispositivo: o Android é executado em uma gama de hardwares aproveitando alguns recursos específicos de segurança destes hardwares.

Sistema operacional Android: o Android é desenvolvido com base no kernel do Linux, grande partes dos recursos do dispositivo, como funcionaldiade das câmera, GPS, Bluetooth... são acessados através do SO.

Android Application Runtime: aplicações android em sua maioria utilizam a linguagem JAVA e são executados no ART.

2 – Comente as fontes principais de aplicativos para o Android.

Os aplicativos pré instalados são aqueles que fornecem as funcionalidades básicas do aparelho e podem ser acessados por outro aplicativos, que por sua vez podem fazer parte da plataforma Android, como o app de chamadas, ou podem ser desenvolvidos pelos fabricantes de dispositivos e adicionados à plataforma.

Aplicativos de terceiros podem ser encontrados em lojas como Google Play, que oferece milhares de aplicações.

3 – O que são API's sensíveis ao custo? Descreva cada uma delas.

São APIs que podem gerar custo ao usuário ou a rede. Essas APis são: Telefonia, SMS / MMS, Rede / Dados, Faturamento na aplicação, Acesso NFC.

Acesso ao cartão SIM: o acesso de baixo nível ao cartão SIM não está disponível para aplicativos de terceiros, ficando a cargo do SO lidar com todas as comunicações com o cartão.

Informações pessoais: o android possui APIs que fornecem acesso a dados do usuário. É normal, os dispositivos Android acumularem dados do usuário em aplicativos de terceiros instalados pelos usuários. Os aplicativos que optarem por

compartilhar essas informações podem usar as verificações de permissões do Android para proteger os dados.

Dispositivos de entrada de dados sensíveis: permitem que os aplicativos interajam com o ambiente ao redor, como câmera, microfone ou GPS. Para um aplicativo de terceiros possa acessar esses dispositivos, primeiro deve ser explicitamente fornecido acesso pelo usuário através do uso das permissões do Android.

Metadados do dispositivo: o Android busca restringir o acesso a dados que não são intrinsecamente sensíveis, mas podem revelar indiretamente características, preferências do usuário, assim como ele usa um dispositivo. Por padrão, os aplicativos não têm acesso aos metadados do dispositivo. Caso um aplicativo solicite acesso no momento da instalação, o instalador perguntará ao usuário se o aplicativo pode acessar a informação. Caso o usuário não conceda o acesso, a instalação é interrompida.

Certificados de Autoridade: o Android inclui um diversos Certificações de Autoridade de Sistema instaladas, que são confiáveis em todo o sistema.

Assinatura do Aplicativo: permite aos desenvolvedores identificar o autor do aplicativo e atualizar seu aplicativo sem criar interfaces e permissões complicadas.

Verificação de Aplicação: alerta o usuário ao tentar instalar um aplicativo que possa ser prejudicial, se um aplicativo é especialmente suspeito, ele pode bloquear a instalação.

Gestão de Direitos Digitais: permite aos aplicativos gerenciar conteúdos protegidos por direitos de acordo com as restrições de licença associadas ao conteúdo. A estrutura DRM suporta muitos esquemas DRM, quais destes sistemas um dispositivo suporta são definidos pelo fabricante do dispositivo.

4 – Suponha que você está desenvolvendo um aplicativo e deseja acessar as informações do cartão SIM do aparelho. Como é possível realizar essa operação?

Acessar informações do cartão SIM não está disponível para aplicativos de terceiros.

5 – O que é o sistema de permissões da plataforma Android?

O sistema de permissões do Android faz com que APPs que necessitem acesso a dados não produzidos por eles mesmo (dentro do sandbox dedicado ao app) ou a funcionalidades não disponíveis nestes apps, definam permissões para que esse consumo seja possível.

6 – Descreva pelo menos 5 permissões normais e as funções de cada uma delas.

NFC	Permite aplicações executar operações de I/O utilizando NFC
BLUETOOTH	Permite aplicações a conectar e parear dispositivos via bluetooth
SET_WALLPAPER	Permite que a aplicação a defina um papel de parede
VIBRATE	Permite acessar a função “vibrar”
SET_TIME_ZONE	Permite definir o fuso horário

7 – Descreva pelo menos 5 permissões perigosas e as funções de cada uma delas.

READ_CALENDAR	Permite que um aplicativo leia os dados do calendário do usuário.
CAMERA	Permite acessar todos os recursos da câmera do dispositivo.
GET_ACCOUNTS	Permite o acesso à lista de contas no Serviço de Contas.
RECORD_AUDIO	Permite que um aplicativo grave um áudio
READ_EXTERNAL_STORAGE	Permite que um aplicativo seja lido a partir de um armazenamento externo

8 – Pesquise quais as outras categorias de permissões além das normais e perigosas

Permissões URI

9 – Qual a função do manifest.xml?

A função do manifest.xml é apresenta informações essenciais sobre o aplicativo ao sistema Android, necessárias para o sistema antes que ele possa executar o código da aplicação.

10 – Avalie o manifest.xml abaixo:

```
<?xml version="1.0" encoding="utf-8"?>
  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.thedigitalsean.examples"
    android:versionCode="1"
    android:versionName="1.0">
  <application android:icon="@drawable/icon"
android:label="@string/app_name">
      <activity android:name=".GetLocation"
                android:label="@string/app_name">
          <intent-filter>
              <action android:name="android.intent.action.MAIN" />
              <category android:name="android.intent.category.LAUNCHER" />
          </intent-filter>
      </activity>
  </application>
  <uses-permission
android:name="android.permisssion.ACCESS_FINE_LOCATION"></uses-permission>
  <uses-permission
android:name="android.permisssion.ACCESS_COARSE_LOCATION"></uses-permission
>
  <uses-permission
android:name="android.permisssion.INTERNET"></uses-permission>
  </manifest>
```

Este aplicativo solicita permissão para realizar o que?

ACCESS_FINE_LOCATION: permite que um aplicativo acesse localização precisa
ACCESS_COARSE_LOCATION: permite que um aplicativo acesse localização aproximada
INTERNET: Permite que os aplicativos abram sockets de rede

11 – Qual a utilidade das classes públicas ContextCompat e ActivityCompat

As classes ContextCompat e ActivityCompat oferecem métodos para que seja realizadas a solicitação de permissões ao usuário, permissões estas necessárias à aplicação.

12 – Qual a utilidade dos Providers?

Um “Content Provider” gerencia o acesso a repositório central de dados. Um “Provider” é parte de uma aplicação Android que muitas vezes prover sua própria UI para trabalhar com os dados. No entanto, content providers são especialmente

utilizados por outras aplicações, que o acessam através do “provider client object”. Juntos, providers e providers clientes oferecem uma interface consistente e padronizada para os dados, que também, gerenciam comunicações “inter-process” e a segurança de acesso a dados.

13 – Demonstre como é implementado uma solicitação de permissão perigosa (de localização).

xml

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

java

```
@Override
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[]
grantResults)
{
    switch (requestCode) {
      case REQUEST_PERMISSIONS_CODE:
        for (int i = 0; i < permissions.length; i++) {
          if (permissions[i].equalsIgnoreCase(Manifest.permission.ACCESS_FINE_LOCATION)
             && grantResults[i] == PackageManager.PERMISSION_GRANTED) {
                readMyCurrentCoordinates();
           }
         }
         super.onRequestPermissionsResult(requestCode, permissions, grantResults);
}
```

14 – Como podemos verificar se uma determinada permissão já foi autorizada?

Utilizando o método checkSelfPermission();