

Questionário - TAG Mobile - Processo Seletivo

GRIS 21.1

David Rodrigues Albuquerque¹

¹davidra@dcc.ufrj.br - 120047390

Abril, 2021

1) O Sistema Operacional Android é um sistema baseado no kernel Linux projetado para funcionar em dispositivos móveis (smartphones, TVs, tablets...). Utiliza também uma JVM para executar as aplicações instaladas, possibilitando que as aplicações para Android sejam desenvolvidas em Java.

2) Dalvik Virtual Machine é a máquina virtual que roda por baixo das aplicações Android, responsável por traduzir o código Java para código legível pela máquina. Diferentemente da JVM, que compila o código para ByteCode para ser interpretado diretamente por uma JVM, a DVM traduz primeiramente para ByteCode e, após, realiza uma compilação do Java ByteCode para Dalvik ByteCode (arquivo .class do Java para .dex), que é um ByteCode menor e otimizado para atender melhor o sistema Android.

3) Enquanto o Dalvik compila o código durante a execução, isso pode tornar os aplicativos mais lentos. O ART é uma evolução do ART, que realiza um processo similar, mas com compilação de código antes da execução do programa. Sendo assim, ao executar a aplicação, ela já está pré-compilada e pronta para execução.

4) O processo de construção da APK reúne o bytecode compilado pela Dalvik (arquivos DEX) e outros recursos compilados em um único arquivo executável ".apk". Essa execução só é válida caso haja um certificado válido e assinado no apk, podendo essa assinatura possuir uma Key de debug ou produção.

5) Smali é uma linguagem de montagem, semelhante ao Assembly para programas em C. Podemos, por exemplo, decompilar um .dex para .smali para conseguir fazer a engenharia reversa da aplicação.

6) `method(I,[[I,Ljava/lang/String,[Ljava/lang/Object)Ljava/lang/String;`

7) O `AndroidManifest.xml` lista diversas informações fundamentais para o compilar, sistema Android e para o publicador do app, como o nomes dos pacotes, permissões necessárias, nomes do app, versões e etc.

8) a) `LoginActivity`

b) Risk permissions: `WRITE_EXTERNAL_STORAGE, READ_CONTACTS, READ_PROFILE, SEND_SMS, GET_ACCOUNTS, READ_PHONE_STATE, READ_EXTERNAL_STORAGE, READ_CALL_LOG, ACCESS_COARSE_LOCATION, USE_CREDENTIALS, READ_PROFILE`

Normal permissions: `ACCESS_WIFI_STATE, SET_WALLPAPER, INTERNET, ACCESS_NETWORK_STATE`

PS: `USE_CREDENTIALS` e `READ_PROFILE` são permissões deprecated, então as classifiquei como de risco. Mas não encontrei informações sobre seu nível de segurança antes de se tornarem deprecated.

c) `"com.android.insecurebankv2"`.

d) Permite ambos backup e debug. Na linha 15, podemos ver `android:allowBackup="true"` e `android:debuggable="true"`.

9) Intents são recursos do app e/ou da arquitetura Android que possibilitam comunicação entre components, podendo serem usados para chamar outras activities, serviços, broadcast receivers e etc.

10) Explícito: Declara explicitamente qual implementação atenderá a Intent, passando como argumento o nome de um pacote ou de uma classe específica qualificada.

Implícito: Não especifica explicitamente alguém qualificado para atender a requisição da Intent, cabendo ao sistema Android cuidar da requisição e entregar a implementação da ação requerida.

11) Intent-filters especificam os tipos de Intents que uma activity/broadcast receiver pode responder.

12) a) Implícito. A classe ou pacote não é passada no argumento.

b) Implícito. A classe ou pacote não é passada no argumento.

c) Explícito. O método declara explicitamente a implementação que deve ser utilizada (`ReceiverActivity.class`).