Permessi in Android

In Android, un **permesso** è una stringa di testo che consente di eseguire specifiche operazioni. Operazioni che possono riguardare l'hardware (come ad esempio la videocamera, il bluetooth, ecc.), oppure servizi messi a disposizione da un'altra applicazione o dal sistema, o la connettività ad Internet, o ancora l'accesso a determinati dati.

Poiché le applicazioni vengono eseguite in una sandbox, di per sé non hanno molta libertà di movimento: possono solo accedere ai propri file, a tutte quelle risorse che sono comunemente accessibili e a un insieme molto limitato di servizi. Una simile applicazione, dal punto di vista delle funzionalità, rischia di non essere molto interessante e utile, da qui il bisogno di interagire direttamente con altre risorse.

Un'applicazione, per poter interagire con l'hardware, con il sistema o con un'altra applicazione ha bisogno di particolari permessi, in modo che sia in grado di eseguire operazioni su risorse non controllate direttamente. A questo scopo, Android ha introdotto i **permessi**, per garantire e controllare l'accesso delle applicazioni a risorse terze e permettere una più ampia varietà di funzionalità.

Il sistema operativo Android viene distribuito con un insieme di permessi predefiniti che un'applicazione può richiedere. Cosa importante, i permessi necessari all'applicazione devono essere richiesti al momento dell'**installazione** e non possono essere più cambiati, salvo delle eccezioni che vedremo in seguito.

Richiesta dei permessi

I permessi, necessari all'applicazione per eseguire alcune sue funzionalità, vanno richiesti nel file *AndroidManifest.xml* (Figura 1).



Figura 1

In *AndroidManifest.xml* i permessi vengono richiesti attraverso il tag <uses-permission> che prende due attributi principali: – android:name: rappresenta il nome del permesso da richiedere; – android:maxSdkVersion: rappresenta la maggiore versione dell'SDK che garantisce la concessione del permesso associato.

Alcuni permessi possono essere concessi solo alle applicazioni di sistema e vengono sistematicamente rifiutati se a richiederli è una normale app.

Come detto, i permessi possono essere assegnati all'applicazione che li richiede solo al momento dell'installazione. Il *package manager* (pm), che si occupa di concedere i permessi, tiene traccia nel file */data/system/packages.xml* (aggiornato ogni volta che un applicazione viene installata, disinstallata o aggiornata) di tutte le informazioni relative ad un pacchetto, ad esempio path, versione, permessi concessi.

Definizione di un nuovo permesso

Le applicazioni di terze parti possono definire dei **permessi personalizzati** utilizzando gli stessi attributi di un permesso di sistema. La definizione avviene attraverso il tag <permission>, sempre nel manifesto dell'applicazione, nel seguente modo:

```
<permission android:description="string resource" android:icon="drawable resource" android:label="string resource" android:name="string" android:permissionGroup="string" android:protectionLevel=["normal" | "dangerous" | "signature" | ...] />
```

I campi interessanti sono *name*, che rappresenta il nome del

permesso da creare, e *protectionLevel*, che vedremo in dettaglio dopo.

Per vedere una lista di possibili permessi, occorre lanciare, una volta eseguito adb shell, il comando pm list permissions sul device o nell'emulatore (Figura 2).



Figura 2

Come funziona

Al momento dell'installazione, Android controlla i permessi richiesti e decide quali concedere automaticamente, quali dopo consenso dell'utente e quali non concedere. Una volta concessi, i permessi non possono essere revocati, tranne in casi particolari, e sono disponibili all'app senza ulteriori passaggi. Il comportamento di un permesso è determinato dal livello di **protezione** associato, ad esempio alcuni permessi richiedono un consenso esplicito da parte dell'utente, anche se all'applicazione è stato concesso già il permesso relativo. Vediamo ora in dettaglio proprio i livelli di protezione disponibili.

Livelli di protezione

Android associa ad ogni permesso un livello di protezione basato sulla natura delle risorse al quale il permesso dà accesso. La concessione o meno di un permesso ad una applicazione dipende dal livello di protezione associato allo specifico permesso richiesto. I possibili valori sono:

- **normal**: è il valore che, se non indicato nessun livello di protezione, viene impostato di default. Viene usato per tutti quei permessi che non servono a svolgere operazioni critiche sulle risorse. Questi permessi vengono automaticamente concessi all'applicazione, e non

richiedono una esplicita conferma da parte dell'utente.

- **dangerous**: i permessi con livello *dangerous* permettono, generalmente, di controllare in qualche modo il dispositivo o i dati presenti in esso. A differenza degli altri permessi, questi non vengono concessi automaticamente all'app che li richiede. Se un'applicazione dichiara che ha bisogno di un permesso indicato come *dangerous*, l'utente deve esplicitamente concedere il permesso all'app. Per poter utilizzare questi permessi, l'app deve mostrare, a runtime, un avviso all'utente, che può o meno accettare (Figura 3). Inoltre, un'app deve controllare i permessi ogni volta che vuole usarli, e, dalla versione 23 delle API (Android 6.0), l'utente può revocare, in qualsiasi momento, i permessi precedentemente concessi all'applicazione.
- **signature**: se il livello è *signature*, il permesso viene concesso solo a quelle applicazioni firmate con la stessa chiave dell'applicazione che definisce i permessi.



Figura 3

I livelli di protezione dei permessi built-in sono definiti nel manifesto del file */system/framework/framework-res.apk*, dove per ogni permesso definito dal sistema è indicato il relativo livello, ad esempio:

android:protectionLevel="dangerous"

Inoltre, è possibile anche indicare, nello stesso campo, dei **flag** che specificano l'azione, ad esempio:

android:protectionLevel="dangerous|privileged"

I principali flag sono:

- **privileged**: nelle API level 23 e precedenti, questo flag

si chiamava *system*. I permessi con questo flag richiedono, per essere concessi, che l'applicazione sia un'app privilegiata installata nell'immagine di sistema, ovvero che risieda sulla partizione, a sola lettura, del sistema (/system/priv-app).
- **development**: usualmente, Android non consente di richiedere o revocare i permessi dopo l'installazione; se invece il flag è impostato come *development*, il permesso in esame diventa parte dei *development permissions*, che vengono gestiti in modo differente dal sistema: sarà possibile da parte dell'applicazione richiedere o revocare il permesso anche dopo l'installazione.

Concessione effettiva dei permessi

Ad ogni applicazione sono assegnati, al momento dell'esecuzione, un UID e un GID, e, in base ai permessi richiesti, uno o più GID supplementari. Android usa questi ID per decidere se una specifica risorsa deve essere concessa all'applicazione.

Il sistema tiene traccia dei permessi necessari ad accedere ad ogni risorsa e, quando un'applicazione vuole accedere ad essa, prima di permetterlo, controlla che il permesso richiesto sia stato concesso.

Conclusioni

Abbiamo visto come vengono richiesti ed usati i permessi da parte di un'applicazione e come vengono concessi e gestiti da parte del sistema. Informazioni utili soprattutto in fase di assessment ed analisi di un'app.

A cura di: **Gianluigi Spagnuolo**

L'articolo Permessi in Android proviene da ICT Security Magazine.

https://www.ictsecuritymagazine.com/articoli/permessi-in-android/