Analyse complète du modèle de sécurité Android : Sandboxing, Permissions et SELinux

Découvrez comment Android protège vos données grâce à une architecture de sécurité multicouche.

Introduction au modèle de sécurité Android

Un modèle multi-parties

Chaque action sur Android nécessite l'accord de tous les acteurs : l'utilisateur, l'application et le système. Cette approche collaborative assure un contrôle granulaire.

Un écosystème immense

Avec plus de 3 milliards d'appareils actifs prévus en 2025, la sécurité est primordiale face à la diversité des usages, des communications aux transactions financières et à la santé.

Objectif clé : l'équilibre

Le défi majeur d'Android est de trouver un équilibre entre une sécurité robuste, la protection de la confidentialité des utilisateurs et une facilité d'usage irréprochable.

Sandboxing : Isolation des applications

L'isolation par UID

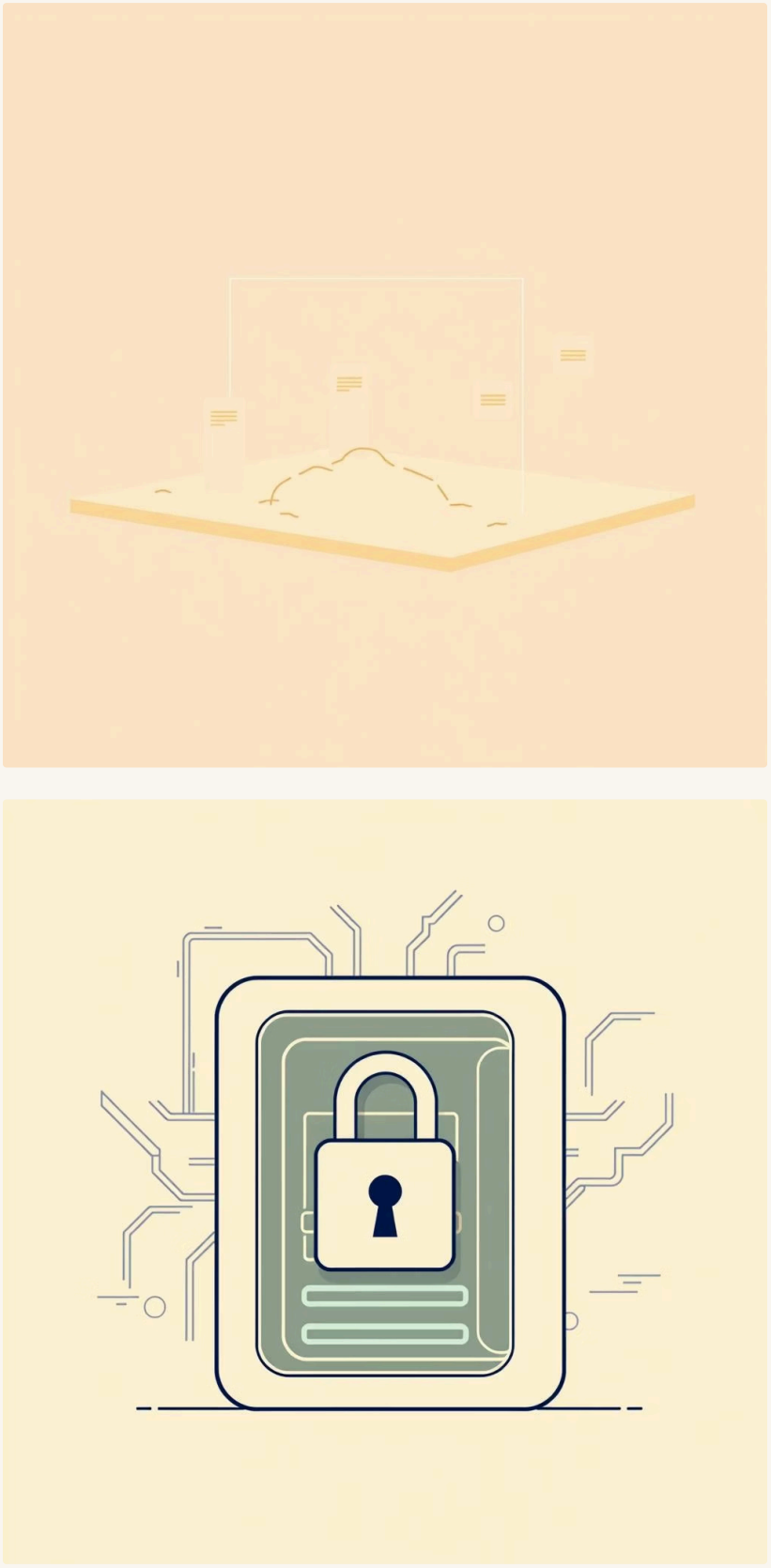
Chaque application Android reçoit un identifiant utilisateur (UID) unique lors de son installation. Cet UID permet d'isoler les processus de l'application et ses fichiers des autres applications et du système.

Contrôle d'accès discrétionnaire (DAC)

Le noyau Linux, sur lequel Android est bâti, utilise le DAC pour restreindre l'accès aux ressources système. Chaque application n'a accès qu'à ce qui lui est explicitement autorisé, limitant ainsi les interactions non souhaitées.

Les limites du DAC

Bien que le DAC soit efficace, il est souvent considéré comme "coarse-grained". Cela signifie qu'il peut manquer de finesse dans ses contrôles, le rendant potentiellement vulnérable à certaines formes d'escalade de privilèges si une application parvient à contourner ses restrictions.



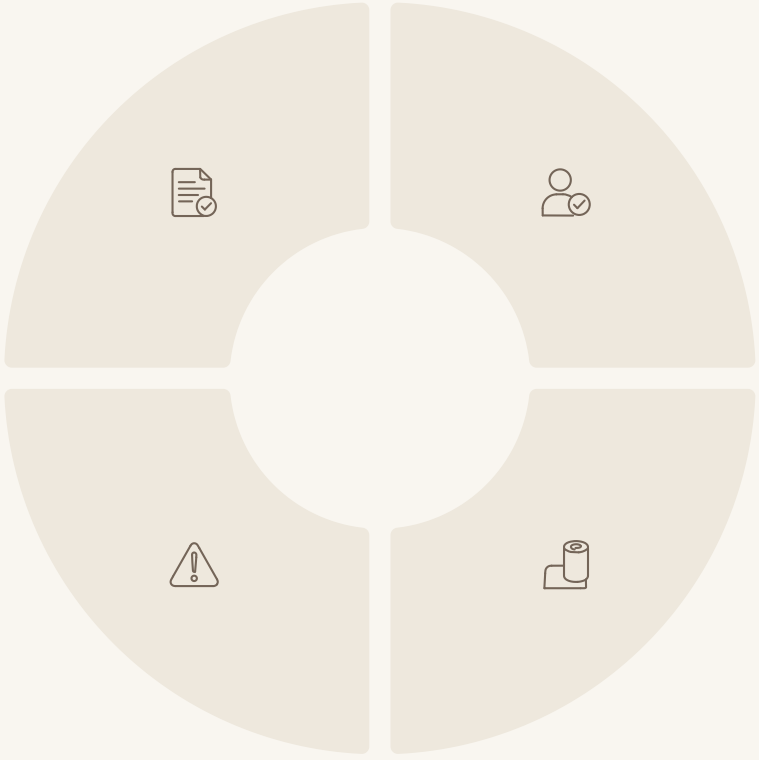
Le modèle des permissions Android

Déclaration par les développeurs

Les développeurs doivent déclarer explicitement les permissions dont leur application a besoin dans le manifeste. Cela inclut l'accès à des fonctionnalités sensibles.

Un point de vigilance

Le principal problème est que les utilisateurs peuvent accorder des permissions sans bien comprendre les implications, ce qui peut créer des risques si elles sont mal gérées ou exploitées par des applications malveillantes.



Acceptation par l'utilisateur

Avant l'installation, l'utilisateur est informé des permissions requises et doit les accepter. Sur les versions récentes d'Android, les permissions sont demandées au moment de leur utilisation.

Contrôle d'accès rigoureux

Ces permissions contrôlent l'accès aux composants d'application critiques et aux ressources sensibles telles que la caméra, la localisation, le microphone ou le stockage externe.

SELinux : Renforcement par contrôle d’accès obligatoire (MAC)

SELinux (Security-Enhanced Linux) est une couche de sécurité supplémentaire qui implémente le Contrôle d'Accès Obligatoire (MAC), renforçant considérablement le modèle de sécurité Android.

1

Politique de sécurité systémique

SELinux ne se base pas sur les autorisations des utilisateurs, mais sur une politique de sécurité définie au niveau du système, qui est appliquée de manière obligatoire.

2

Confinement des processus

Il confine les processus et les ressources en leur attribuant des labels de sécurité. Seules les interactions explicitement autorisées par la politique SELinux peuvent avoir lieu.

3

Protection avancée

Même les processus root sont soumis aux restrictions de SELinux, ce qui signifie qu'une application obtenant des privilèges root ne peut pas nécessairement faire ce qu'elle veut si SELinux l'en empêche. Il protège contre les escalades de privilèges.

4

Intégration clé dans Android

SELinux est intégré à Android depuis la version 5 (Lollipop), il a transformé la sécurité en limitant les interactions non autorisées entre les applications et les composants du système.

Fonctionnement de SELinux dans Android

Confinement des démons système

SELinux confiner les démons système privilégiés (comme zygote ou mediaserver). En cas de compromission, les dégâts sont minimisés car le démon ne peut accéder qu'à un ensemble très limité de ressources.

Séparation stricte des applications

Il garantit une séparation stricte non seulement entre les applications et le système, mais aussi entre les applications elles-mêmes. Une application ne peut pas lire ou écrire dans l'espace mémoire d'une autre sans permission explicite.

Politique centralisée et modifiable

La politique SELinux est centralisée et peut être analysée pour comprendre les flux de données et d'accès. Elle est également modifiable, permettant aux fabricants d'appareils d'adapter les règles aux besoins spécifiques de leurs implémentations Android.

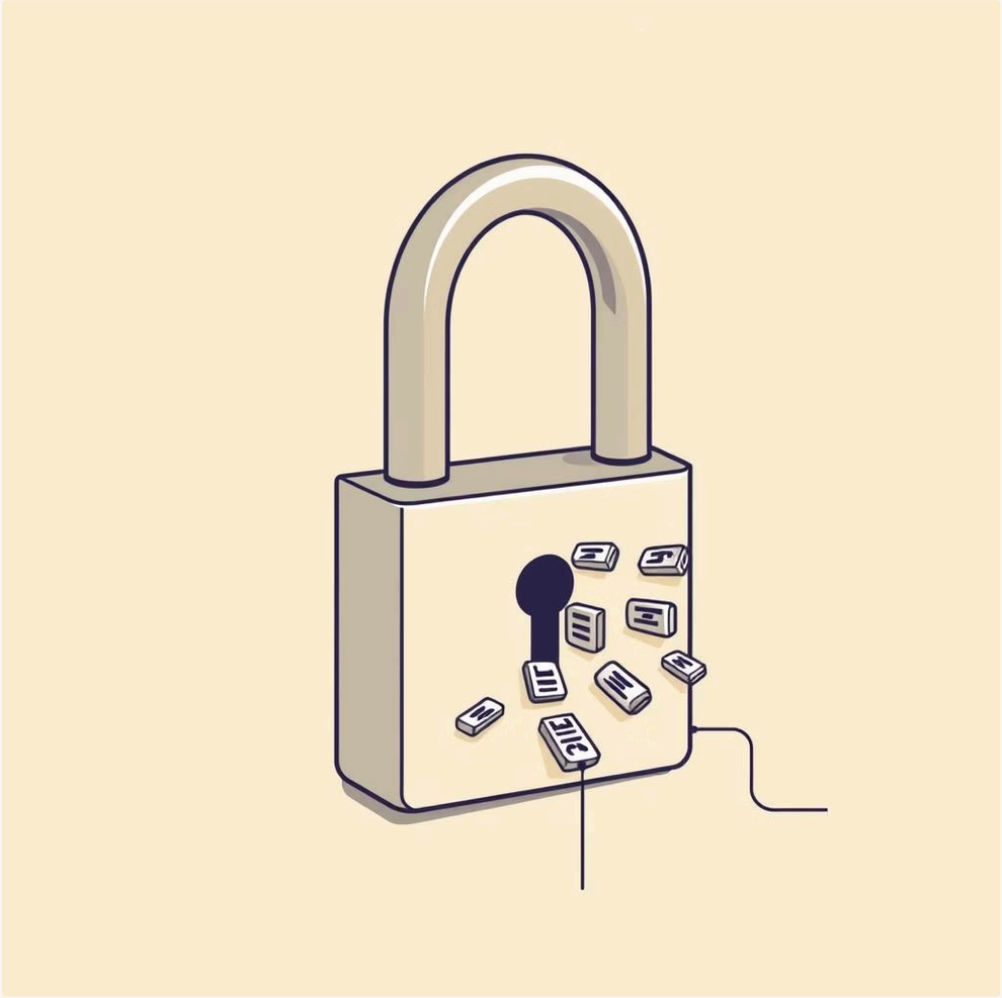
Étude de cas : Attaque ANDROSCOPE (2024)

La vulnérabilité : canal auxiliaire sur cache CPU

L'attaque ANDROSCOPE, découverte en 2024, a mis en lumière une faiblesse dans l'isolation du sandboxing. Elle exploitait un canal auxiliaire lié au cache CPU, permettant à des applications isolées de faire fuiter des informations entre leurs sandboxes respectives.

Exploitation du DICI

Cette attaque tire parti du concept de "Dynamic Inter-Component Interaction" (DICI), où des composants partagés entre différentes applications peuvent devenir des vecteurs de fuite d'informations, même avec des applications isolées par UID.



Cette attaque illustre un point critique : les limites du sandboxing purement logiciel face aux attaques sophistiquées qui exploitent les interactions au niveau matériel.

Réponse du modèle Android face aux attaques

SELinux, une première ligne

SELinux joue un rôle crucial en bloquant de nombreuses tentatives d'escalade de privilèges, même si le DAC ou la sandbox initiale est compromise. Il ne peut cependant pas contrer toutes les vulnérabilités du noyau.

Améliorations continues

Android évolue constamment avec des fonctionnalités comme le Scoped Storage (restreignant l'accès au stockage), des permissions plus granulaires (accès temporaire à la localisation, etc.) et des renforcements du noyau Linux.

Politique rigoureuse et moindre privilège

Une politique SELinux bien conçue, combinée à une architecture applicative qui adhère strictement au principe du moindre privilège, sont essentiels pour minimiser les risques.

Synthèse : Forces et limites du modèle de sécurité Android

Le modèle de sécurité Android est une architecture robuste qui combine plusieurs couches de protection pour offrir une défense en profondeur.

Forces : Défense en profondeur

1

L'association du sandboxing, du modèle de permissions et de SELinux crée un système résilient face à de nombreuses menaces.

2

Limites : Nouvelles menaces

Malgré sa robustesse, le modèle doit faire face aux nouvelles menaces, notamment celles qui exploitent les interactions matérielles ou les vulnérabilités de bas niveau.

Vigilance constante requise

3

La course à l'armement entre attaquants et défenseurs exige une vigilance constante et des mises à jour régulières des politiques et des mécanismes de sécurité.

4

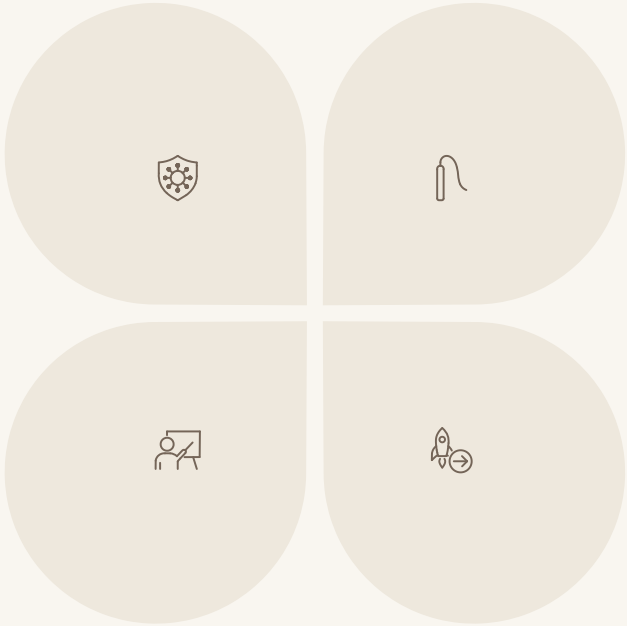
Collaboration essentielle

La sécurité de l'écosystème Android dépend d'une collaboration étroite entre les développeurs d'applications, les fabricants d'appareils et les utilisateurs.

Conclusion & Perspectives

Un modèle robuste mais perfectible

Android a mis en place un système de sécurité solide basé sur l'isolation des applications, le contrôle granulaire des permissions et les politiques MAC de SELinux.



Responsabilité partagée

Un écosystème mobile réellement sûr est le fruit d'efforts conjoints de toutes les parties prenantes, des ingénieurs aux utilisateurs finaux.

L'impact majeur de SELinux

L'intégration de SELinux a été une avancée fondamentale pour la protection des données et la prévention des escalades de privilèges, même pour les processus critiques.

Vers un futur plus sûr

Les perspectives incluent le renforcement de la sécurité matérielle, l'affinement des politiques de sécurité et une sensibilisation accrue des utilisateurs.