

### Élévation de privilège :

Octroyer à un individu des autorisations supérieures à celles initialement accordées.

Par exemple, passer de droits "en lecture seule" à des droits "en lecture et en écriture" d'une façon ou d'une autre.

## Élévation de privilèges

- Élévation de privilèges



### Élévation de privilège :

Octroyer à un individu des autorisations supérieures à celles initialement accordées.

Par exemple, passer de droits "en lecture seule" à des droits "en lecture et en écriture" d'une façon ou d'une autre.

## Élévation de privilèges

• Élévation de privilèges

Qu'est ce qu'une élévation de privilège ?



### Élévation de privilège :

Octroyer à un individu des autorisations supérieures à celles initialement accordées.

Par exemple, passer de droits "en lecture seule" à des droits "en lecture et en écriture" d'une façon ou d'une autre.

## Élévation de privilèges

• Élévation de privilèges

Qu'est ce qu'une élévation de privilège ?

Obtention de permissions accordées à un utilisateur supérieures aux permissions qu'il possède



### Élévation de privilège :

Octroyer à un individu des autorisations supérieures à celles initialement accordées.

Par exemple, passer de droits "en lecture seule" à des droits "en lecture et en écriture" d'une façon ou d'une autre.<

### Intérêts :

- Android restreints les droits utilisateurs (on ne peut pas désinstaller certaines application)
- On va chercher à utiliser davantage de possibilités qui ne sont pas accessible spar défaut
- Un des intèrés peut par exemple être la moification en profondeur le fonctionnement des applications

## Élévation de privilèges

• Élévation de privilèges

Qu'est ce qu'une élévation de privilège ?

Obtention de permissions accordées à un utilisateur supérieures aux permissions qu'il possède

Intérêt :



### Élévation de privilège :

Octroyer à un individu des autorisations supérieures à celles initialement accordées.

Par exemple, passer de droits "en lecture seule" à des droits "en lecture et en écriture" d'une façon ou d'une autre.<

### Intérêts :

- Android restreint les droits utilisateurs (on ne peut pas désinstaller certaines applications)
- On va chercher à utiliser davantage de possibilités qui ne sont pas accessibles par défaut
- Un des intérêts peut par exemple être la modification en profondeur le fonctionnement des applications

## Élévation de privilèges

• Élévation de privilèges

**Qu'est ce qu'une élévation de privilège ?**

Obtention de permissions accordées à un utilisateur supérieures aux permissions qu'il possède

**Intérêt :**

- ▶ Android est un système qui restreint l'utilisateur



### Élévation de privilège :

Octroyer à un individu des autorisations supérieures à celles initialement accordées.

Par exemple, passer de droits "en lecture seule" à des droits "en lecture et en écriture" d'une façon ou d'une autre.<

### Intérêts :

- Android restreint les droits utilisateurs (on ne peut pas désinstaller certaines applications)
- On va chercher à utiliser davantage de possibilités qui ne sont pas accessibles par défaut
- Un des intérêts peut par exemple être la modification en profondeur le fonctionnement des applications

## Élévation de privilèges

• Élévation de privilèges

### Qu'est ce qu'une élévation de privilège ?

Obtention de permissions accordées à un utilisateur supérieures aux permissions qu'il possède

### Intérêt :

- ▶ Android est un système qui restreint l'utilisateur
- ▶ Accéder aux fonctionnalités bloquées



### Élévation de privilège :

Octroyer à un individu des autorisations supérieures à celles initialement accordées.

Par exemple, passer de droits "en lecture seule" à des droits "en lecture et en écriture" d'une façon ou d'une autre.<

### Intérêts :

- Android restreint les droits utilisateurs (on ne peut pas désinstaller certaines applications)
- On va chercher à utiliser davantage de possibilités qui ne sont pas accessibles par défaut
- Un des intérêts peut par exemple être la modification en profondeur le fonctionnement des applications

## Élévation de privilèges

• Élévation de privilèges

### Qu'est ce qu'une élévation de privilège ?

Obtention de permissions accordées à un utilisateur supérieures aux permissions qu'il possède

### Intérêt :

- ▶ Android est un système qui restreint l'utilisateur
- ▶ Accéder aux fonctionnalités bloquées
- ▶ Modifier en profondeur le fonctionnement des applications



### **Root :**

Utilisation de privilèges avancés, permettant de limiter des limitations imposées par le système

Par exemple, cela permet de supprimer les applications système, qui ne sont pas désinstallables en tant que simple utilisateur.

## **Élévation de privilèges : Root**

• Élévation de privilèges





### **Root :**

Utilisation de privilèges avancés, permettant de limiter des limitations imposées par le système

Par exemple, cela permet de supprimer les applications systèmes, qui ne sont pas désinstallables en tant que simple utilisateur.

## Élévation de privilèges : Root

• Élévation de privilèges

Qu'est ce que le root ?



### **Root :**

Utilisation de privilèges avancés, permettant de limiter des limitations imposées par le système

Par exemple, cela permet de supprimer les applications systèmes, qui ne sont pas désinstallables en tant que simple utilisateur.

## **Élévation de privilèges : Root**

• Élévation de privilèges

### **Qu'est ce que le root ?**

Obtention de permissions avancées pour l'utilisateur ("droits super-utilisateurs"), permettant de contourner les limitations constructeurs



### **Root :**

Utilisation de privilèges avancés, permettant de limiter des limitations imposées par le système

Par exemple, cela permet de supprimer les applications systèmes, qui ne sont pas désinstallables en tant que simple utilisateur.

### **Principe du root :**

- Utilisation d'une faille d'android, ou alors du mode récupération d'android pour obtenir temporairement un uid à 0, c'est à dire root
- Remontage de la partition système en écriture, afin de pouvoir la modifier

- Copie de nouveaux binaires, tels que su, busybox
- Remontage de la partition système en lecture seule

## Élévation de privilèges : Root

• Élévation de privilèges

### Qu'est ce que le root ?

Obtention de permissions avancées pour l'utilisateur ("droits super-utilisateurs"), permettant de contourner les limitations constructeurs

### Principe du root : /system



### **Root :**

Utilisation de privilèges avancés, permettant de limiter des limitations imposées par le système

Par exemple, cela permet de supprimer les applications système, qui ne sont pas désinstallables en tant que simple utilisateur.

### **Principe du root :**

- Utilisation d'une faille d'android, ou alors du mode récupération d'android pour obtenir temporairement un uid à 0, c'est à dire root
- Remontage de la partition système en écriture, afin de pouvoir la modifier

- Copie de nouveaux binaires, tels que su, busybox
- Remontage de la partition système en lecture seule

## Élévation de privilèges : Root

• Élévation de privilèges

### Qu'est ce que le root ?

Obtention de permissions avancées pour l'utilisateur ("droits super-utilisateurs"), permettant de contourner les limitations constructeurs

### Principe du root : /system

1. Utilisation d'une vulnérabilité par un processus pour changer son uid à 0



### **Root :**

Utilisation de privilèges avancés, permettant de limiter des limitations imposées par le système

Par exemple, cela permet de supprimer les applications système, qui ne sont pas désinstallables en tant que simple utilisateur.

### **Principe du root :**

- Utilisation d'une faille d'android, ou alors du mode récupération d'android pour obtenir temporairement un uid à 0, c'est à dire root
- Remontage de la partition système en écriture, afin de pouvoir la modifier

- Copie de nouveaux binaires, tels que su, busybox
- Remontage de la partition système en lecture seule

## Élévation de privilèges : Root

• Élévation de privilèges

### Qu'est ce que le root ?

Obtention de permissions avancées pour l'utilisateur ("droits super-utilisateurs"), permettant de contourner les limitations constructeurs

### Principe du root : /system

1. Utilisation d'une vulnérabilité par un processus pour changer son uid à 0
2. Remontage de la partition /system en écriture



### **Root :**

Utilisation de privilèges avancés, permettant de limiter des limitations imposées par le système

Par exemple, cela permet de supprimer les applications systèmes, qui ne sont pas désinstallables en tant que simple utilisateur.

### **Principe du root :**

- Utilisation d'une faille d'android, ou alors du mode récupération d'android pour obtenir temporairement un uid à 0, c'est à dire root
- Remontage de la partition système en écriture, afin de pouvoir la modifier

- Copie de nouveaux binaires, tels que su, busybox

- Remontage de la partition système en lecture seule

## **Élévation de privilèges : Root**

• Élévation de privilèges

### **Qu'est ce que le root ?**

Obtention de permissions avancées pour l'utilisateur ("droits super-utilisateurs"), permettant de contourner les limitations constructeurs

### **Principe du root : /system**

1. Utilisation d'une vulnérabilité par un processus pour changer son uid à 0
2. Remontage de la partition /system en écriture
3. Copie des binaires su, busybox



### **Root :**

Utilisation de privilèges avancés, permettant de limiter des limitations imposées par le système

Par exemple, cela permet de supprimer les applications système, qui ne sont pas désinstallables en tant que simple utilisateur.

### **Principe du root :**

- Utilisation d'une faille d'android, ou alors du mode récupération d'android pour obtenir temporairement un uid à 0, c'est à dire root
- Remontage de la partition système en écriture, afin de pouvoir la modifier

- Copie de nouveaux binaires, tels que su, busybox
- Remontage de la partition système en lecture seule

## Élévation de privilèges : Root

• Élévation de privilèges

### Qu'est ce que le root ?

Obtention de permissions avancées pour l'utilisateur ("droits super-utilisateurs"), permettant de contourner les limitations constructeurs

### Principe du root : /system

1. Utilisation d'une vulnérabilité par un processus pour changer son uid à 0
2. Remontage de la partition /system en écriture
3. Copie des binaires su, busybox
4. Remontage de /system en lecture seule



## **Root :**

Utilisation de privilèges avancés, permettant de limiter des limitations imposées par le système

Par exemple, cela permet de supprimer les applications systèmes, qui ne sont pas désinstallables en tant que simple utilisateur.

## **Principe du root :**

- Utilisation d'une faille d'Android, ou alors du mode récupération d'Android pour obtenir temporairement un uid à 0, c'est à dire root
- Remontage de la partition système en écriture, afin de pouvoir la modifier

- Copie de nouveaux binaires, tels que su, busybox

- Remontage de la partition système en lecture seule

## **Exemples d'utilisation :**

- Accéder aux partitions systèmes
- Installation de busybox
- Sauvegarder une application en conservant l'état de l'application au moment de la sauvegarde
- Modifier des propriétés systèmes (densité d'écran, adresse mac...)

# Élévation de privilèges : Root

• Élévation de privilèges

## Qu'est ce que le root ?

Obtention de permissions avancées pour l'utilisateur ("droits super-utilisateurs"), permettant de contourner les limitations constructeurs

## Exemples d'utilisation





### Root :

Utilisation de privilèges avancés, permettant de limiter des limitations imposées par le système

Par exemple, cela permet de supprimer les applications systèmes, qui ne sont pas désinstallables en tant que simple utilisateur.

### Principe du root :

- Utilisation d'une faille d'Android, ou alors du mode récupération d'Android pour obtenir temporairement un uid à 0, c'est à dire root
- Remontage de la partition système en écriture, afin de pouvoir la modifier

- Copie de nouveaux binaires, tels que su, busybox

- Remontage de la partition système en lecture seule

### Exemples d'utilisation :

- Accéder aux partitions systèmes
- Installation de busybox
- Sauvegarder une application en conservant l'état de l'application au moment de la sauvegarde
- Modifier des propriétés systèmes (densité d'écran, adresse mac...)

## Élévation de privilèges : Root

• Élévation de privilèges

### Qu'est ce que le root ?

Obtention de permissions avancées pour l'utilisateur ("droits super-utilisateurs"), permettant de contourner les limitations constructeurs

### Exemples d'utilisation

- ▶ Accéder aux partitions systèmes



### **Root :**

Utilisation de privilèges avancés, permettant de limiter des limitations imposées par le système

Par exemple, cela permet de supprimer les applications systèmes, qui ne sont pas désinstallables en tant que simple utilisateur.

### **Principe du root :**

- Utilisation d'une faille d'Android, ou alors du mode récupération d'Android pour obtenir temporairement un uid à 0, c'est à dire root
- Remontage de la partition système en écriture, afin de pouvoir la modifier

- Copie de nouveaux binaires, tels que su, busybox

- Remontage de la partition système en lecture seule

### **Exemples d'utilisation :**

- Accéder aux partitions systèmes
- Installation de busybox
- Sauvegarder une application en conservant l'état de l'application au moment de la sauvegarde
- Modifier des propriétés systèmes (densité d'écran, adresse mac...)

## Élévation de privilèges : Root

• Élévation de privilèges

### Qu'est ce que le root ?

Obtention de permissions avancées pour l'utilisateur ("droits super-utilisateurs"), permettant de contourner les limitations constructeurs

### Exemples d'utilisation

- ▶ Accéder aux partitions systèmes
- ▶ Ajouter un binaire BusyBox



### **Root :**

Utilisation de privilèges avancés, permettant de limiter des limitations imposées par le système

Par exemple, cela permet de supprimer les applications systèmes, qui ne sont pas désinstallables en tant que simple utilisateur.

### **Principe du root :**

- Utilisation d'une faille d'Android, ou alors du mode récupération d'Android pour obtenir temporairement un uid à 0, c'est à dire root
- Remontage de la partition système en écriture, afin de pouvoir la modifier

- Copie de nouveaux binaires, tels que su, busybox

- Remontage de la partition système en lecture seule

### **Exemples d'utilisation :**

- Accéder aux partitions systèmes
- Installation de busybox
- Sauvegarder une application en conservant l'état de l'application au moment de la sauvegarde
- Modifier des propriétés systèmes (densité d'écran, adresse mac...)

## Élévation de privilèges : Root

• Élévation de privilèges

### Qu'est ce que le root ?

Obtention de permissions avancées pour l'utilisateur ("droits super-utilisateurs"), permettant de contourner les limitations constructeurs

### Exemples d'utilisation

- ▶ Accéder aux partitions systèmes
- ▶ Ajouter un binaire BusyBox
- ▶ Sauvegarder l'état actuel d'une application



### **Root :**

Utilisation de privilèges avancés, permettant de limiter des limitations imposées par le système

Par exemple, cela permet de supprimer les applications systèmes, qui ne sont pas désinstallables en tant que simple utilisateur.

### **Principe du root :**

- Utilisation d'une faille d'Android, ou alors du mode récupération d'Android pour obtenir temporairement un uid à 0, c'est à dire root
- Remontage de la partition système en écriture, afin de pouvoir la modifier

- Copie de nouveaux binaires, tels que su, busybox

- Remontage de la partition système en lecture seule

### **Exemples d'utilisation :**

- Accéder aux partitions systèmes
- Installation de busybox
- Sauvegarder une application en conservant l'état de l'application au moment de la sauvegarde
- Modifier des propriétés systèmes (densité d'écran, adresse mac...)

## Élévation de privilèges : Root

• Élévation de privilèges

### Qu'est ce que le root ?

Obtention de permissions avancées pour l'utilisateur ("droits super-utilisateurs"), permettant de contourner les limitations constructeurs

### Exemples d'utilisation

- ▶ Accéder aux partitions systèmes
- ▶ Ajouter un binaire BusyBox
- ▶ Sauvegarder l'état actuel d'une application
- ▶ Modifier les propriétés systèmes

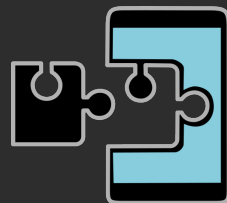


### Xposed :

- Framework permettant d'intercepter toutes méthodes d'une application, pour injecter du code supplémentaire
- Ne fonctionne qu'avec les applications java, pas avec les bibliothèques natives, par exemple

## Élévation de privilèges : Xposed

- Élévation de privilèges



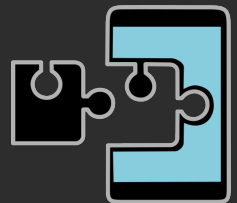
### Xposed :

- Framework permettant d'intercepter toutes méthodes d'une application, pour injecter du code supplémentaire
- Ne fonctionne qu'avec les applications java, pas avec les bibliothèques natives, par exemple

## Élévation de privilèges : Xposed

• Élévation de privilèges

Qu'est ce que le module Xposed ?



### Xposed :

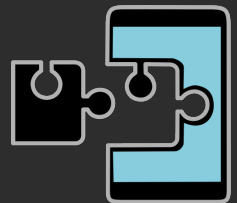
- Framework permettant d'intercepter toutes méthodes d'une application, pour injecter du code supplémentaire
- Ne fonctionne qu'avec les applications java, pas avec les bibliothèques natives, par exemple

## Élévation de privilèges : Xposed

• Élévation de privilèges

### Qu'est ce que le module Xposed ?

Framework permettant d'intercepter toutes méthodes d'une application, pour injecter du code supplémentaire



### Xposed :

- Framework permettant d'intercepter toutes méthodes d'une application, pour injecter du code supplémentaire
- Ne fonctionne qu'avec les applications java, pas avec les bibliothèques natives, par exemple

### Exemples d'utilisation de Xposed :

- Lire les paramètres des applications
- Désactiver la vérification SSL, pour par exemple, pouvoir déchiffrer le trafic
- Modifier son IMEI
- Simuler sa position GPS

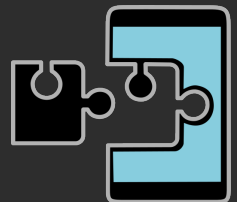
## Élévation de privilèges : Xposed

• Élévation de privilèges

### Qu'est ce que le module Xposed ?

Framework permettant d'intercepter toutes méthodes d'une application, pour injecter du code supplémentaire

### Exemple d'utilisation





### Xposed :

- Framework permettant d'intercepter toutes méthodes d'une application, pour injecter du code supplémentaire
- Ne fonctionne qu'avec les applications java, pas avec les bibliothèques natives, par exemple

### Exemples d'utilisation de Xposed :

- Lire les paramètres des applications
- Désactiver la vérification SSL, pour par exemple, pouvoir déchiffrer le trafic
- Modifier son IMEI
- Simuler sa position GPS

## Élévation de privilèges : Xposed

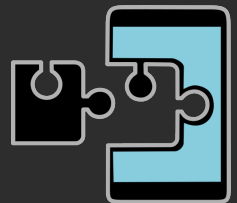
• Élévation de privilèges

### Qu'est ce que le module Xposed ?

Framework permettant d'intercepter toutes méthodes d'une application, pour injecter du code supplémentaire

### Exemple d'utilisation

- ▶ Lire les préférences



### Xposed :

- Framework permettant d'intercepter toutes méthodes d'une application, pour injecter du code supplémentaire
- Ne fonctionne qu'avec les applications java, pas avec les bibliothèques natives, par exemple

### Exemples d'utilisation de Xposed :

- Lire les paramètres des applications
- Désactiver la vérification SSL, pour par exemple, pouvoir déchiffrer le trafic
- Modifier son IMEI
- Simuler sa position GPS

## Élévation de privilèges : Xposed

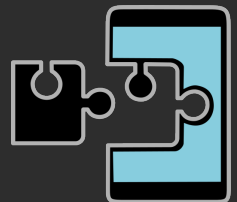
• Élévation de privilèges

### Qu'est ce que le module Xposed ?

Framework permettant d'intercepter toutes méthodes d'une application, pour injecter du code supplémentaire

### Exemple d'utilisation

- ▶ Lire les préférences
- ▶ Désactiver la vérification des certificats SSL/TLS



### Xposed :

- Framework permettant d'intercepter toutes méthodes d'une application, pour injecter du code supplémentaire
- Ne fonctionne qu'avec les applications java, pas avec les bibliothèques natives, par exemple

### Exemples d'utilisation de Xposed :

- Lire les paramètres des applications
- Désactiver la vérification SSL, pour par exemple, pouvoir déchiffrer le trafic
- Modifier son IMEI
- Simuler sa position GPS

## Élévation de privilèges : Xposed

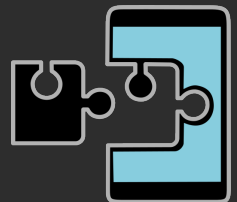
• Élévation de privilèges

### Qu'est ce que le module Xposed ?

Framework permettant d'intercepter toutes méthodes d'une application, pour injecter du code supplémentaire

### Exemple d'utilisation

- ▶ Lire les préférences
- ▶ Désactiver la vérification des certificats SSL/TLS
- ▶ Modifier son IMEI



### Xposed :

- Framework permettant d'intercepter toutes méthodes d'une application, pour injecter du code supplémentaire
- Ne fonctionne qu'avec les applications java, pas avec les bibliothèques natives, par exemple

### Exemples d'utilisation de Xposed :

- Lire les paramètres des applications
- Désactiver la vérification SSL, pour par exemple, pouvoir déchiffrer le trafic
- Modifier son IMEI
- Simuler sa position GPS

## Élévation de privilèges : Xposed

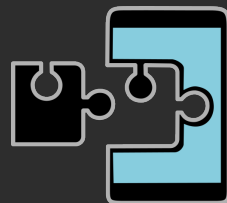
• Élévation de privilèges

### Qu'est ce que le module Xposed ?

Framework permettant d'intercepter toutes méthodes d'une application, pour injecter du code supplémentaire

### Exemple d'utilisation

- ▶ Lire les préférences
- ▶ Désactiver la vérification des certificats SSL/TLS
- ▶ Modifier son IMEI
- ▶ Modifier sa position GPS



### Le démarrage d'Android :

- On s'intéresse au processus une fois le lancement du kernel
- Le kernel initialise le processus Init
- Init lance à son tour des démons (usb, adb, ril), et le runtime
- Init lance aussi Zygote

## Élévation de privilèges : Xposed

- Élévation de privilèges

### Le démarrage d'Android :

- On s'intéresse au processus une fois le lancement du kernel
- Le kernel initialise le processus Init
- Init lance à son tour des démons (usb, adb, ril), et le runtime
- Init lance aussi Zygote

## Élévation de privilèges : Xposed

• Élévation de privilèges

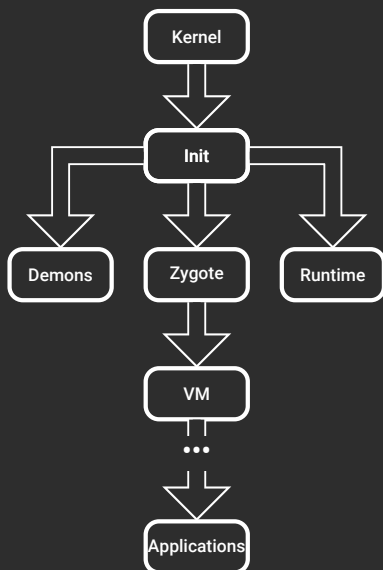


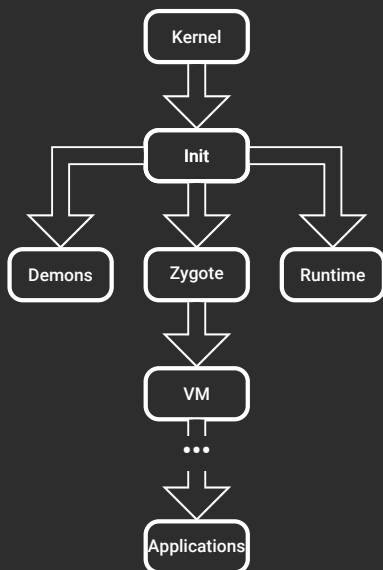
FIGURE – Initialisation d'Android

## Le démarrage d'Android :

- On s'intéresse au processus une fois le lancement du kernel
- Le kernel initialise le processus Init
- Init lance à son tour des démons (usb, adb, ril), et le runtime
- Init lance aussi Zygote

## Élévation de privilèges : Xposed

• Élévation de privilèges



### Démarrage d'Android

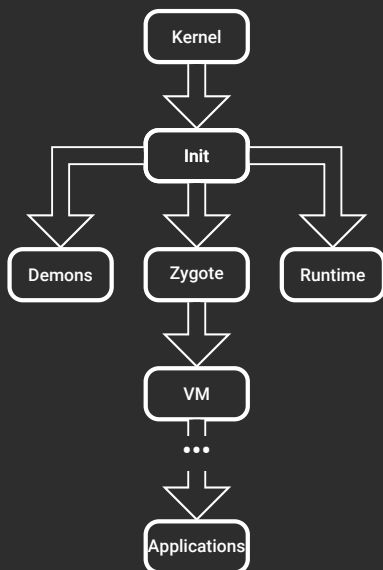
FIGURE – Initialisation d'Android

## Le démarrage d'Android :

- On s'intéresse au processus une fois le lancement du kernel
- Le kernel initialise le processus Init
- Init lance à son tour des démons (usb, adb, ril), et le runtime
- Init lance aussi Zygote

## Élévation de privilèges : Xposed

• Élévation de privilèges



### Démarrage d'Android

#### 1. Le kernel lance le processus init

FIGURE – Initialisation d'Android

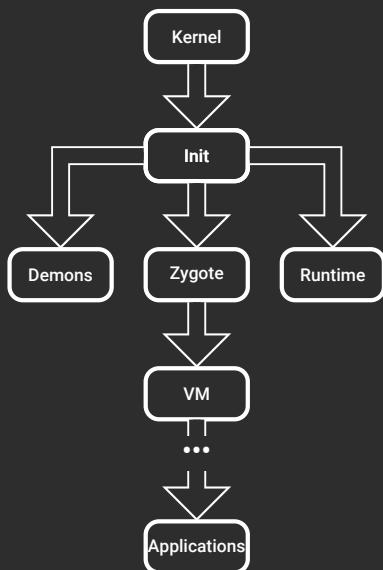


## Le démarrage d'Android :

- On s'intéresse au processus une fois le lancement du kernel
- Le kernel initialise le processus Init
- Init lance à son tour des démons (usb, adb, ril), et le runtime
- Init lance aussi Zygote

## Élévation de privilèges : Xposed

• Élévation de privilèges



### Démarrage d'Android

1. Le kernel lance le processus init
2. Init lance des demons, runtime

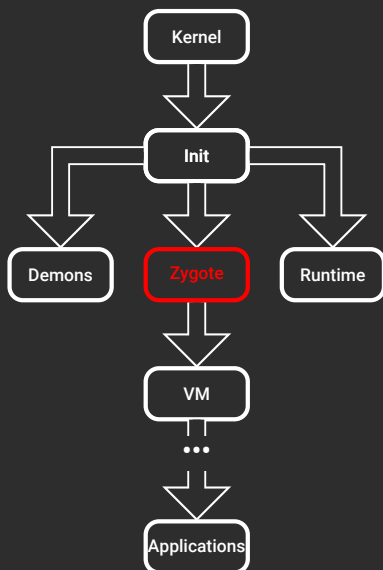
FIGURE – Initialisation d'Android

## Le démarrage d'Android :

- On s'intéresse au processus une fois le lancement du kernel
- Le kernel initialise le processus Init
- Init lance à son tour des démons (usb, adb, ril), et le runtime
- Init lance aussi Zygote

## Élévation de privilèges : Xposed

• Élévation de privilèges



### Démarrage d'Android

1. Le kernel lance le processus init
2. Init lance des demons, runtime
3. Init lance Zygote

FIGURE – Initialisation d'Android

### Le démarrage d'Android :

- On s'intéresse au processus une fois le lancement du kernel
- Le kernel initialise le processus Init
- Init lance à son tour des démons (usb, adb, ril), et le runtime
- Init lance aussi Zygote

### Zygote :

Zygote est un processus primordiale pour Android :

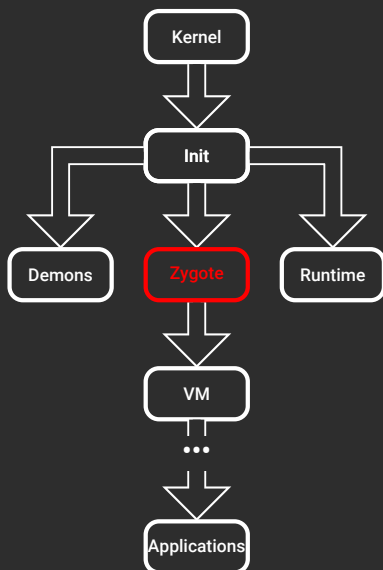
- Il initialise la machine virtuelle
- Pré-charge des classes communes aux applications
- Se fork pour chaque nouvelle application

lancée

- Partage une partie de sa mémoire avec les applications forkées

## Élévation de privilèges : Xposed

• Élévation de privilèges



### Démarrage d'Android

1. Le kernel lance le processus init
2. Init lance des demons, runtime
3. Init lance Zygote

### Le processus Zygote :

FIGURE – Initialisation d'Android

### Le démarrage d'Android :

- On s'intéresse au processus une fois le lancement du kernel
- Le kernel initialise le processus Init
- Init lance à son tour des démons (usb, adb, ril), et le runtime
- Init lance aussi Zygote

### Zygote :

Zygote est un processus primordiale pour Android :

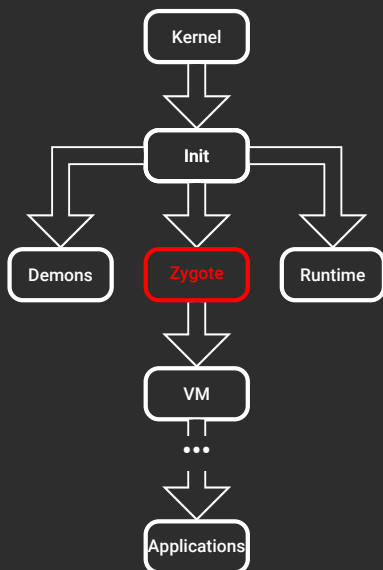
- Il initialise la machine virtuelle
- Pré-charge des classes communes aux applications
- Se fork pour chaque nouvelle application

lancée

- Partage une partie de sa mémoire avec les applications forkées

## Élévation de privilèges : Xposed

• Élévation de privilèges



### Démarrage d'Android

1. Le kernel lance le processus init
2. Init lance des demons, runtime
3. Init lance Zygote

### Le processus Zygote :

1. Initialise une instance de la VM

FIGURE – Initialisation d'Android

### Le démarrage d'Android :

- On s'intéresse au processus une fois le lancement du kernel
- Le kernel initialise le processus Init
- Init lance à son tour des démons (usb, adb, ril), et le runtime
- Init lance aussi Zygote

### Zygote :

Zygote est un processus primordiale pour Android :

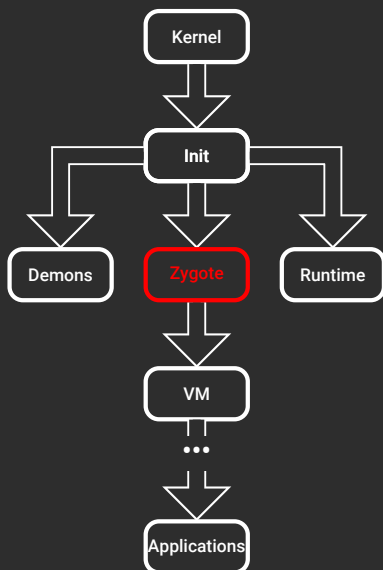
- Il initialise la machine virtuelle
- Pré-charge des classes communes aux applications
- Se fork pour chaque nouvelle application

lancée

- Partage une partie de sa mémoire avec les applications forkées

## Élévation de privilèges : Xposed

• Élévation de privilèges



### Démarrage d'Android

1. Le kernel lance le processus init
2. Init lance des demons, runtime
3. Init lance Zygote

### Le processus Zygote :

1. Initialise une instance de la VM
2. Pré-charge des classes

FIGURE – Initialisation d'Android

### Le démarrage d'Android :

- On s'intéresse au processus une fois le lancement du kernel
- Le kernel initialise le processus Init
- Init lance à son tour des démons (usb, adb, ril), et le runtime
- Init lance aussi Zygote

### Zygote :

Zygote est un processus primordial pour Android :

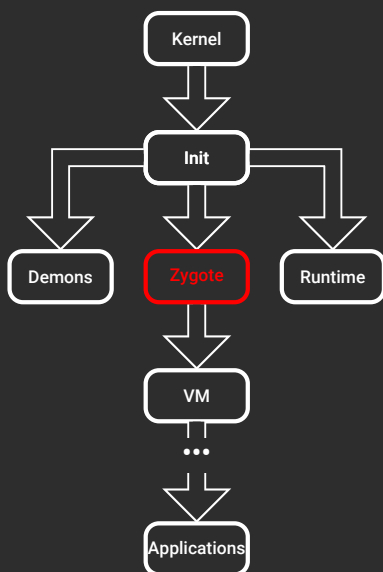
- Il initialise la machine virtuelle
- Pré-charge des classes communes aux applications
- Se fork pour chaque nouvelle application

lancée

- Partage une partie de sa mémoire avec les applications forkées

## Élévation de privilèges : Xposed

• Élévation de privilèges



### Démarrage d'Android

1. Le kernel lance le processus init
2. Init lance des demons, runtime
3. Init lance Zygote

### Le processus Zygote :

1. Initialise une instance de la VM
2. Pré-charge des classes
3. Fork pour chaque application

FIGURE – Initialisation d'Android

### Le démarrage d'Android :

- On s'intéresse au processus une fois le lancement du kernel
- Le kernel initialise le processus Init
- Init lance à son tour des démons (usb, adb, ril), et le runtime
- Init lance aussi Zygote

### Zygote :

Zygote est un processus primordiale pour Android :

- Il initialise la machine virtuelle
- Pré-charge des classes communes aux applications
- Se fork pour chaque nouvelle application

lancée

- Partage une partie de sa mémoire avec les applications forkées

## Élévation de privilèges : Xposed

• Élévation de privilèges

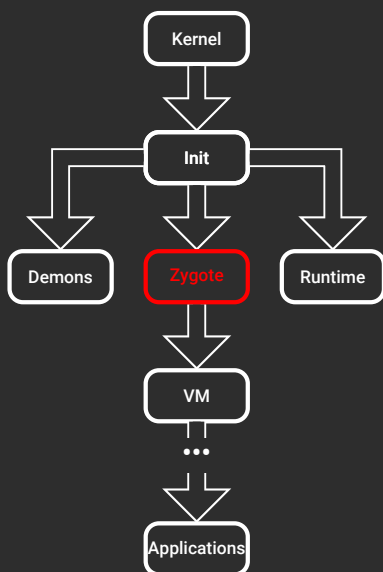


FIGURE – Initialisation d'Android

### Démarrage d'Android

1. Le kernel lance le processus init
2. Init lance des demons, runtime
3. Init lance Zygote

### Le processus Zygote :

1. Initialise une instance de la VM
2. Pré-charge des classes
3. Fork pour chaque application
4. Partage une partie de sa mémoire avec ses fils

### Le démarrage d'Android :

- On s'intéresse au processus une fois le lancement du kernel
- Le kernel initialise le processus Init
- Init lance à son tour des démons (usb, adb, ril), et le runtime
- Init lance aussi Zygote

### Zygote :

Zygote est un processus primordial pour Android :

- Il initialise la machine virtuelle
- Pré-charge des classes communes aux applications
- Se fork pour chaque nouvelle application

lancée

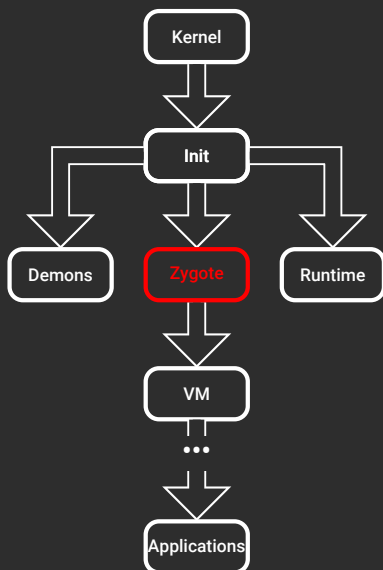
- Partage une partie de sa mémoire avec les applications forkées

### Fonctionnement de Xposed :

- Le processus init est modifié pour changer le comportement de Zygote en ajoutant des bibliothèques au classpath
- Ajout de bibliothèques à Zygote permettant de détecter le lancement d'application
- A chaque lancement d'une application, Zygote va remplacer le code de l'application pour injecter du code externe

## Élévation de privilèges : Xposed

• Élévation de privilèges



### Fonctionnement

FIGURE – Initialisation d'Android



### Le démarrage d'Android :

- On s'intéresse au processus une fois le lancement du kernel
- Le kernel initialise le processus Init
- Init lance à son tour des démons (usb, adb, ril), et le runtime
- Init lance aussi Zygote

### Zygote :

Zygote est un processus primordial pour Android :

- Il initialise la machine virtuelle
- Pré-charge des classes communes aux applications
- Se fork pour chaque nouvelle application

lancée

- Partage une partie de sa mémoire avec les applications forkées

### Fonctionnement de Xposed :

- Le processus init est modifié pour changer le comportement de Zygote en ajoutant des bibliothèques au classpath
- Ajout de bibliothèques à Zygote permettant de détecter le lancement d'application
- A chaque lancement d'une application, Zygote va remplacer le code de l'application pour injecter du code externe

## Élévation de privilèges : Xposed

• Élévation de privilèges

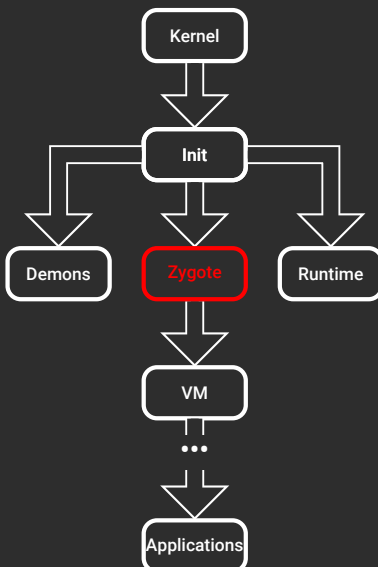


FIGURE – Initialisation d'Android

### Fonctionnement

1. Modification du processus init pour ajouter des bibliothèques au classpath

### Le démarrage d'Android :

- On s'intéresse au processus une fois le lancement du kernel
- Le kernel initialise le processus Init
- Init lance à son tour des démons (usb, adb, ril), et le runtime
- Init lance aussi Zygote

### Zygote :

Zygote est un processus primordial pour Android :

- Il initialise la machine virtuelle
- Pré-charge des classes communes aux applications
- Se fork pour chaque nouvelle application

lancée

- Partage une partie de sa mémoire avec les applications forkées

### Fonctionnement de Xposed :

- Le processus init est modifié pour changer le comportement de Zygote en ajoutant des bibliothèques au classpath
- Ajout de bibliothèques à Zygote permettant de détecter le lancement d'application
- A chaque lancement d'une application, Zygote va remplacer le code de l'application pour injecter du code externe

## Élévation de privilèges : Xposed

• Élévation de privilèges

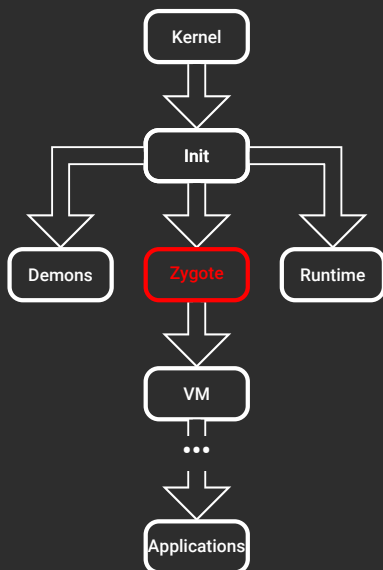


FIGURE – Initialisation d'Android

### Fonctionnement

1. Modification du processus init pour ajouter des bibliothèques au classpath
2. Ajout de bibliothèques à Zygote pour détecter le lancement d'applications

### Le démarrage d'Android :

- On s'intéresse au processus une fois le lancement du kernel
- Le kernel initialise le processus Init
- Init lance à son tour des démons (usb, adb, ril), et le runtime
- Init lance aussi Zygote

### Zygote :

Zygote est un processus primordial pour Android :

- Il initialise la machine virtuelle
- Pré-charge des classes communes aux applications
- Se fork pour chaque nouvelle application

lancée

- Partage une partie de sa mémoire avec les applications forkées

### Fonctionnement de Xposed :

- Le processus init est modifié pour changer le comportement de Zygote en ajoutant des bibliothèques au classpath
- Ajout de bibliothèques à Zygote permettant de détecter le lancement d'application
- A chaque lancement d'une application, Zygote va remplacer le code de l'application pour injecter du code externe

## Élévation de privilèges : Xposed

• Élévation de privilèges

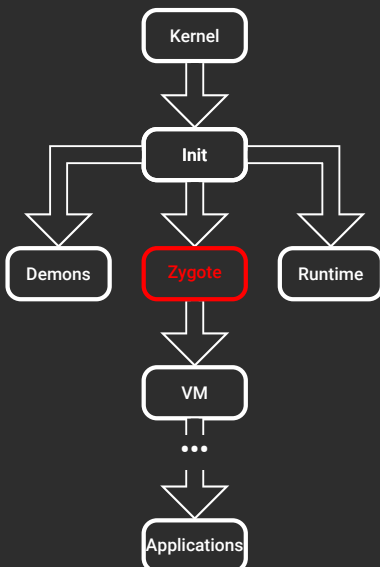


FIGURE – Initialisation d'Android

### Fonctionnement

1. Modification du processus init pour ajouter des bibliothèques au classpath
2. Ajout de bibliothèques à Zygote pour détecter le lancement d'applications
3. A chaque nouvelle application forkée de Zygote, il est possible de modifier le code exécuté par la VM