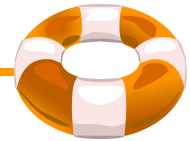
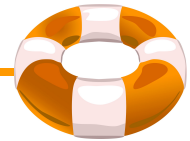


SOS - Laboratoire 3

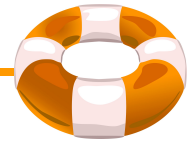
Permissions DAC et mécanismes de hardening

Rayane Annen, Hugo Ducommun, Alexis Martins
21.01.2024



Rappel : Niveaux de protection

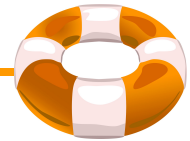
- Normales
  - Faibles risques
  - Accepté tacitement par l'utilisateur à l'**installation**
- Dangereuses
  - CIA sensible
  - Accepté par l'utilisateur à l'**exécution**
- Signées
  - Accorde la permission aux applications signées avec la même clé que la permission elle-même
  - Partage de fonctionnalités entre applications d'un même fournisseur



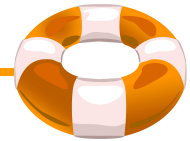
Rappel : Niveaux de protection

Via ADB, possibilité de voir les catégories de permissions :

```
# -d : Permissions dangereuses
# -g : Groupé par permission-group

$ adb shell pm list permissions -g -d

Dangerous Permissions:

group:com.google.android.gms.permission.CAR_INFORMATION
  permission:com.google.android.gms.permission.CAR_VENDOR_EXTENSION
  permission:com.google.android.gms.permission.CAR_MILEAGE
  permission:com.google.android.gms.permission.CAR_FUEL

group:android.permission-group.CONTACTS

group:android.permission-group.PHONE

group:com.example.permissiondemo.SOS_PERMISSION_GROUP
  permission:com.example.permissiondemo.SOS_PERMISSION

group:android.permission-group.CALENDAR

group:android.permission-group.CALL_LOG

group:android.permission-group.CAMERA
```

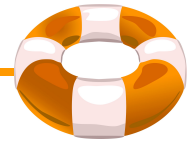


Permissions classiques



Permissions d’un package

```
$ adb shell dumpsys package com.example.permissiondemo

Permissions:
  Permission [com.example.permissiondemo.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMISSION] (95dcc92):
    sourcePackage=com.example.permissiondemo
    uid=10190 gids=[] type=0 prot=signature
    perm=PermissionInfo{c084b4a
com.example.permissiondemo.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMISSION}
    flags=0x0

    requested permissions:
      android.permission.VIBRATE
      android.permission.CAMERA
      com.example.permissiondemo.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMISSION
    install permissions:
      com.example.permissiondemo.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMISSION: granted=true
      android.permission.VIBRATE: granted=true

    runtime permissions:
        android.permission.CAMERA: granted=false, flags=[
USER_SET|USER_SENSITIVE_WHEN_GRANTED|USER_SENSITIVE_WHEN_DENIED]
```

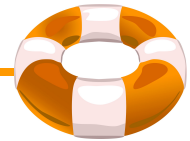


ADB : Modifier les permissions

Via ADB, possibilité de modifier les permissions accordées :

```
$ adb shell pm revoke com.example.permissiondemo android.permission.VIBRATE


Exception occurred while executing 'revoke':
java.lang.SecurityException: Permission android.permission.VIBRATE requested
by com.example.permissiondemo is not a changeable permission type


$ adb shell pm revoke com.example.permissiondemo android.permission.CAMERA


# Pas d'erreur -> C'est passé
```

Permissions personnalisées



Permissions personnalisées

- Déclaration dans le Manifest
  - Nom
  - Description
  - Niveau de protection
  - Groupe de permissions
    - Privilégier les groupes existants (bonnes pratiques)

```
$ adb shell pm list permissions -g -d

Dangerous Permissions:

…
group:com.example.permissiondemo.SOS_PERMISSION_GROUP
  permission:com.example.permissiondemo.SOS_PERMISSION
…
```

Exemple de permission personnalisée

AppA, AndroidManifest.xml:

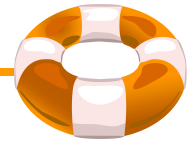
```
<permission android:name="com.example.myapp.ACCESS_ADMIN"
            android:label="@string/perm_admin_access"
            android:description="@string/perm_admin_description"
            android:protectionLevel="signature"/>
```

AppB, AndroidManifest.xml:

```
<uses-permission android:name="com.example.myapp.ACCESS_ADMIN"/>
```

AppB, MainActivity.kt:

```
if (checkCallingOrSelfPermission("com.example.myapp.ACCESS_ADMIN")
    == PackageManager.PERMISSION_GRANTED) {
    // Autoriser l'accès au module d'administration
} else {
    // Refuser l'accès
}
```



Hardening

Voir les mécanismes de hardening

Via ADB il est possible de voir les mécanismes activés :

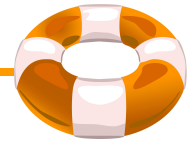
- cgroups: `adb shell cat /proc/<pid>/cgroup`
- Hardening activé : `adb shell cat /proc/<pid>/status`
- Politique seccomp (code) :
  https://android.googlesource.com/platform/bionic/+/master/libc/seccomp/seccomp_policy.cpp
- Blocklist par défaut de seccomp:
  https://github.com/aosp-mirror/platform_bionic/blob/master/libc/SECCOMP_BLOCKLIST_APP.TXT
  - Exemples : swapon, swapoff (désactiver / activer le swapping), chroot, etc.

Zygote

- Démarré quand le système boot, charge les librairies et frameworks communs (e.g. thème des activités).
- Chaque application est fork à partir de ce processus.
- Permet de partager la mémoire des frameworks et leurs ressources à toutes les apps.
- **La politique seccomp est appliquée à Zygote, ainsi tout fork (application) de ce processus héritera de cette politique.**

Sources :

- https://developer.android.com/topic/performance/memory-overview
- https://stackoverflow.com/questions/48802321/oreo-how-to-find-all-restricted-syscalls-at-source-code



Exploration de l’isolation de l’app

- Utilisateur lié à notre application :

```
$ adb shell ps -A | grep permissiondemo

USER          PID   PPID  VSZ         RSS         WCHAN ADDR  S     NAME
u0_a190       3382  360   13718888    44864       0     0     S     com.example.permissiondemo
```

- Le *working folder* de l’app est restreint à cet utilisateur :

```
$ adb shell ls -lisa /data/data/com.example.permissiondemo/

total 48
320138  8 drwx------   5 u0_a190 u0_a190        4096 2024-01-20 09:34 .
311297 16 drwxrwx--x 224 system   system       12288 2024-01-20 09:34 ..
320140  8 drwxrws--x   2 u0_a190 u0_a190_cache  4096 2024-01-20 09:34 cache
320141  8 drwxrws--x   2 u0_a190 u0_a190_cache  4096 2024-01-20 09:34 code_cache
320264  8 drwxrwx--x   2 u0_a190 u0_a190        4096 2024-01-20 09:34 files
```

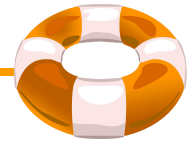
Exploration de l’isolation de l’app

- PID de notre application :

```
$ adb shell pidof com.example.permissiondemo

8382
```

- Le *working folder* de l’app est restreint à cet utilisateur :

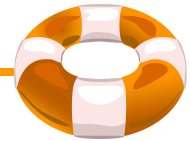
```
$ adb shell cat /proc/8382/cgroup

4:memory:/
3:cpuset:/top-app
2:cpu:/top-app
1:blkio:/
0::/uid_10190/pid_8382
```



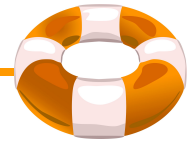
Exploration de l’isolation de l’app

- Statuts du processus :

```
$ adb shell cat /proc/8382/status

Name:  .permissiondemo
. . .
Pid:  8382
PPid:  360
TracerPid:  0
Uid:  10190  10190  10190  10190
Gid:  10190  10190  10190  10190
. . .
Threads:  19
. . .
Seccomp:  2
Seccomp_filters:  1
Speculation_Store_Bypass:  vulnerable
SpeculationIndirectBranch:  always enabled
Cpus_allowed:  f
Cpus_allowed_list:  0-3
Mems_allowed:  1
Mems_allowed_list:  0
voluntary_ctxt_switches:  301
nonvoluntary_ctxt_switches:  503
```



Demo



Références

Permissions : https://developer.android.com/guide/topics/permissions/overview?hl=fr

Hardening : https://source.android.com/docs/security/features?hl=fr

ChatGPT : https://chat.openai.com/

ADB Cheat Sheet : https://www.automatetheplanet.com/adb-cheat-sheet/#tab-con-11