ESUP-NFC-TAG-DROID

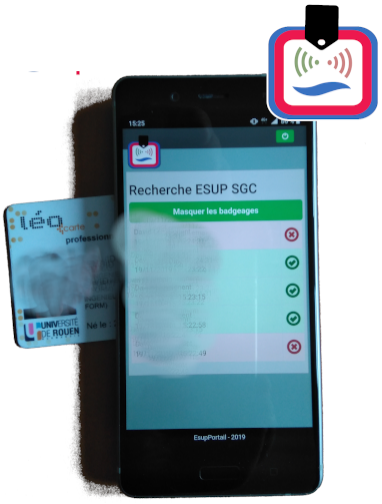
Esup-nfc-tag-droid permet de lire (et potentiellement encoder) les cartes Mifare Desfire.

Le client s'appuie sur la platefome https://github.com/EsupPortail/esup-nfc-tag-server qui calcule les commandes (APDU) à transmettre à la carte.

Esup-nfc-tag-droid permet d'utiliser un smartphone Android pour badger, en utilisant l'UID (CSN) ou en faisant une lecture d'un fichier Desfire (avec autentification AES)

L'application est packagée sous la forme d'un apk qu'il faudra installer sur un smartphone Android en autorisant les applciations de sources inconnues.

L'application peut être installée et debugée depuis Android-Studio ou complilée directement à l 'aide de Gradle



ESUP-NFC-TAG-DROID

https://esup-portail.org

Fonctionnalités

1 - L'application esup-nfc-tag-droid se comporte de la même manière que l'application Java [esup-nfc-tag-desktop] (https://github.com/EsupPortail/esup-nfc-tag-desktop "esup-nfc-tag-desktop")

2 - L'application repose sur un composant webview qui se connecte et affiche la vue fournie par esup-nfc-tag-server

3 - Après l'authentification Shibboleth il faut choisir la salle de badgeage

4 - Pour badger il suffit de poser une carte sur le lecteur nfc (à l'arrière du smartphone)

Environnement

Logiciel

L'application est prévue pour tourner sous Android 5 minimun

Materiel

Un smartphone Android équipé d'un lecteur NFC et disposant d'un accès Internet

APK officielle ESUP sur Google Play

Vous pouvez utiliser directement l'APK officielle et générique diffusée depuis Google Play par ESUP :

https://play.google.com/store/apps/details?id=org.esupportail.esupnfctagdroid

Si vous êtes responsable d'un serveur esup-nfc-tag d'un établissement de l'ESR que vous souhaitez voir apparaître votre serveur esup-nfc-tag dans la liste des serveurs disponibles depuis cette version esup-nfc-tag-droid fournie sur le google play, vous pouvez en faire la demande au travers d'un Pull Request proposant la modification du listing des urls disponibles :
https://github.com/EsupPortail/esup-nfc-tag-droid/blob/master/src/main/assets/urls

Sources

https://github.com/EsupPortail/esup-nfc-tag-droid

```
git clone https://github.com/EsupPortail/esup-nfc-tag-droid.git
```

Génération de l'APK

Si vous souhaitez proposer votre propre ESUP-NFC-TAG-DROID pointant directement sur votre serveur esup-nfc-tag, il vous faut construire votre propre APK.

Vous pouvez le faire de 3 manières :

1. en utilisant Android Studio **(https://developer.android.com/studio) ; c'est la manière la plus simple à mettre en oeuvre. C'est la méthode recommandée.**
2. en ligne de commande depuis un linux

Génération de l'APK en ligne de commandes

Pré-requis

- Android Studio 2
- Android SDK API level 28

Android SDK API level 28

En dehors d'android studio, la récupération du SDK d'Android se fait via le sdk-tools.

https://developer.android.com/studio/#command-tools

On le télécharge et on le dézippe en tant qu'utilisateur (esup ici pour nous) qui se chargera de compiler/packager l'applciation android.

Le sdk-tools est donc disponible ici : /home/esup/sdk-android-tools

On installe mainetant le sdk d'android , level 28 :

```
[esup@carbonne bin]$ ./sdkmanager 'platforms;android-28'
[esup@carbonne bin]$ ./sdkmanager 'build-tools;28.0.3'
```

On accepte au passage la licence proposée.

Le SDK et outils associés ont ainsi été installés dans le home d'esup.

On pourra positionner ANDROID_HOME ainsi dans le .bashrc de l'utilisateur esup :

```
[esup@carbonne ~]$ echo 'export ANDROID_HOME=/home/esup' >>  ~/.bashrc
```

Autres ...

Si à la compilation (cf ci-dessous compilation via gradlew clean assemble), vous trouvez des erreurs types

```
java.io.IOException: Cannot run program "/home/esup/build-tools/22.0.1/aapt": error=2, Aucun fichier ou dossier
de ce type
```

Il vous manque sans doute des librairies.

En exécutant /home/esup/build-tools/22.0.1/aapt directement on voit quelle librairie il cherche.

Ensuite un

```
yum whatprovides 'libz.so.1'
```

nous donne le paquet à installer avec yum install.

```
[root@carbonne ~]# yum install libstdc++* zlib*
```

Des librairies 32 bits seront à installer notamment.

Compilation esup-nfc-tag-droid

- esup-nfc-tag-droid génère des logs à destination d'un fichier de logs local au téléphone, à destination d'esupNfcTagServer (envoi de logs par POST au serveur) et à destination d'une adresse mail système. Les éléments paramétrables (mail systeme, serveur esupNfcTagServer) sont à configurer dans ce fichier src/main/assets/logback.xml
- modifier src/main/assets/esupnfctag.properties pour spécifier l'adresse de votre esup-nfc-tag-server
- Vous pouvez spécifier les paramètres de signature de votre APK dans build.gradle, si vous ne souhaitez pas utiliser ceux donnés par défaut (connus de tous). Vous devrez alors créer un keystore :

```
keytool -genkey -v -keystore esup-android-apps.keystore -alias LeoDroidApp -keyalg RSA -keysize 2048 -validity
10000
```

- build de l'APK

```
gradlew clean assemble
```

Integration dans esup-nfc-tag-server

- copier l'APK dans EsupNfcTagServer pour le mettre à disposition des utilisateurs :

```
cp ./build/outputs/apk/release/esup-nfc-tag-droid-release.apk /<path to>/esup-nfc-tag-server/src/main/resources
/apk/esupnfctagdroid.apk
```

- recompiler et redéployer esup-nfc-tag-server. Au redémarrage d'esup-nfc-tag-server la nouvelle version de l'apk sera prise en compte