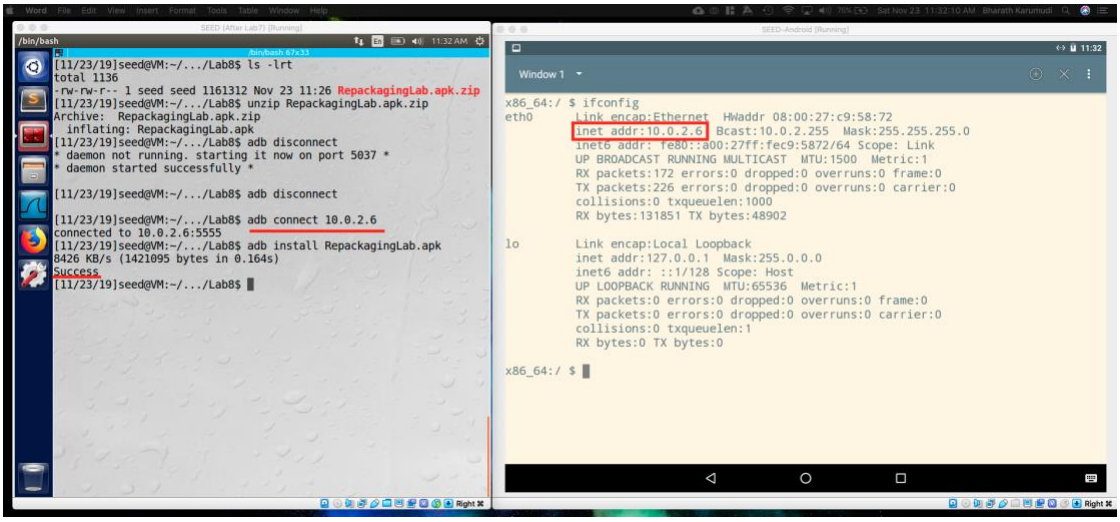
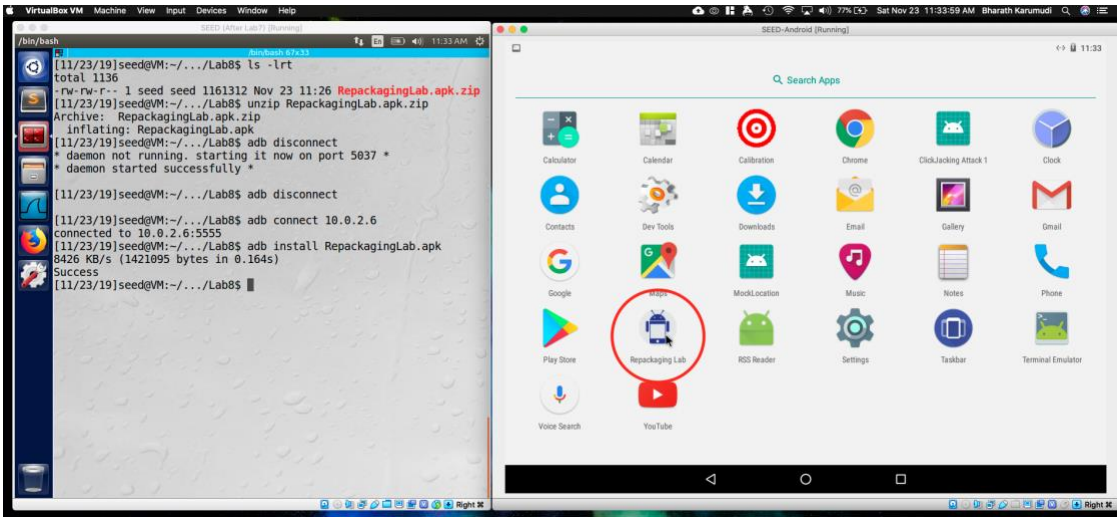
Name: Bharath Karumudi
Lab: Android Repackaging Attack

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

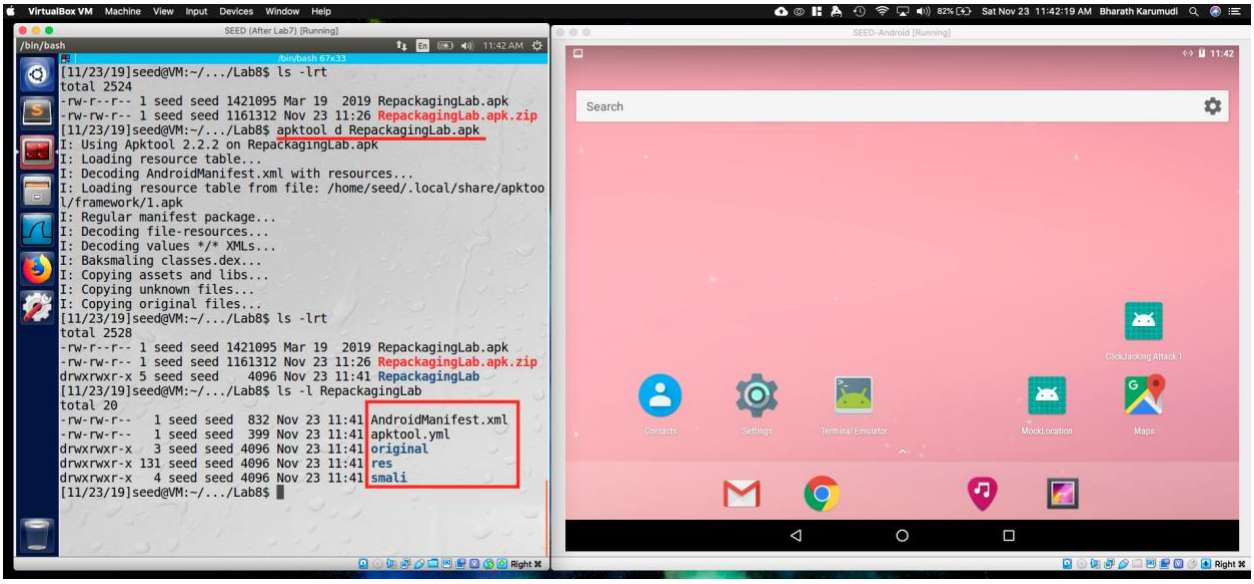




Observation: Downloaded the “RepackagingLab.apk.zip” from lab site and using adb connected to Android VM and installed the apk file. After installation, the “Repackaging Lab” can be seen in the Android applications menu.

Explanation: Using terminal emulator, found the IP address of the Android VM and it is 10.0.2.6. With the help of adb connected to the Android VM on the IP and ran the install command using the RepackagingLab.apk file. The connect and installation was successful and so the app can be seen in the menu.

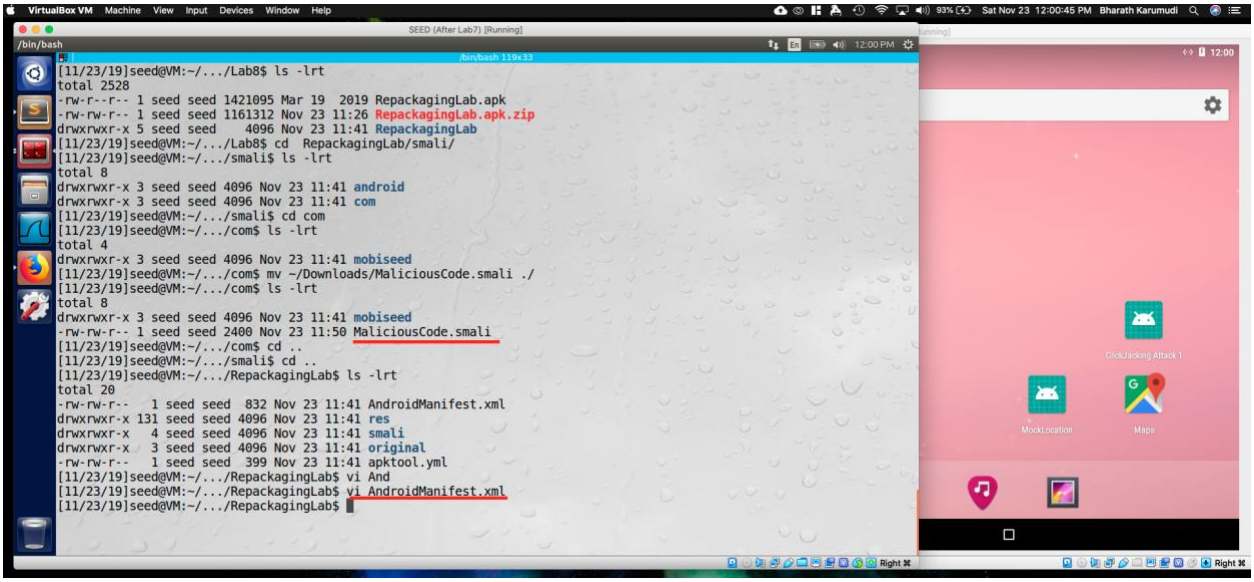
Task 2: Disassemble Android App

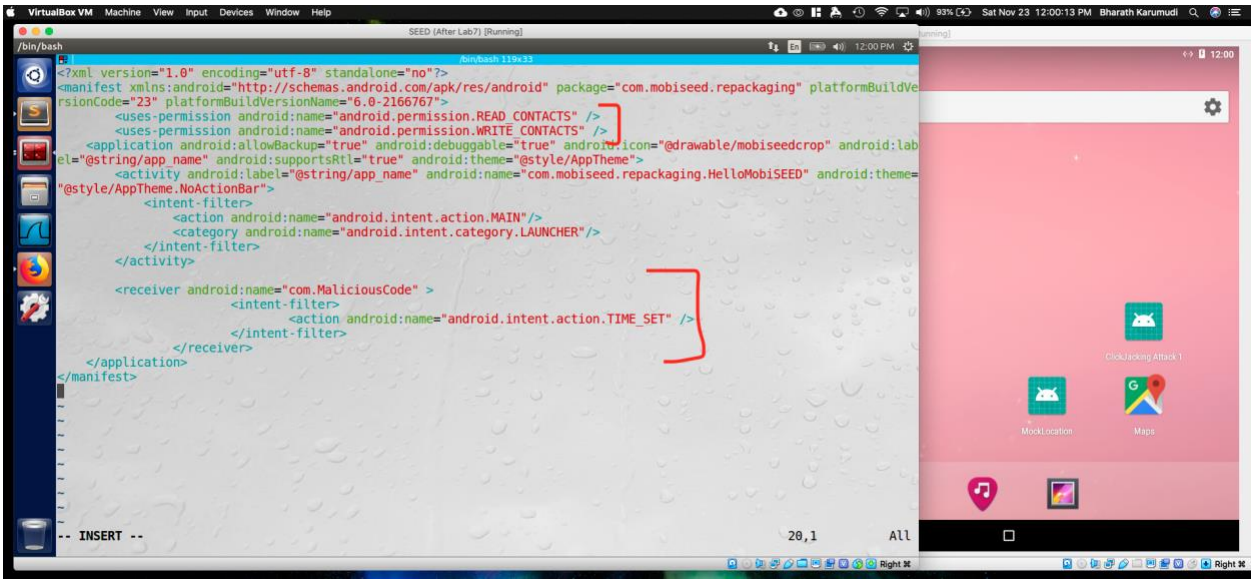


Observation: Disassembled the android app using the *apktool* and a new directory with files has been created.

Explanation: apktool helps in doing the reverse engineering and with this disassembled the apk file and can be used for further attacks.

Task 3: Inject Malicious Code



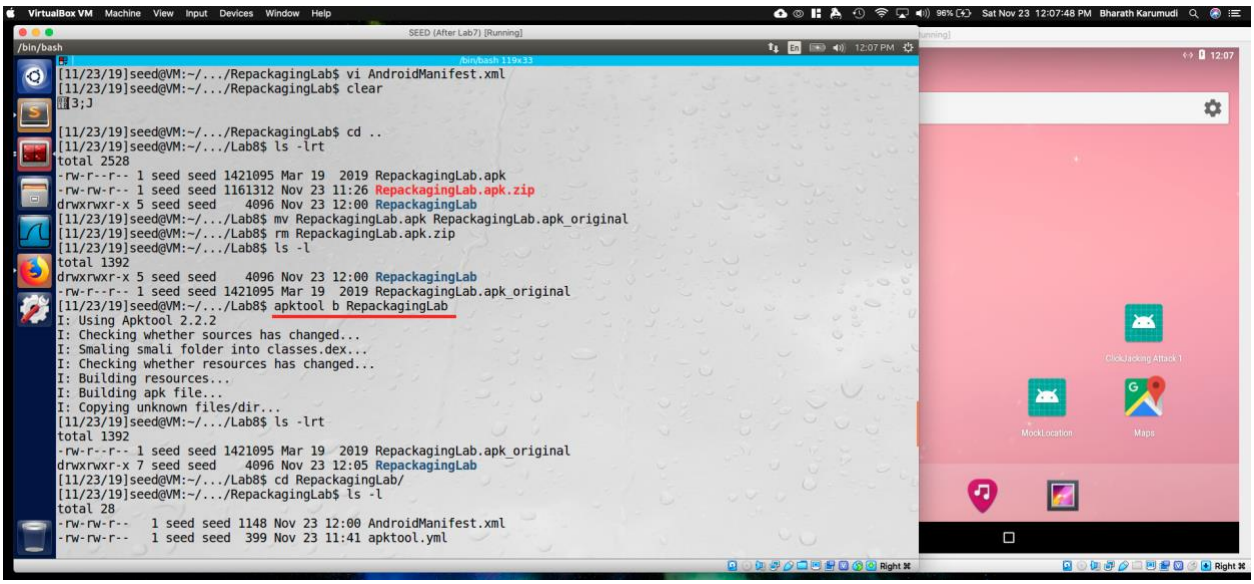


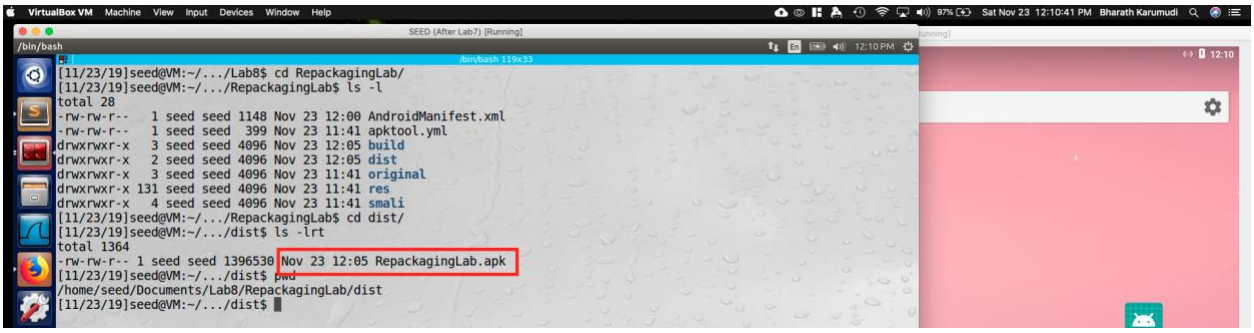
Observation: Added the “MaliciousCode.smali” to the smali/ directory and edited the AndroidManifest.xml file to add the permissions and to register the receiver for broadcast.

Explanation: The malicious code file has been injected to the application, without disturbing its functionality and the AndroidManifest.xml was modified to get the malicious module triggered with time was changed. Thus, when a user modifies the time, the malicious code will get executed, as it is subscribed to the TIME_SET broadcast.

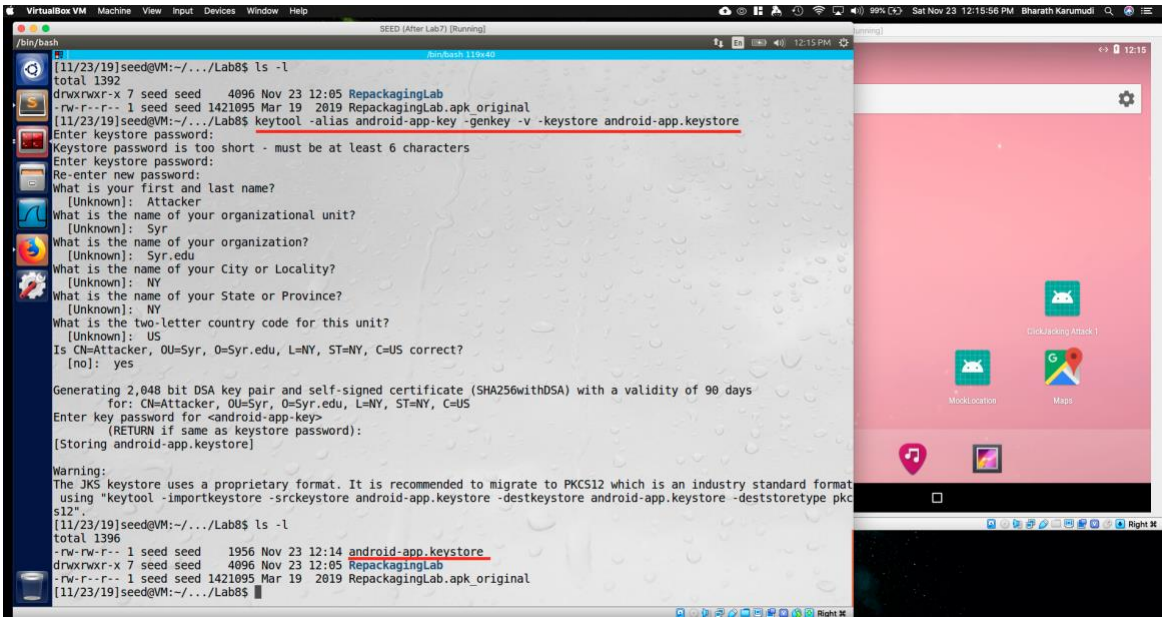
Task 4: Repack Android App with Malicious Code

Step 1:



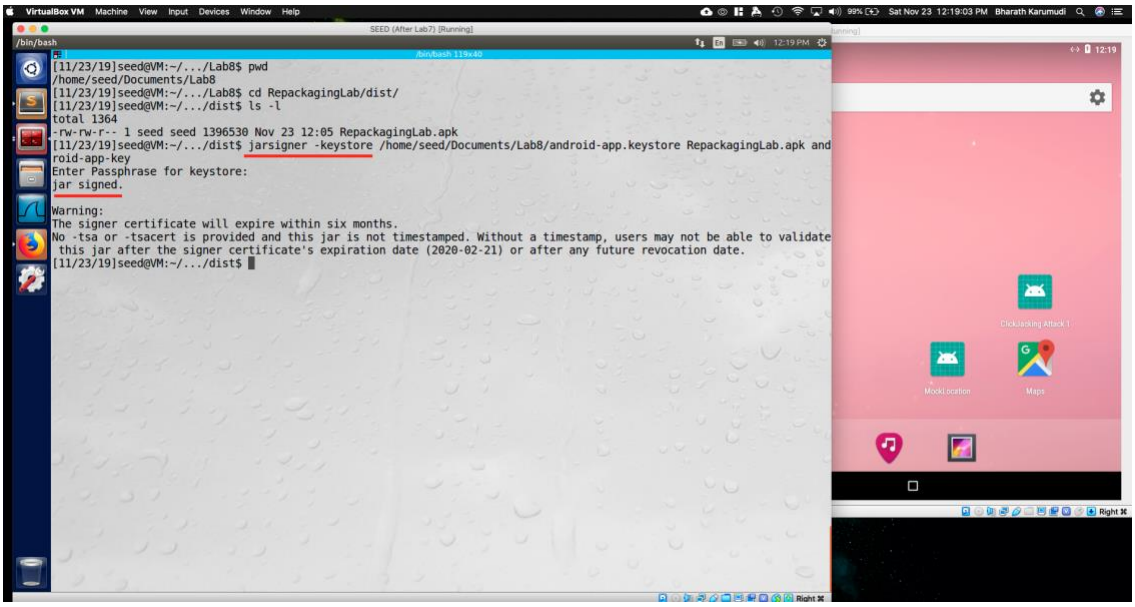

[11/23/19]seed@VM:~/.../Lab8$ cd RepackagingLab/
[11/23/19]seed@VM:~/.../RepackagingLab$ ls -l
total 28
-rw-rw-r-- 1 seed seed 1148 Nov 23 12:00 AndroidManifest.xml
-rw-rw-r-- 1 seed seed 399 Nov 23 11:41 apktool.yml
drwxrwxr-x 3 seed seed 4096 Nov 23 12:05 build
drwxrwxr-x 2 seed seed 4096 Nov 23 12:05 dist
drwxrwxr-x 3 seed seed 4096 Nov 23 11:41 original
drwxrwxr-x 131 seed seed 4096 Nov 23 11:41 res
drwxrwxr-x 4 seed seed 4096 Nov 23 11:41 smali
[11/23/19]seed@VM:~/.../RepackagingLab$ cd dist/
[11/23/19]seed@VM:~/.../dist$ ls -lrt
total 1364
-rw-rw-r-- 1 seed seed 1396530 Nov 23 12:05 RepackagingLab.apk
[11/23/19]seed@VM:~/.../dist$ pwd
/home/seed/Documents/Lab8/RepackagingLab/dist
[11/23/19]seed@VM:~/.../dist$

Step 2:


[11/23/19]seed@VM:~/.../Lab8$ ls -l
total 1392
drwxrwxr-x 7 seed seed   4096 Nov 23 12:05 RepackagingLab
-rw-r--r-- 1 seed seed 1421095 Mar 19 2019 RepackagingLab.apk_original
[11/23/19]seed@VM:~/.../Lab8$ keytool -alias android-app-key -genkey -v -keystore android-app.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]: Attacker
What is the name of your organizational unit?
  [Unknown]: Syr
What is the name of your organization?
  [Unknown]: Syr.edu
What is the name of your City or Locality?
  [Unknown]: NY
What is the name of your State or Province?
  [Unknown]: NY
What is the two-letter country code for this unit?
  [Unknown]: US
Is CN=Attacker, OU=Syr, O=Syr.edu, L=NY, ST=NY, C=US correct?
  [no]: yes

Generating 2,048 bit DSA key pair and self-signed certificate (SHA256withDSA) with a validity of 90 days
        for: CN=Attacker, OU=Syr, O=Syr.edu, L=NY, ST=NY, C=US
Enter key password for <android-app-key>
        (RETURN if same as keystore password):
[Storing android-app.keystore]

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format
 using "keytool -importkeystore -srckeystore android-app.keystore -destkeystore android-app.keystore -deststoretype pkc
s12".
[11/23/19]seed@VM:~/.../Lab8$ ls -l
total 1396
-rw-rw-r-- 1 seed seed   1956 Nov 23 12:14 android-app.keystore
drwxrwxr-x 7 seed seed   4096 Nov 23 12:05 RepackagingLab
-rw-r--r-- 1 seed seed 1421095 Mar 19 2019 RepackagingLab.apk_original
[11/23/19]seed@VM:~/.../Lab8$


[11/23/19]seed@VM:~/.../Lab8$ pwd
/home/seed/Documents/Lab8
[11/23/19]seed@VM:~/.../Lab8$ cd RepackagingLab/dist/
[11/23/19]seed@VM:~/.../dist$ ls -l
total 1364
-rw-rw-r-- 1 seed seed 1396530 Nov 23 12:05 RepackagingLab.apk
[11/23/19]seed@VM:~/.../dist$ jarsigner -keystore /home/seed/Documents/Lab8/android-app.keystore RepackagingLab.apk and
roid-app-key
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 timestamp, users may not be able to validate
 this jar after the signer certificate's expiration date (2020-02-21) or after any future revocation date.
[11/23/19]seed@VM:~/.../dist$

Observation: The modified application has been rebuilt to a new apk file using apktool and then signed the apk file with the key, using keytool and jarsigner utilities.

Explanation: apktool helps in building the files into the apkfile and as per Android specs, the apk file has to be signed. So followed the below steps to get the new signed apk file.

1. Built the apk using apktool b RepackagingLab
2. As I don't have a keystore, created a new keystore with the name android-app.keystore
3. Signed the new apk file using jarsigner utility and with my key.

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

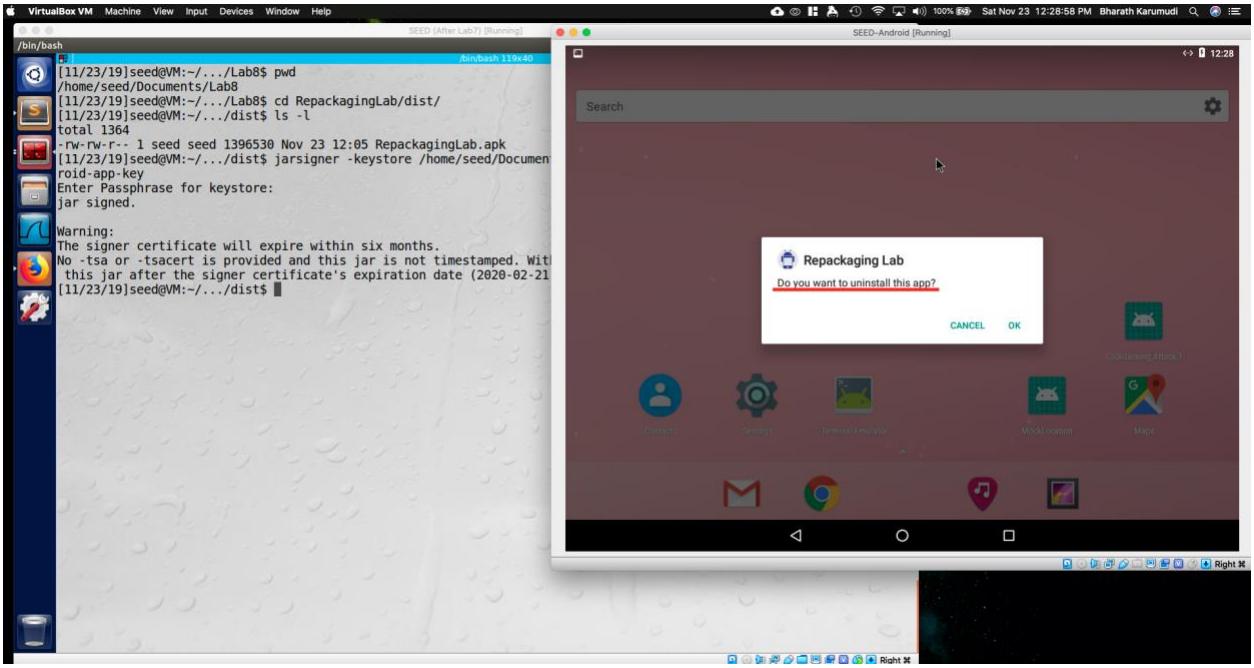


Fig: Uninstalling the existing app

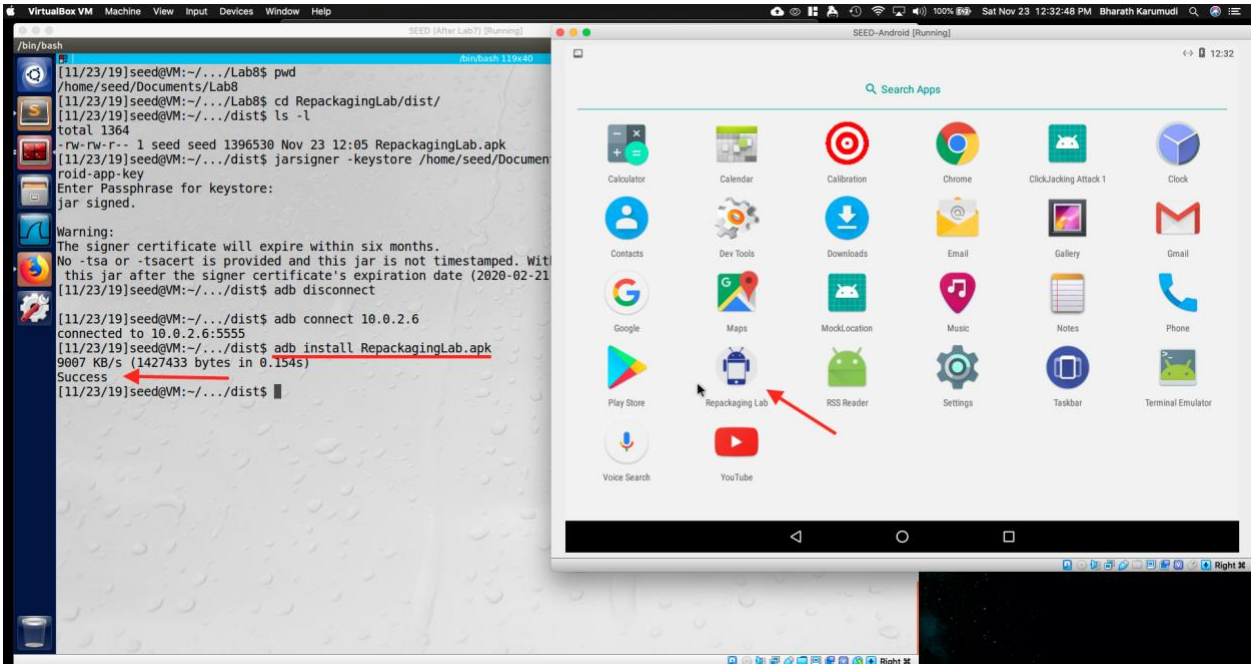


Fig: Installed the malicious app

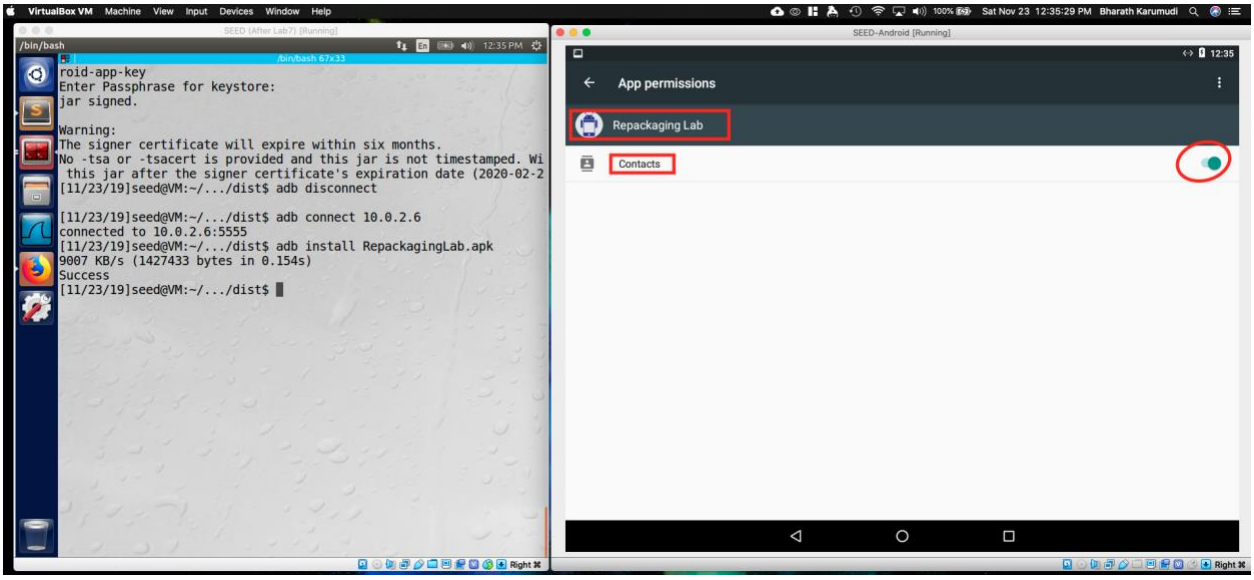


Fig: Granted the Contacts permissions to malicious app

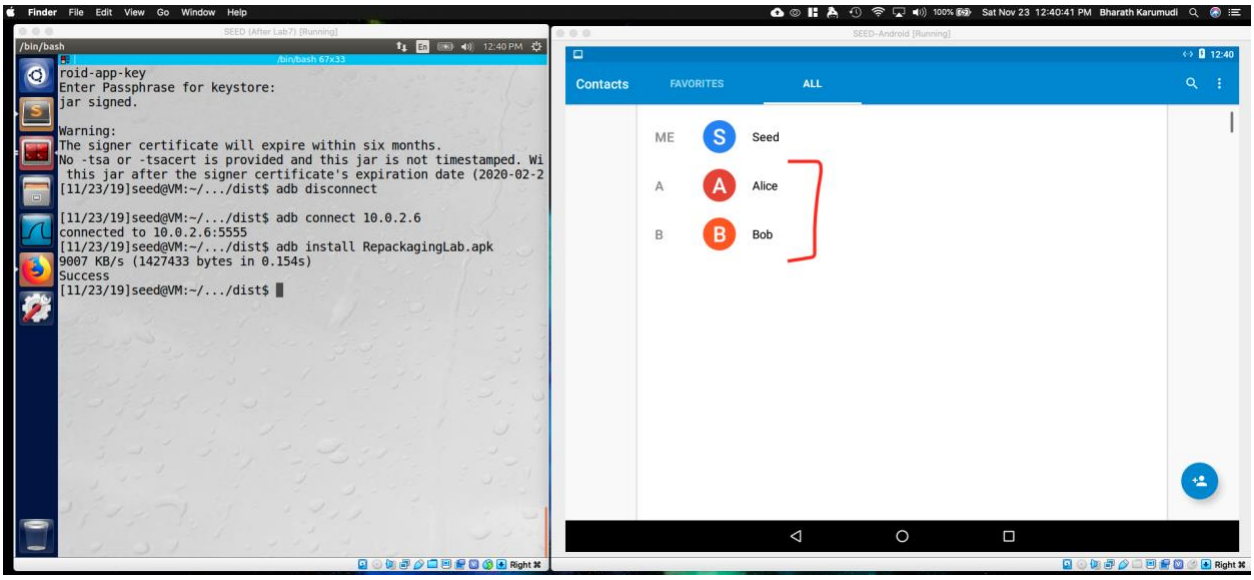


Fig: Added three contacts to the “Contacts”.

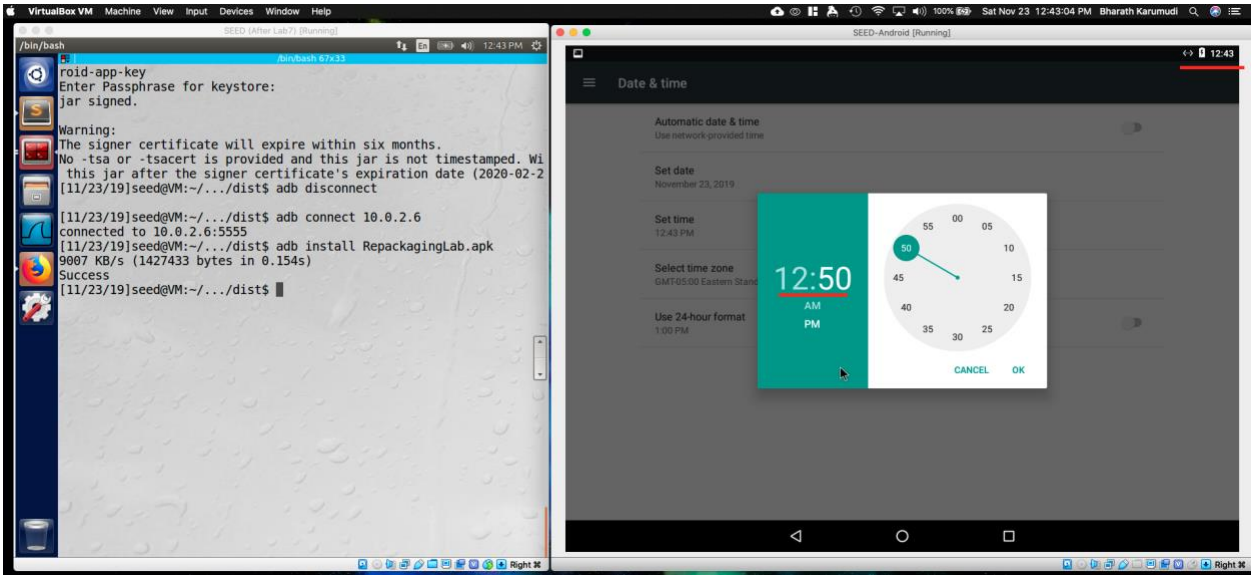


Fig: Modified the time (to 12:50 from 12:43)

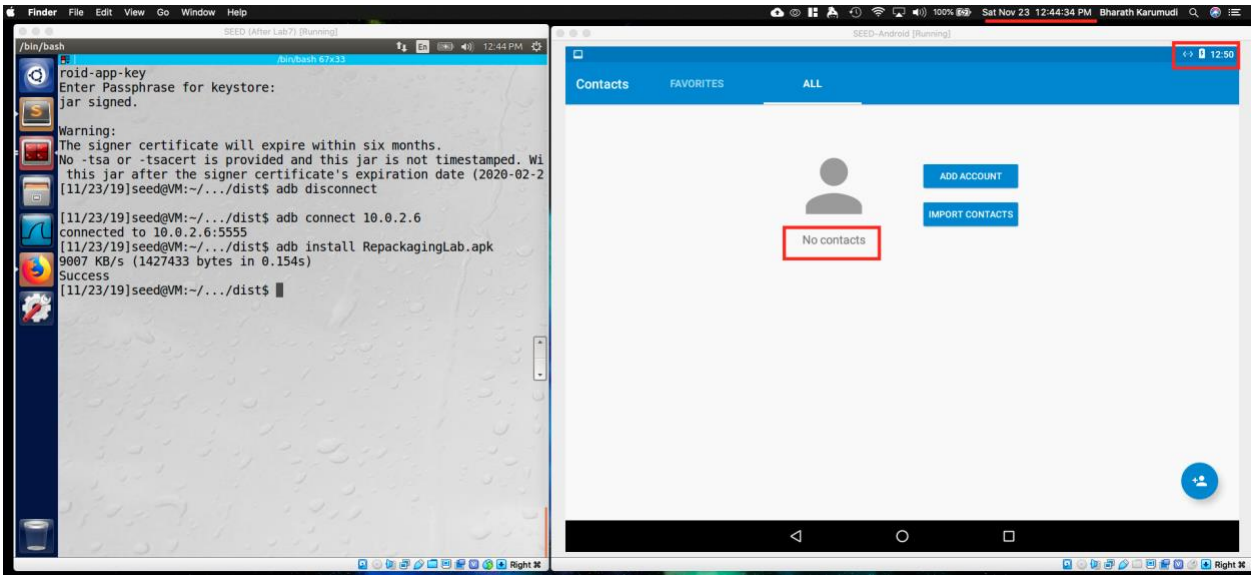


Fig: Zero Contacts – Attack successful

Observation: Installed the malicious app to the Android and granted the permissions on the "Contacts" to the malicious app. Added the contacts to the "Contacts" and when changed the time and once came back to the Contacts app, there were no contacts. Attack was successful.

Explanation: Using adb connected to the Android VM and installed the malicious "RepackagingLab.apk" which we built. The application manifest has the permissions defined to use the Contacts Read and Write. So toggled the permissions to the app and the app is also registered to the broadcast for change in time. When contacts are added and changed the time, the broadcast was sent that time was changed, so the app executed the contacts remove and all the contacts are removed from the device.

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

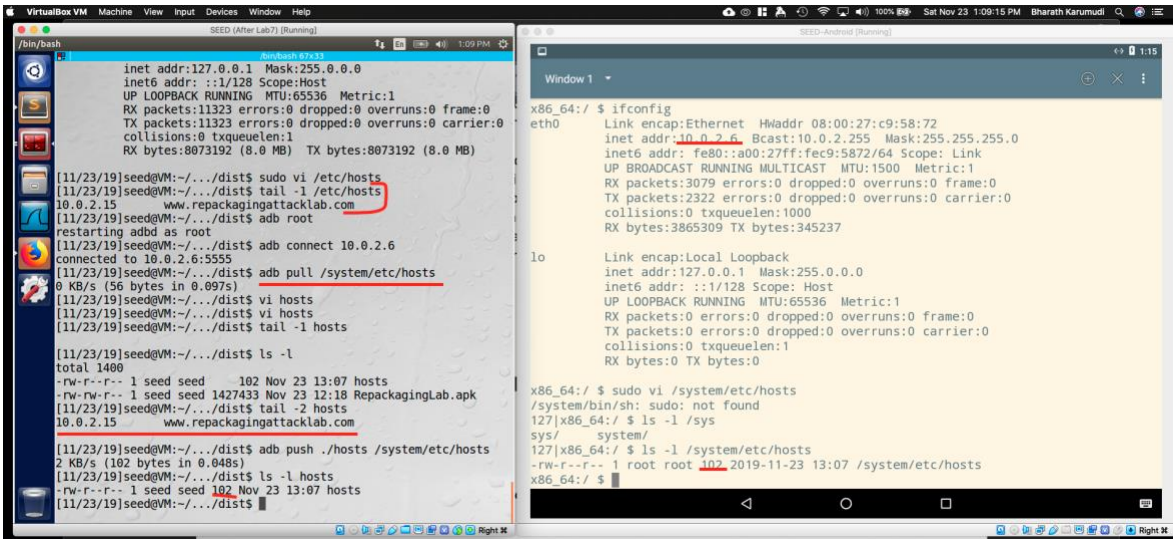


Fig: Step 2 - Modified the hosts file in Android VM.

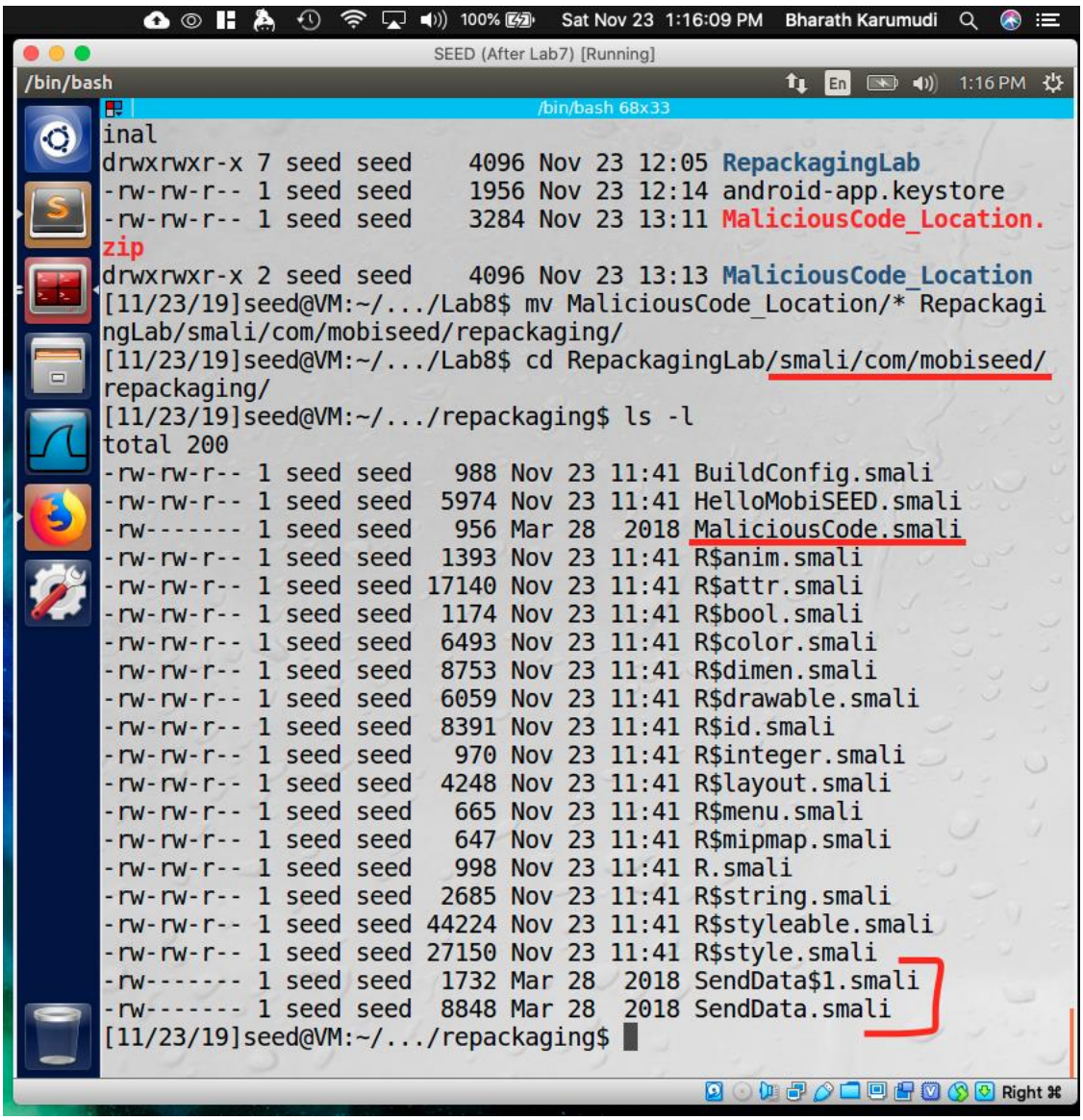


Fig: Step 3a - Added the three malicious code files to the package

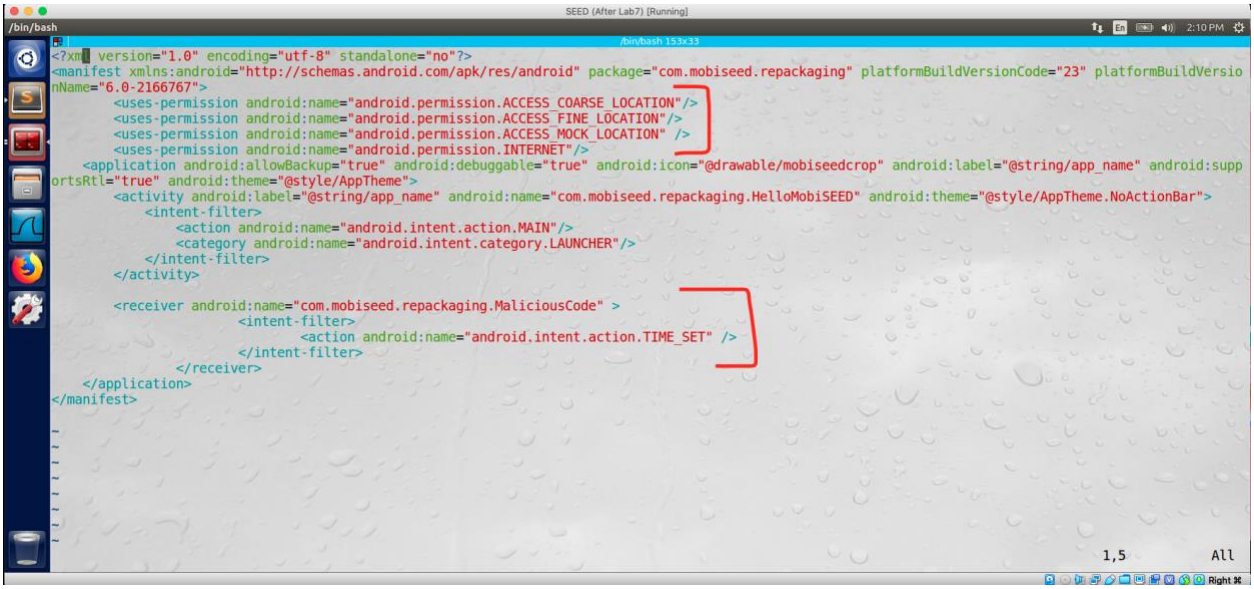


Fig: Step 3b - Modified the Manifest file for location access and broadcast receiver

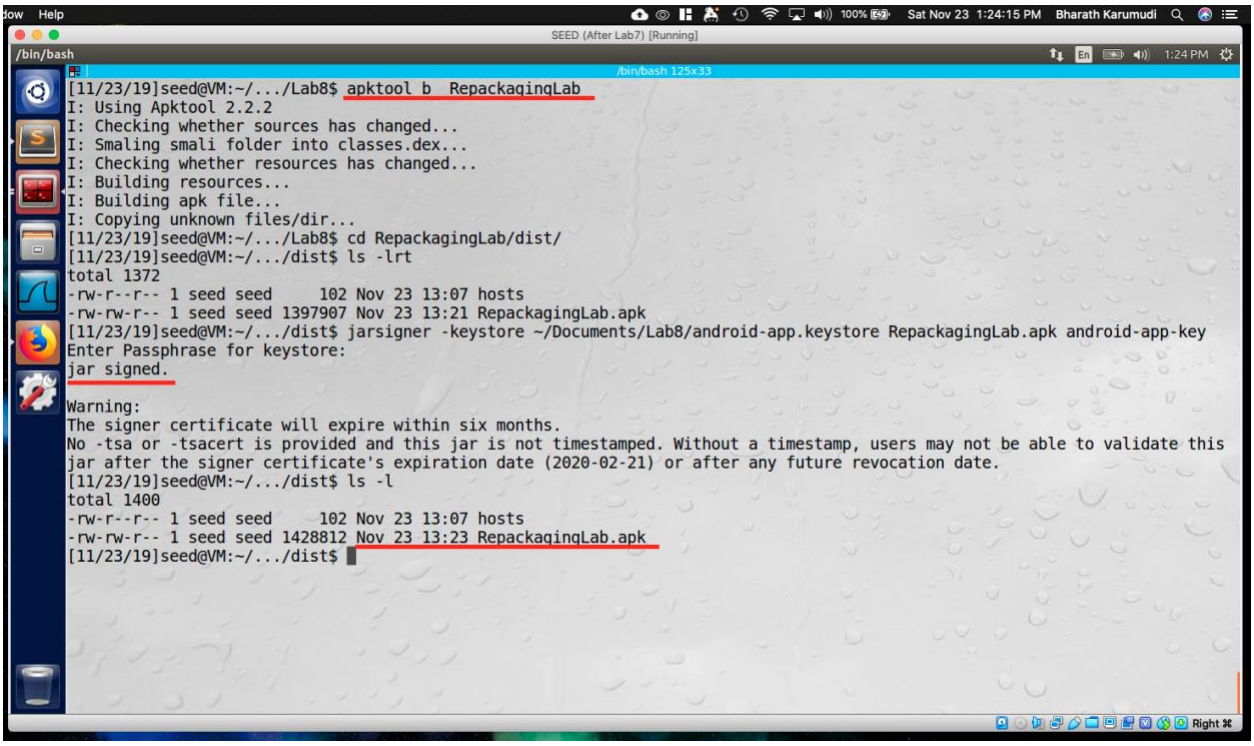


Fig: Step 3c- Build and Signed the malicious apk file

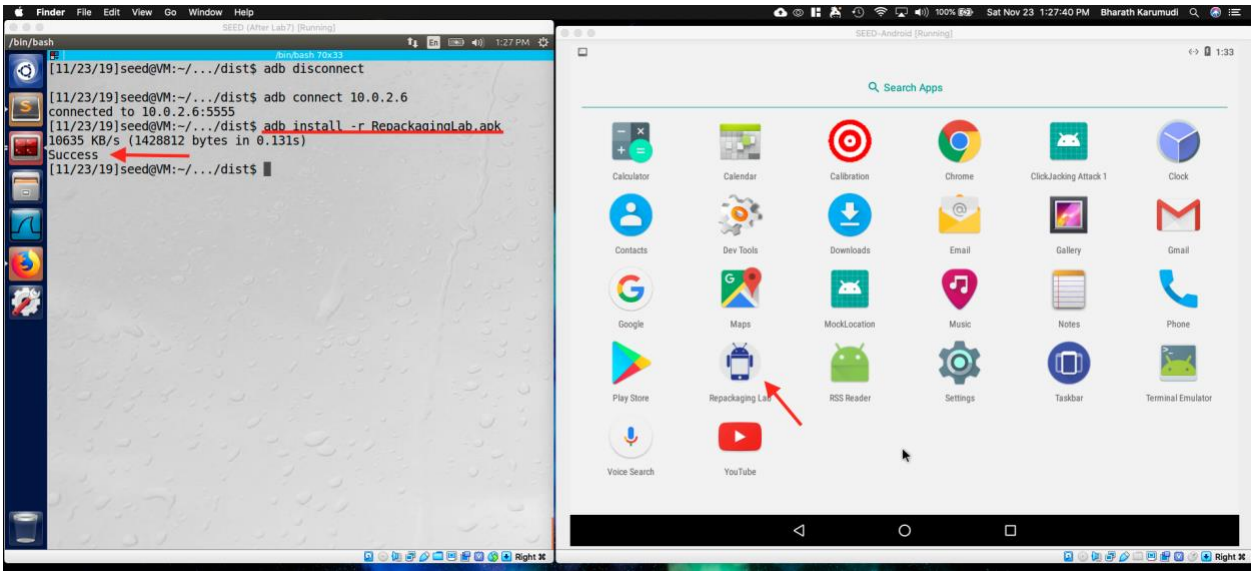


Fig: Step 3d - Installed the app on Android

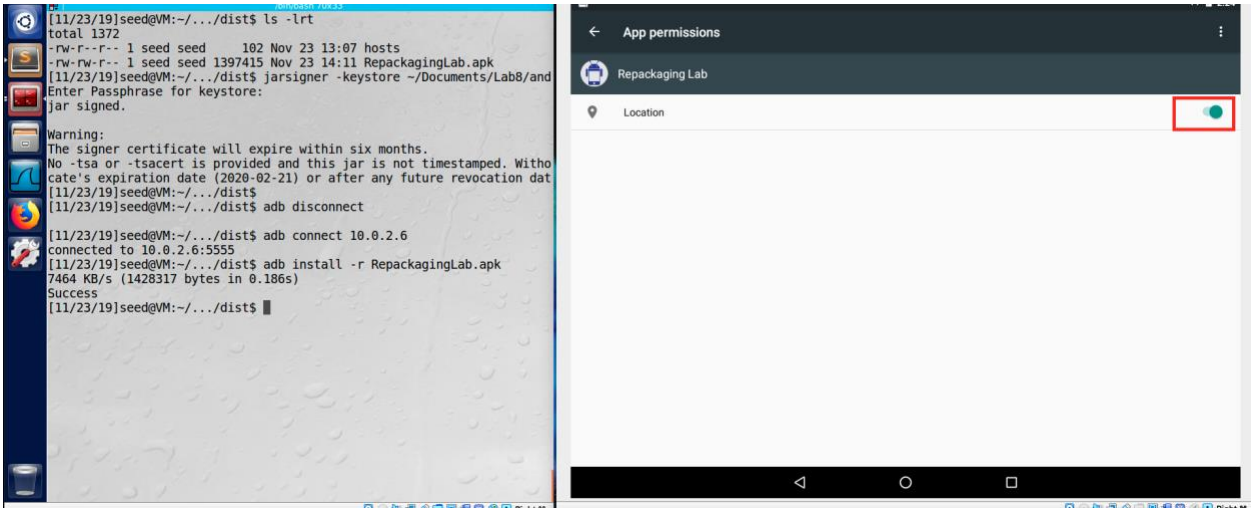


Fig: Step 4 - Granted the Location access to the app

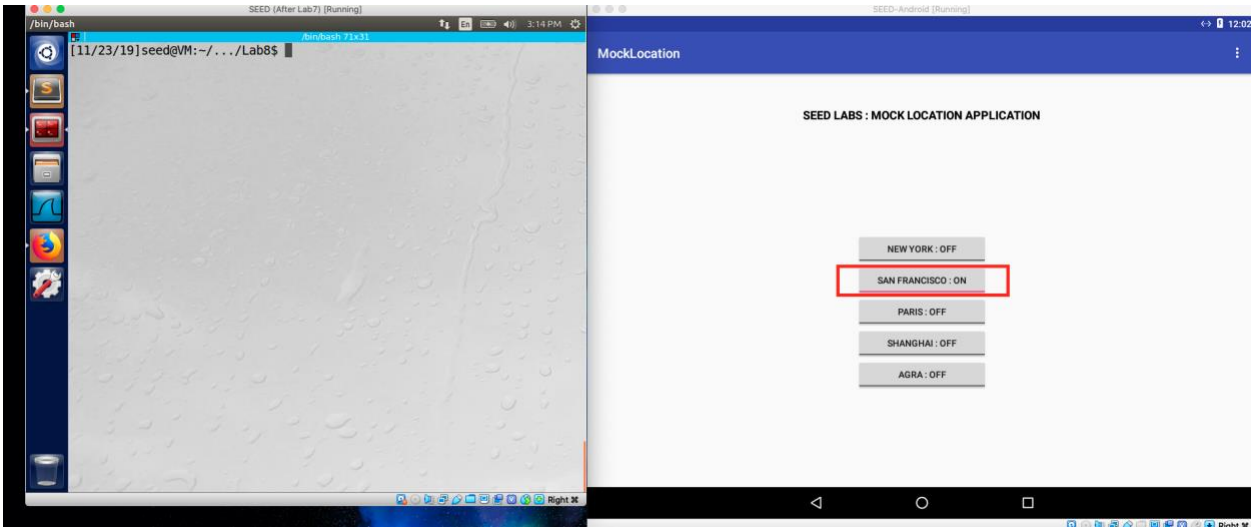


Fig: Step 5a - Device Location set to “San Francisco”

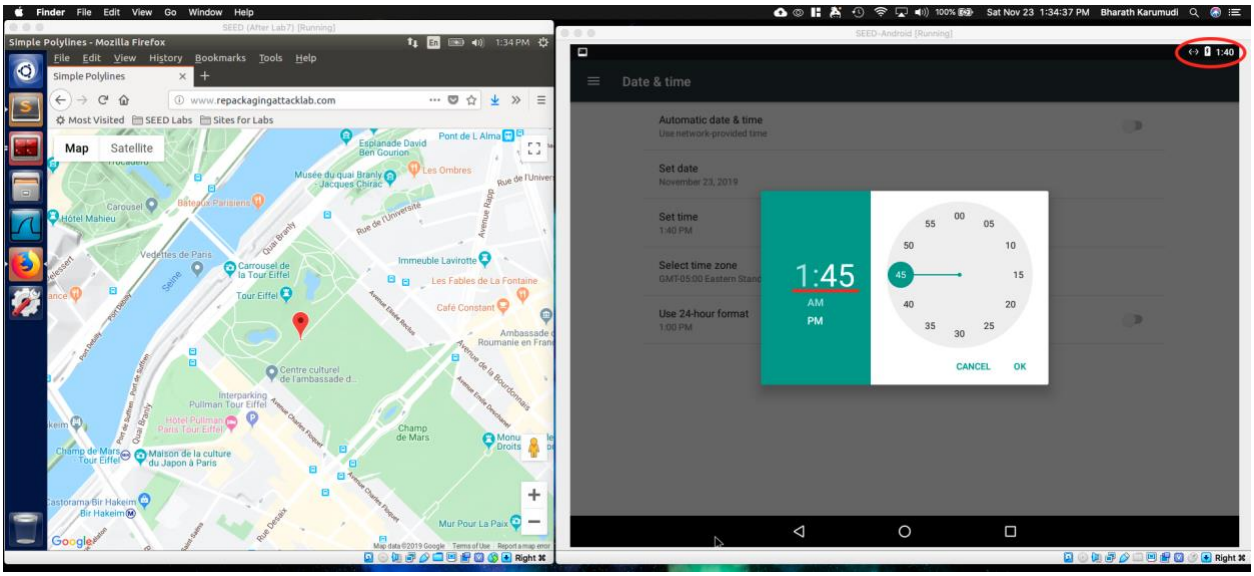


Fig: Step 5b - Changing time to perform the attack and default location is in Europe

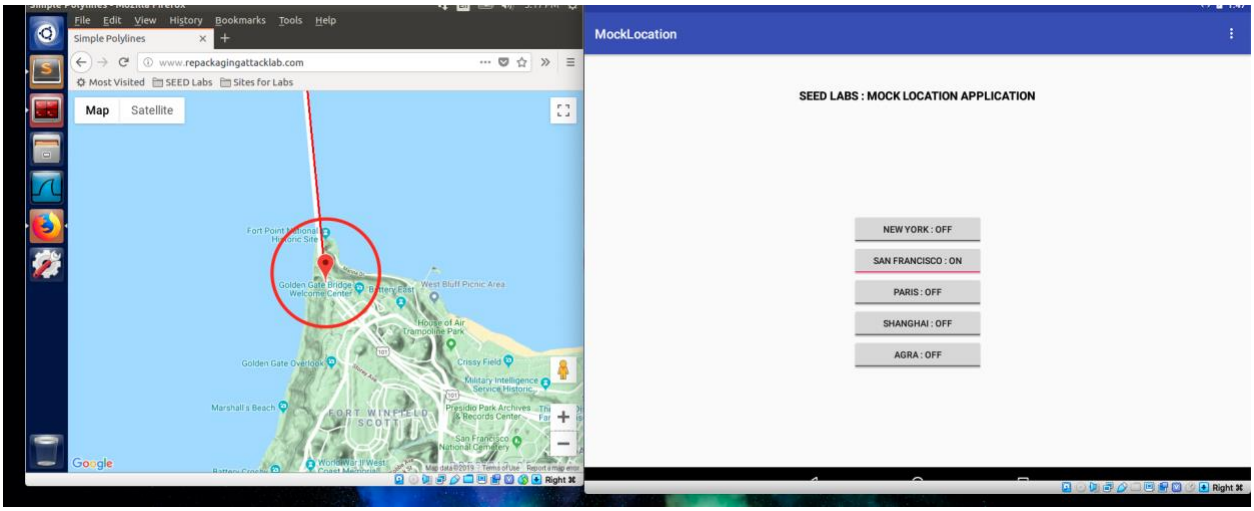


Fig: Step 5c -Location updated to San Francisco

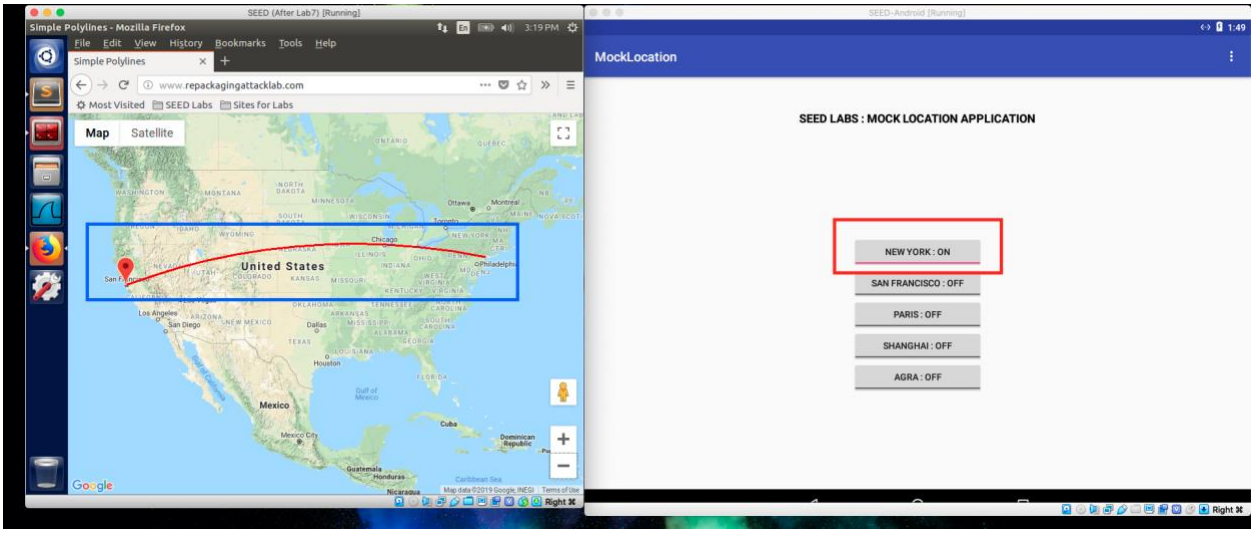


Fig: Step 6 - Location changed to New York and Updated in map

Observation: Configured the Android VM hosts file to reach the Ubuntu VM for "www.repackagingattacklab.com", added the malicious code files and modified the Android Manifest file. Later, built the apk file and signed the apk file. The apk file was installed on the Android and granted the location preferences. The location was set to San Francisco and once the time was changed, the location was updated in the www.repackagingattacklab.com website. The location then changed to "New York" and the location was updated in the web page and victim location is tracked.

Explanation: A malicious code was introduced into the package and updated the manifest file for location access and broadcast receiver when time was changed. Once changed, repackaged the apk file and signed with the key and installed on the victim Android.

On the Android, location access has been granted to the app and using the Mockup location app, selected the San Francisco. Once the time was changed on the Android, the device sent the broadcast to the subscriber – malicious app module, the malicious code was executed, which sends the location of the device to the attacker.
To track the victim location, location changed this time to “New York” and the location was updated on the attacker side and able to track the victim location.