MOBILE PENETRATION TESTING

DEKSHINAMURTHY MEENAKSHI
Sstudy2016@gmail.com

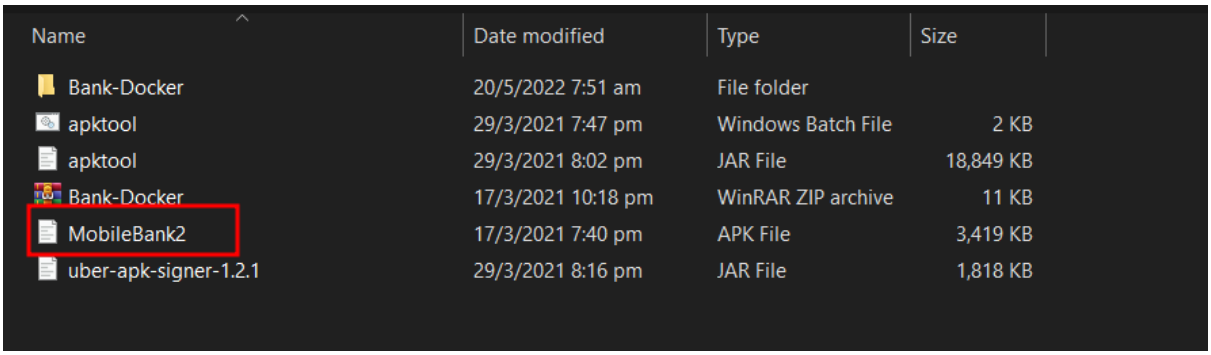
1 Table of Contents

1. SETUP

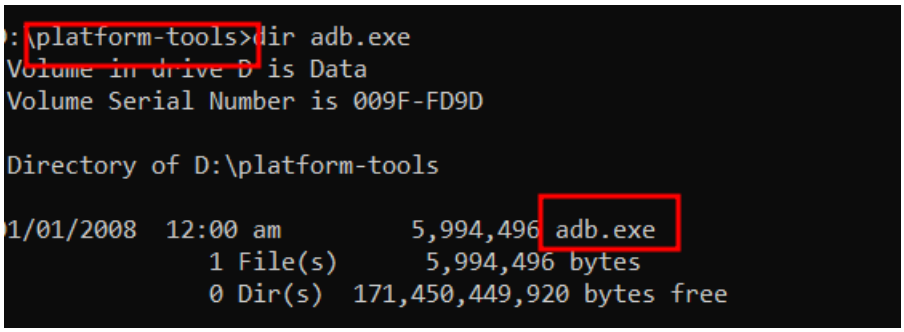
1.1 Setup Genymotion

Download the project file.

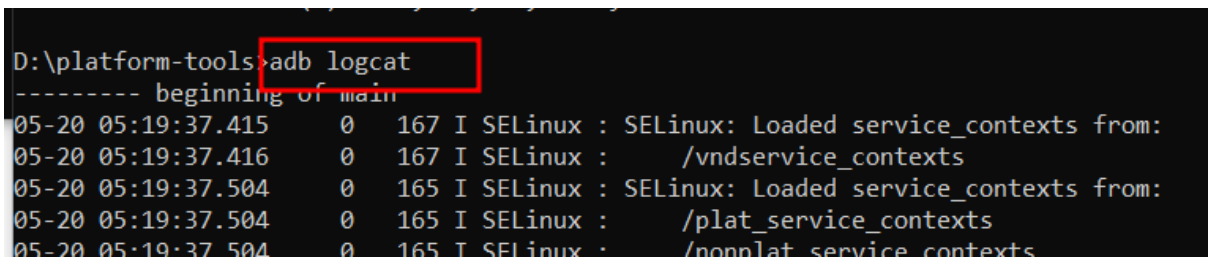
Extract the MobileBank.rar in Host



Check if ADB is installed



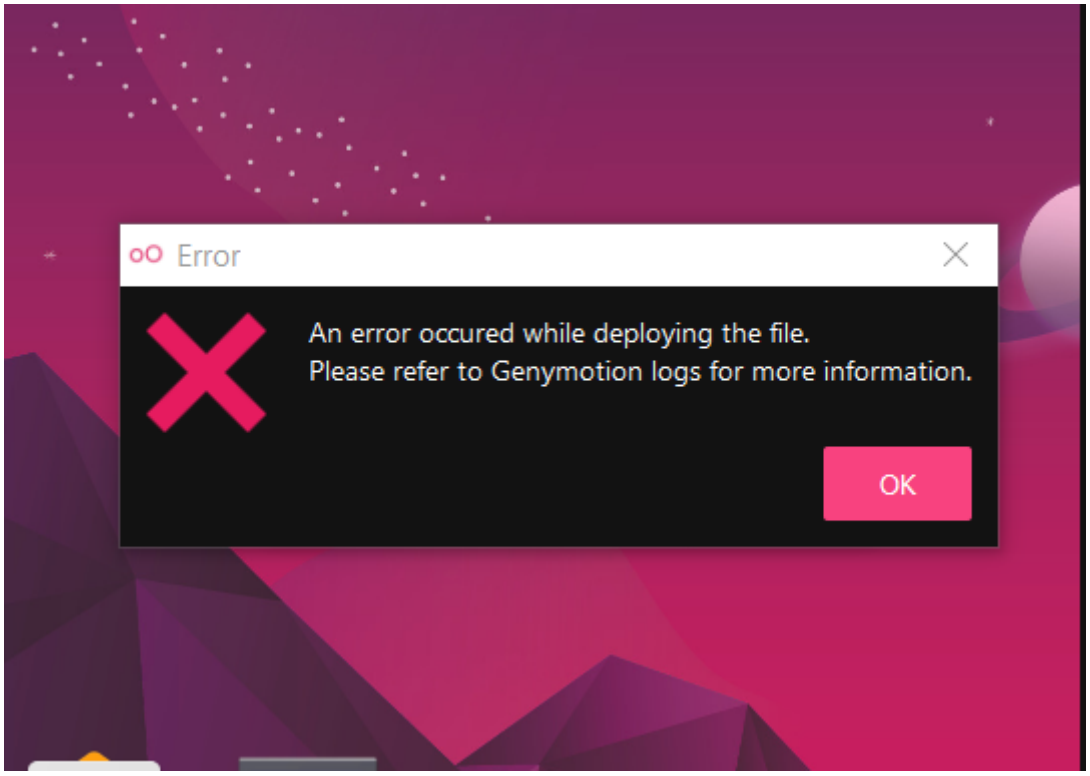
Start adb logcat to catch the log messages/ errors



Copy the MobilebankApk to the genymotion emulator by dragging from the host windows explorer to the emulator

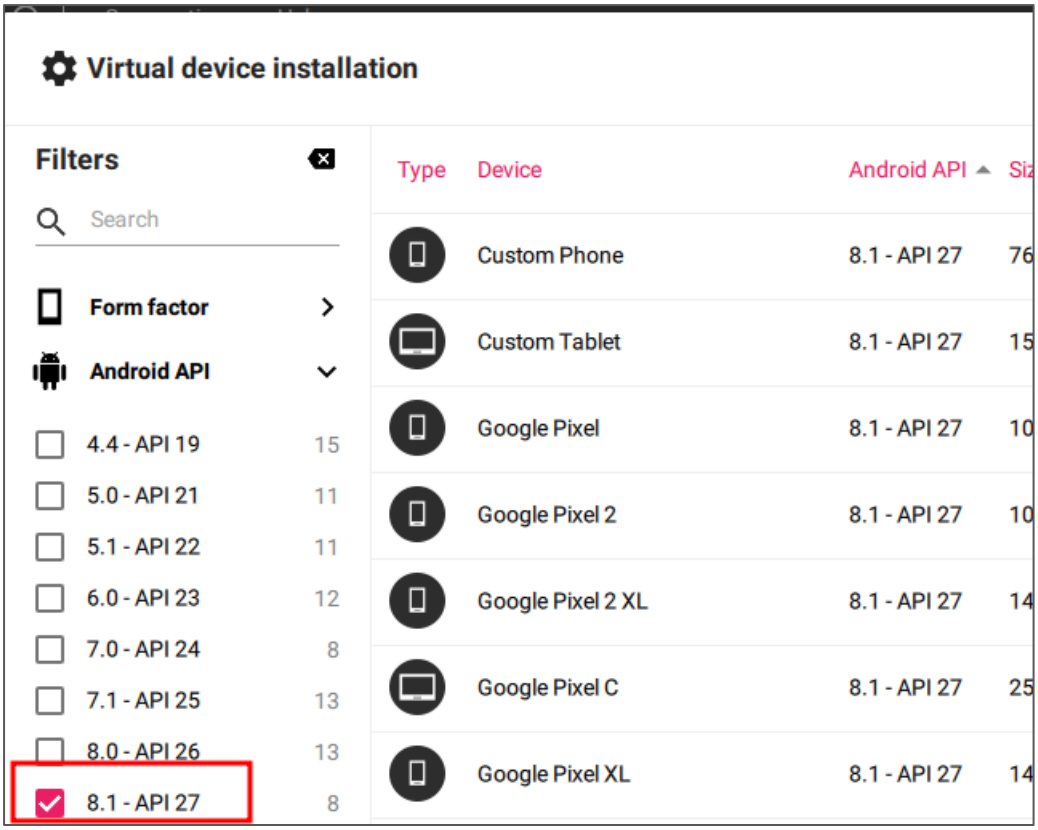
The current version in 26 so, the log message shows err.

Install a new device with api 27

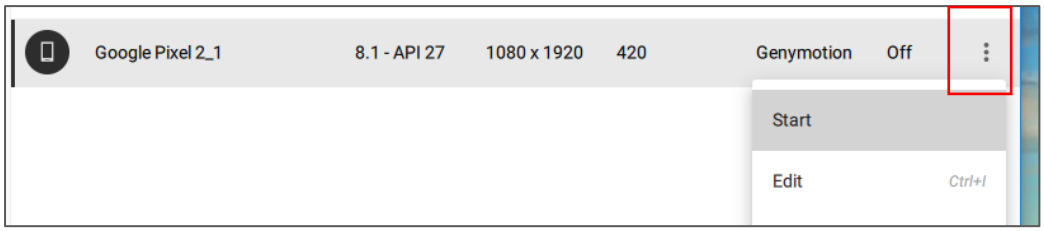



```
context=u:r:system_server:s0 tclass=binder permissive=1
05-20 06:14:23.499   509   542 W PackageManager: Failed parse during installPackageLI
05-20 06:14:23.499   509   542 W PackageManager: android.content.pm.PackageParser$PackageParserException: /data/app/vm
061744663.tmp/base.apk (at Binary XML file line #0): Requires newer sdk version #27 (current version is #26)
05-20 06:14:23.499   509   542 W PackageManager:        at android.content.pm.PackageParser.parseBaseApk(PackageParser
ava:1265)
05-20 06:14:23.499   509   542 W PackageManager:        at android.content.pm.PackageParser.parseClusterPackage(Packag
```

Click on the “+” sign to add a virtual device and Choose Api27

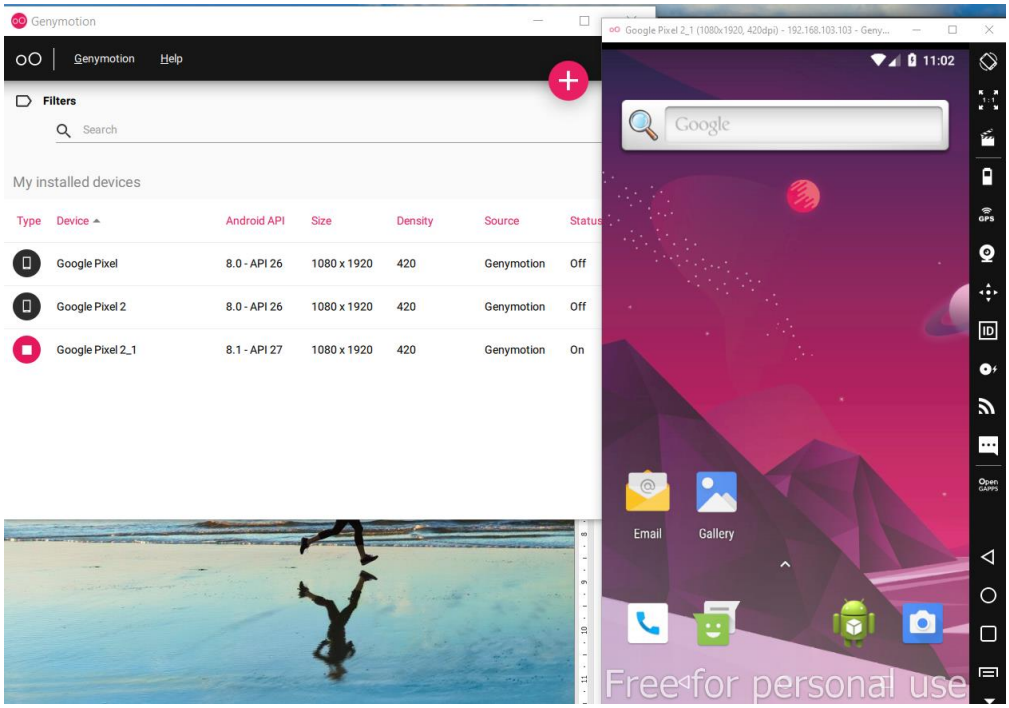
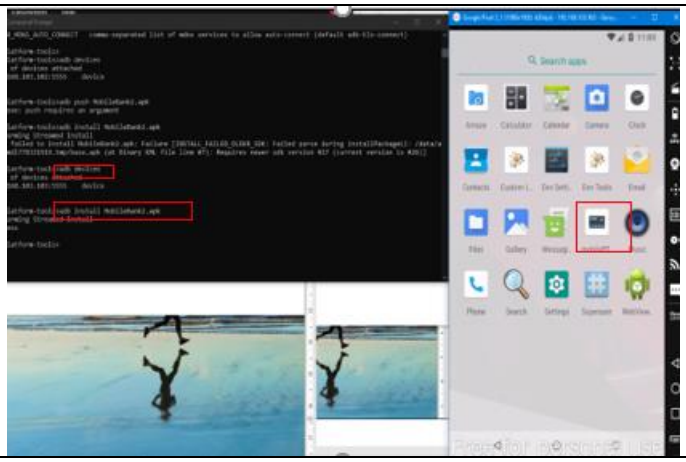


Choose device and Click “Next” to install.

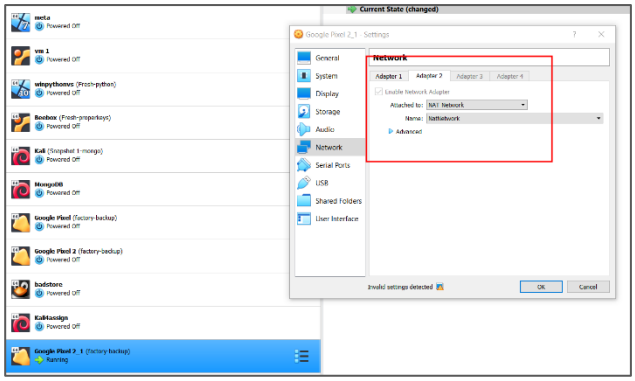


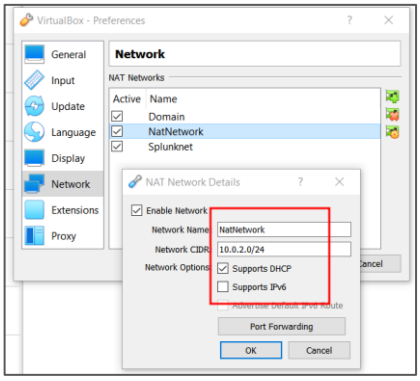
The device is listed upon install.

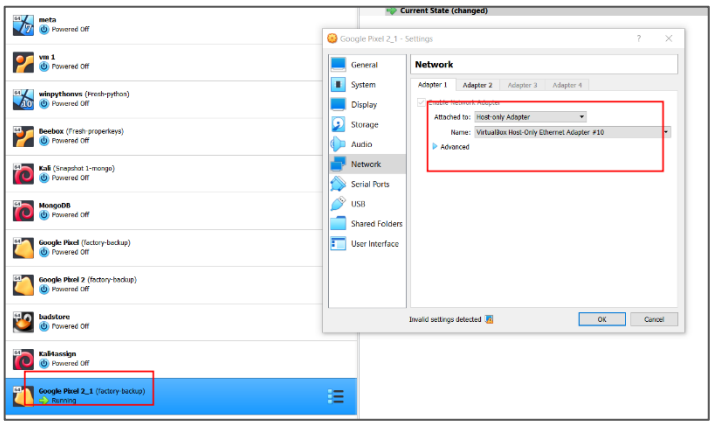
Start it by clicking on the three dots as shown below

	
	
	The Device is installed and running , connected through adb We can see the device listed by adb device command and the MobileBankApi2 is installed by adb install MobilBank2.apk command
	The mobile bank app is seen in the emulator as shown

Set the Emulator network configuration. The first adapter in bridged mode with host. The second adapter in NAtNetwork



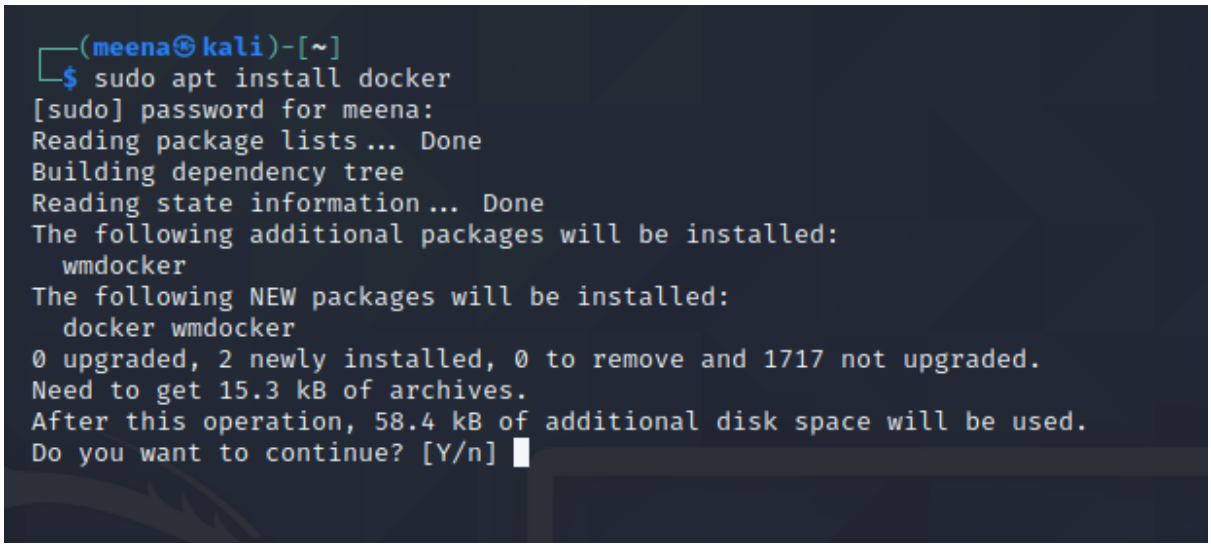




1.2 Kali Setup

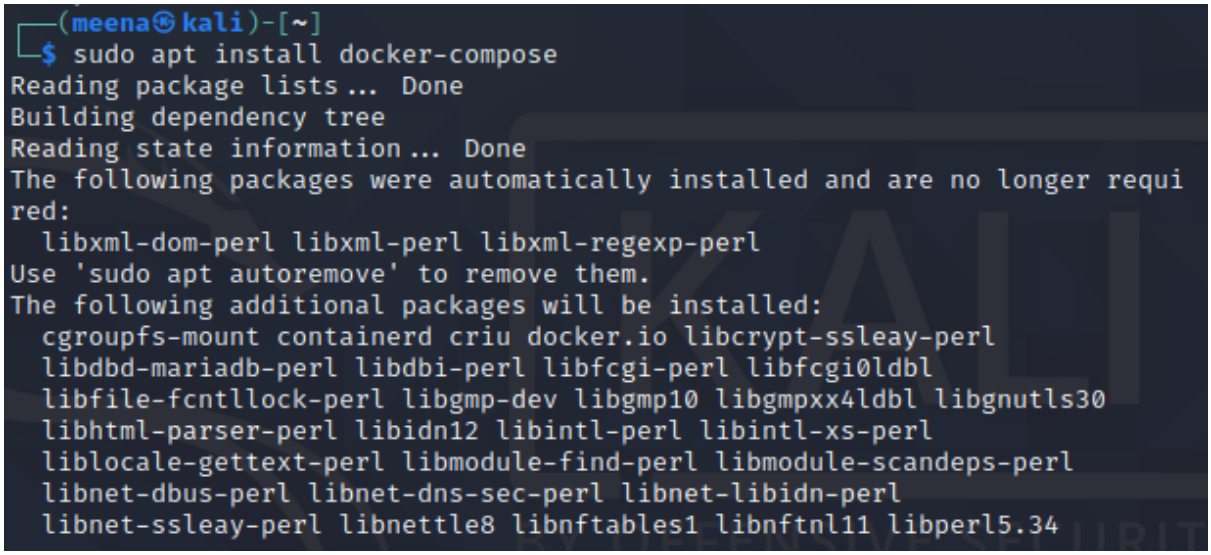
1.2.1 Install Docker:

Docker is an open source containerization platform. It enables developers to package applications into containers—standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment.



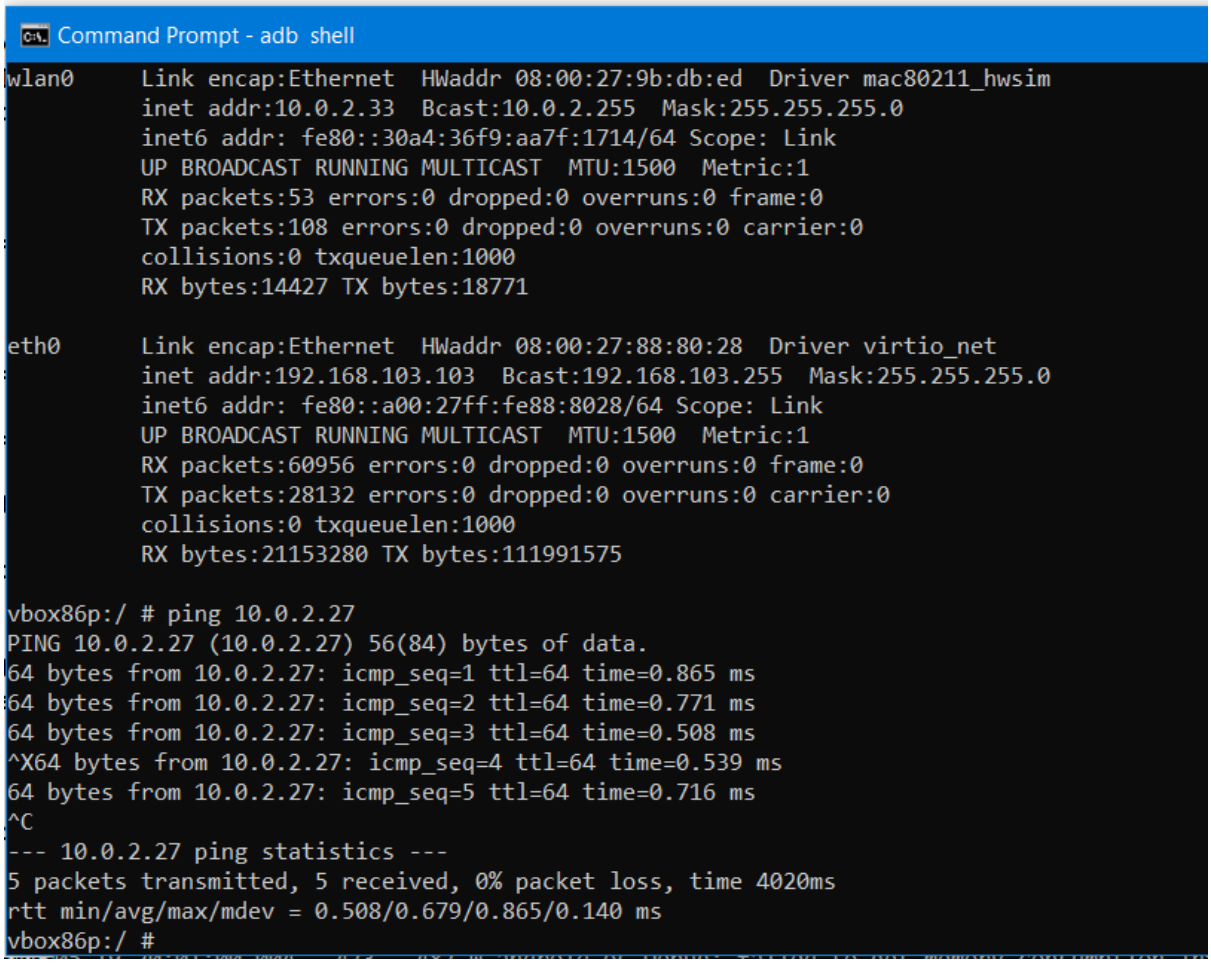
1.2.2 Install Docker-compose

Docker Compose is a tool that was developed to help define and share multi-container applications. With Compose, we can create a YAML file to define the services and with a single command



set the Kali machine to be in the NAT network as genymotion

Connectivity between Genymotion VM (10.0.2.33) and Kali VM (10.0.2.27)



Move Banker.zip to Kali VM

Unzip Banker.zip

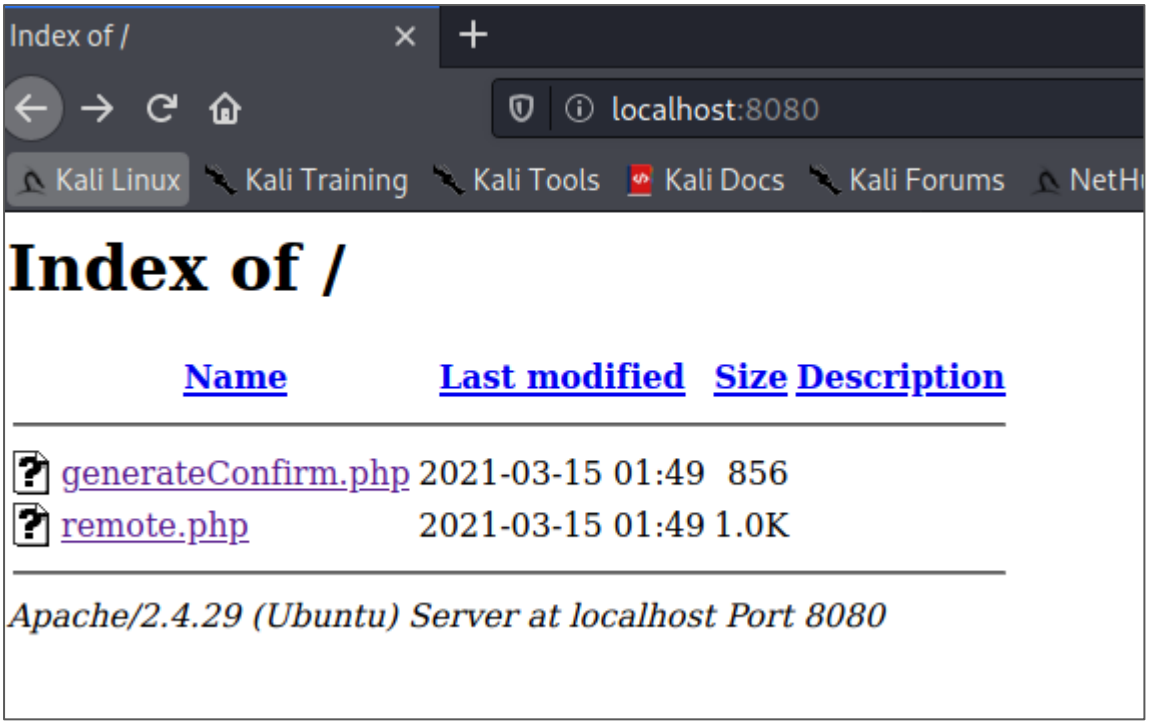
```
┌──(meena㉿kali)-[~]
└─$ unzip Bank-Docker.zip
Archive:  Bank-Docker.zip
   creating: Bank-Docker/
  inflating: Bank-Docker/.prettierrc
  inflating: Bank-Docker/docker-compose.yml
  inflating: Bank-Docker/README.txt
   creating: Bank-Docker/services/
   creating: Bank-Docker/services/apache/
   creating: Bank-Docker/services/apache/config/
  inflating: Bank-Docker/services/apache/config/.default-ssl.conf.swp
  inflating: Bank-Docker/services/apache/config/default-ssl.conf
  inflating: Bank-Docker/services/apache/config/my.cnf
  inflating: Bank-Docker/services/apache/dockerfile
   creating: Bank-Docker/services/apache/scripts/
  inflating: Bank-Docker/services/apache/scripts/data-dump.sql
  inflating: Bank-Docker/services/apache/scripts/run.sh
   creating: Bank-Docker/services/apache/src/
  inflating: Bank-Docker/services/apache/src/.htaccess
  inflating: Bank-Docker/services/apache/src/generateConfirm.php
  inflating: Bank-Docker/services/apache/src/remote.php
```

Start the service with “PORT1=8080 docker-compose up –build” in the Bank-Docker Folder containing the docker.yaml file

```
┌──(root💀kali)-[/home/meena]
└─# cd Bank-Docker

┌──(root💀kali)-[/home/meena/Bank-Docker]
└─# PORT1=8080 docker-compose up --build
Creating network "bank-docker_default" with the default driver
Building apache
Sending build context to Docker daemon  24.58kB
Step 1/14 : FROM ubuntu:18.04
18.04: Pulling from library/ubuntu
40dd5be53814: Pull complete
Digest: sha256:d21b6ba9e19feffa328cb3864316e6918e30acfd55e285b5d3df1d8ca3c7
fd3f
Status: Downloaded newer image for ubuntu:18.04
 ---> c6ad7e71ba7d
Step 2/14 : ENV DEBIAN_FRONTEND=noninteractive
 ---> Running in a4ba0f512877
Removing intermediate container a4ba0f512877
 ---> 48e305566953
Step 3/14 : RUN apt-get update && apt-get install gnupg2 -yq
 ---> Running in c5d5c592dffc
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:3 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages
```

```
Successfully built 84e079a206fd
Successfully tagged the_students_apache:latest
Creating bank-docker_apache_1 ... done
Attaching to bank-docker_apache_1
apache_1  |  * Restarting Apache httpd web server apache2
o suppress this message
apache_1  |                                                          [ OK ]
apache_1  |  * Starting MariaDB database server mysqld                [ OK ]
apache_1  |  * Restarting Apache httpd web server apache2
o suppress this message
apache_1  |                                                          [ OK ]
```

We see 3 Oks with no red alerts



Launch app on the localhost

1.2.3 Network Configuration

Change Ip addr

Configure default route for 10.0.2.0/24 net

Add default gateway

```
┌──(root💀kali)-[/home/meena]
└─# ifconfig eth0 10.0.2.4 netmask 255.255.255.0 up
```

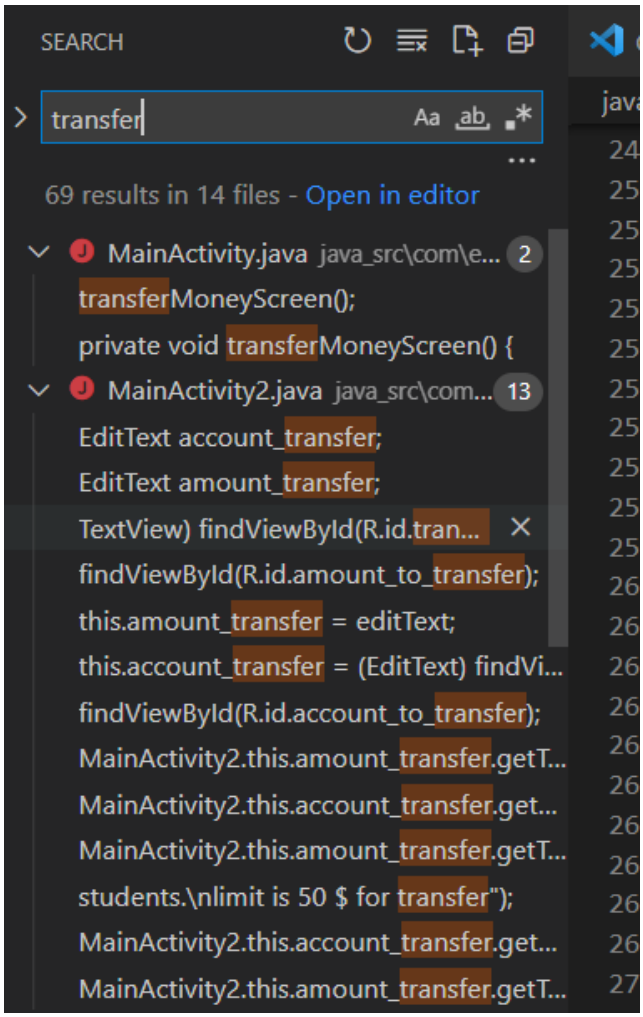
```
┌──(root💀kali)-[/home/meena]
└─# ip route list
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown
172.18.0.0/16 dev br-8b42833430a2 proto kernel scope link src 172.18.0.1

┌──(root💀kali)-[/home/meena]
└─# ip route add 10.0.2.0/24 dev eth0

┌──(root💀kali)-[/home/meena]
└─# route add default gw 10.0.2.1
```

Ping the emulator from Kali

```
┌──(root💀kali)-[/home/meena]
└─# ping  10.0.2.33
PING 10.0.2.33 (10.0.2.33) 56(84) bytes of data.
64 bytes from 10.0.2.33: icmp_seq=1 ttl=64 time=1.01 ms
64 bytes from 10.0.2.33: icmp_seq=2 ttl=64 time=0.637 ms
^C
--- 10.0.2.33 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
```

WE are good!

1.3 De compiling the Apk



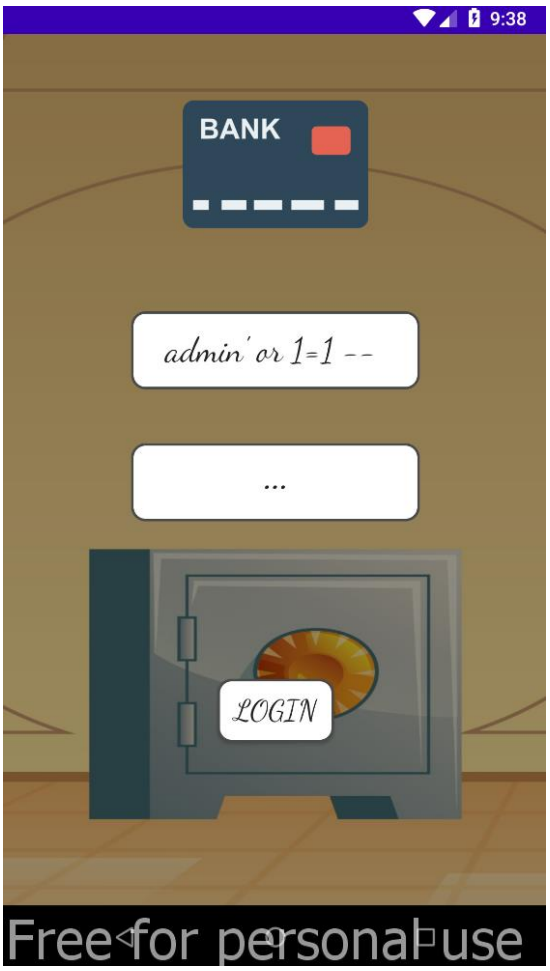
To Check the code.. , open visual studio. decompile the apk by loading it into the Visual studio with APKlab installed

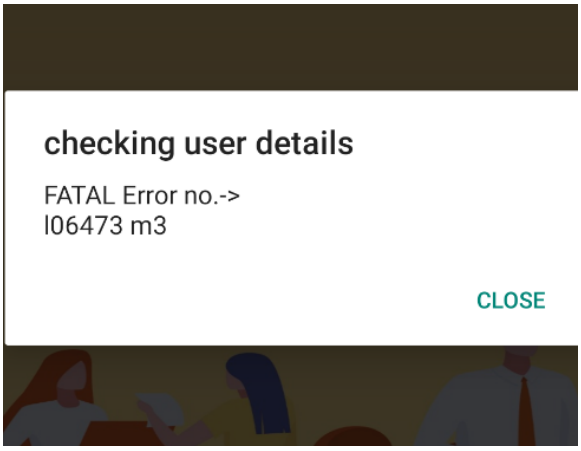
We can see the code for Money transfer

1.4 Functionality of APP and bypassing by sqli

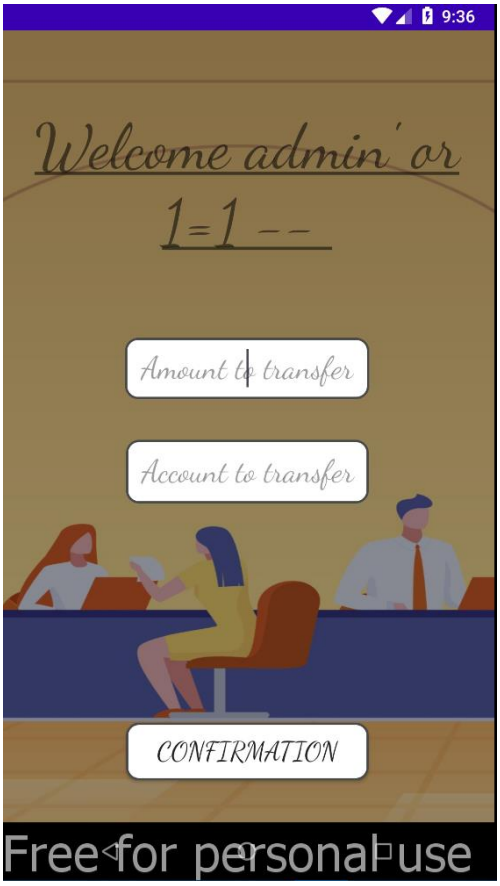
Click open the Mobile bank App. The login page is displayed.

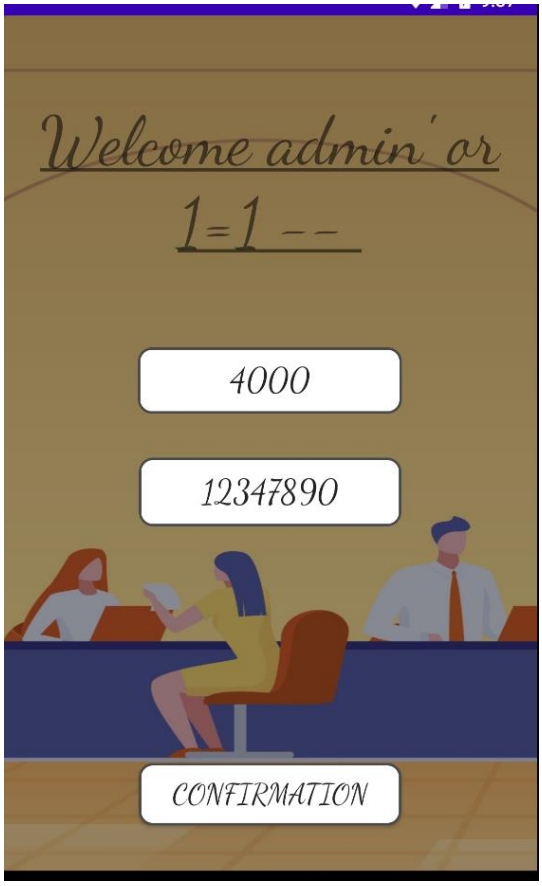
Trying the Boolean sql 1 = 1condition with admin login, an error message is seen





Close the error message and the screen for money transfer is seen.





Watch the adb logcat window when you do the Boolean sql injection

The log message display a set of username s and passwords!!!

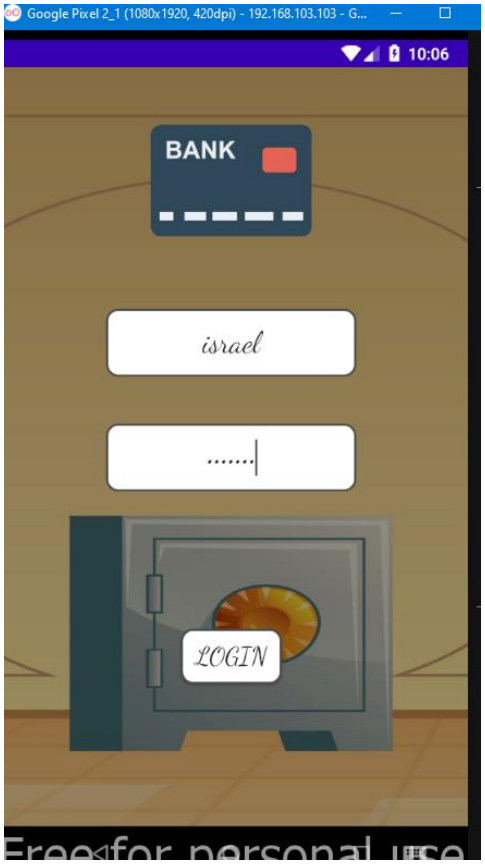
```
05-19 22:04:08.758   473  1534 D DhcpClient: Scheduling renewal in 299s
05-19 22:04:08.758   473  1534 D DhcpClient: Scheduling rebind in 524s
05-19 22:04:08.758   473  1534 D DhcpClient: Scheduling expiry in 599s
05-19 22:04:46.269  1606  3081 D BACKGROUND RETURN -> : class java.lang.StringBuilder
05-19 22:04:46.369   473   574 I ActivityManager: START u0 {cmp=com.example.mobileptfinal/.MainActivity2 (has extras)} f
rom uid 10067
05-19 22:04:46.502  1606  3084 D BACKGROUND RETURN -> : class java.lang.StringBuilder
05-19 22:04:46.678  1606  1606 D stock -> : israel isis447
05-19 22:04:46.678  1606  1606 D stock -> : nadav shna467
05-19 22:04:46.678  1606  1606 D stock -> : asaf moas823
05-19 22:04:46.678  1606  1606 D stock -> : eliran klel139
05-19 22:04:46.678  1606  1606 D stock -> : oded ocod669
05-19 22:04:46.882   448   448 D SurfaceFlinger: duplicate layer name: changing com.example.mobileptfinal/com.example.mo
bileptfinal.MainActivity2 to com.example.mobileptfinal/com.example.mobileptfinal.MainActivity2#1
05-19 22:04:47.186   473   495 I ActivityManager: Displayed com.example.mobileptfinal/.MainActivity2: +736ms
05-19 22:04:49.958   473   624 E TaskPersister: File error accessing recents directory (directory doesn't exist?).
```

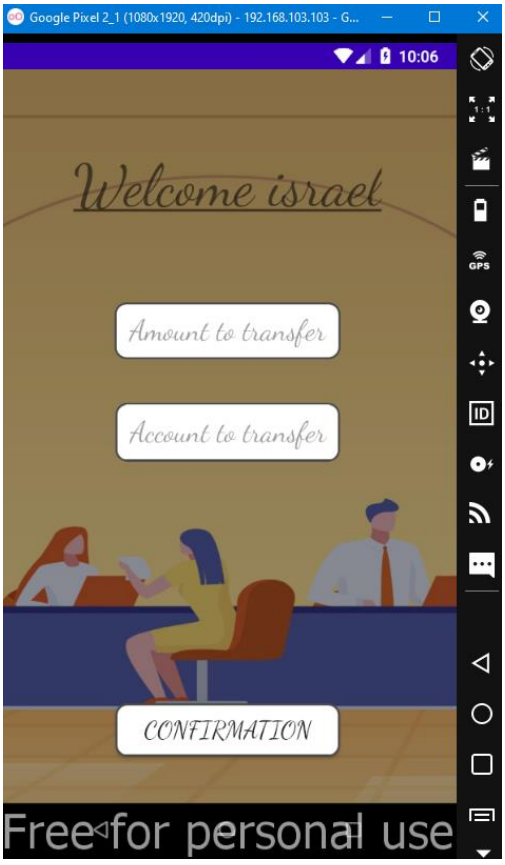
The first username Israel and the password isis447 is chosen to try out in the app.

1.5 Hack to transfer money

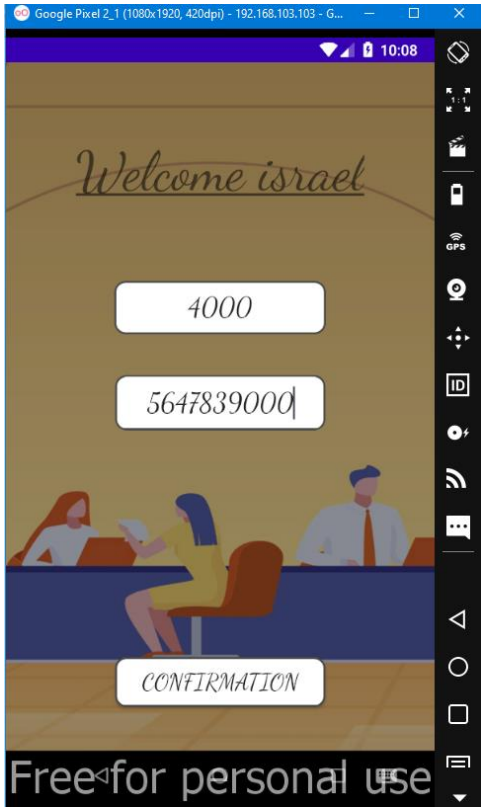
Enter the username passwd as shown below. The app shows ‘Welcome Israel screen and asks for amt and account to transfer money

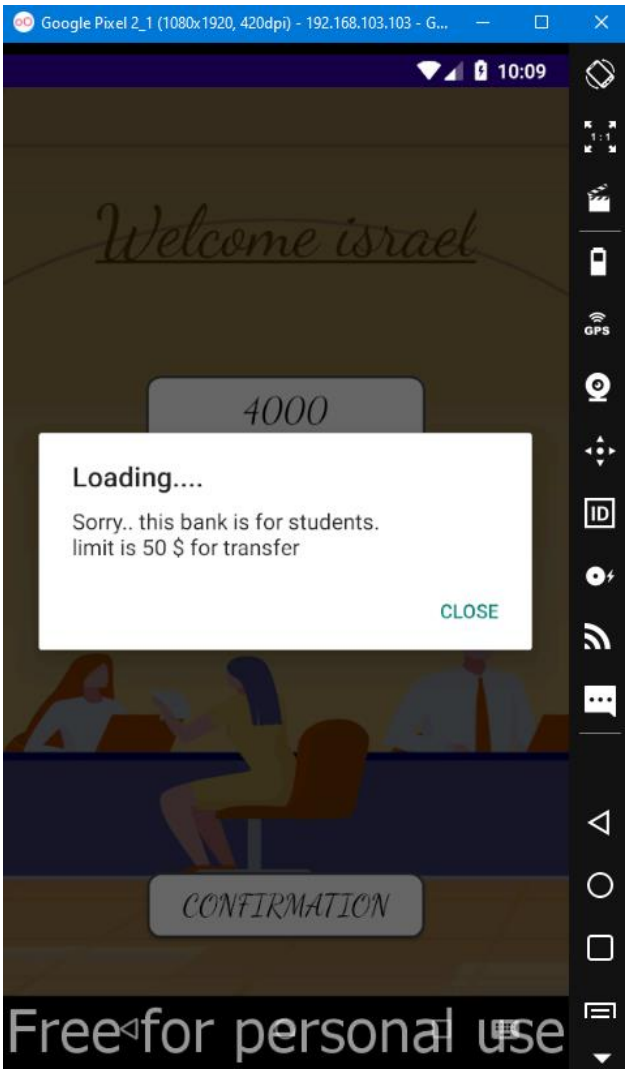
Refer to the screen shots below


Google Pixel 2_1 (1080x1920, 420dpi) - 192.168.103.103 - G...
10:06
BANK
israel
.....
LOGIN
Free for personal use


Google Pixel 2_1 (1080x1920, 420dpi) - 192.168.103.103 - G...
10:06
Welcome israel
Amount to transfer
Account to transfer
CONFIRMATION
Free for personal use

Enter 4000 at the text message Amount to transfer, 4000 and the account number. Press “Confirmation”



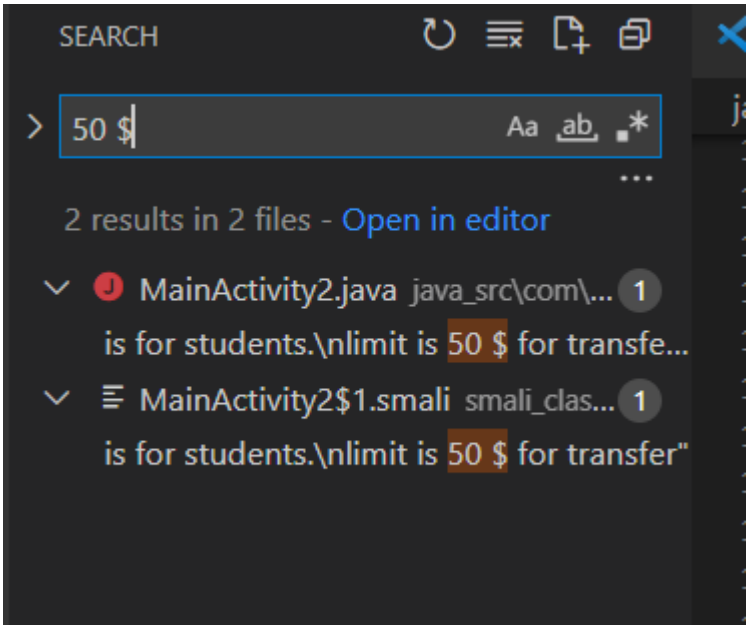


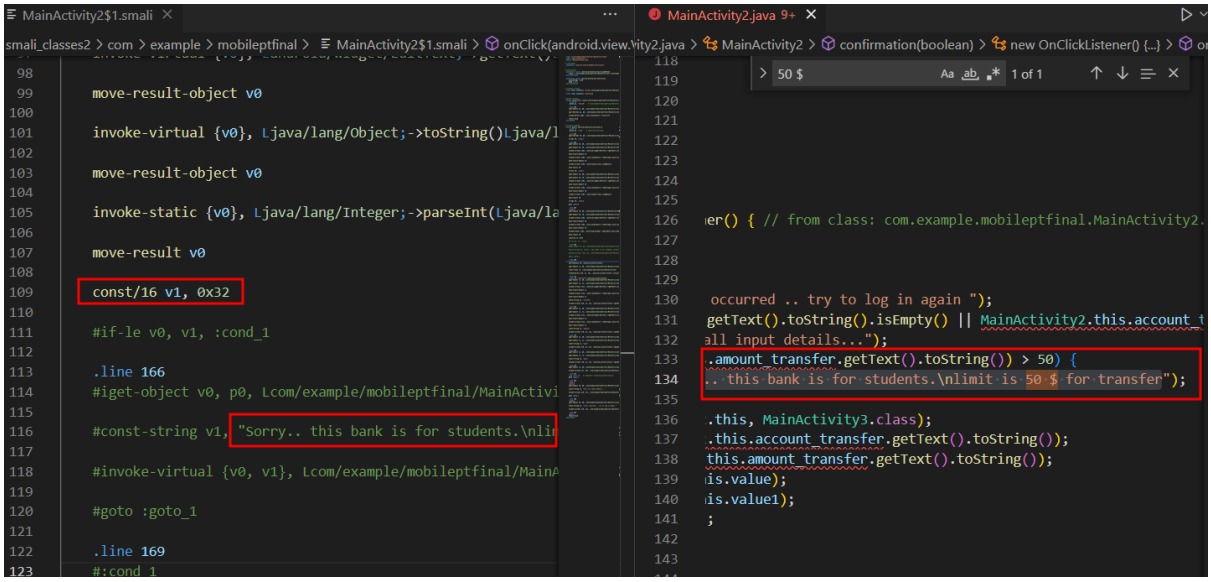
An error message that “ Sorry.. this bank is for students.\nlimit is 50 $ for transfer” is shown. From the error message, search for string $50 for transfer in the Visual studio workspace.

Two files are listed. One java and one smali file.

Check MainActivity2.java. Locate the code in java and compare with small code.

Search Comment in the MainActivity2$1.smali. Open the two files side by side. Search for the error msg in smali file. Just few lines above the error message the value 50 is defined as constant. The program checks for the amount transferred and limit it to $50. If anybody tries to transfer more, the check is successful and the app gives err message.





The related lines are commented with # in front. So the app will not check if the amount is > 50 anymore. Any amount can now be transferred now.

```
const/16 v1, 0x32

#if-le v0, v1, :cond_1

.line 166
#iget-object v0, p0, Lcom/example/mobileptfinal/MainActivity2$1;->this$0:Lcom/

#const-string v1, "Sorry.. this bank is for students.\nlimit is 50 $ for trans

#invoke-virtual {v0, v1}, Lcom/example/mobileptfinal/MainActivity2;->confirm(L

#goto :goto_1

.line 169
#:cond_1
new-instance v0, Landroid/content/Intent;
```

Re-compile the changed smali code using the option b

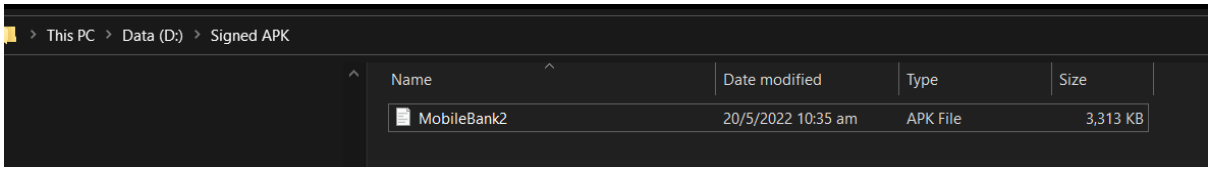
```
D:\platform-tools\MobileBank2>cd ..

D:\platform-tools>apktool b MobileBank2
```

All the files are recompiled and the new apk is put under <project folder>/dist

```
D:\platform-tools>E:\MobileFinalBank\apktool b MobileBank2
I: Using Apktool 2.5.0
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes2 folder into classes2.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...
```

```
D:\platform-tools>cd MobileBank2

D:\platform-tools\MobileBank2>dir
 Volume in drive D is Data
 Volume Serial Number is 009F-FD9D

 Directory of D:\platform-tools\MobileBank2

20/05/2022  10:24 am    <DIR>          .
20/05/2022  10:24 am    <DIR>          ..
20/05/2022  08:50 am                13 .gitignore
20/05/2022  08:49 am             1,296 AndroidManifest.xml
20/05/2022  08:49 am             2,181 apktool.yml
20/05/2022  10:23 am    <DIR>          build
20/05/2022  10:24 am    <DIR>          dist
20/05/2022  08:50 am    <DIR>          java_src
20/05/2022  08:49 am    <DIR>          original
20/05/2022  08:49 am    <DIR>          res
20/05/2022  08:49 am    <DIR>          smali
20/05/2022  08:49 am    <DIR>          smali_classes2
               3 File(s)          3,490 bytes
               9 Dir(s)  171,460,636,672 bytes free

D:\platform-tools\MobileBank2>cd dist

D:\platform-tools\MobileBank2\dist>dir
 Volume in drive D is Data
 Volume Serial Number is 009F-FD9D

 Directory of D:\platform-tools\MobileBank2\dist

20/05/2022  10:24 am    <DIR>          .
20/05/2022  10:24 am    <DIR>          ..
20/05/2022  10:24 am         3,328,270 MobileBank2.apk
               1 File(s)      3,328,270 bytes
               2 Dir(s)  171,460,636,672 bytes free

D:\platform-tools\MobileBank2\dist>
```

Sign the new APK file, Using java -jar uber-apk-signer.jar -a /path/to/apks --out /path/to/apks/out

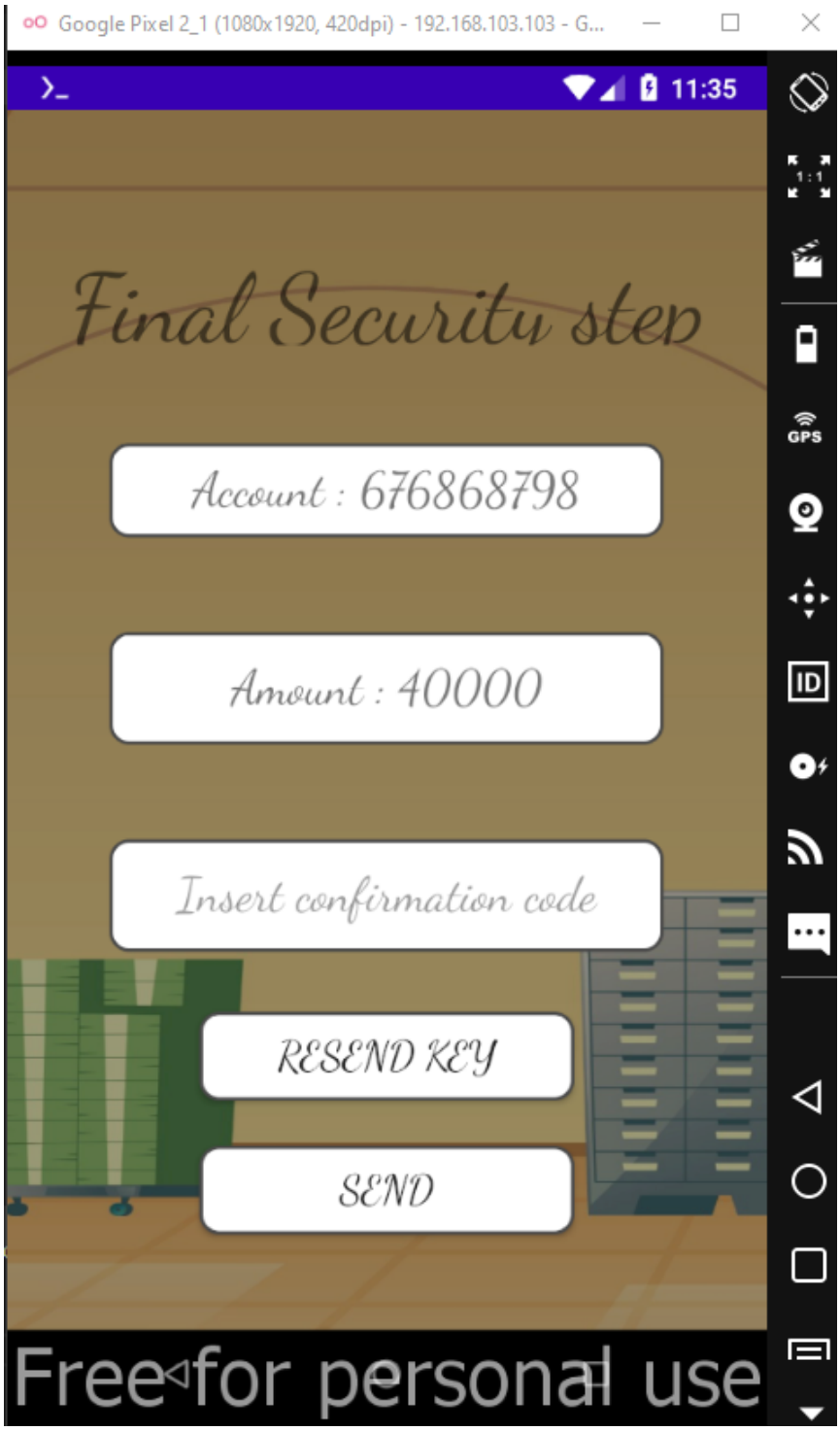


Drag and drop the signed APK in to the emulator. Check the functionality of the APK. Login to the MobileBank app and login again using Israel user. Enter details in the money transfer screen and click confirmation. “Final Security Step” Screen is shown. And it asks for the Confirmation code, which would have been sent to the user.

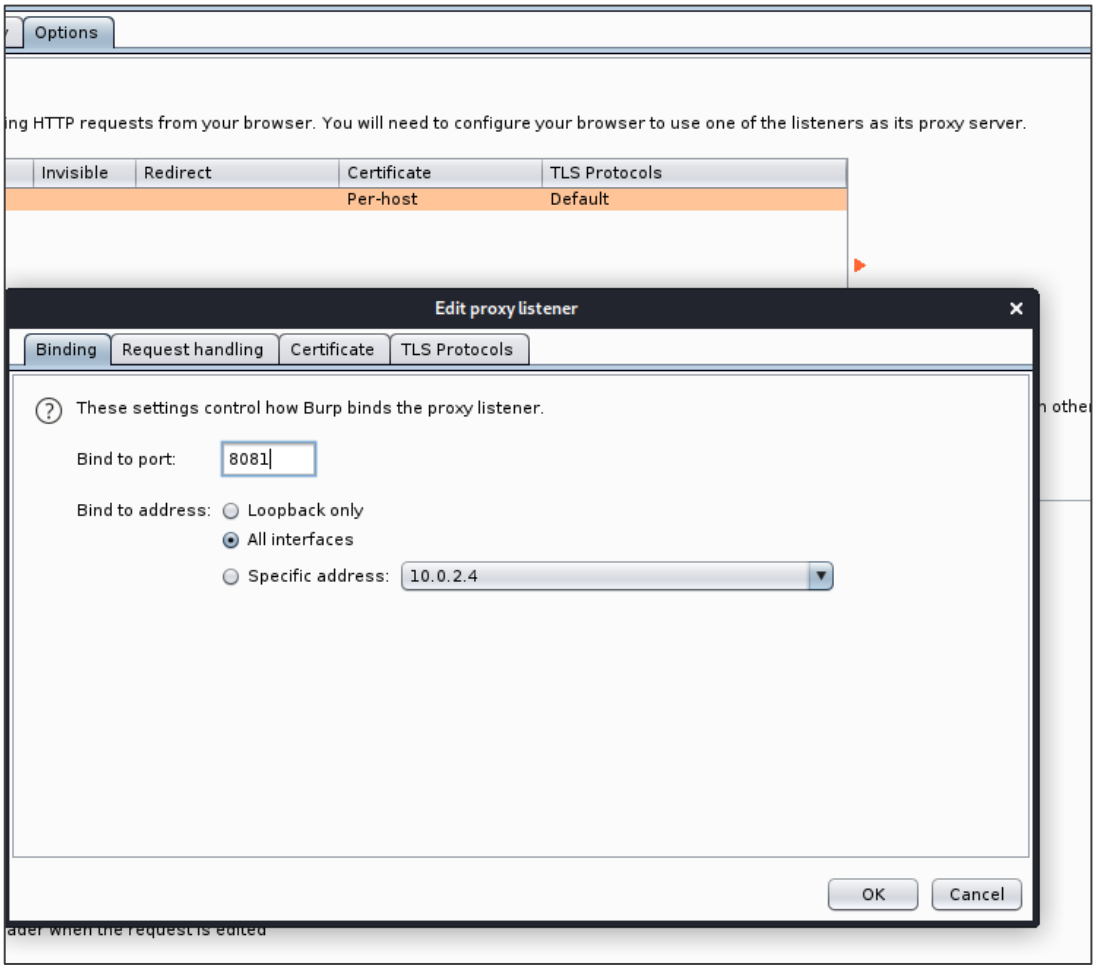
Now we have to capture the Confirmation code and enter it in the bix “Insert confirmation code” and Click “Send”.

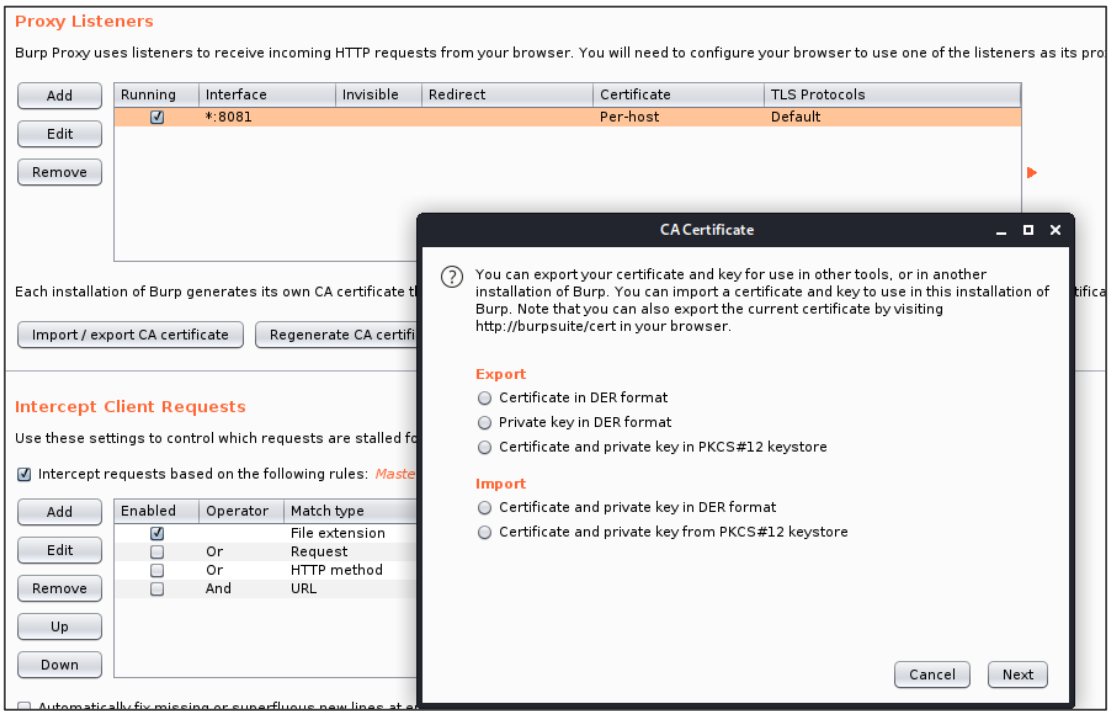
The lab instruction suggests using burp. Configure burp in kali to listen for the app. The following screenshots show it.

Final security test screen:

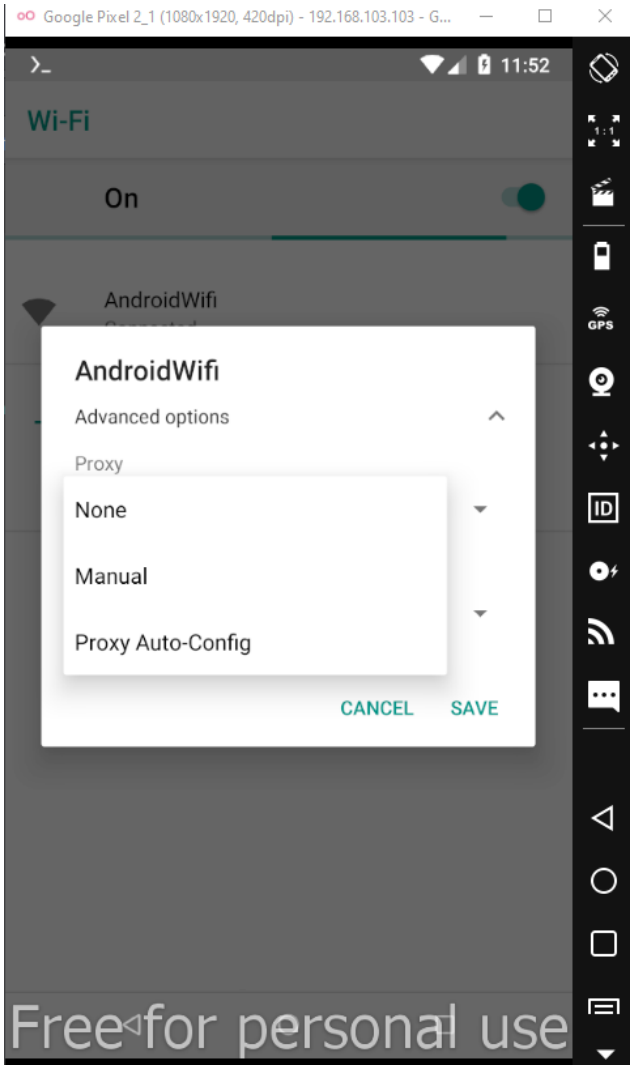


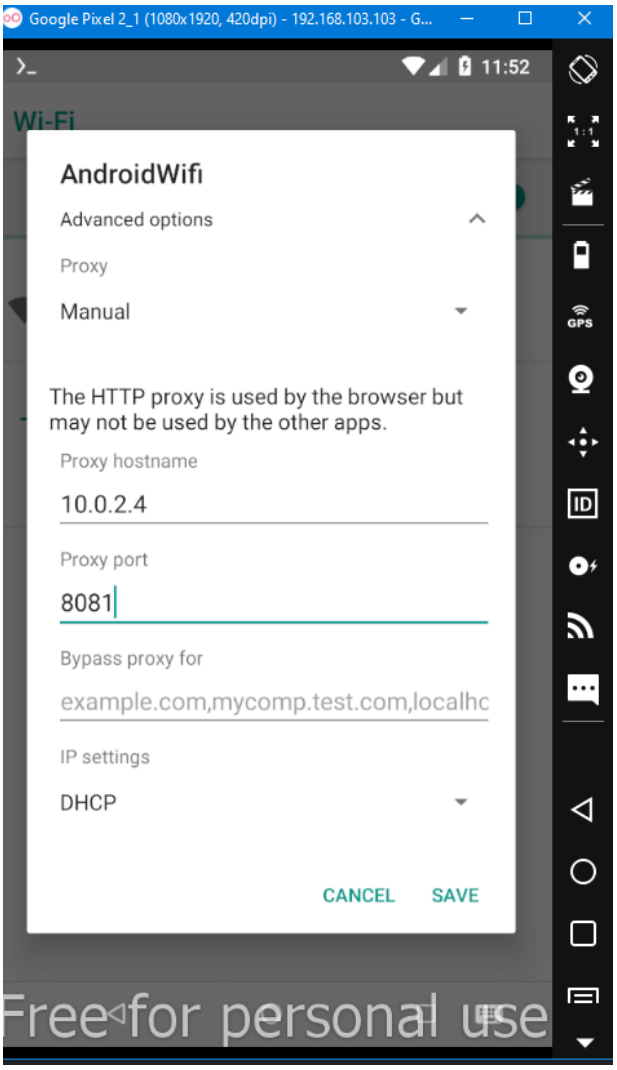
Burp proxy configuration:



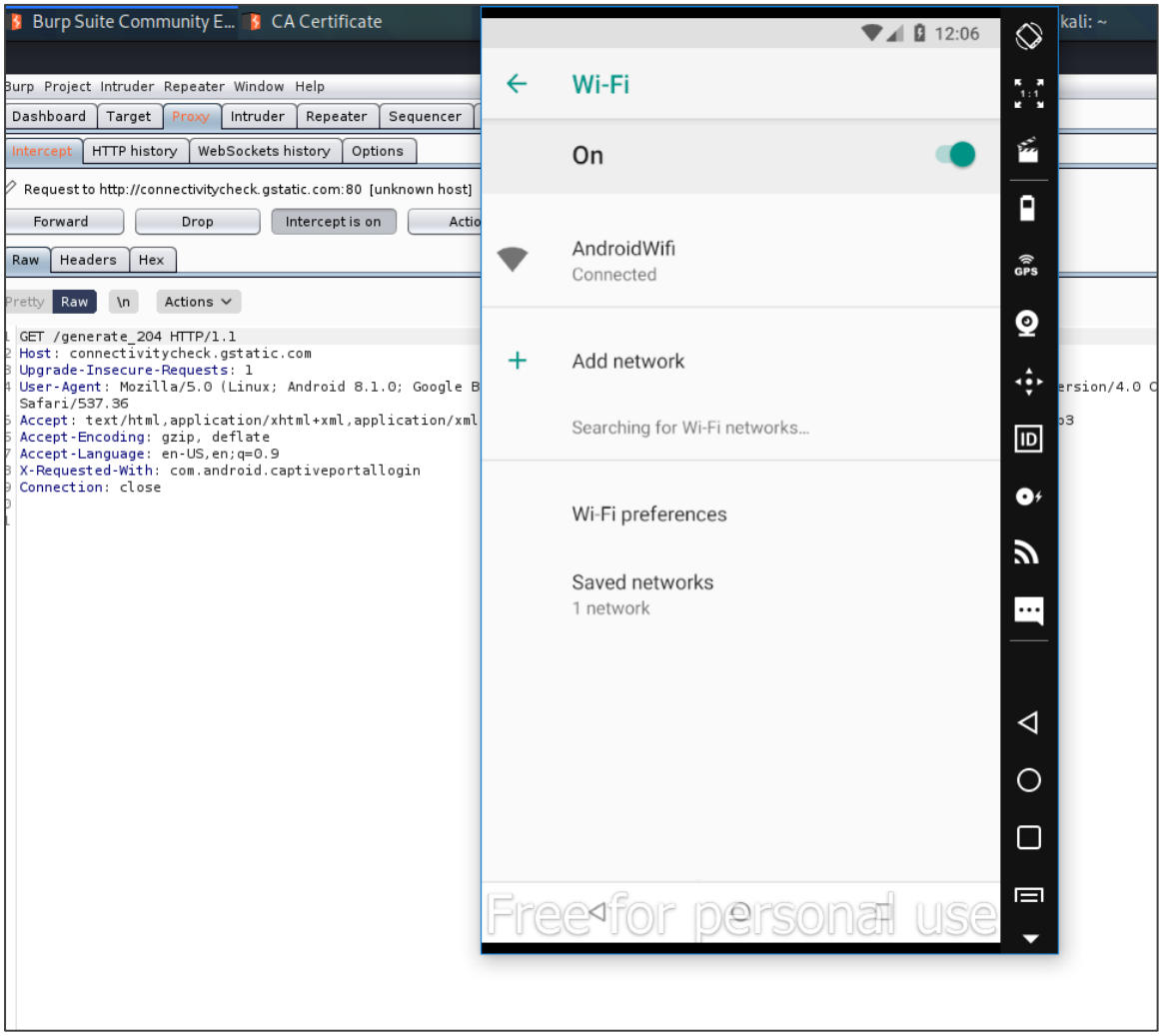


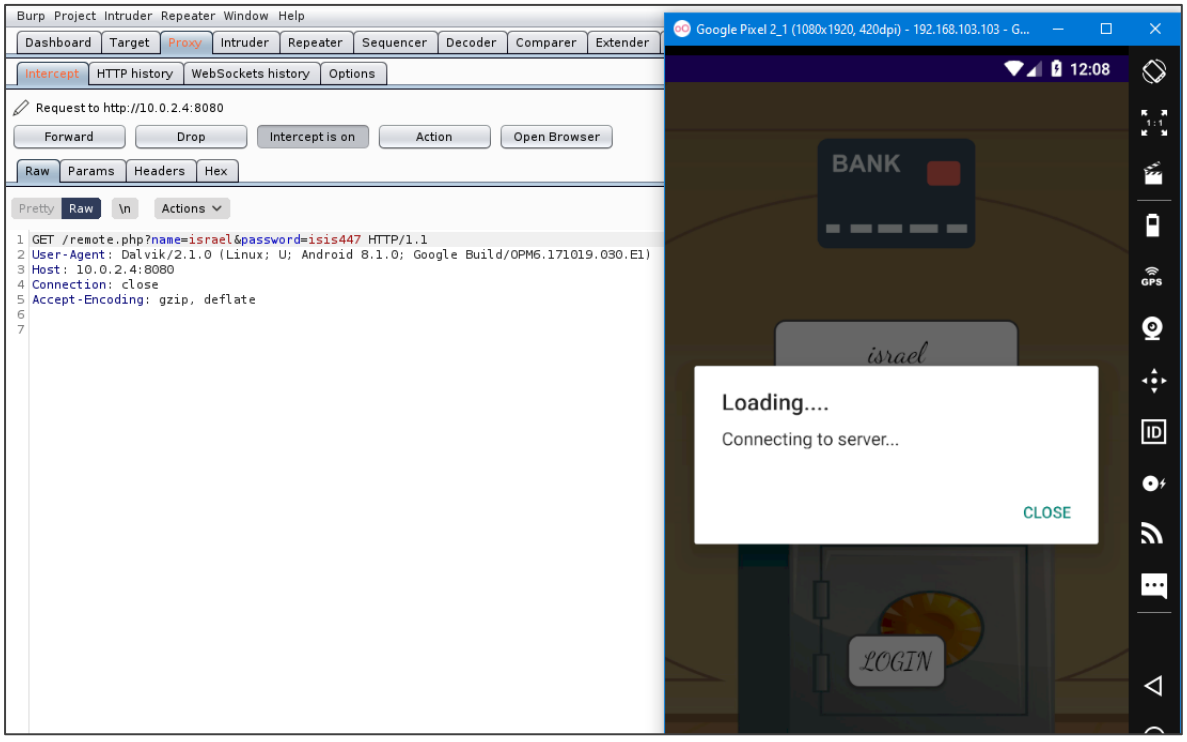
Emulator proxy configuration



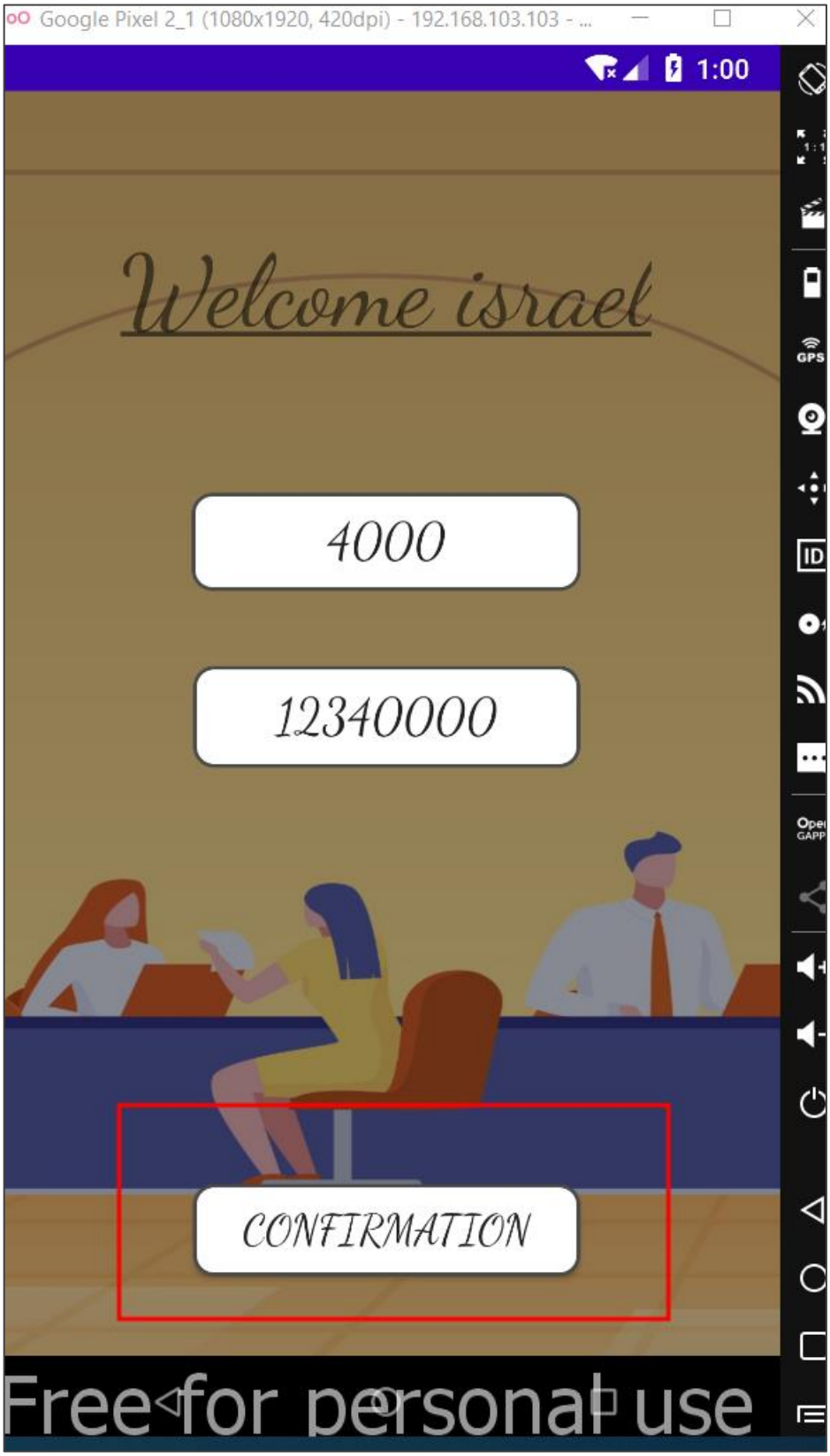


Burp catures the tracfic from the mobile bank app as shown below:



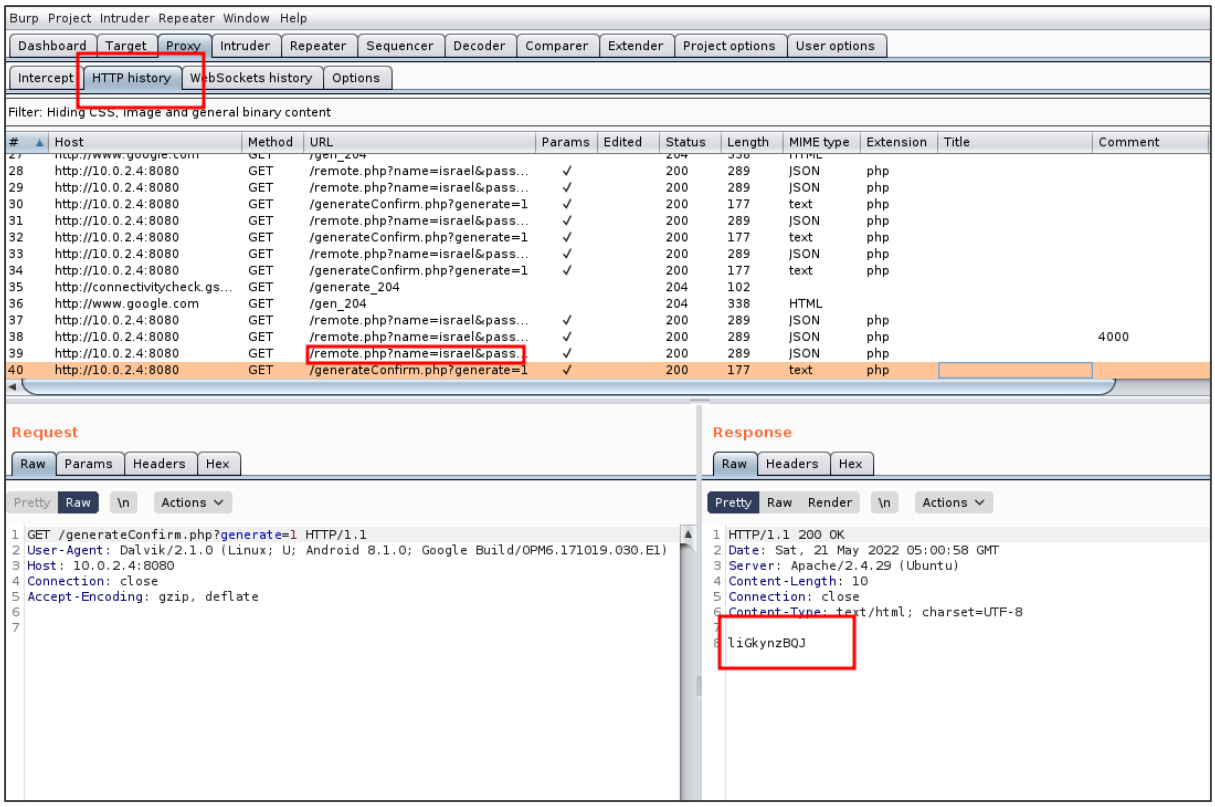


Forward the request and check the http history tab under Proxy

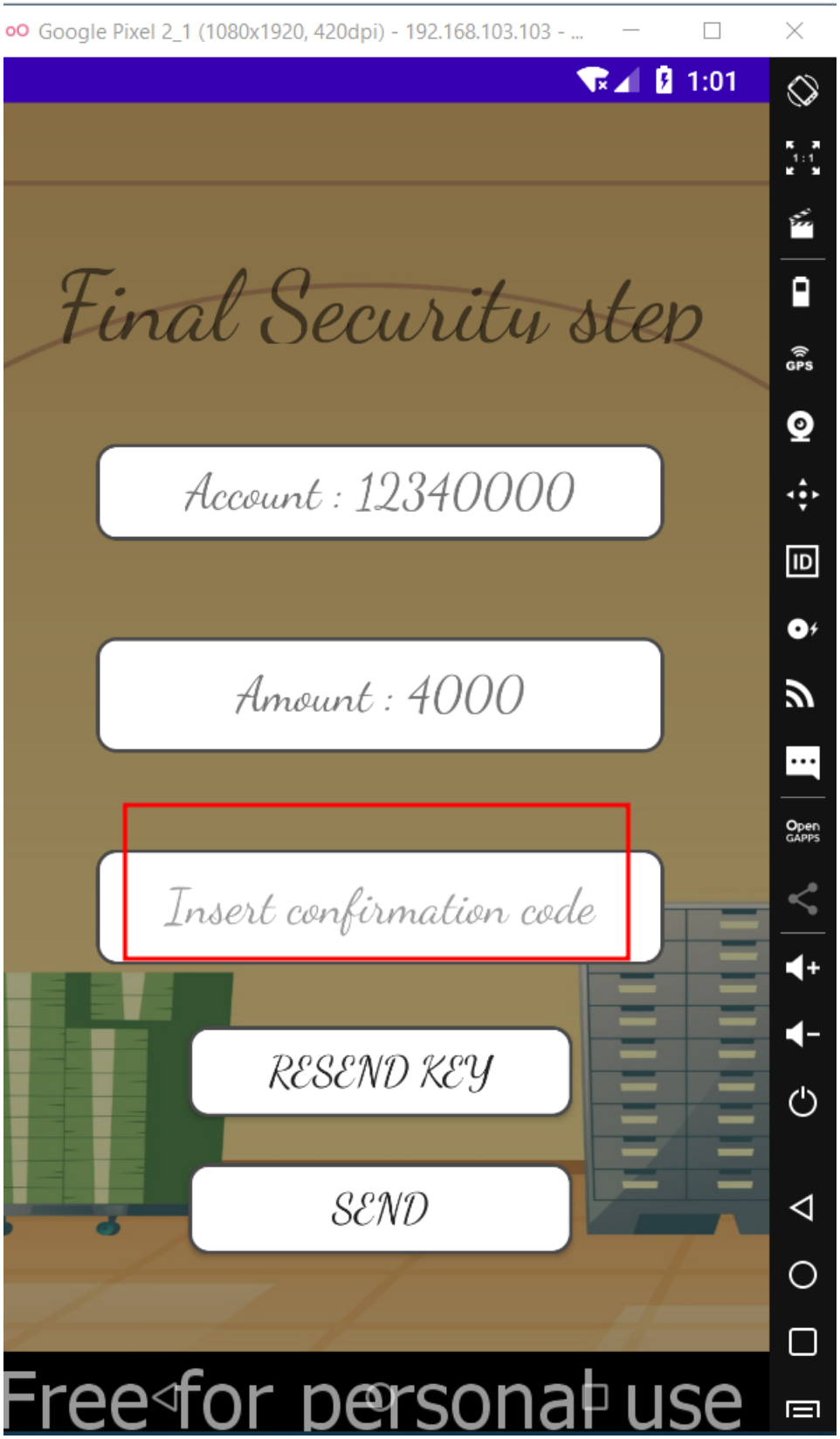


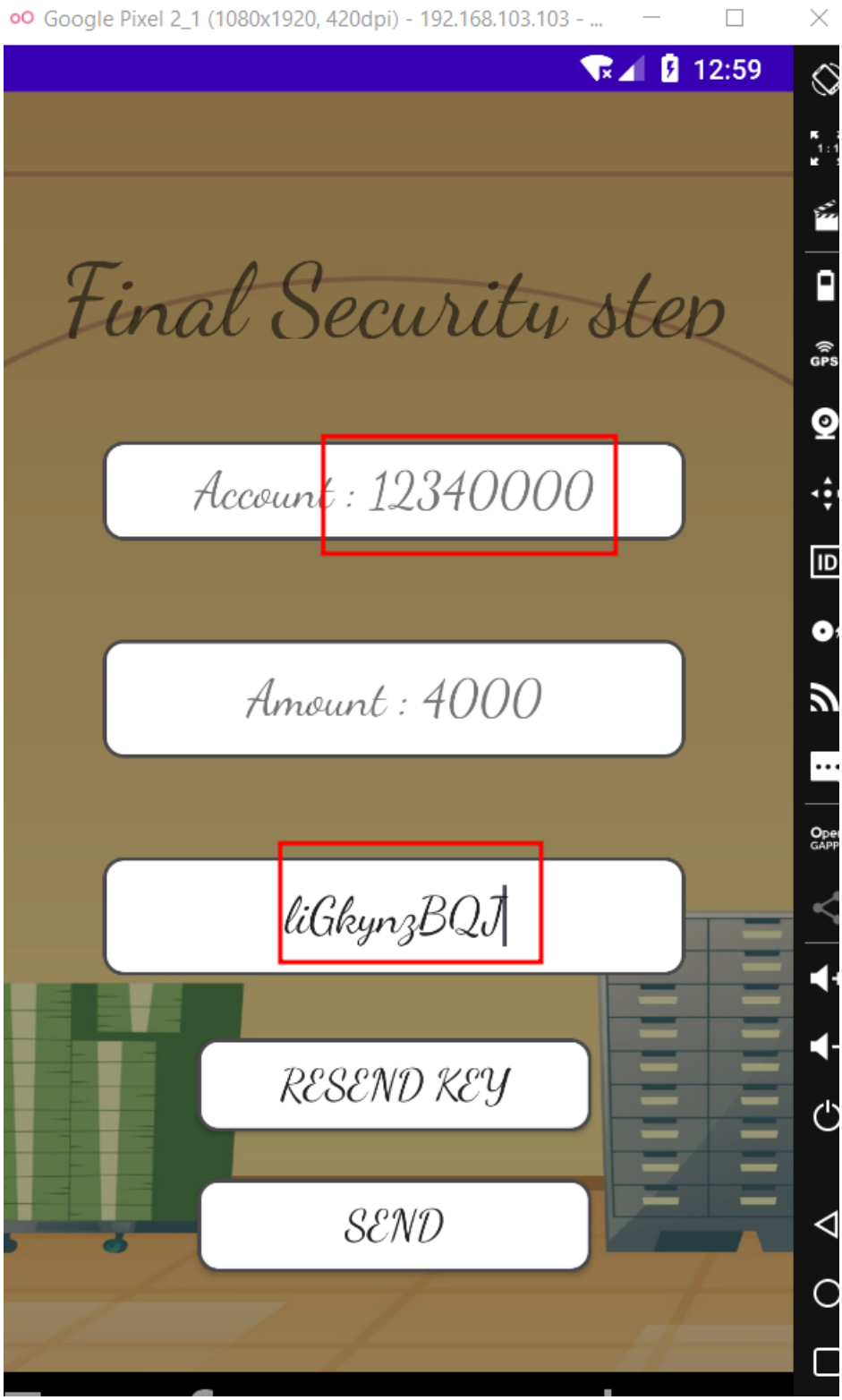
Press confirmation and forward the requests

Check the HTTP history . The confirmtion code generated is seen.

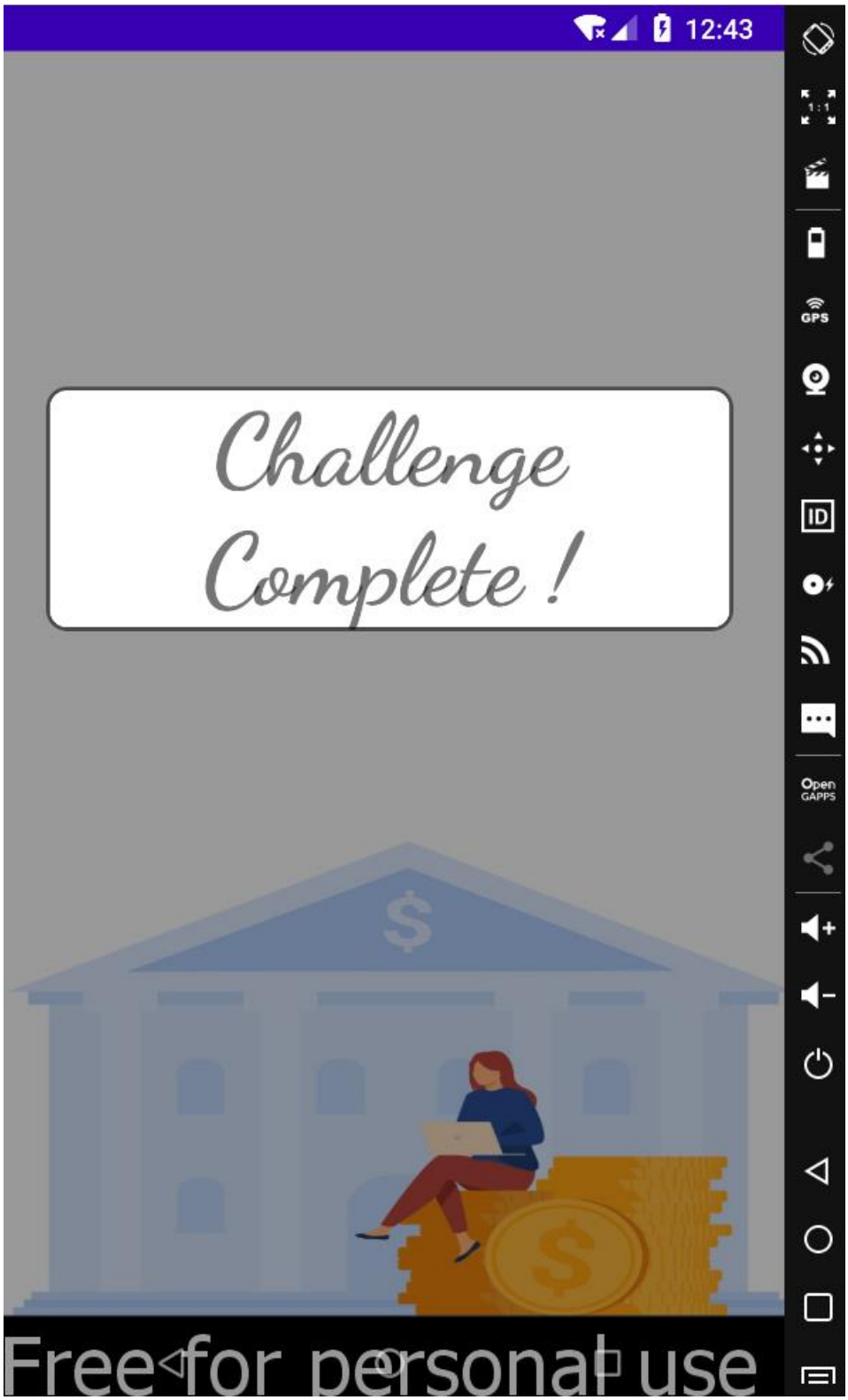


Enter it in the below screen


Google Pixel 2_1 (1080x1920, 420dpi) - 192.168.103.103 - ...
1:01
Final Security step
Account : 12340000
Amount : 4000
Insert confirmation code
RESEND KEY
SEND
Free for personal use


Google Pixel 2_1 (1080x1920, 420dpi) - 192.168.103.103 - ...
12:59
Final Security step
Account : 12340000
Amount : 4000
liGkynzBQJ
RESEND KEY
SEND

Press “Send”


12:43
Challenge
Complete !
Free for personal use