Modifying Game Apps

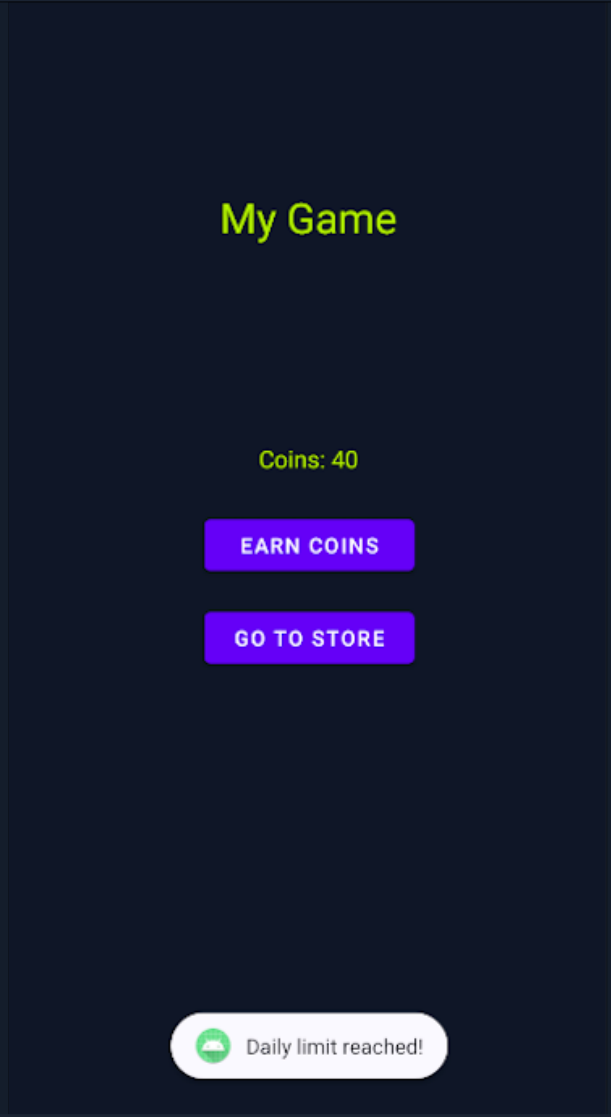
Let's examine another instance of how application patching can change its intended functionality. We'll be using an AVD for demonstration, but these steps work just as well on other Android emulators or physical devices. After starting the emulator, install the app using the following `ADB` commands.

Modifying Game Apps

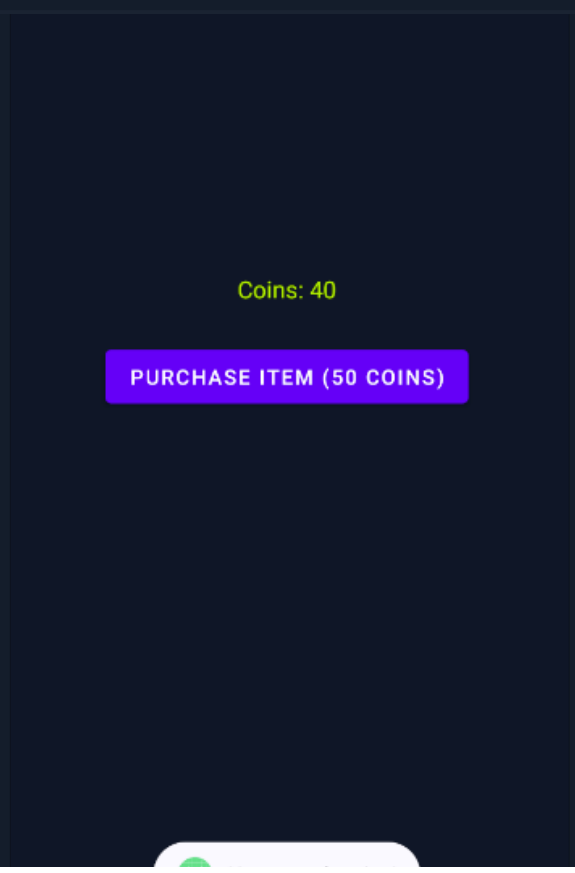
```
rl1k@htb[/htb]$ adb connect
rl1k@htb[/htb]$ adb install myapp.apk

Performing Streamed Install
Success
```

The following is a gaming application where users can claim a set amount of coins daily. These coins can be used to purchase items from the in-game store. Starting the app and tapping multiple times on the `EARN COINS` button eventually produces the message `Daily limit reached!`.



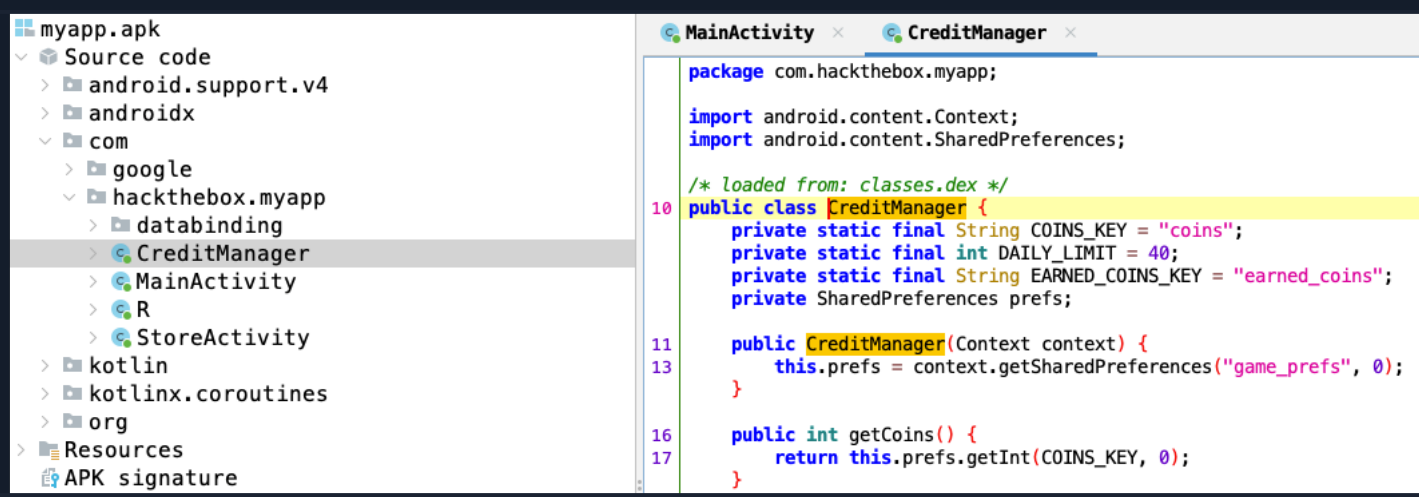
Navigating to the in-game store by tapping the `GO TO STORE` button presents us with the following screen.



As shown in the image above, tapping the `PURCHASE ITEM (50 COINS)` button returns the message `Not enough coins!`. Let's reverse the app and check if we're able to change its intended flow. First, let's inspect the code with JADX.

Modifying Game Apps

```
rl1k@htb[/htb]$ jadx-gui myapp.apk
```



Decompiling the app using JADX reveals the `MainActivity` and `CreditManager` activities. Reading the content of `CreditManager`, we can see the variable `private static final int DAILY_LIMIT = 40;`. There is a strong chance that this variable controls the user's daily limit. Now, let's use APKTool to extract the smali files and see if we can adjust this limit.

Modifying Game Apps

```
rl1k@htb[/htb]$ apktool d myapp.apk

I: Using Apktool 2.7.0 on myapp.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /Users/bertolis/Library/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...
I: Copying META-INF/services directory
```

Listing the content of the directory `myapp/smali/com/hackthebox/myapp/` reveals the `CreditManager.smali`, among other files.

Modifying Game Apps

```
rl1k@htb[/htb]$ ls -l myapp/smali/com/hackthebox/myapp

-rw-r--r--  1 bertolis  bertolis  4502 Nov 14 01:00 CreditManager.smali
-rw-r--r--  1 bertolis  bertolis  1009 Nov 14 01:00 MainActivity$$ExternalSyntheticLambda0.smali
-rw-r--r--  1 bertolis  bertolis  1009 Nov 14 01:00 MainActivity$$ExternalSyntheticLambda1.smali
-rw-r--r--  1 bertolis  bertolis  5623 Nov 14 01:00 MainActivity.smali
-rw-r--r--  1 bertolis  bertolis   803 Nov 14 01:00 R$color.smali
-rw-r--r--  1 bertolis  bertolis   607 Nov 14 01:00 R$drawable.smali
<SNIP>
```

Reading the first lines of the file, it's easy to spot the `DAILY_LIMIT` variable set to the hexadecimal value `0x28,` which is the decimal value `40`.

Modifying Game Apps

```
rl1k@htb[/htb]$ vim myapp/smali/com/hackthebox/myapp/CreditManager.smali
```

Code: smali

```
.class public Lcom/hackthebox/myapp/CreditManager;
.super Ljava/lang/Object;
.source "CreditManager.java"


# static fields
.field private static final COINS_KEY:Ljava/lang/String; = "coins"

.field private static final DAILY_LIMIT:I = 0x28

.field private static final EARNED_COINS_KEY:Ljava/lang/String; = "earned_coins"
<SNIP>
```

Let's see see what happens if we change this value with `0x32` (which represents the decimal value `50`). This way, the daily amount to claim will theoretically be set to 50. Hopefully, we can then be able to purchase the item from the in-game store. The updated snippet looks like this.

Code: smali

```
<SNIP>
# static fields
.field private static final COINS_KEY:Ljava/lang/String; = "coins"

.field private static final DAILY_LIMIT:I = 0x32
<SNIP>
```

Once we finish editing, we can follow the usual process to recompile and sign the APK.

Modifying Game Apps

```
rl1k@htb[/htb]$ apktool b myapp
rl1k@htb[/htb]$ echo -e "password\npassword\njohn doe\ntest\ntest\ntest\ntest\ntest\nyes" > params.txt
rl1k@htb[/htb]$ cat params.txt | keytool -genkey -keystore key.keystore -validity 1000 -keyalg RSA -alias john
rl1k@htb[/htb]$ zipalign -p -f -v 4 myapp/dist/myapp.apk myapp_aligned.apk
rl1k@htb[/htb]$ echo password | apksigner sign --ks key.keystore myapp_aligned.apk
rl1k@htb[/htb]$ adb uninstall com.hackthebox.myapp
rl1k@htb[/htb]$ adb install myapp_aligned.apk

Performing Incremental Install
Serving...
All files should be loaded. Notifying the device.
Success
Install command complete in 381 ms
```

Running the application and trying to purchase the item from the in-game store will result in the same error. A closer inspection of the smali code within `CreditManager.smali` reveals the line `const/16 v3, 0x28`, which sets the hex value `0x28` (40 in decimal) to the local variable `v3` before being used in a conditional check.

Modifying Game Apps

```
rl1k@htb[/htb]$ vim myapp/smali/com/hackthebox/myapp/CreditManager.smali
```

Code: smali

```
<SNIP>
const/16 v3, 0x28

if-ge v0, v3, :cond_0

.line 23
invoke-virtual {p0}, Lcom/hackthebox/myapp/CreditManager;->getCoins()I
<SNIP>
```

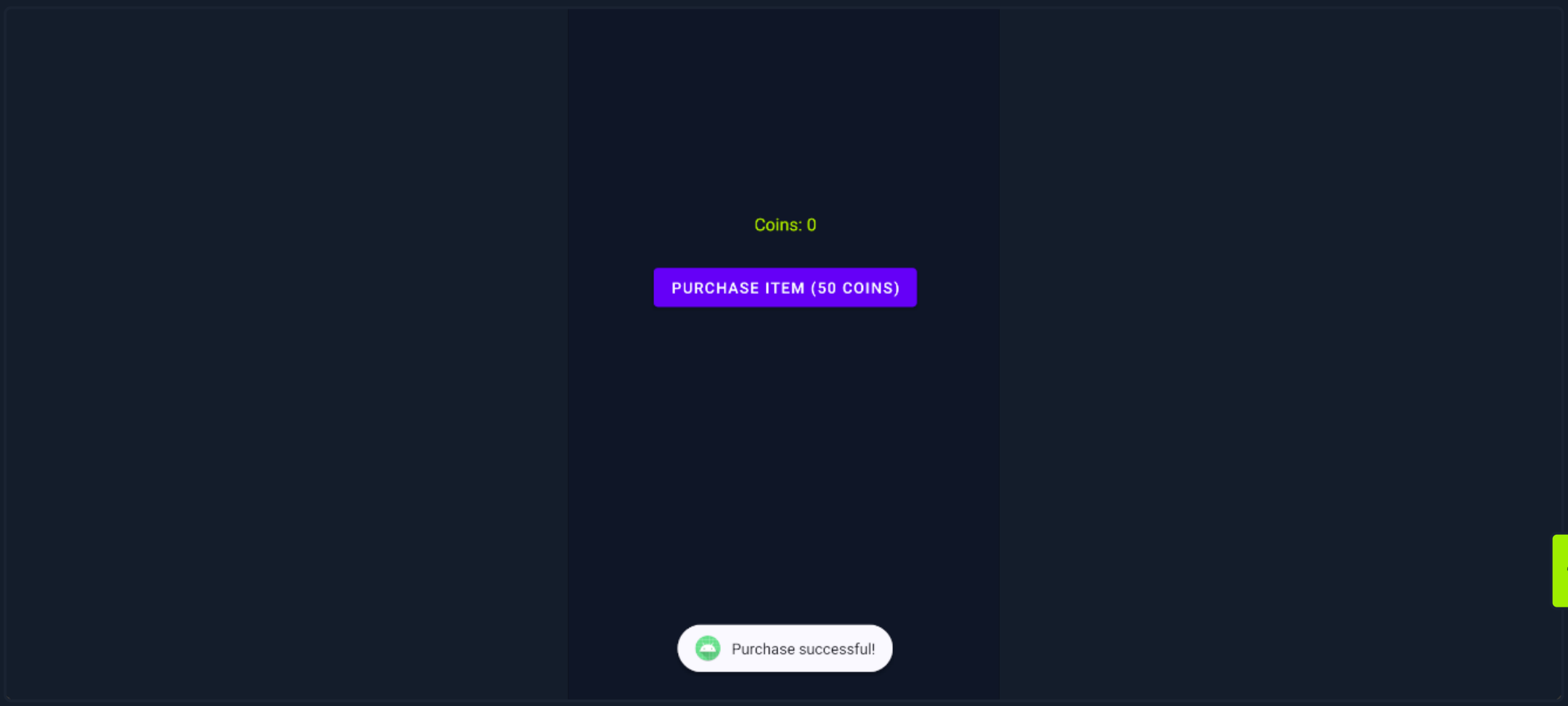
We should update this line as well. The updated code looks like this:

Code: smali

```
<SNIP>
const/16 v3, 0x32

if-ge v0, v3, :cond_0
<SNIP>
```

Let's recompile, sign, and install the application again using the same steps as earlier. Once it's installed, we will start the app and press the EARN COINS button five times. After navigating to the in-game store screen (by tapping the `GO TO STORE` button), we can finally try to purchase the item.



The patching is successful, and the product is purchased.

Waiting to start...

Enable step-by-step solutions for all questions

Questions

Cheat Sheet

Answer the question(s) below to complete this Section and earn cubes!

+ 3 What is the message displayed on the screen after successfully purchasing the item?

Submit your answer here...

+10 Streak pts Submit myapp_mod_games.zip

Previous Next

Cheat Sheet

Go to Questions

Table of Contents

Root Detection Bypass

Skills Assessment

Skills Assessment

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left