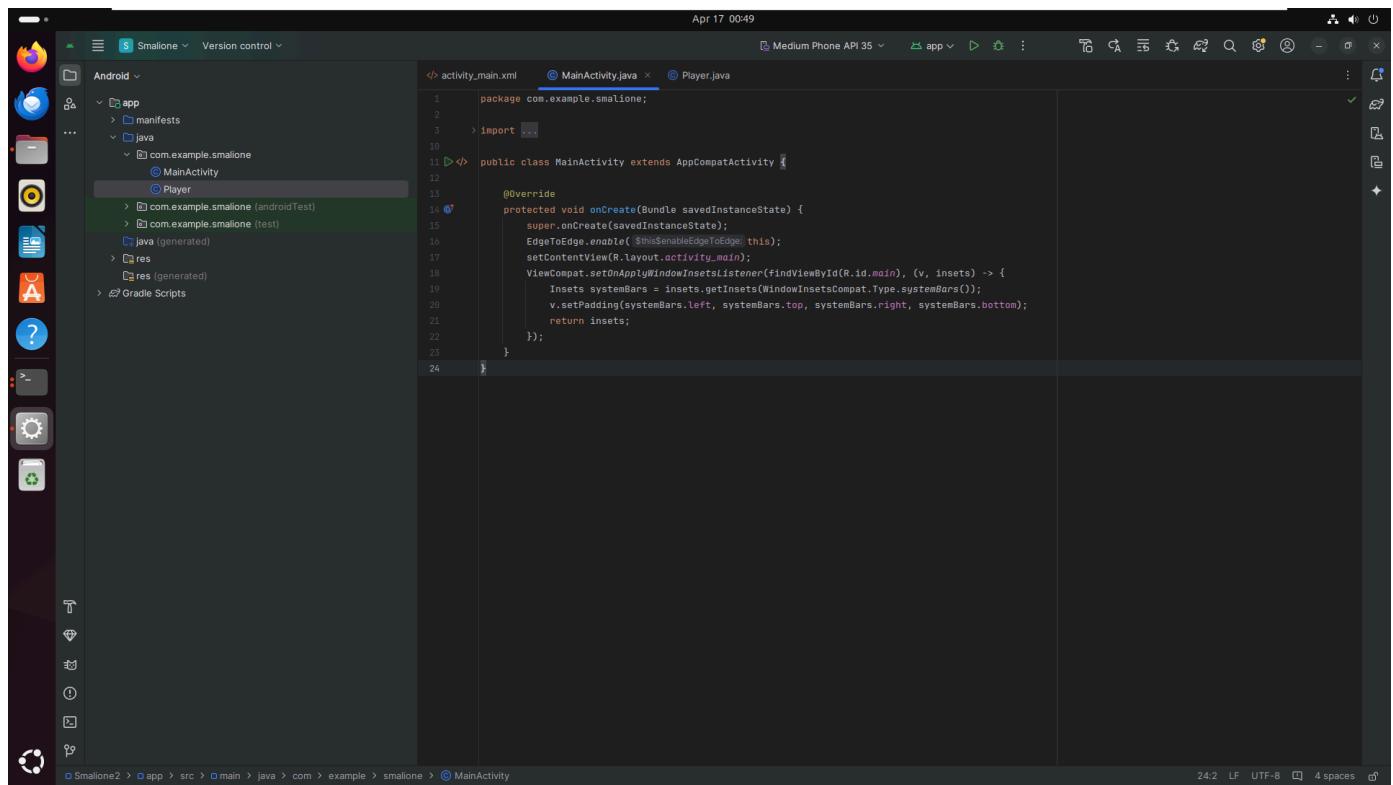


4- Smali

51-Introduction

هنا احنا هننشء apk باستخدام android studio وبعد ما ننشء apk هنفكه باستخدام apktool علشان حوله smali code



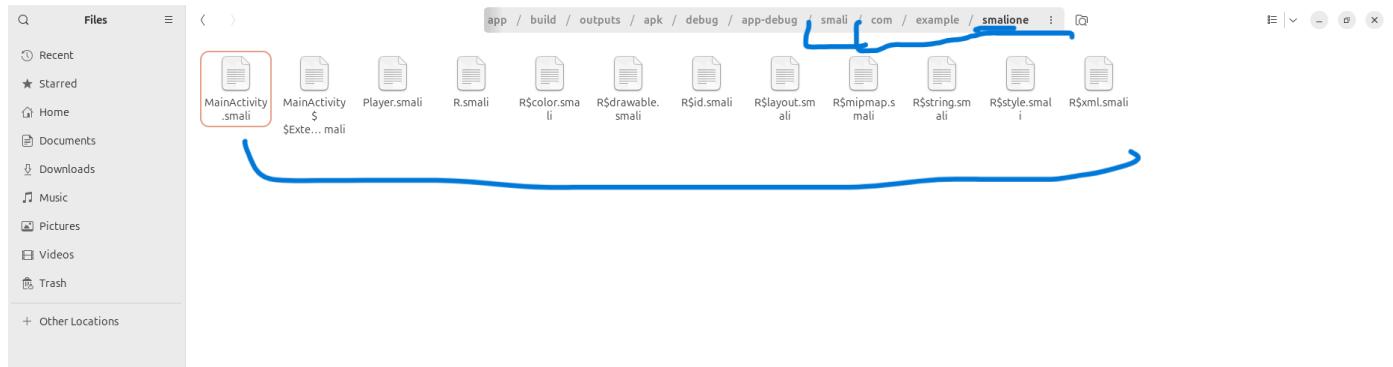
```
→ debug apktool d app-debug.apk
I: Using Apktool 2.11.0 on app-debug.apk with 4 threads
I: Baksmaling classes.dex...
I: Loading resource table...
I: Decoding file-resources...
I: Loading resource table from file:
/home/ubuntu/.local/share/apktool/framework/1.apk
I: Decoding values */* XMLs...
I: Decoding AndroidManifest.xml with resources...
```

دلوتي بعد ما فكنا هنلاقي كون smali اسمه folder وهنلاقي في folder ده بيكون من 3 عناصر هما

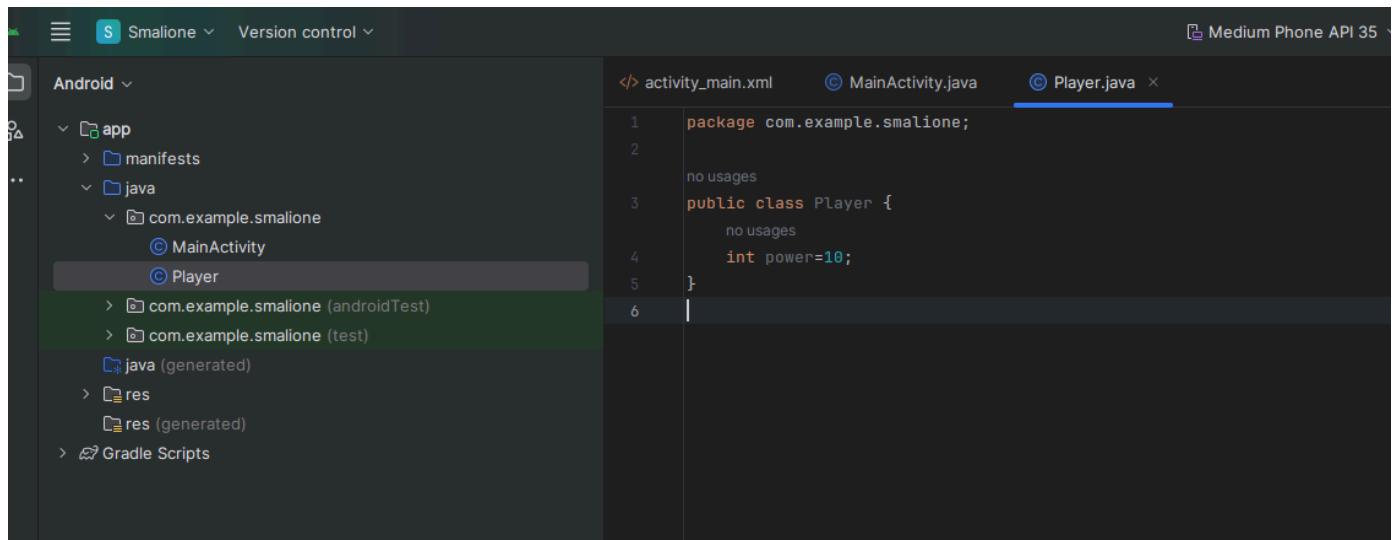
- android
- androidx
- com

خاصين ببنا التطبيق نفسه android, androidx ولكن اللي هنستخدمه هو com

هنا اه ملفات التطبيق نفسه مكتوبة بلغة smali تحت com



create simple class with java code and see it with smali code



smali code for java

```

.class public Lcom/example/smalione/Player;
.super Ljava/lang/Object;
.source "Player.java"

# instance fields
.field power:I

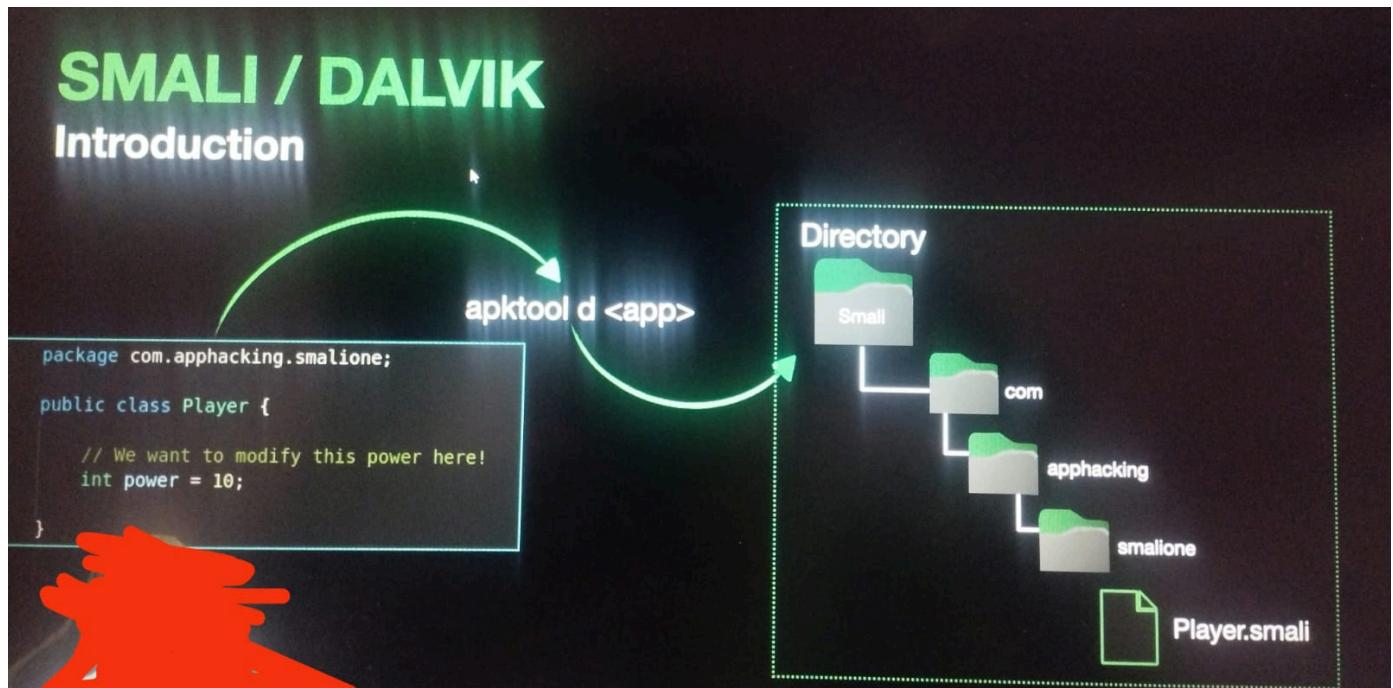
# direct methods
.method public constructor <init>()V
    .locals 1

    .line 3
    invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    .line 4
    const/16 v0, 0xa
    input v0, p0, Lcom/example/smalione/Player;->power:I

    return-void
.end method

```



لو بقى نشوف الفرق بين java و smali ممكن الاختلاف اللي نلاحظه ان القيمة اتعرفت في smali قيمه hexa --> const/16

v0.0xa

Feature	Java Code	Smali Code
Class Declaration	public class Player	.class public Lcom/example/smali_code/Player;
Inheritance	Implicit (extends Object)	Explicit: .super Ljava/lang/Object;
Field	int score = 0;	.field power:I (field named power)
Field Initialization	score = 0; (directly assigned)	input v0, p0, Lcom/example/smali_code/Player;->power:I (done in the constructor)
Constructor	Implicit default constructor	Explicit constructor with invoke-direct for super()
Syntax	High-level, readable	Low-level, assembly-like

52- Smali - Patching

دلوتي بقى من اللي اتعلمناه عرفنا ان القيمة بتبقى في smali بتبقى hexa لو بقى عاوزين نغير في قيمة في تطبيق معين من خلال كود smali

steps to change any thing on apk app using smali

1- decompile apk using apktool d app.apk

2- open smali file and change the value

3- build application

apktool b folder

4- signing application using this method

- 1- Generate Key

- keytool -genkey -v -keystore ~/android-app-hack.keystore -alias alias_name -keyalg RSA -keysize 2048 -validity 365

```
keytool -genkey -v -keystore ~/android-app-hack.keystore -alias alias_name -keyalg RSA -keysize 2048 -validity 365
```

- 2- configer lines

- zipalign -v 4 sieve.apk out.apk

```
zipalign -v 4 sieve.apk out.apk
```

- 3-sign app with apksigner
- apksigner sign --ks-key-alias my_key -ks ~/android-app-hack.keystore out.apk

```
apksigner sign --ks-key-alias my_key -ks ~/android-app-hack.keystore
out.apk
```

Solving challenge

1-install app to emulator

```
→ first_chellange_smali adb install smaliOne.apk
```

2-do logcat to show the value of power

```
→ first_chellange_smali adb logcat |grep power
```

```
17 02:31:22.944 847 891 I ActivityManager: Skipped duplicated ANR, pfd=1107 executing service com.android.systemui/com.google.android.systemui.p
rvice, waited 27850ms
17 02:31:27.134 2978 2978 I System.out: Player power = 10
17 02:31:46.166 847 847 W Looper : Slow delivery took 248ms main h=com.android.server.power.Notifier$NotifierHandler c=null m=7
17 02:32:13.879 847 847 W Looper : Slow delivery took 221ms main h=com.android.server.power.Notifier$NotifierHandler c=null m=7
17 02:32:16.900 3975 4244 W PowerStateManagerModule: <DWB> Context manager broadcast triggered power state fence restoration
17 02:32:25.496 3975 4244 W PowerStateManagerModule: <DWB> Context manager broadcast triggered power state fence restoration
17 02:32:46.830 847 879 W Looper : Slow delivery took 440ms android.fg h=android.os.Handler c=com.android.server.power_hint.HintManagerServ
ice1@bce3581 m=0
17 02:33:05.065 847 847 W Looper : Slow delivery took 233ms main h=com.android.server.power.Notifier$NotifierHandler c=null m=7
```

here power = 10 we need to change it to 20 using smali

3-remove app from emulator

```
→ ~ adb uninstall com.apphacking.smaliOne
```

Success

4-decompile app with apktool

```
→ first_chellange_smali apktool d smaliOne.apk
```

5-open the file of class player on smali and change the value from 0xa to 0x14

```
→ apphacking cd smaliOne
→ smaliOne ls
BuildConfig.smali      'R$color.smali'          'R$ipmap.smali'
MainActivity.smali    'R$dimen.smali'          'R$plurals.smali'
Player.smali           'R$drawable.smali'        'R$string.smali'
'R$animator.smali'    'R$id.smali'             'R$styleable.smali'
'R$anim.smali'         'R$integer.smali'        'R$style.smali'
'R$attr.smali'         'R$interpolator.smali'   'R$xml.smali'
'R$bool.smali'         'R$layout.smali'         R.smali
```

before

```
method public constructor <init>()V
    .locals 1

    .line 3
    invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    .line 5
    const/16 v0, 0xa

    iput v0, p0, Lcom/apphacking/smaliOne/Player;->power:I
    [ Read 23 lines ]
```

after

```
.line 3
invoke-direct {p0}, Ljava/lang/Object;-><init>()V

.line 5
const/16 v0, 0x14

iput v0, p0, Lcom/apphacking/smaliOne/Player;->power:I
```

6- build the file after change the value

```
→ first_chellange_smali apktool b smaliOne
I: Using Apktool 2.11.0 on smaliOne.apk with 4 threads
I: Checking whether sources have changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources have changed...
```

the new apk on I: Built apk into: **smaliOne/dist/smaliOne.apk**

لو جينا بقى نعمله install مش هيرضي علشان احنا غيرنا فيه د

```
db: install requires an argument
: dist adb install smaliOne.apk
Performing Streamed Install
db: failed to install smaliOne.apk: Failure [INSTALL_PARSE_FAILED_NO_CERTIFICATES: Failed to collect certificates from /data/app/vmdl1815250047.tmp/base.apk: Attempt to get length of null array]
: dist
```

7-signing the application

1- create the key

```
→ dist keytool -genkey -v -keystore ~/android-app-hack.keystore -keysize
2048 -alias smali_chellange -keyalg RSA -validity 365
Enter keystore password:
```

Enter the distinguished name. Provide a single dot (.) to leave a **sub-component** empty **or** press ENTER to use the default value in braces.

check it with list-key

```
→ dist Keytool -list -keystore ~/android-app-hack.keystore
Enter keystore password:
Keystore type: PKCS12
Keystore provider: SUN

Your keystore contains 5 entries

alias_name, Mar 17, 2025, PrivateKeyEntry,
Certificate fingerprint (SHA-256): 06:3A:E7:6F:0F:2A:E3:02:A0:BA:59:04:B7:06:14:98:95:D8:87:E2:76:D7:3A:F8:0A:A0:BD:7D:20:19:D7:01
my_ass, Apr 12, 2025, PrivateKeyEntry,
Certificate fingerprint (SHA-256): 51:D6:DE:0D:63:5A:EA:DB:9D:17:C6:68:BF:5E:73:95:22:C2:BD:99:C3:6B:40:D1:5B:2D:1E:5E:09:D3:20:12
my_key, Mar 18, 2025, PrivateKeyEntry,
Certificate fingerprint (SHA-256): 10:11:7A:C5:55:C1:24:1A:49:DA:36:66:51:51:0B:B5:8F:B5:4B:CF:48:EF:AB:4C:ED:5D:60:A2:49:DC:48:CB
smali_chellange, Apr 17, 2025, PrivateKeyEntry,
Certificate fingerprint (SHA-256): F4:F4:D4:B7:94:89:1C:26:7B:C3:7F:E3:16:90:21:E6:37:59:36:18:BE:1F:44:CB:96:95:69:1B:59:D7:1C:A9
udemy, Mar 17, 2025, PrivateKeyEntry,
Certificate fingerprint (SHA-256): 88:4F:DB:C5:E5:AE:81:7D:FA:59:55:D3:36:0D:DA:DE:9E:1A:C3:7C:13:05:ED:78:E6:0E:9F:9B:04:5B:C4:4D

Warning:
<alias_name> uses the SHA1withRSA signature algorithm which is considered a security risk.
<my_key> uses the SHA1withRSA signature algorithm which is considered a security risk.
<udemy> uses the SHA1withRSA signature algorithm which is considered a security risk.
→ dist
```

2- configer the lines

```
→ dist zipalign -v 4 smaliOne.apk out.apk
Verifying alignment of out.apk (4)...
    41 classes.dex (OK - compressed)
```

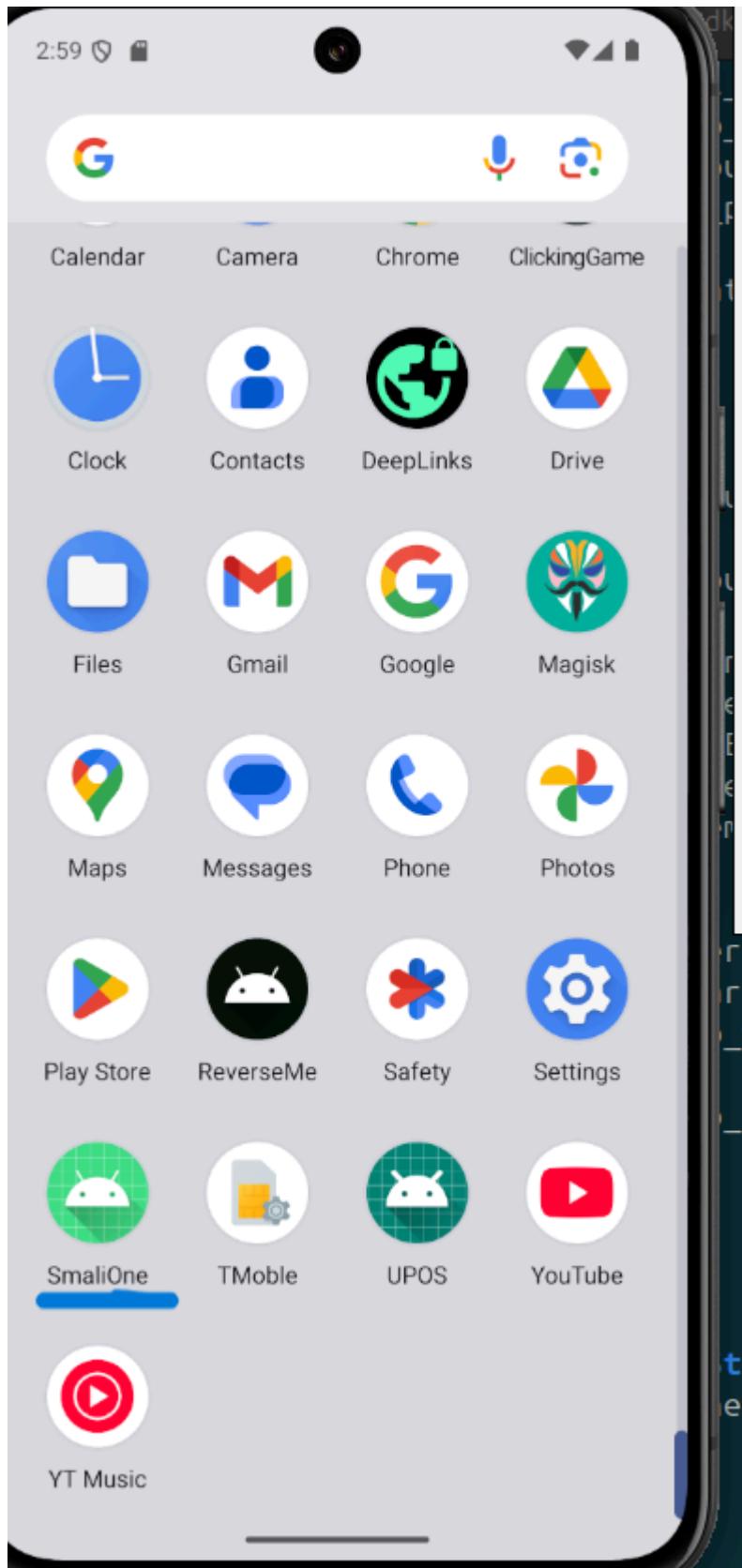
3- sign with

```
→ dist apksigner sign --ks-key-alias smali_chellange --ks ~/android-app-hack.keystore entry smali_chellange does not contain a key
→ dist apksigner sign --ks-key-alias smali_chellange --ks ~/android-app-hack.keystore out.apk
WARNING: A restricted method in java.lang.System has been called
WARNING: java.lang.System::loadLibrary has been called by org.conscrypt.NativeLibraryUtil in an unnamed module (file:/home/ubuntu/Android/Sdk/build-tools/35.0.1/lib/apksigner.jar)
WARNING: Use --enable-native-access=ALL-UNNAMED to avoid a warning for callers in this module
WARNING: Restricted methods will be blocked in a future release unless native access is enabled
keystore password for signer #1:
→ dist
```

8- upload the apk after change it and check it and we success to change the value because when we upload he give me success

```
→ dist adb install out.apk
Performing Incremental Install
Serving...
All files should be loaded. Notifying the device.
Success ↵
Install command complete in 892 ms
→ dist
```

9- check the app is installed or not



10- check the logcat the value is changed or not

finally the value is changed to 20

```
04-17 02:43:08.729 1062 1062 I BrowserPlayerWrapper: getChildrenItems: Connected to browsable Player: com.google.android.bluetooth
04-17 02:43:14.459 1062 1062 I AvrcpBrowsablePlayerConnector: Successfully connected to 0 browsable players.
04-17 02:43:14.687 1062 1062 I MediaPlayerList: init: Browsable Player list size is 0
04-17 02:43:29.590 2502 2502 I ExoPlayerImpl: Init b4f33c1 [AndroidXMedia3/1.6.0-beta01] [emu64xa, sdk_gphone64_x86_64, Google, 35]
04-17 02:43:29.623 2502 2502 I ExoPlayerImpl: Init 6f5f8c0 [AndroidXMedia3/1.6.0-beta01] [emu64xa, sdk_gphone64_x86_64, Google, 35]
04-17 03:02:48.276 6171 6171 I System.out: Player power = 20
```

54-Registers

هنا هنشرح ازاي smali بتشتغل وهنشرح شوية حاجات مهمة وهنبدأ بأول حاجة وهي ايه هي --> high level language ودي عباره عن ان الانسان بيقدر يفهمها وستخدمها واسهل في التعامل معها زي المثال ده كده :

هنا دي عباره عن function بتجمع رقمين ودي بكتوبه ايه ان الانسان يقدر يفهمها عادي ونروح لموضوع تاني وهو CPU

```
int add(int a,int b) {
    int sum=0;
    sum=a+b;
    return sum
}
```

نجي لـ CPU هنا هو مش بيفهم assembly هو بتترجم ل high level language و هو يقدر يفهم وتحول علشان OS يقدر يفهمها بس خلينا نتكلم عن CPU هو بيقي جواه register كل register يخزن القيمة اللي هي زي sum , a , b وهكذا وبيقدر يتعامل معها علشان يخزنها في memory عن طريق stack وبيستخدم LIFO (Last-In-First-Out) وهكذا نيجي بقى smali بتعامل كأنها زي register بالظبط وده لما نستخدم apktool يكون كود smali وبيقسم حاجتين local و parameters وان

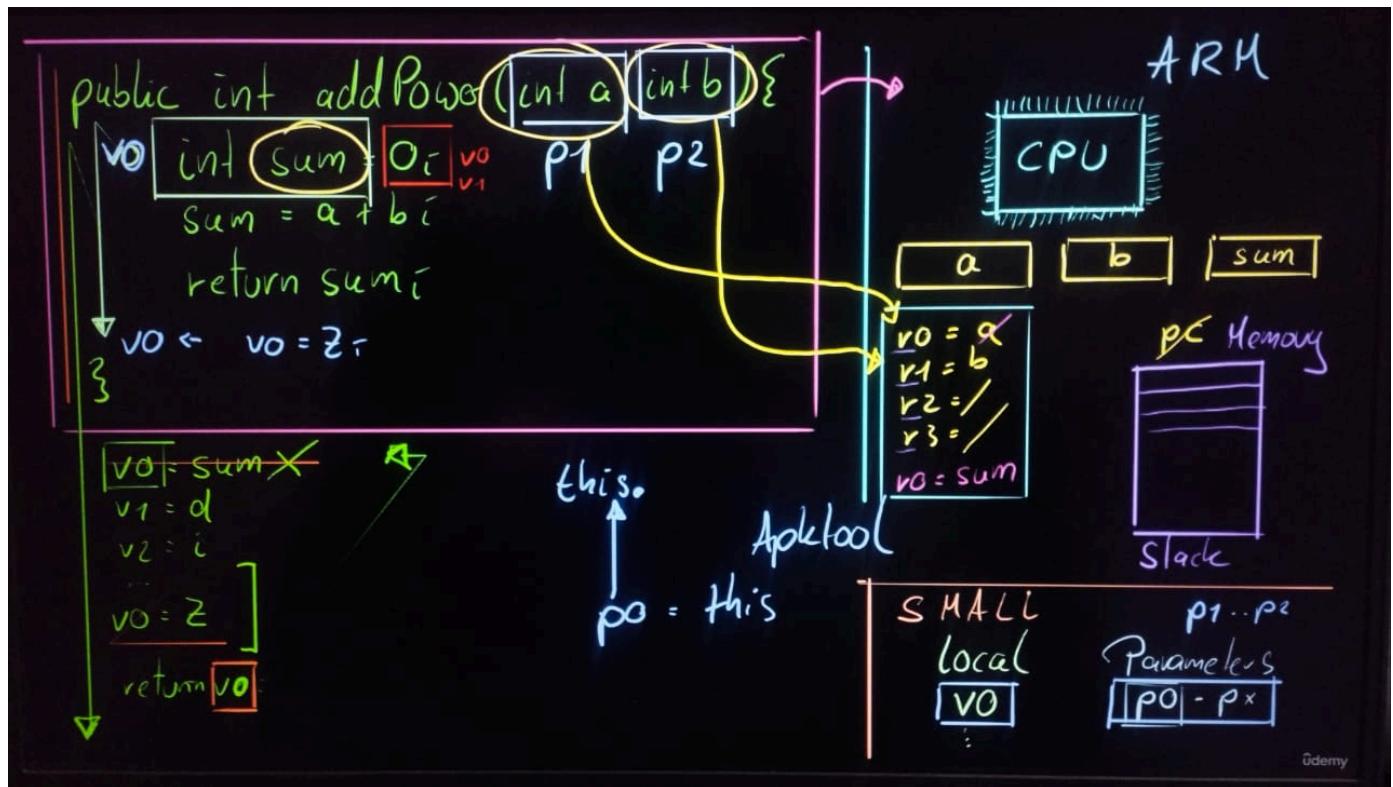
local --> sum --> on smali (V0 -->Vn)

parameters --> a ,b on smali (P0 --> Pn) and P0 assign to this on java

جدول مقارنة بين **Smali** و **Java**

العنصر	Java	Smali
الكائن الحالي	this	p0
المعامل الأول	a	p1
المعامل الثاني	b	p2
المتغير المحلي	sum	v0

وهنا بقى الصورة دي بنوضح كل حاجة



if we create the add function by java and build app and decompile it to show the smali code

```

package com.example.smali_code;

public class Player {
    int score=0;
    public int addPower(int a,int b) {
        int sum=0;
        sum=a+b;
        return sum;
    }
}

```

.method public addPower()I ---> public int addPower

.param p1, "a"

.param p2, "b" two mean parameters for function addPower

const/4 v0, 0x0 ---> v0=sum and 0x0 =0

```
1.class public Lcom/example/smali_code/Player;
2.super Ljava/lang/Object;
3.source "Player.java"
4.
5.# instance fields
6.field score:I
7.
8.
9.
10.# direct methods
11.method public constructor <init>()V
12    .locals 1
13.
14    .line 3
15    invoke-direct {p0}, Ljava/lang/Object;-><init>()V
16.
17    .line 4
18    const/4 v0, 0x0
19.
20    input v0, p0, Lcom/example/smali_code/Player;->score:I
21.
22    return-void
23.end method
24.
25.
26# virtual methods
27.method public addPower(II)I
28    .locals 1
29    .param p1, "a"    # I
30    .param p2, "b"    # I
31.
32    .line 6
33    const/4 v0, 0x0
34.
35    .line 7
36    local v0, "sum":I
37    add-int v0, p1, p2
38.
39    .line 8
40    return v0
41.end method
```

شرح الكود خطوة بخطوة

1. تعريف الدالة

```
.method public addPower(II)I
```

- **public**: الدالة عامة ويمكن استدعاؤها من خارج الصف.

- **addPower**: اسم الدالة.

- **(II)I**:

○ الجزء **II** يشير إلى أن الدالة تأخذ مدخلين من النوع **int**.

○ الجزء **I** يشير إلى أن الدالة ترجع قيمة من النوع **int**.

2. عدد المتغيرات المحلية

```
.locals 1
```

- يحدد عدد المتغيرات المحلية المستخدمة في هذه الدالة.

- هنا يتم استخدام متغير محلي واحد فقط، وهو **v0**.

3. تعريف المعاملات

```
.param p1, "a"    # I
.param p2, "b"    # I
```

- **p1** و **p2** هما المعاملات التي تستقبلها الدالة.

- **"a"** و **"b"** هما أسماء وصفية لهذه القيم لتوضيح دورها.

- النوع **I** يشير إلى أن هذه القيم من النوع **int**.

4. تهيئة المتغير المحلي

```
.line 6  
const/4 v0, 0x0
```

- يتم تهيئة المتغير **v0** بقيمة صفر.
- **const/4** تُستخدم لتخزين القيم الصغيرة (0-15) بكفاءة.

5. عملية الجمع

```
.line 7  
.local v0, "sum":I  
add-int v0, p1, p2
```

- يتم استخدام التعليمة **add-int** لإضافة المعاملين **p1** و **p2** و تخزين النتيجة في **v0**.
- **v0** و **a** يُستخدم هنا لتمثيل مجموع القيم **b**.

6. إرجاع النتيجة

```
.line 8  
return v0
```

- إلى الدالة المستدعاة **v0** يتم إرجاع النتيجة المخزنة في.

القيم اللي هي **.line**. دي يعني بقىول رقم **line** اللي في **java** بتحده في **smali**

```
.class public Lcom/example/small_code/Player;
.super Ljava/lang/Object;
.source "Player.java"

# instance fields
.field score:I

# direct methods
.method public constructor <init>()V
    .locals 1
    .line 3
    invoke-direct {p0}, Ljava/lang/Object;-><init>()V
    .line 4
    const/4 v0, 0x0

    input v0, p0, Lcom/example/small_code/Player;->score:I

    return-void
.end method

# virtual methods
.method public addPower(II)I
    .locals 1
    .param p1, "a"      # I
    .param p2, "b"      # I
    .line 6
    const/4 v0, 0x0

    .line 7
    .local v0, "sum":I
    add-int v0, p1, p2

    .line 8
    return v0
```

```
1 package com.example.smali_code;
2
3     2 usages
4     public class Player {
5         1 usage
6         int score=0;
7         no usages
8         public int addPower(int a, i
9             int sum=0;
10            sum=a+b;
11            return sum;
12        }
13    }
```

يبقى هنا local variable بنعرف ب local. و parameter variable بنعرفهم p1 to pn . وببدأ بقيمة local وان

.const ثابتة قيمة تعرف عاوز وان

another example this code with smali so we want to convert it to java code

The screenshot shows the contents of a file named "Player.smali". The code is written in the Smali assembly language, which is used to represent Java bytecode. The code defines a public method named "addSomePower" that takes three integer parameters (a, b, c) and returns an integer. The implementation calculates the sum of a, b, and c, then multiplies it by a factor of 2. The assembly code includes comments for virtual methods, locals, parameters, and various assembly instructions like const, add-int, and return.

```
1 # virtual methods
2 .method public addSomePower(III)I
3     .locals 2
4     .param p1, "a"    # I
5     .param p2, "b"    # I
6     .param p3, "c"    # I
7
8     .line 17
9     const/4 v0, 0x2
10
11    .line 20
12    .local v0, "factor":I
13    add-int v1, p1, p2
14
15    add-int/2addr v1, p3
16
17    add-int/2addr v1, v0
18
19    .line 22
20    .local v1, "sum":I
21    return v1
22 .end method
```

java code

خلي بالك : كده بقولك ان v0 هيكون فيها المتغير اللي اسمه factor و برضه هكذا مع sum

```
public int addSomePower(int a,int b,int c) {
    int factor=2;
    sum=a+b;
    sum+=c;
    sum+=factor;
    return sum;
} this is my analyze
the real code
public int addSomePower(int a,int b,int c) {
    int factor=2;
    int sum;
    sum=a+b+c+factor;
    return sum;
}
```

55-Types

Syntax	Meaning
V	Void
B	Byte
Z	Boolean
C	Char
I	Int
J	Long
F	Float
D	Double
[descriptor (e.g [B ---> Byte	array
L	Class name
S	Short

```
# static data
.field public static final SPRITE_NAME:Ljava/lang/String;="alien"
.field public static final SPRITE_NAME_BOSS:Ljava/lang/String;="alienboss"

this mean there are variables:
1- public static final SPRITE_NAME="alien"    -> Ljava/lang/String --> the
data type is string
2- public static final SPRITE_NAME_BOSS="alienboos"  Ljava/lang/String -->
the data type is string
```

از ای نعرف اسم class

smali code

```
.class public Lde/fgerbig/spacepng/components/ColorAnimatoin;
.super Lcom/artemis/Component;
.source "ColorAnimation.java"
```

java

```
public class ColorAnimation extends Componet{}
```

Inner classes / enum (\$)

in java

```
public class Animation{  
    public class PlayMode{  
        } //or  
        enum PlayMode{  
        }  
}
```

in smali

```
.filed public Play:Mode:Lcom/badlogic.gdx/graphics/g2d/Animation$PlayMode;
```

56- P0 - Register

دلوقي هر نوع نصيف function getpower ونخليها ترجع ليما قيمة power ب استخدام this ونشوفها في كود smali بعد ما نعمل له decompile

get function

```
public int getPower() {  
    return this.power;  
}
```

smali

```
method public getPower() I  
.locals 1  
  
.line 6  
iget v0, p0, Lcom/example/smali_code/Player;->power:I  
  
return v0  
.end method
```

دلوقي بقى خلينا نشرح كود smali

method public getPower()I : here I mean return integer	public int getpower(){
.locals 1	
.line 6 --> for line 6 on code java	line 6 return this.power;

method public getPower()I : here I mean return integer	public int getpower(){
iget v0,p0. Lcom/example/smali_code/Player; ->power:I	iget-->get value and v0 --> register , p0 --> this , class_name = player and -> for value on class
return v0	return

خلي بالك هنا في حاجة مهمة هو ان احنا مش معرفين **function** داخل **variable** ولكن في **smali** استخدم **locals 1** وهذا بيجي لشرح ان زي مثلاً لغة **assembly** ان **cpu** مش بيعامل مع قيم معينة ولكن بيعامل مع **power** وان القيم لازم تكون مخزنة في **register** وهذا هو استخدام **v0** كأنه **register** هيرجع قيمة **register** وهنا برد القيمه اللي هي **p0** علشان قولنا ان **this =p0** وجنبها بقى اسم **class** وبি�شاور على القيمه اللي جواه اللي هي **Power**

57-Delvik Opcodes

هنكلم ان operation code اللي في smali ودي زي word اللي في الدرس اللي فات اللي هي **Iget** ودول عباره عن العمليات اللي عياز تعملاها زي ما قولنا تحط قيمة في REGISTER وعمليات تانية زي **add,mul,....**,**add,mul** pdf في كل **Iget** operation مثلًا

52	iget vx, vy, field_id	Reads an instance field into vx. The instance is referenced by vy.	5210 0300 - iget v0, v1, Test2.i6:I // field@0003 Reads field@0003 into v0 (entry #3 in the field id table). The instance is referenced by v1.
53	iget-wide vx,vy,field_id	Reads an instance field into vx ¹ . The instance is referenced by vy.	5320 0400 - iget-wide v0, v2, Test2.I0:J // field@0004 Reads field@0004 into v0 and v1 registers (entry #4 in the field id table). The instance is referenced by v2.
54	iget-object vx,vy,field_id	Reads an object reference instance field into vx. The instance is referenced by vy.	iget-object v1, v2, LineReader.fis:Ljava/io/FileInputStream; // field@0002 Reads field@0002 into v1 (entry #2 in the field id table). The instance is referenced by v2.
55	iget-boolean vx,vy,field_id	Reads a boolean instance field into vx. The instance is referenced by vy.	55FC 0000 - iget-boolean v12, v15, Test2.b0:Z // field@0000 Reads the boolean field@0000 into v12 register (entry #0 in the field id table). The instance is referenced by v15.
56	iget-byte vx,vy,field_id	Reads a byte instance field into vx. The instance is referenced by vy.	5632 0100 - iget-byte v2, v3, Test3.bi1:B // field@0001 Reads the char field@0001 into v2 register (entry #1 in the field id table). The instance is referenced by v3.

57	iget-char vx,vy,field_id	Reads a char instance field into vx. The instance is referenced by vy.	5720 0300 - iget-char v0, v2, Test3.ci1:C // field@0003 Reads the char field@0003 into v0 register (entry #3 in the field id table). The instance is referenced by v2.
----	-----------------------------	--	---

ثال لکود smali و نشوف الی جواه

```
.method public addPower(II)I
.locals 1
.param p1,"a" #I
.param p2,"b" #I
.line 13
.const/v4 v0,0x0
.line 15
.local v0,"sum":I
add-int v0,p1,p2
.line 17
return v0
.end method
```

هنا اب add-int بتجمع رقمين وتحطّم في register ويرده هنا const. دي عباره عن Ocode

90	add-int vx,vy,vz	Calculates vy+vz and puts the result into vx.	9000 0203 - add-int v0, v2, v3 Adds v3 to v2 and puts the result into v0 ⁴ .
----	---------------------	---	--

12	const/4 vx,lit4	Puts the 4 bit constant into vx	1221 - const/4 v1, #int2 Moves literal 2 into v1. The destination register is in the lower 4 bit in the second byte, the literal 2 is in the higher 4 bit.
----	--------------------	---------------------------------	---

java code

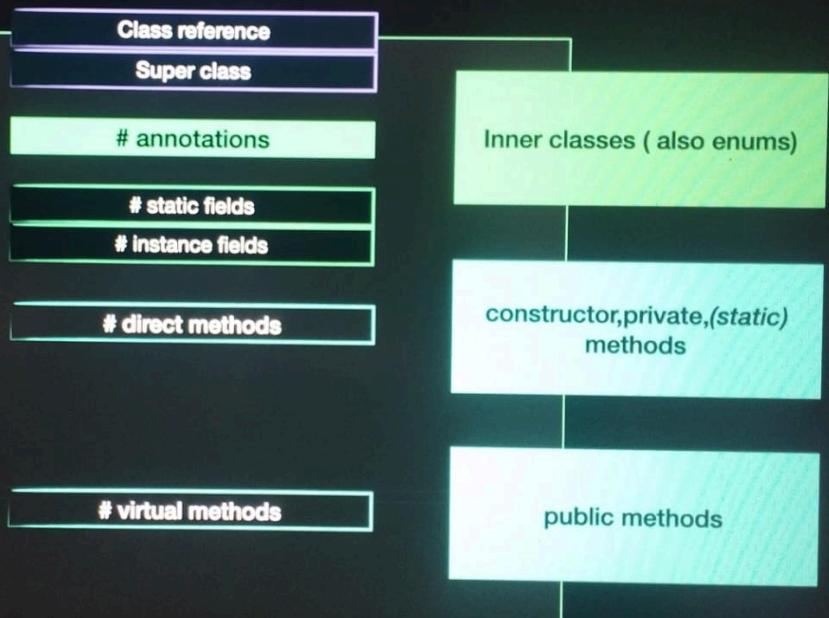
```
public int addPower(int a,int b) {
    int sum=0;
    sum=a+b;
    return sum;
}
```

58- Smali File Structure

SMALI File Structure

Dalvik Opcodes

```
1 .class public Lcom/apphacking/smalione/Player;
2 .super Ljava/lang/Object;
3 .source "Player.java"
4
5 # static fields
6 .field public static lives:I
7
8
9 # instance fields
10 .field power:I
11
12 .field shield:Z
13
14
15 # direct methods
16 .method public constructor <init>()V
17 .locals 1
18
19 .line 3
20 invoke-direct {p0}, Ljava/lang/Object;::>init<()
21
22 .line 5
23 const/16 v0, 0xc
24
25 input v0, p0, Lcom/apphacking/smalione/Player;::power:I
26
27 .line 7
28 const/4 v0, 0x0
29
30 input-boolean v0, p0, Lcom/apphacking/smalione/Player;::shield:Z
31
32 return-void
33
34 .end method
35
36
37 # virtual methods
38 .method public addPower(III)I
39 .locals 2
40 .param p1, "a" # I
41 .param p2, "b" # I
42 .param p3, "factor" # I
43
44 .line 13
45 const/4 v0, 0x0
46
```



1. تعريف الكلاس (Class Reference)

- `.class` يبدأ الملف بتحديد الكلاس باستخدام
- `.(public, final)` يتم تحديد اسم الكلاس مع الـ modifiers (مثل

```
.class public Lcom/apphacking/smalione/Player;
```

2. الكلاس الأب (Super Class)

- يحدد الكلاس الأب الذي يرث منه الكلاس الحالي باستخدام `.super`.

```
.super Ljava/lang/Object;
```

3. التعليقات التوضيحية (Annotations)

- يمكن أن تحتوي الملفات على تعليقات توضيحية تُستخدم لتوثيق أو إضافة خصائص إضافية.

4. الحقول (Fields)

أ. الحقول الثابتة (Static Fields)

- يتم تعيين المتغيرات العامة أو الثابتة الخاصة بالكلاس.

```
.field public static lives:I
```

ب. الحقول غير الثابتة (Instance Fields)

- تُعرف المتغيرات الخاصة بكل نسخة من الكائن.

```
.field power:I.field shield:Z
```

- `I`: تشير إلى نوع Integer.

- `Z`: تشير إلى نوع Boolean.

5. الدوال (Methods)

A. الدوال المباشرة (Direct Methods)

- أو الدوال الخاصة والدوال الثابتة Constructors تتضمن الـ.

```
.method public constructor <init>() V
.locals 1

invoke-direct {p0}, Ljava/lang/Object;-><init>() V
return-void
.end method
```

- `.locals 1`: يحدد عدد السجلات المستخدمة في الدالة.
- `invoke-direct`: يستدعي الدالة من الكلاس الأب.

B. الدوال الافتراضية (Virtual Methods)

- تُستخدم للدوال العامة التي يمكن استدعاؤها على الكائنات.

```
.method public addPower(III) I
.locals 2

const/4 v0, 0x0
return v0
.end method
```

- `addPower(III) I`: وتُعيد قيمة من نوع Integer تشير إلى دالة تأخذ 3 معامالت من نوع Integer.
- `const/4`: تُستخدم لتحميل قيمة ثابتة إلى سجل.

6. Inner Classes:

- يتم تضمينها في نفس الملف أو كملف مستقل، إذا كانت هناك كلاسات داخلية أو

ال التقسيم:

- **Static Fields**: الحقول الثابتة.
- **Instance Fields**: الحقول الخاصة بالكائنات.
- **Direct Methods**: الدوال الخاصة Constructors تتضمن الـ.
- **Virtual Methods**: تشمل الدوال العامة.

هنا هنتعلم ازاي نفهم code smali ونحاول نفهم اللي جواه ونقارنه بـ java code

1- class initializing

```
.class public Lcom/apphacking/smalione/Player;
.super Ljava/lang/Object;
.source "Player.java"
```

```
mean : public class Player extends Object() {}
```

2- initializing static and instance variable

```
# static fields
.field public static lives:I
```

```
# instance fields
```

```
.field power:I
```

```
.field shield:Z
```

```
mean :
```

```
public int lives;
```

```
int power;
```

```
boolean shield;
```

3- initializing constructor to set values for static and instance variable

```
.method public constructor <init>()V
    .registers 2
```

```
.line 3
```

```
invoke-direct {p0}, Ljava/lang/Object;-><init>()V
```

```
.line 5
```

```
const/16 v0, 0xc
```

```
input v0, p0, Lcom/apphacking/smalione/Player;->power:I
```

```
.line 7
```

```
const/4 v0, 0x0
```

```
input-boolean v0, p0, Lcom/apphacking/smalione/Player;->shield:Z
```

```
return-void
```

```
.end method
```

```
mean in line 5 --> power=12
```

```
in line 7 --> change the value from 12 to 0 mean (false) --> shield=false;
```

4-initializing public function

```
.method public addPower(III)D
```

```
.registers 9
```

```
.param p1, "a"      # I
```

```
.param p2, "b"      # I
```

```
.param p3, "factor"    # I
```

```
.line 13
```

```
const/4 v0, 0x0
```

```
.line 15
```

```
.local v0, "sum":I
```

```
const-wide v1, 0x400266666666666L      # 2.3
```

```
.line 17
```

```
.local v1, "bias":D
```

```
add-int v3, p1, p2
```

```
mul-int v3, v3, p3
```

```
.line 19
```

```
.end local v0      # "sum":I
```

```
.local v3, "sum":I
```

```
add-int/lit8 v3, v3, -0x1
```

```
.line 21
```

```
div-int p1, v3, p2
```

```
.line 23
```

```
xor-int v0, v3, p1
```

```
.line 25
```

```
.end local v3      # "sum":I
```

```
.restart local v0      # "sum":I
```

```
int-to-double v3, v0
```

```
return-wide v3
```

```

.end method

mean

1- .method public addPower(III)D -->    public Double addPower(int a,int
b,int factor)

2-      .param p1, "a"      # I
      .param p2, "b"      # I
      .param p3, "factor"   # I

      take three parameter a, b, factor and the data type for each one
is int

3- in line 13 : create register v0 and the value of it =0

4- in line 15 : set the value of v0 to sum

5- const-wide v1, 0x400266666666666L      # 2.3 -->  set value of data type
long to register v1

6- in line 17 : create register v1      =bias the data type of it is double
v1=2.3

7-add-int v3, p1, p2 --> v3 =p1+p2  --> v3=a+b

8-mul-int v3, v3, p3      --> v3=v3*p3      ---> v3=v3*factor           v3 became v3=
(a+b)*factor

9-.local v3, "sum":I --> v3=sum and the data type is int

10- add-int/lit8 v3, v3, -0x1 --> v3=v3-1 ---> v3 became v3=
(((a+b)*factor)-1)

11- in line 21 --> p1=v3/p2 ---> a=sum/b

12- in line 23 --> v0=sum^p1  --> v0 = sum^a  --> xor  if the difference is
1 else 0

13-      int-to-double v3, v0 --> convert the value from v0 int to v3 to be
double

14- return v3 (double)

```

if we set the number for each variable --> a=1 , b=33 , factor=7 the final result will be 234

double addPower(int p1, int ³³ p2, int ⁷ p3) {
 v0 = 0 ✓
 v1 = 2. ³ ✓
 v3 = ³⁴ $p_1 + p_2$ ✓
 v3 = $v_3 \times p_3$ ✓
 $v_3 = v_3 - 1$ ✓
 $v_3 = v_3 / p_2$ ✓
 $v_0 = v_3 ^ p_1$ ✓
 v3 = (double) v0 ✓
 v0 = 237 ✓
 v0 = 234 (double) ✓
 return v3 ✓
 }

this code by java is

```

package com.apphacking.smalione;

/* loaded from: classes.dex */
public class Player {
    public static int lives;
    int power = 12;
    boolean shield = false;

    public double addPower(int a, int b, int factor) {
        int sum = ((a + b) * factor) - 1;
        int a2 = sum / b;
        return sum ^ a2;
    }
}

```

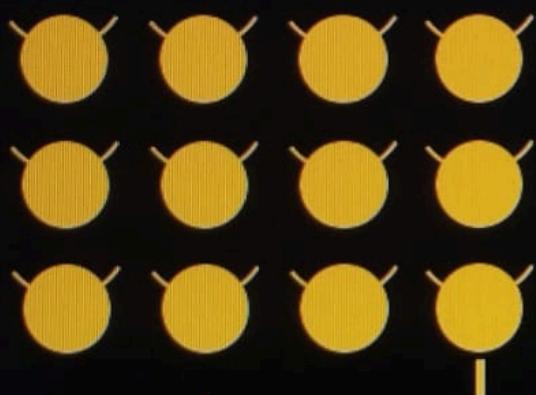
62- Oragne Belt chellange

لوقتي هو عاوز في التحدي دي ان اغير قيمة Lives من 5 الى 7 ولكن باستخدام كود smali

Can we increase

This game is way too

Lives	Score	High Score
5	1715	9005

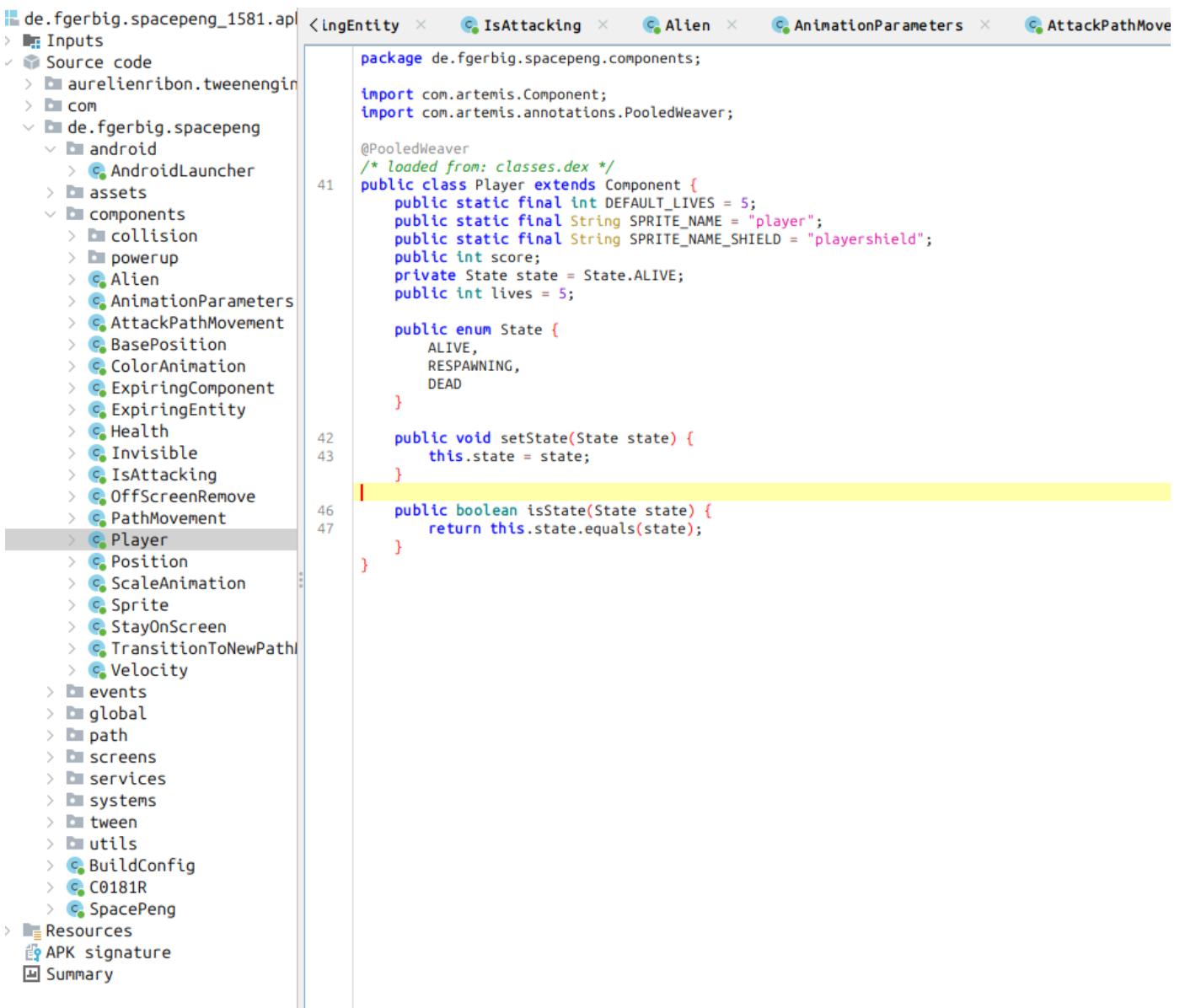


1- decompile apk app

```
→ challange_smali apktool d de.fgerbig.spacepeng_1581.apk
I: Using Apktool 2.11.0 on de.fgerbig.spacepeng_1581.apk with 4 threads
I: Baksmaling classes.dex...
I: Loading resource table...
I: Decoding file-resources...
I: Decoding values /* XMLs...
I: Loading resource table from file:
/home/ubuntu/.local/share/apktool/framework/1.apk
I: Decoding AndroidManifest.xml with resources...
I: Regular manifest package...
```

2- open app with jadx-gui tool, and after open with jadx and analyze find the file contain the value of live

here file /components/player ---> the default value is 5



```
de.fgerbig.spacepeng_1581.apk
Inputs
Source code
  aurelienribon.tweenengine
  com
    de.fgerbig.spacepeng
      android
        AndroidLauncher
      assets
      components
        collision
        powerup
        Alien
        AnimationParameters
        AttackPathMovement
        BasePosition
        ColorAnimation
        ExpiringComponent
        ExpiringEntity
        Health
        Invisible
        IsAttacking
        OffScreenRemove
        PathMovement
        Player
        Position
        ScaleAnimation
        Sprite
        StayOnScreen
        TransitionToNewPath
        Velocity
      events
      global
      path
      screens
      services
      systems
      tween
      utils
      BuildConfig
      C0181R
      SpacePeng
Resources
APK signature
Summary

LingEntity × IsAttacking × Alien × AnimationParameters × AttackPathMove

package de.fgerbig.spacepeng.components;

import com.artemis.Component;
import com.artemis.annotations.PooledWeaver;

@PooledWeaver
/* loaded from: classes.dex */
public class Player extends Component {
    public static final int DEFAULT_LIVES = 5;
    public static final String SPRITE_NAME = "player";
    public static final String SPRITE_NAME_SHIELD = "playershield";
    public int score;
    private State state = State.ALIVE;
    public int lives = 5;

    public enum State {
        ALIVE,
        RESPawning,
        DEAD
    }

    public void setState(State state) {
        this.state = state;
    }

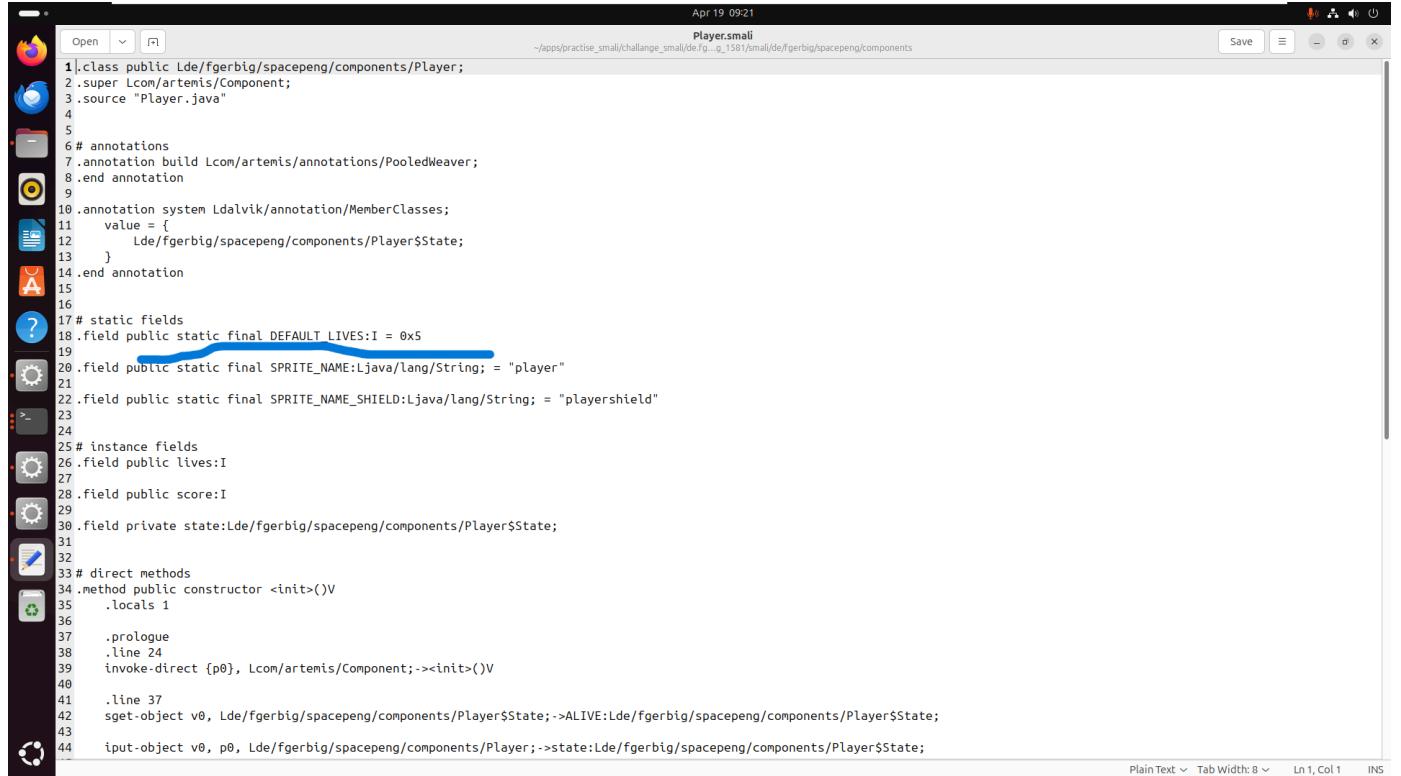
    public boolean isState(State state) {
        return this.state.equals(state);
    }
}
```

3-find the file location after decompile the app

```
+ components ls
Alien.smali      collision      Health.smali      PathMovement.smali  powerup      StayOnScreen.smali
AnimationParameters.smali  ColorAnimation.smali  Invisible.smali  'Player$State.smali'  ScaleAnimation.smali  TransitionToNewPathMovement.smali
AttackPathMovement.smali  ExpiringComponent.smali  IsAttacking.smali  Player.smali      'Sprite$Layer.smali'  Velocity.smali
BasePosition.smali    ExpiringEntity.smali    OffScreenRemove.smali Position.smali     Sprite.smali

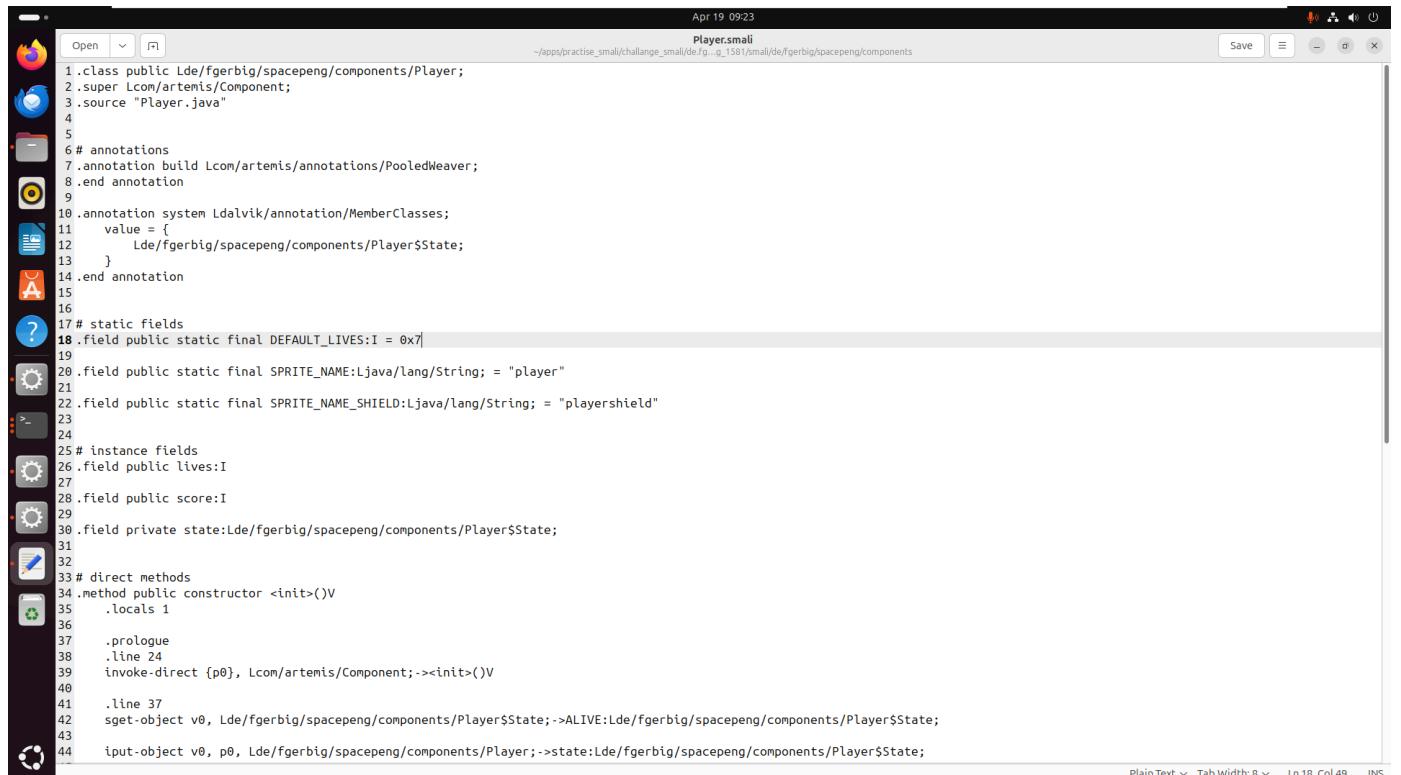
+ components pwd
/home/ubuntu/apps/practise_smali/challange_smali/de/fgerbig/spacepeng_1581/smali/de/fgerbig/spacepeng/components
+ components
```

4- open Player.smali and change the value, here the value by hexa = 0x5, so we will change it to 0x7 --> to be 7 by decimal



```
1.class public Lde/fgerbig/spacepeng/components/Player;
2.super Lcom/artemis/Component;
3.source "Player.java"
4
5
6# annotations
7.annotation build Lcom/artemis/annotations/PooledWeaver;
8.end annotation
9
10.annotation system Ldalvik/annotation/MemberClasses;
11    value = {
12        Lde/fgerbig/spacepeng/components/Player$State;
13    }
14.end annotation
15
16
17# static fields
18.field public static final DEFAULT_LIVES:I = 0x5
19
20.field public static final SPRITE_NAME:Ljava/lang/String; = "player"
21
22.field public static final SPRITE_NAME_SHIELD:Ljava/lang/String; = "playershield"
23
24
25# instance fields
26.field public lives:I
27
28.field public score:I
29
30.field private state:Lde/fgerbig/spacepeng/components/Player$State;
31
32
33# direct methods
34.method public constructor <init>()V
35    .locals 1
36
37    .prologue
38    .line 24
39    invoke-direct {p0}, Lcom/artemis/Component;-><init>()V
40
41    .line 37
42    sget-object v0, Lde/fgerbig/spacepeng/components/Player$State;->ALIVE:Lde/fgerbig/spacepeng/components/Player$State;
43
44    input-object v0, p0, Lde/fgerbig/spacepeng/components/Player;->state:Lde/fgerbig/spacepeng/components/Player$State;
```

change it to 0x7 and save it



```
1.class public Lde/fgerbig/spacepeng/components/Player;
2.super Lcom/artemis/Component;
3.source "Player.java"
4
5
6# annotations
7.annotation build Lcom/artemis/annotations/PooledWeaver;
8.end annotation
9
10.annotation system Ldalvik/annotation/MemberClasses;
11    value = {
12        Lde/fgerbig/spacepeng/components/Player$State;
13    }
14.end annotation
15
16
17# static fields
18.field public static final DEFAULT_LIVES:I = 0x7
19
20.field public static final SPRITE_NAME:Ljava/lang/String; = "player"
21
22.field public static final SPRITE_NAME_SHIELD:Ljava/lang/String; = "playershield"
23
24
25# instance fields
26.field public lives:I
27
28.field public score:I
29
30.field private state:Lde/fgerbig/spacepeng/components/Player$State;
31
32
33# direct methods
34.method public constructor <init>()V
35    .locals 1
36
37    .prologue
38    .line 24
39    invoke-direct {p0}, Lcom/artemis/Component;-><init>()V
40
41    .line 37
42    sget-object v0, Lde/fgerbig/spacepeng/components/Player$State;->ALIVE:Lde/fgerbig/spacepeng/components/Player$State;
43
44    input-object v0, p0, Lde/fgerbig/spacepeng/components/Player;->state:Lde/fgerbig/spacepeng/components/Player$State;
```

5- bulid the app with apktool

```
→ challange_smali apktool b de.fgerbig.spacepeng_1581
I: Using Apktool 2.11.0 on de.fgerbig.spacepeng_1581.apk with 4 threads
I: Checking whether sources have changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources have changed...
I: Building resources with aapt2...
I: Building apk file...
I: Importing assets...
I: Importing lib...
I: Importing unknown files...
I: Built apk into:
de.fgerbig.spacepeng_1581/dist/de.fgerbig.spacepeng_1581.apk
```

```
→ challange_smali cd de.fgerbig.spacepeng_1581
→ de.fgerbig.spacepeng_1581 cd dist
→ dist ls
de.fgerbig.spacepeng_1581.apk
→ dist
```

6- create key with alias name to use it in signing the app

```
→ dist keytool -genkey -v -keystore ~/android-app-hack.keystore -keysize 2048 -alias smalie_challange_key -keyalg RSA -validity 365
Enter keystore password:
Enter the distinguished name. Provide a single dot (.) to leave a sub-component empty or press ENTER to use the default value in braces.
What is your first and last name?
[Unknown]:
What is the name of your organizational unit?
[Unknown]:
What is the name of your organization?
[Unknown]:
What is the name of your City or Locality?
[Unknown]:
What is the name of your State or Province?
[Unknown]:
What is the two-letter country code for this unit?
[Unknown]:
Is CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correct?
[no]: y

Generating 2048-bit RSA key pair and self-signed certificate (SHA384withRSA) with a validity of 365 days
    for: CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
[Storing /home/ubuntu/android-app-hack.keystore]
→ dist
```

check the key was created

```
→ dist keytool -list -keystore ~/android-app-hack.keystore
Enter keystore password: →
Keystore type: PKCS12
Keystore provider: SUN

Your keystore contains 6 entries

alias_name, Mar 17, 2025, PrivateKeyEntry,
Certificate fingerprint (SHA-256): 06:3A:E7:6F:0F:2A:E3:02:A0:BA:59:04:B7:06:14:98:95:D8:87:E2:76:D7:3A:F8:0A:A0:BD:7D:20:19:D7:01
my_ass, Apr 12, 2025, PrivateKeyEntry,
Certificate fingerprint (SHA-256): 51:D6:DE:0D:63:5A:EA:DB:9D:17:C6:68:BF:5E:73:95:22:C2:BD:99:C3:6B:40:D1:5B:2D:1E:5E:09:D3:20:12
my_key, Mar 18, 2025, PrivateKeyEntry,
Certificate fingerprint (SHA-256): 10:11:7A:C5:55:C1:24:1A:49:DA:36:66:51:51:0B:B5:8F:B5:4B:CF:48:EF:AB:4C:ED:5D:60:A2:49:DC:48:CB
smali_chellange, Apr 17, 2025, PrivateKeyEntry,
Certificate fingerprint (SHA-256): F4:F4:D4:B7:94:89:1C:26:7B:C3:7F:E3:16:9D:21:E6:37:59:36:18:BE:1F:44:CB:96:95:69:1B:59:D7:1C:A9
smalie_challange_key, Apr 19, 2025, PrivateKeyEntry,
Certificate fingerprint (SHA-256): 09:AE:13:29:3D:2C:A5:7F:63:6E:8F:D4:F1:94:14:69:48:DD:0D:9F:3F:E4:C9:C4:0A:39:F5:1E:6C:AF:D2:FD
udemy, Mar 17, 2025, PrivateKeyEntry,
Certificate fingerprint (SHA-256): 88:4F:DB:C5:E5:AE:81:7D:FA:59:55:D3:36:0D:DA:DE:9E:1A:C3:7C:13:05:ED:78:E6:0E:9F:9B:04:5B:C4:4D

Warning:
<alias_name> uses the SHA1withRSA signature algorithm which is considered a security risk.
<my_key> uses the SHA1withRSA signature algorithm which is considered a security risk.
<udemy> uses the SHA1withRSA signature algorithm which is considered a security risk.
→ dist
```

7- configer the lines zipalign tool

```
ubuntu@AndroidAppHacking:~/apps/practise_small/challange_small/de.ferberbig.spacepong_1581/dist
zipalign -v 4 de.ferberbig.spacepong_1581.apk out.apk
Verifying alignment of out.apk (4)...
    41 classes.dex (OK - compressed)
889904 resources.arsc (OK)
891409 AndroidManifest.xml (OK - compressed)
892260 res/drawable-hdpi-v4/ic_launcher.png (OK)
898880 res/drawable-mdpi-v4/ic_launcher.png (OK)
902520 res/drawable-xhdpi-v4/ic_launcher.png (OK)
912584 res/drawable-xxhdpi-v4/ic_launcher.png (OK)
930916 assets/music/game.ogg (OK)
1552196 assets/music/game.ogg (OK)
2973252 assets/sound/boring.ogg (OK)
2985296 assets/sound/click.ogg (OK)
2999984 assets/sound/alienexplosion.ogg (OK)
2999384 assets/sound/alienexplosion_ogg (OK)
3006544 assets/sound/alienbossexplosion.ogg (OK)
3033568 assets/sound/playershot.ogg (OK)
3041956 assets/sound/alienbosshot.ogg (OK)
3047440 assets/sound/playerexplosion.ogg (OK)
3086492 assets/sound/alienshot.ogg (OK)
3094536 assets/skin/uiskin.png (OK)
3132131 assets/skin/default.fnt (OK - compressed)
3133500 assets/skin/default.png (OK)
3159732 assets/skin/uiskin.json (OK - compressed)
3160495 assets/skin/uiskin.atlas (OK - compressed)
3161120 assets/font/small.png (OK)
3195659 assets/font/large.fnt (OK - compressed)
3200092 assets/font/medium.fnt (OK - compressed)
3202488 assets/font/medium.png (OK)
3228282 assets/font/small.fnt (OK - compressed)
3230484 assets/font/large.png (OK)
3383836 assets/splash.jpg (OK)
3458892 assets/image/page0.png (OK)
4018662 assets/image/page0.pack (OK - compressed)
4019152 assets/COPYING (OK - compressed)
4031342 lib/armeabi-v7a/libgdx.so (OK - compressed)
4136907 lib/armeabi/libgdx.so (OK - compressed)
4252724 lib/x86/libgdx.so (OK - compressed)
4347146 META-INF/buildserverid (OK - compressed)
4347255 META-INF/frodoServerId (OK - compressed)
4347366 com/badlogic/gdx.gwt.xml (OK - compressed)
4350929 com/badlogic/gdx/utils/JsonReader.rl (OK - compressed)
4354272 com/badlogic/gdx/utils/arial-15.png (OK)
```

8-Sign the key with apksigner

successful sign the app

```
Failed to load signer signer #1: /home/ubuntu/android-app-hack.keystore entry small_challange_key does not contain a key
→ dist apksigner sign --ks-key-alias small_challange_key -ks ~/android-app-hack.keystore out.apk
WARNING: A restricted method in java.lang.System has been called
WARNING: java.lang.System::loadLibrary has been called by org.conscrypt.NativeLibraryUtil in an unnamed module (file:/home/ubuntu/Android/Sdk/build-tools/35.0.1/lib/apksigner.jar)
WARNING: Use --enable-native-access=ALL-UNNAMED to avoid a warning for callers in this module
WARNING: Restricted methods will be blocked in a future release unless native access is enabled

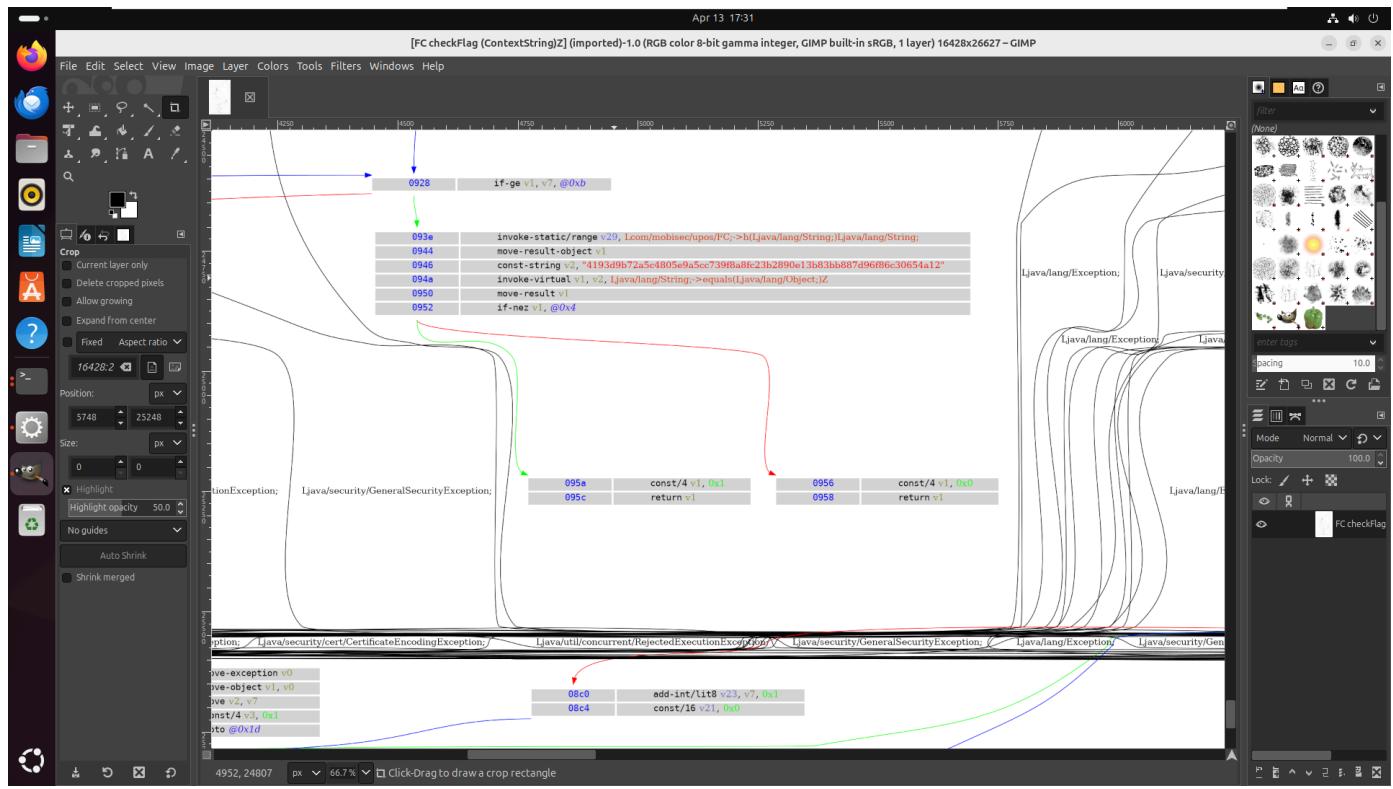
Keystore password for signer #1:
Failed to load signer "signer #1"
java.io.IOException: keystore password was incorrect
    at java.base/sun.security.pkcs12.PKCS12KeyStore.engineLoad(PKCS12KeyStore.java:2109)
    at java.base/sun.security.util.KeyStoreDelegate.engineLoad(KeyStoreDelegate.java:226)
    at java.base/java.security.KeyStore.load(KeyStore.java:1497)
    at com.android.apksigner.SignerParams.loadKeyStoreFromFile(SignerParams.java:423)
    at com.android.apksigner.SignerParams.loadPrivateKeyAndCertsFromKeyStore(SignerParams.java:309)
    at com.android.apksigner.SignerParams.loadPrivateKeyAndCerts(SignerParams.java:237)
    at com.android.apksigner.ApksignerTool.getSignerConfig(ApkSignerTool.java:449)
    at com.android.apksigner.ApksignerTool.sign(ApkSignerTool.java:362)
    at com.android.apksigner.ApksignerTool.main(ApkSignerTool.java:94)
Caused by: java.security.UnrecoverableKeyException: failed to decrypt safe contents entry: javax.crypto.BadPaddingException: Given final block not properly padded. Such issues can arise if a bad key is used during decryption.
... 9 more
→ dist
```

9- upload it with adb shell

```
adb install out.apk
```

63- IF - intro

هندنا نتكلم في topics اللي جاية عن if statement وازاي بنسخدمها و ازاي بنعرفها في Flow Graph فمثلاً في FG هي الخطوط الحمراء والخضراء

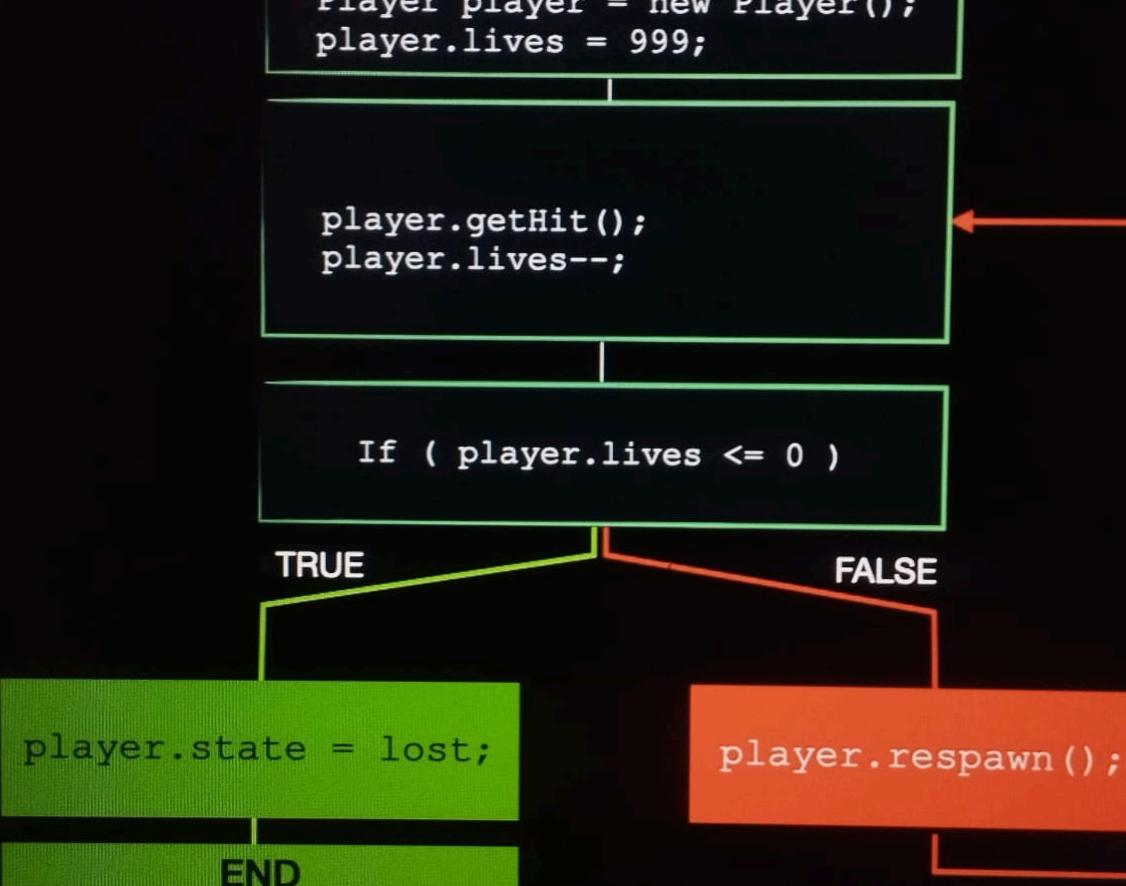


64 - IF-ELSE

هنا اه مثلا نفترض ان ده FG فلو ان statement is true هبيقي اللون اخضر ولو العكس احمر وهنا هنا بيوضح حالة lives وان لو ما بتخسر بنقل واحد وان لو هي اصغر من او تساوي صفر هتبقي خسرت ويخرج ولو اكتر من 0 هيكمي عادي ودلوقتي هشنوف smalli code for it

Introduction

IF / ELSE / GOTO



لوقتي هنروح بقى نعمل التطبيق اللي يعملنا الشرط ده من خلال الاكرواد دي

1-Player.java

</> activity_main.xml

© MainActivity.java

© Player.java ×

```
1 package com.example.smalitwo;
2
3     2 usages
4     public class Player {
5         no usages
6         int power=10;
7         2 usages
8         int lives=5;
9         2 usages
10        boolean state=true;
11        1 usage
12        public void getHit(){
13            this.lives--;
14            if(this.lives>0) {
15                this.state=true;
16            } else {
17                this.state=false;
18            }
19        }
20    }
```

2-MainActivity.java

```
</> activity_main.xml      © MainActivity.java ×   © Player.java
1 package com.example.smalitwo;
2
3 > import ...
11
12 </> public class MainActivity extends AppCompatActivity {
13     2 usages
14     Player player=new Player();
15     2 usages
16     TextView textView;
17     1 usage
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         EdgeToEdge.enable($this$enableEdgeToEdge: this);
22         setContentView(R.layout.activity_main);
23         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
24             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
25             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
26             return insets;
27         });
28         textView=(TextView) findViewById(R.id.textView);
29     }
30     1 usage
31     public void simulateCollision(View view){
32
33         player.getHit();
34         textView.setText("Lives : " + player.lives);
35     }
36 }
```

3- activity_main.xml ده من خالله مثلا لو عاوزين نتشيء button or text

The screenshot shows the Android Studio interface with two tabs open: `activity_main.xml` and `MainActivity.java`. The `activity_main.xml` tab displays the XML configuration for the activity's layout. The `MainActivity.java` tab is currently inactive.

activity_main.xml Content:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#000000"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:textColor="@color/white"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

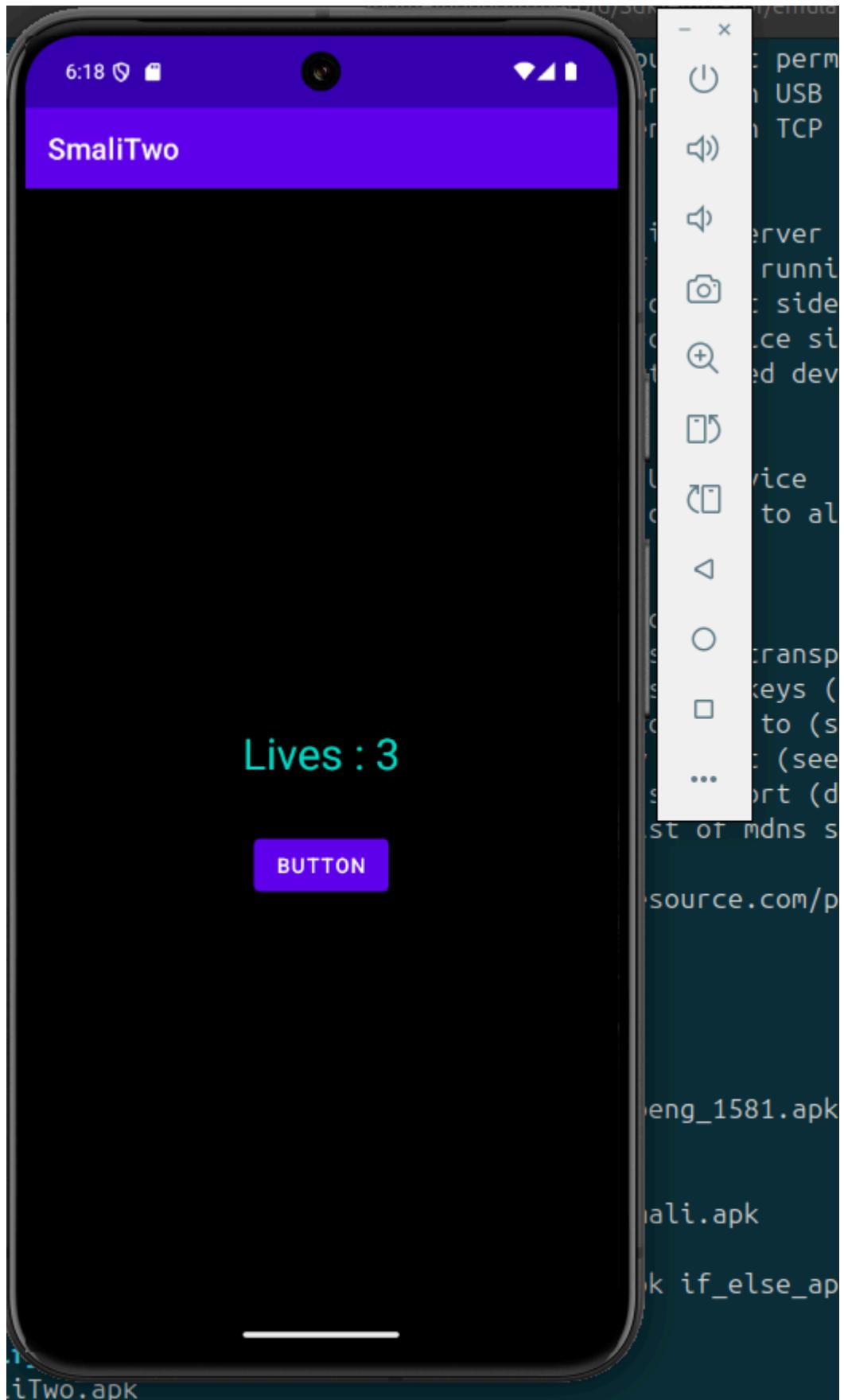
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"
        android:onClick="simulateCollision"
        android:layout_marginTop="34dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/textView" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Layout Editor View:

The Layout Editor shows a ConstraintLayout with a black background. It contains a `TextView` with the text "Hello World!" and a `Button` labeled "Button". The `Button` has a purple background. The `TextView` is positioned above the `Button`. Both views are constrained to the parent layout using `app:layout_constraintTop_toTopOf` and `app:layout_constraintBottom_toBottomOf` attributes. The `Button` is also constrained to the right edge of the parent and has a vertical margin of 34dp from the top of the `TextView`.

app after install it on emulator and each time click on the button the value of lives decrease by 1



show Smali code for this app after decompile it or using jadx-gui + app.apk

```

c Player ×
1 ##### Class com.apphacking.smalitwo.Player (com.apphacking.smalitwo.Player)
2 .class public Lcom/apphacking/smali/Player;
3 .super Ljava/lang/Object;
4 .source "Player.java"
5
6
7 # instance fields
8 .field lives:I
9
10 .field power:I
11
12 .field state:Z
13
14
15 # direct methods
16 .method public constructor <init>()V
17     .registers 2
18
19     .line 3
20     invoke-direct {p0}, Ljava/lang/Object; -><init>()V
21
22     .line 5
23     const/16 v0, 0xa
24
25     input v0, p0, Lcom/apphacking/smali/Player; ->power:I
26
27     .line 6
28     const/4 v0, 0x5
29
30     input v0, p0, Lcom/apphacking/smali/Player; ->lives:I
31
32     .line 11
33     const/4 v0, 0x1
34
35     input-boolean v0, p0, Lcom/apphacking/smali/Player; ->state:Z
36
37     return-void
38 .end method
39
40
41 # virtual methods
42 .method public getHit()V
43     .registers 3
44
45     .line 17
46     ige v0, p0, Lcom/apphacking/smali/Player; ->lives:I
47
48     const/4 v1, 0x1
49
50     sub-int/2addr v0, v1
51
52     input v0, p0, Lcom/apphacking/smali/Player; ->lives:I
53
54     .line 19
55     if-lez v0, :cond_b
56
57     .line 21
58     input-boolean v1, p0, Lcom/apphacking/smali/Player; ->state:Z
59
60     goto :goto_e
61
62     .line 24
63     :cond_b
64     const/4 v0, 0x0
65
66     input-boolean v0, p0, Lcom/apphacking/smali/Player; ->state:Z
67
68     .line 26
69     :goto_e
70     return-void
71 .end method
72

```

65- IF-Else code analysis

لوقتي بقى عاوزين نحل الكود ده (smali)

1- classe name

```

.class public Lcom/apphacking/smalitwo/Player;
.super Ljava/lang/Object;
.source "Player.java"

public class Player extends Object(){}
```

2- instance field

```

# instance fields
.field lives:I

.field power:I

.field state:Z

int lives
int power
boolean state
```

3- direct method --> constructor for initializing the variable

```

# direct methods
.method public constructor <init>()V
    .registers 2

    .line 3
    invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    .line 5
    const/16 v0, 0xa           ----> set 10 for v0 --> v0 =10

    input v0, p0, Lcom/apphacking/smalitwo/Player;->power:I   --->
this.live=v0      --> this.live=10

    .line 6
    const/4 v0, 0x5            ----> set 5 for v0 --> v0=5

    input v0, p0, Lcom/apphacking/smalitwo/Player;->lives:I     --->
this.power=v0      --> this.power=5

    .line 11
    const/4 v0, 0x1      ---> set 1 for v0 --> v0=1

    input-boolean v0, p0, Lcom/apphacking/smalitwo/Player;->state:Z  --->
```

```
state=v0 --> state=1 ---> state=true
```

```
    return-void  
.end method
```

4- virtual method --> public method on code we create it

```
.method public getHit()V ---> public void getHit()  
.registers 3  
  
.line 17  
    igure v0, p0, Lcom/apphacking/smalitwo/Player;->lives:I      --> get the  
this.live and set it on v0  
  
    const/4 v1, 0x1      ---> set 1 to v1 --> v1=1  
  
    sub-int/2addr v0, v1      --> v0=v0-v1 --->           v0--  
  
    igure v0, p0, Lcom/apphacking/smalitwo/Player;->lives:I      --> put the  
value from v0 to this.lives --> this.lives--  
  
.line 19  
    if-lez v0, :cond_b      ---> if <= 0 it true jump to cond_d else  
complete      --> v0 <= 0  
  
.line 21  
    igure-boolean v1, p0, Lcom/apphacking/smalitwo/Player;->state:Z      -->  
put the value of v1=1 to state --> this.state=1 = true  
  
    goto :goto_e      ---> jump to goto_e  
  
.line 24  
    :cond_b      ---> else  
    const/4 v0, 0x0      --> v0=0  
  
    igure-boolean v0, p0, Lcom/apphacking/smalitwo/Player;->state:Z      ---> put  
value from v0 to this.state --> this.state=false  
  
.line 26  
    :goto_e  
    return-void  
.end method
```

28	goto target	Unconditional jump by short offset ² .	28F0 - goto 0005 // -0010 Jumps to current position-16 words (hex 10). 0005 is the label of the target instruction.
----	-------------	---	--

3A	if-ltz vx,target	Checks vx and jumps if vx<0 ² .	3A00 1600 - if-ltz v0, 002d // +0016 Jumps to the current position+16H words if v0<0. 002d is the label of the target instruction.
----	------------------	--	---

هنا بقول ان لو $v0 < 0$ if $v0 < 0$ كده بشوف الشرط ولو else بنحط دي target: ودي مهناها ان هو jump to target

```

if-lez v0, :cond_b      ---> if v0<= 0 true jump to cond_d and and
if false complete

.line 21
input-boolean v1, p0, Lcom/apphacking/smalitwo/Player;->state:Z    -->
put the value of v1=1 to state --> this.state=1 = true

goto :goto_e    ---> jump to goto_e

.line 24
:cond_b      ---> else
const/4 v0, 0x0    --> v0=0

input-boolean v0, p0, Lcom/apphacking/smalitwo/Player;->state:Z    --> put
value from v0 to this.state --> this.state=false

.line 26
:goto_e
if v0 < 0:cond_d  ---> else
:cond_b      ---> else
const/4 v0, 0x0    --> v0=0  ---> else

```

اني واحده وهي goto target هي ان لو الشرط ده صح ننتقل لل target وخلی بالك في الاتنين لو عاوز نتنقل للحاجة اللي هي مثلا target: تحددها في الكود تبقي target:

```

goto :goto_e    --> if ture jump goto_e

:goto_e      --> here goto_e before return-void because when jump it end
the class and return void
return-void

```

خلی بالك هان هو في التحقق من الشرط هو بيعمل عكس java بس بنفس المعنى يعني في java كنا بنتتحقق ان $this.lives > 0$ لا اهو بيعمل العكس بس بنفس المعنى if-lez v0 فلو فال من او يساوي 0 هيعمل jump to target ولو الشرط false هيكلم

Syntax	Description	
if-eqz vx,target	Equal Zero	if vx == 0
if-nez vx,target	Not Equal Zero	if vx != 0
if-ltz vx,target	Less Than Zero	if vx < 0
if-gez vx,target	Greater Equal Zero	if vx >= 0
if-gtz vx,target	Greater Than Zero	if vx > 0
if-lez vx,target	Less Equal Zero	if vx <= 0

Comparison against 0.

target = Location to JMP after this check.

Syntax	Description	
if-eq vx,vy,target	Equal	if vx == vy
if-ne vx,vy,target	Not Equal	if vx != vy
if-lt vx,vy,target	Less Than	if vx < vy
if-ge vx, vy,target	Greater Equal	if vx >= vy
if-gt vx,vy,target	Greater Than	if vx > vy
if-le vx,vy,target	Less Equal	if vx <= vy

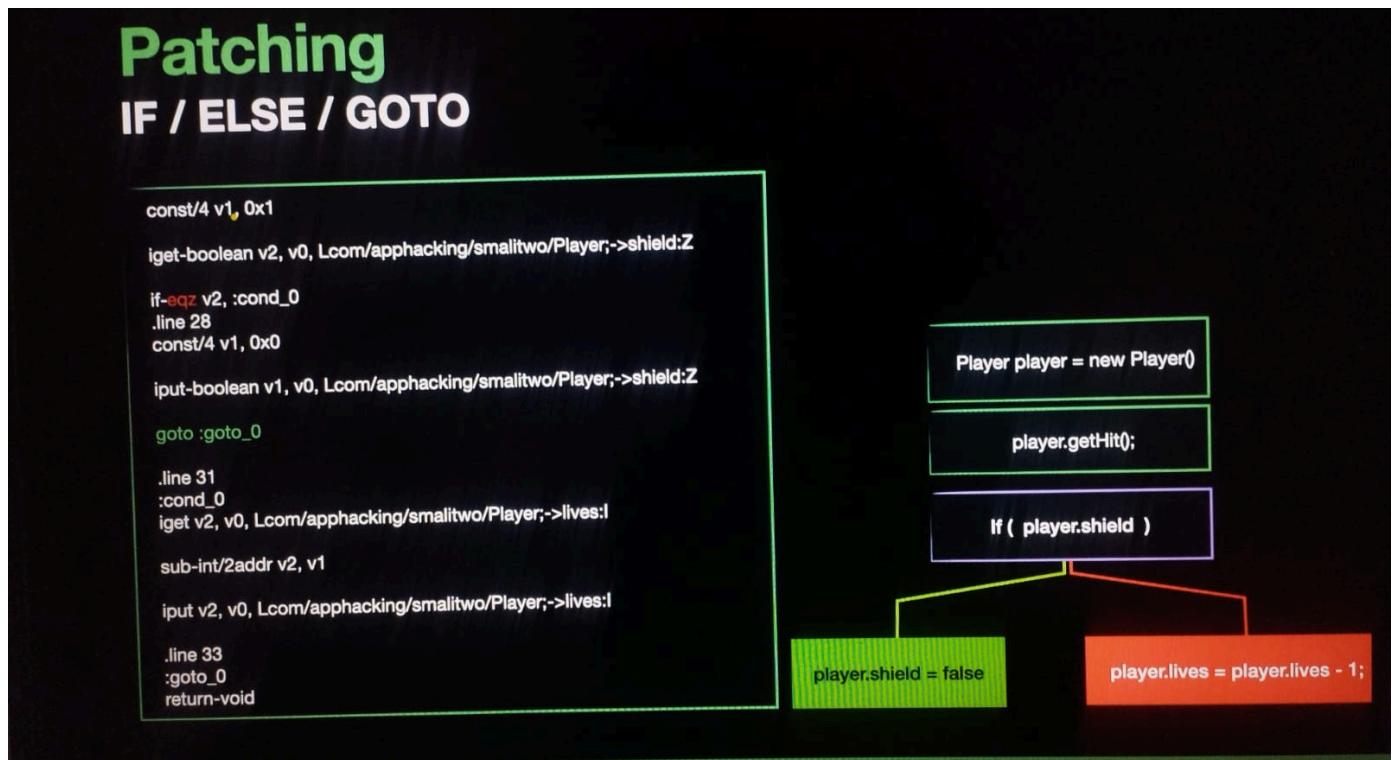
Comparison against a local register.

target = Location to JMP after this check.

68- Flipping the logic

1- method1 logic flipping

هنا مثلا الكود ده ان لو هو لايس درع فالدرع ده هيشهيله ولو بس لايس فهو هيموت



```
const/4 v1, 0x1 ---> v1 = 1
    igeget-boolean v2, p0, Lcom/apphacking/smalltwo/Player;->shiled:Z      --> v2
    =this.shiled
    if-eqz v2:cond_0  --> if v2 == 0   jump to cond_0 else complete
    .line 28
    const/4 v1, 0x0  --> v1=0
    igeput-boolean v1, p0, Lcom/apphacking/smalltwo/Player;->shiled:Z      --> put the
    value of v1 on this.shiled --> this.shiled=0 = false
    goto: goto_0  --> jump goto_0
    .line 31
    :cond_0  --> the condition      if v2==0
    igeget v2, p0, Lcom/apphacking/smalltwo/Player;->lives:I  --> get the value from
    this.lives and set it into v2 --> v2=this.lives
    sub-int/2addr v2, v1 --> v2=v2-v1
    igeput v2, p0, Lcom/apphacking/smalltwo/Player;->lives:I  --> put the value of
    v2 into this.lives
    .line 33
    :goto_0
    return-void
```

علشان نحاول نستغل دي نحوال الشرط زي كده مثلا : هنا اه مثلا بعكس الشرط لو انا هبda مش لابس درع كده هبيقي (false)! كده هبيقي true ويروح ينفذ الشرط انه يشيب درع وهكذا وتبقى infinite loop بس مش هينقص live

Patching IF / ELSE / GOTO

```

const/4 v1, 0x1
iget-boolean v2, v0, Lcom/apphacking/smallitwo/Player;.>shield:Z
nez
if <= v2, :cond_0
.line 28
const/4 v1, 0x0

input-boolean v1, v0, Lcom/apphacking/smallitwo/Player;.>shield:Z
goto :goto_0

.line 31
:cond_0
iget v2, v0, Lcom/apphacking/smallitwo/Player;.>lives:I
sub-int/2addr v2, v1
input v2, v0, Lcom/apphacking/smallitwo/Player;.>lives:I
.line 33
:goto_0
return-void

```

```

IF V2 EQZ == 0 :cond_0
Player player = new Player()
player.getHit();
if (player.shield)
    player.shield = false
else
    player.lives = player.lives - 1;

```

نروح بقى نغير القيمة عادي ل nez في smali code ونعمل نفس الخطوات اللي نعمل signing built من ونعمل نفس الخطوات اللي نعمل signing وبكده بيبقى احنا غيرنا app وبكده دي اول طريقة من الطرق اللي من خلله ازاي نستغل logic smali code

2-method 2 Deleting Code

هنا في الكود ده هو بيشفوف هل الاعب عند MasterCap power ب 100 ولا لا ولو عنده يزود power ب 10 لو معندوش يزودها ب 10

```
5  public void processGame() {  
6      // Create new Player Object  
7      Player player = new Player();  
8  
9      if (player.hasItem(MasterCap)) {  
10         player.power += 100;  
11     }  
12     else {  
13         player.power += 10;  
14     }  
15 }
```

اللي بقى نعمله هو ان احنا نشيل الشرط بناء IF ولكن لو شيلنها مش هيبقى في else} في java علشان هيبقى في else} ولكن في smali هيبقى عادي زي ما هنشوف دلوقتي وبعد ما نعمله هنحصل على power من غير اي شرط

```
5 public void processGame() {  
6     // Create new Player Object  
7     Player player = new Player();  
8  
9     if (player.hasItem(masterCap)) {  
10         player.power += 100;  
11     }  
12     else {  
13         player.power += 10;  
14     }  
15 }
```

code java for player

هنا اه هو قيمة masterCap =false فكده هو هيزد ال power ب 10

```

package com.apphacking.smalitwo;

/* loaded from: classes2.dex */
public class Player {
    int highscore;
    int power = 10;
    int lives = 5;
    int score = 100;
    boolean shield = true;
    boolean state = true;
    boolean masterCap = false;
}

public void hasItem(Player player) {
    if (player.masterCap) {
        player.power += 100;
    } else {
        player.power += 10;
    }
}

public void checkHighScore(Player player)
    int i = player.score;
    if (i < player.highscore) {
        return;
    }
    player.highscore = i;
    if (player.lives >= 5) {
        player.score = i + 1000;
    }
}

public void getHit() {
    if (this.shield) {
        this.shield = false;
        return;
    }
    int i = this.lives - 1;
    this.lives = i;
    if (i > 0) {
        this.state = true;
    } else {
        this.state = false;
    }
}

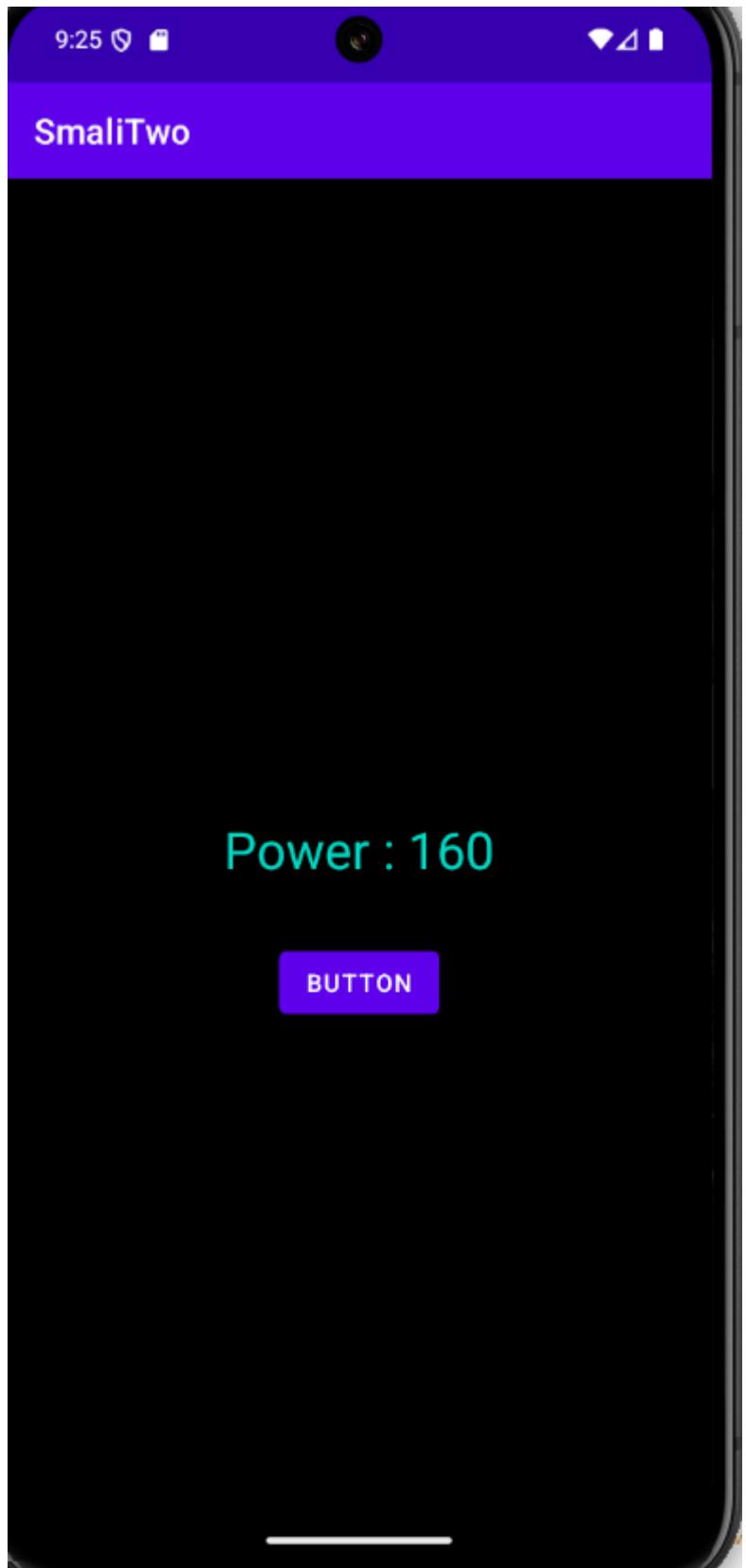
```

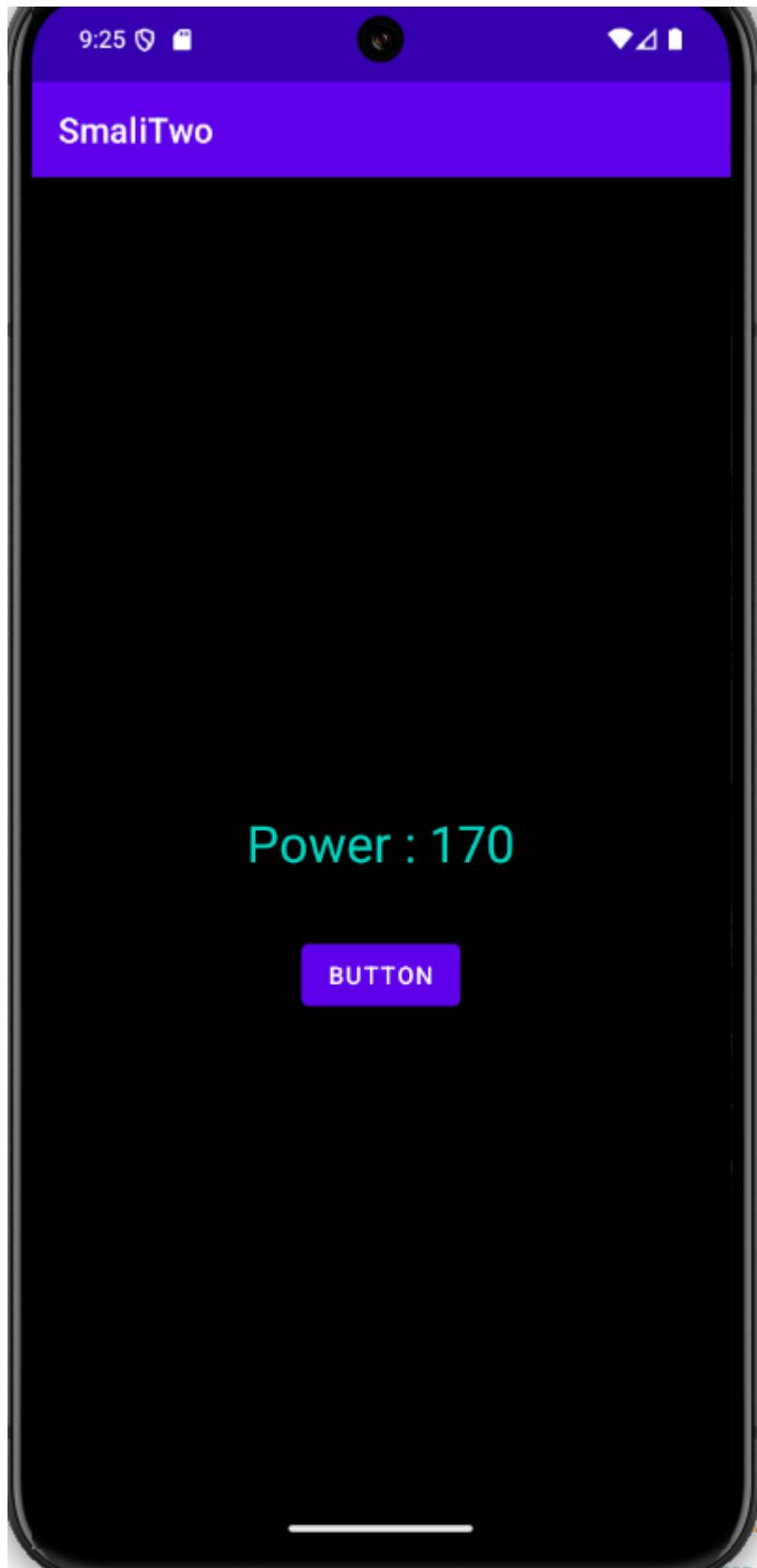
.method public hasItem(Lcom/apphacking/smali/Player;)V
 .registers 3
 .param p1, "player" # Lcom/apphacking/smali/Player;
 .line 28
 igeq v0, p1, Lcom/apphacking/smali/Player; ->masterCap:Z
 if-eqz v0, :cond_b
 .line 22
 igeq v0, p1, Lcom/apphacking/smali/Player; ->power:I
 add-int/lit8 v0, v0, 0x64
 igure v0, p1, Lcom/apphacking/smali/Player; ->power:I
 goto :goto_11
 .line 25
 :cond_b
 igeq v0, p1, Lcom/apphacking/smali/Player; ->power:I
 add-int/lit8 v0, v0, 0xa
 igure v0, p1, Lcom/apphacking/smali/Player; ->power:I
 .line 28
 :goto_11
 return-void
.end method

هنا بقى في smali لو حزفنا if-eqz v0,:cond_b هيكمل حاجة من غير ما يتحقق من حاجة ويزودها ب 100

if-eqz v0, :cond_b --> if we remove this line

هنا اه بيزود ب 10





دلوقتني هنحذف الامر بنطاع if statement on smalli

```

.method public hasItem(Lcom/apphacking/smalitwo/Player;)V
    .locals 1
    .param p1, "player"    # Lcom/apphacking/smalitwo/Player;

    .line 20
    igure-boolean v0, p1, Lcom/apphacking/smalitwo/Player;->masterCap:Z

    .line 22
    igure v0, p1, Lcom/apphacking/smalitwo/Player;->power:I
    add-int/lit8 v0, v0, 0x64

    igure v0, p1, Lcom/apphacking/smalitwo/Player;->power:I
    goto :goto_0

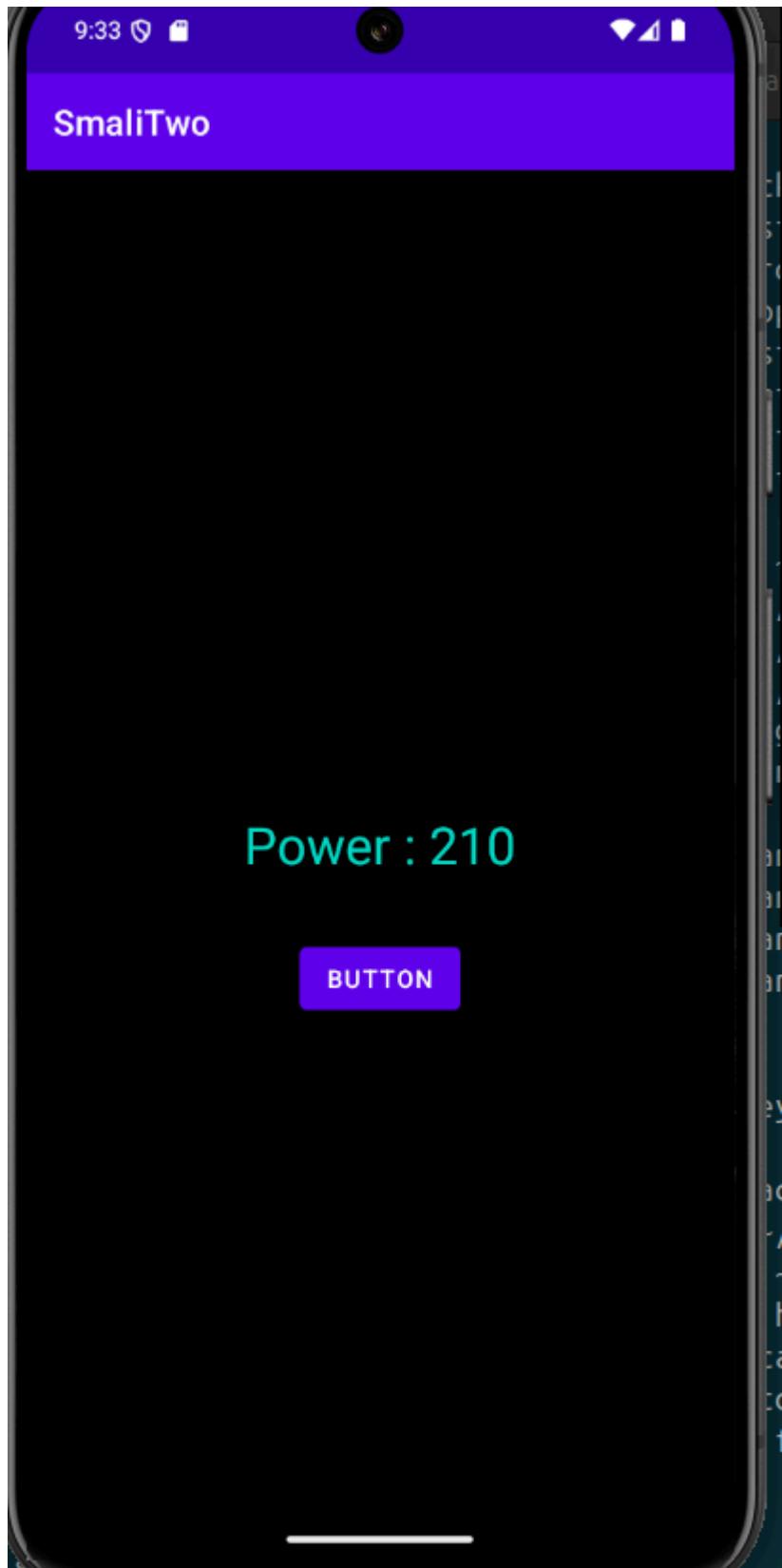
    .line 25
    :cond_0
    igure v0, p1, Lcom/apphacking/smalitwo/Player;->power:I
    add-int/lit8 v0, v0, 0xa

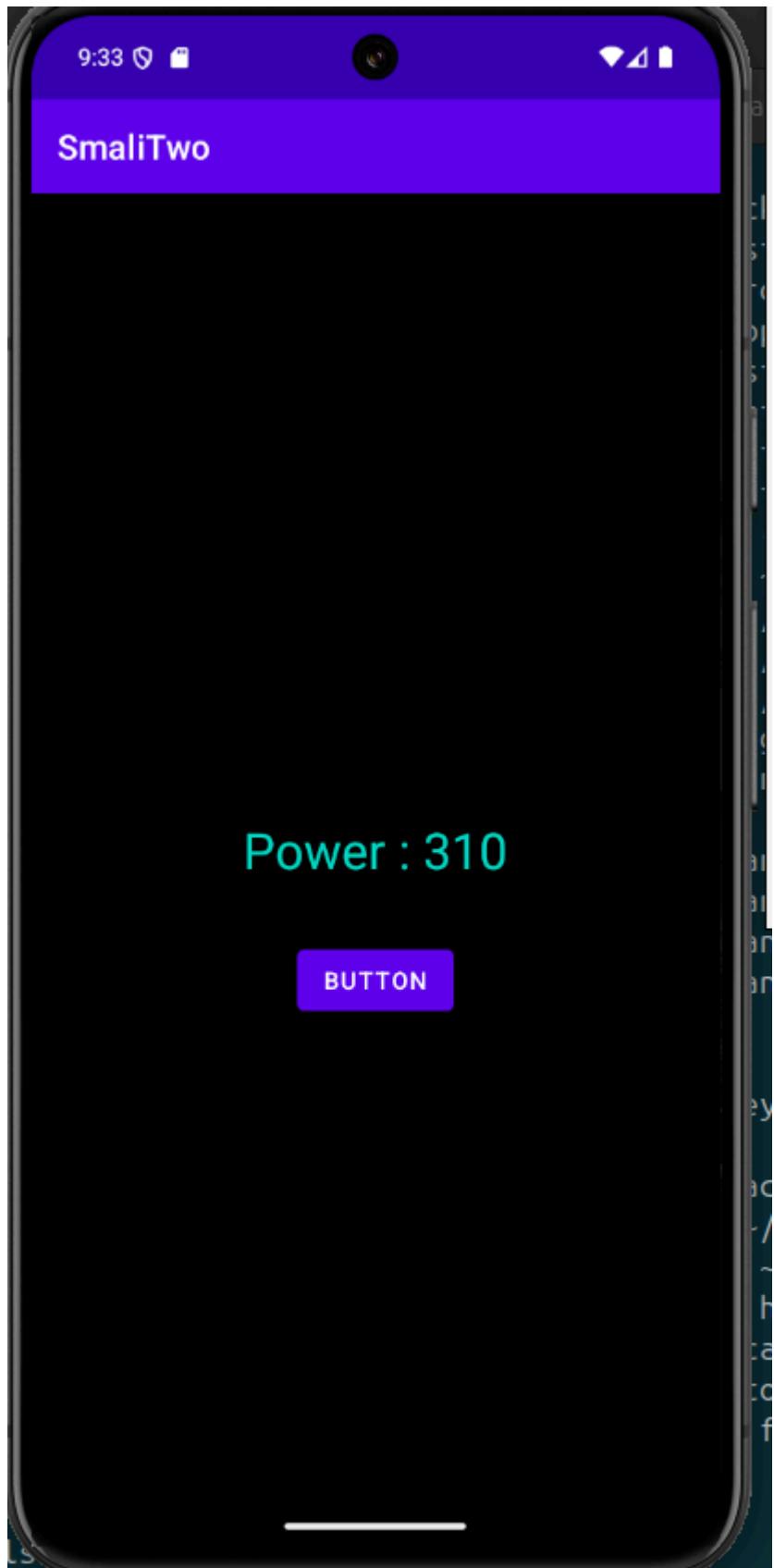
    igure v0, p1, Lcom/apphacking/smalitwo/Player;->power:I

    .line 28
    :goto_0
    return-void
.end method

```

نعمل بقى و signing ونحمله ونشوف دلوقتى انا هعمله كده و هحمله دلوقتى انا عملته sign وبعد ما حذفنا السطر فعلا بقى يزود القيمة بتاعت power ب 100





او برضه ممكن نحذا اي شرط ونخليه يزود وخلاص المهم هنا هو بنعدل في الكود علشان نغير function اللي هو بيعملها

```
Player X
package com.apphacking.smalitwo;

13 public class Player {
    int highscore;
    int lives = 5;
    boolean masterCap = false;
    int power = 10;
    int score = 100;
    boolean state = true;

14     public void hasItem(Player player) {
15         boolean z = player.masterCap;
16         player.power += 100;
17     }

31     public void checkHighScore(Player player) {
32         int i = player.score;
33         if (i >= player.highscore) {
34             player.highscore = i;
35             if (player.lives >= 5) {
36                 player.score = i + 1000;
37             }
38         }
39     }

49     public void getHit() {
50         int i = this.lives - 1;
51         this.lives = i;
52         if (i > 0) {
53             this.state = true;
54         } else {
55             this.state = false;
56         }
57     }
58 }
```

70 -Method 3 : Jumps

هنا ان احنا نعمل jump على command `goto` تاني باستخدام `if` فهنا لما نتسخدمها مثلا الشرط هيتتحقق ونزوود `power` ب 10 وفي نفس `scope` بتاع الشرط نعمل `jump` للشرط الثاني اللي هو بيزود `power` ب 100 وبكده بيقي في كل مرة هنزوودها بقيمتين هما 10 و 100

```
goto: target
```

```
.line 61
:goto_0
return-void
.end method

.method public hasItem(Lcom/apphacking/smalitwo/Player;)V
.locals 1
.param p1, "player"    # Lcom/apphacking/smalitwo/Player; ✓

.line 15
iget-boolean v0, p1, Lcom/apphacking/smalitwo/Player;->masterCap:Z
if-eqz v0, :cond_0 •

.line 17
iget v0, p1, Lcom/apphacking/smalitwo/Player;->power:I
add-int/lit8 v0, v0, 0xa + 10

iput v0, p1, Lcom/apphacking/smalitwo/Player;->power:I
goto :goto_0 :cond_0
.line 20
:cond_0
iget v0, p1, Lcom/apphacking/smalitwo/Player;->power:I
add-int/lit8 v0, v0, 0x64 + 100

iput v0, p1, Lcom/apphacking/smalitwo/Player;->power:I
.line 23
:goto_0
return-void
.end method
```

```
    input-boolean v0, p0, Lcom/apphacking/smalitwo/Player; >state.z

.line 61
:goto_0
return-void
.end method

.method public hasItem(Lcom/apphacking/smalitwo/Player;)V
.locals 1
.param p1, "player" # Lcom/apphacking/smalitwo/Player;

.line 15
iget-boolean v0, p1, Lcom/apphacking/smalitwo/Player;->masterCap:Z

if-eqz v0, :cond_0

.line 17
iget v0, p1, Lcom/apphacking/smalitwo/Player;->power:I

add-int/lit8 v0, v0, 0xa

input v0, p1, Lcom/apphacking/smalitwo/Player;->power:I

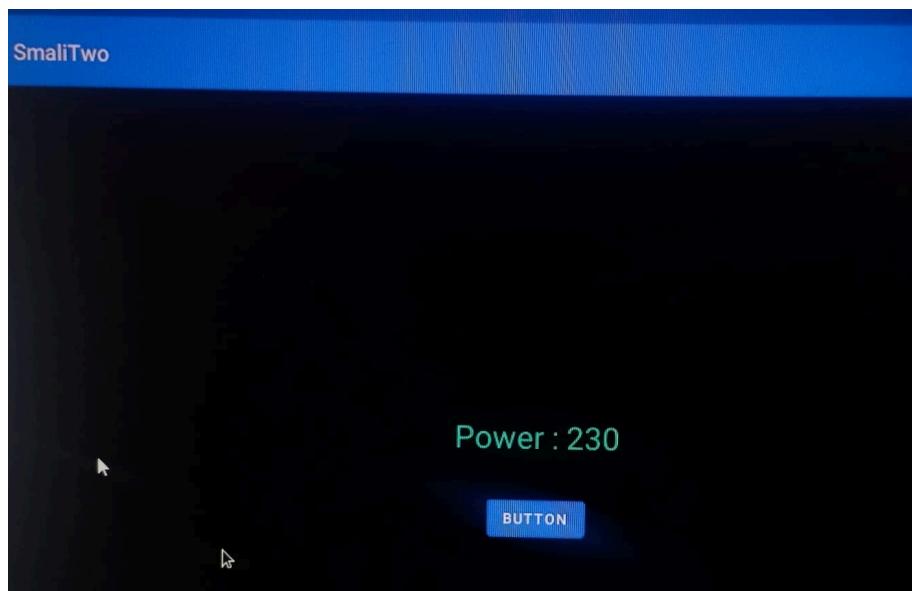
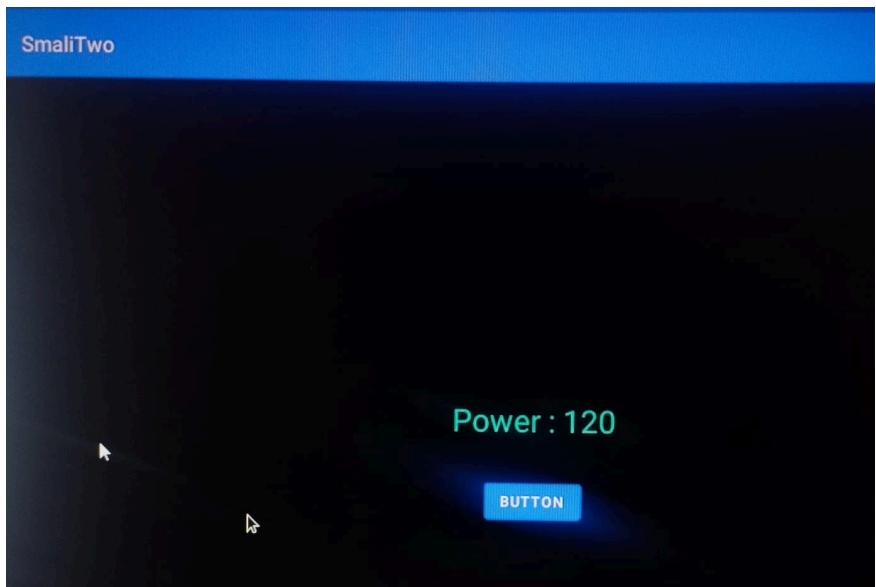
goto :cond_0

.line 20
:cond_0
iget v0, p1, Lcom/apphacking/smalitwo/Player;->power:I

add-int/lit8 v0, v0, 0x64

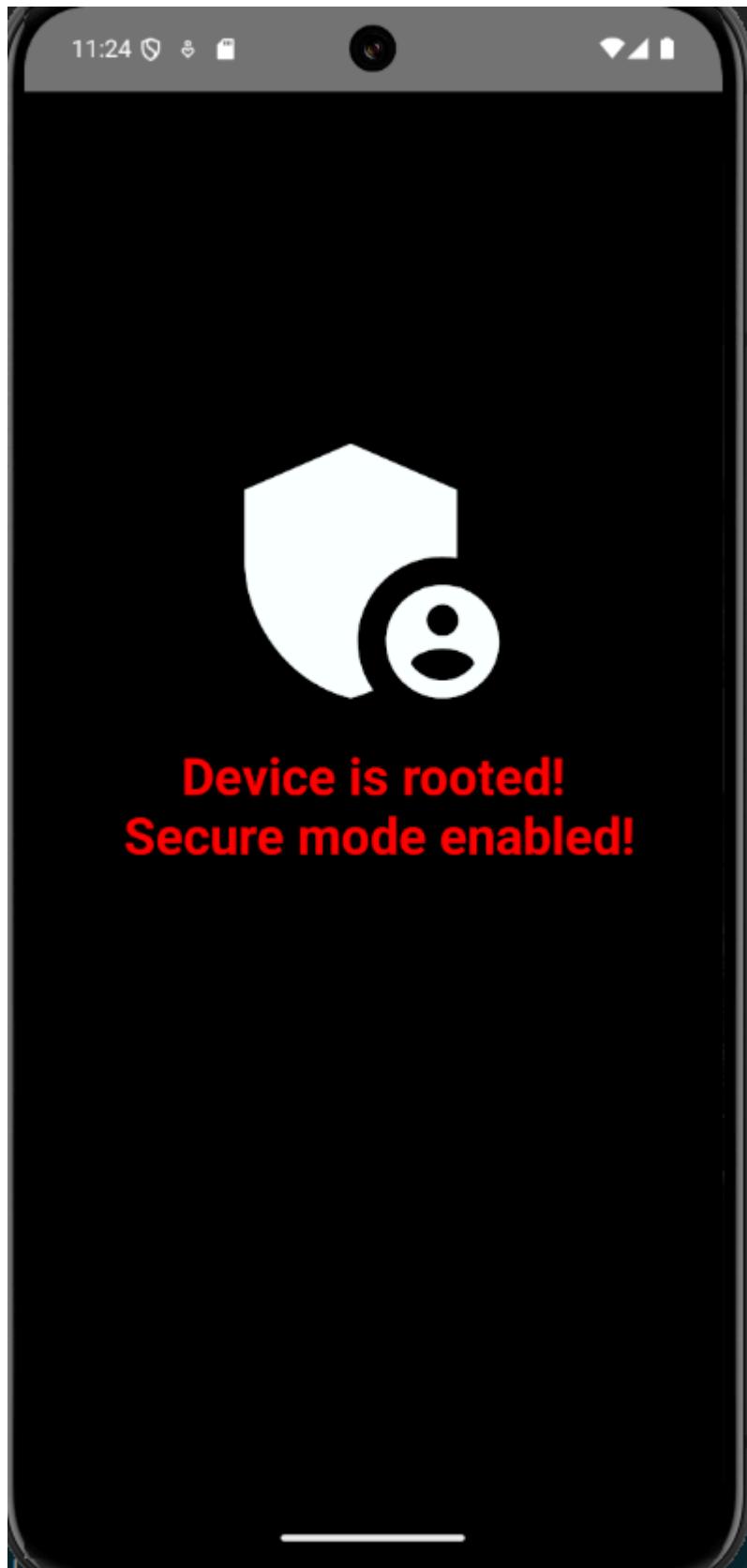
input v0, p1, Lcom/apphacking/smalitwo/Player;->power:I

.line 23
:goto_0
return-void
.end method
```



Assignment

هنا في challenge ده هو عاوزيني اعمل bypass root detectoin



1- decompile the app and analyze it

هنا بقى هنشوف فی Main activity function Check ان هو root من خلال هو بيعمل RootUtil فی

```
if (RootUtil.isDeviceRooted()) {
    this.btn.setVisibility(4);
    this.txtview.setTextColor(Color.parseColor("#FF0000"));
    this.txtview.setText("Device is rooted!\n Secure mode enabled!");
} else {
    this.btn.setVisibility(0);
    this.txtview.setTextColor(Color.parseColor("#00FF00"));
    this.txtview.setText("Device is NOT rooted!\n No login possible!");
}
}

public void login(View view) {
    Intent intent = new Intent(this, (Class<?>) Login.class);
    startActivity(intent);
}
```

2- open file RootUtil

هنا اه هو check root من خلال three method كل واحدة بترجع true

```
package com.apphacking.rootcheck;

import android.os.Build;
import java.io.BufferedReader;
import java.io.File;
import java.io.InputStreamReader;

/* loaded from: classes.dex */
public class RootUtil {
    public static boolean isDeviceRooted() {
        return checkRootMethod1() || checkRootMethod2() || checkRootMethod3();
    }

    private static boolean checkRootMethod1() {
        String buildTags = Build.TAGS;
        return buildTags != null && buildTags.contains("test-keys");
    }

    private static boolean checkRootMethod2() {
        String[] paths = {"/system/app/Superuser.apk", "/sbin/su", "/system/bin/su", "/system/xbin/su", "/data/local/xbin/su", "/data/local/bin/su", "/system/sd/xbin/su"};
        for (String path : paths) {
            if (new File(path).exists()) {
                return true;
            }
        }
        return false;
    }

    private static boolean checkRootMethod3() {
        Process process = null;
        try {
            process = Runtime.getRuntime().exec(new String[]{"./system/xbin/which", "su"});
            BufferedReader in = new BufferedReader(new InputStreamReader(process.getInputStream()));
            if (in.readLine() != null) {
                if (process != null) {
                    process.destroy();
                }
                return true;
            }
            if (process != null) {
                process.destroy();
            }
            return false;
        } catch (Throwable th) {
            if (process != null) {
                process.destroy();
            }
            return false;
        }
    }
}
```

علشان بقى متخلهوش ان check root ممكن نعمل two scenario هو نعكس condition بتاع root في RootUtil والسناريو الثاني ان نحاول تغير في RootUtil بحيث نخله three method return false

1- بس اللي هعمله هو ان احنا في condition بتاع RootUtil اللي هو check isRooted نعس بقى

```
public class RootUtil {
    public static boolean isDeviceRooted() {
        return checkRootMethod1() || checkRootMethod2() || checkRootMethod3();
    }
}
```

في كود smali نحاول ان هو return false علشان لما بروح لل Main Activitiy يبقى قيمة ان هو root =false وبالنالي يبقى عرفنا نعمل bypass root detection

smali code for this function

دلوقي هنا اه هو بيروح يجيب الحاجة اللي بنعملها return من كل method وطبعا علشان موجودة هتبقي true فلما تبقي true يرجع اللي هو بيرجع to cond_1

```
.method public static isDeviceRooted()Z
.locals 1

.line 10
.invoke-static {}, Lcom/apphacking/rootcheck/RootUtil;-
>checkRootMethod1()Z

move-result v0

if-nez v0, :cond_1

.invoke-static {}, Lcom/apphacking/rootcheck/RootUtil;-
>checkRootMethod2()Z

move-result v0

if-nez v0, :cond_1

.invoke-static {}, Lcom/apphacking/rootcheck/RootUtil;-
>checkRootMethod3()Z

move-result v0

if-eqz v0, :cond_0

goto :goto_0

:cond_0
const/4 v0, 0x0

goto :goto_1

:cond_1
:goto_0
```

```

const/4 v0, 0x1

:goto_1
return v0
.end method

```

to solving this

في حلین بقی اول حل هو ان نخلیه یعمل jump to cond_0 من غیر ما یخشن عليها اصلا زی کده هیروح ل و cond_0

```

.method public static isDeviceRooted()Z
.locals 1
goto :cond_0
.line 10
.invoke-static {}, Lcom/apphacking/rootcheck/RootUtil;-
>checkRootMethod1()Z

move-result v0

if-nez v0, :cond_1

.invoke-static {}, Lcom/apphacking/rootcheck/RootUtil;-
>checkRootMethod2()Z

move-result v0

if-nez v0, :cond_1

.invoke-static {}, Lcom/apphacking/rootcheck/RootUtil;-
>checkRootMethod3()Z

move-result v0

if-eqz v0, :cond_0

goto :goto_0

:cond_0
const/4 v0, 0x0

goto :goto_1

:cond_1
:goto_0

```

```
const/4 v0, 0x1
```

```
:goto_1
```

```
return v0
```

الحل الثاني هو ان نعدل condition logic بـ`if_eqz v0 :cond_1` ففي كل مرة هيلقي ان الشرط غلط فمش هي عمل jump to `cond_1` ونعكس بـ`if_nez v0, :cond_0` نخليها if-eqz v0, :cond_0 علشان kده هييفي في `v0 contain value zero` فلما يلاقى فعلا ان هي مش بتتساوي jump to `cond_0` والكود هييفي كده

```
.method public static isDeviceRooted() Z
.locals 1

.line 10
.invoke-static {}, Lcom/apphacking/rootcheck/RootUtil;-
>checkRootMethod1() Z

move-result v0

if-eqz v0, :cond_1

.invoke-static {}, Lcom/apphacking/rootcheck/RootUtil;-
>checkRootMethod2() Z

move-result v0

if-eqz v0, :cond_1

.invoke-static {}, Lcom/apphacking/rootcheck/RootUtil;-
>checkRootMethod3() Z

move-result v0

if-nqz v0, :cond_0

goto :goto_0

:cond_0
const/4 v0, 0x0

goto :goto_1

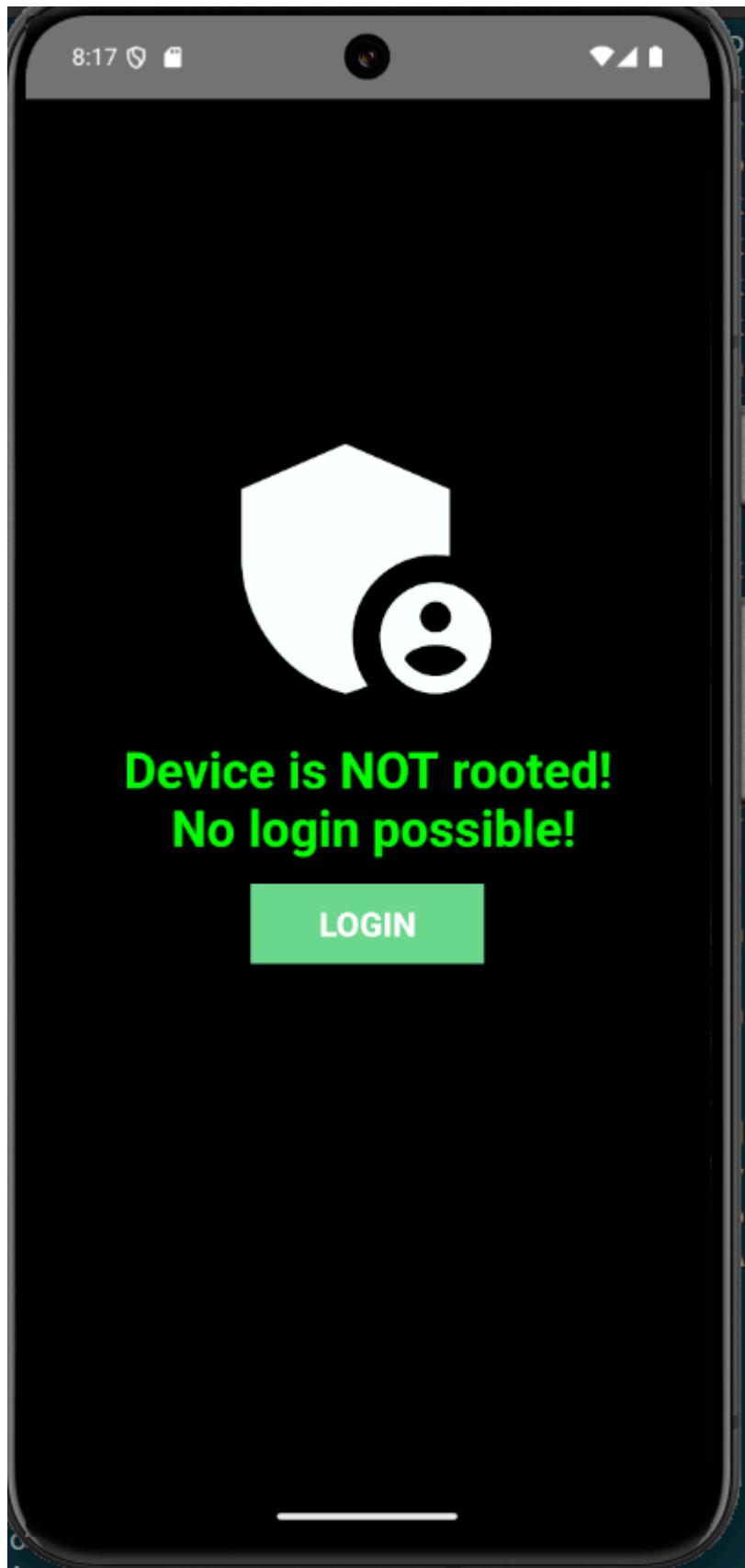
:cond_1
:goto_0
```

```
const/4 v0, 0x1
```

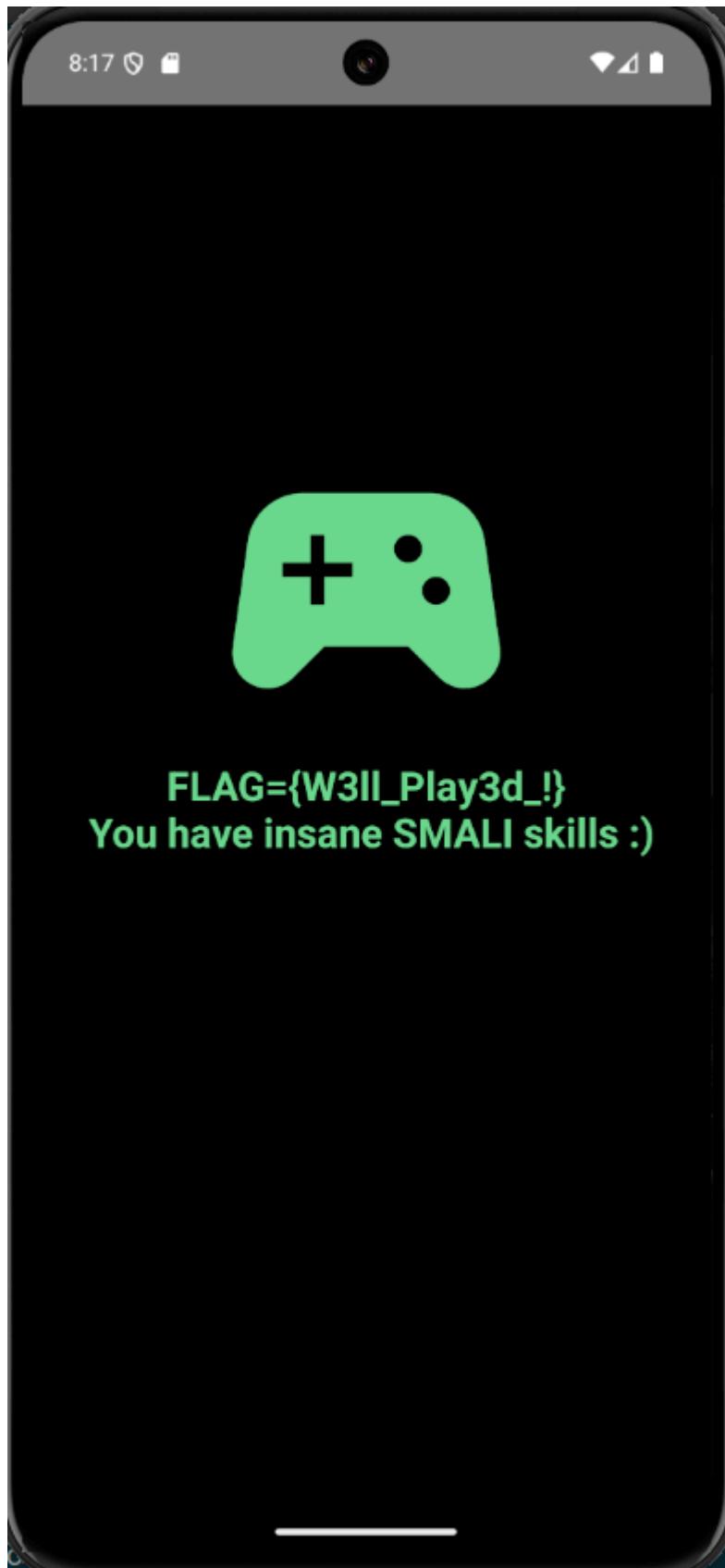
```
:goto_1  
return v0
```

هنجير بقى و نعمل signing ونشوف ايه اللي هيحصل : خلي بالم لما نعمله signing احنا بنحذف التطبيق القديم قبل ما نعمل install للتطبيق الجديد

successfully we change it



ing...
files should be loaded. Notifying the de



في اكثر من حل ولكن ده اسهل حل

خلي بالك كان في حل ممكن نعمله بقى من غير ما نغير اي حاجة وهو ان start activity for login activity وده علشان لو منا لاقينا ان login activity have one of this option 1- exported="true" or have <intent-filter>f

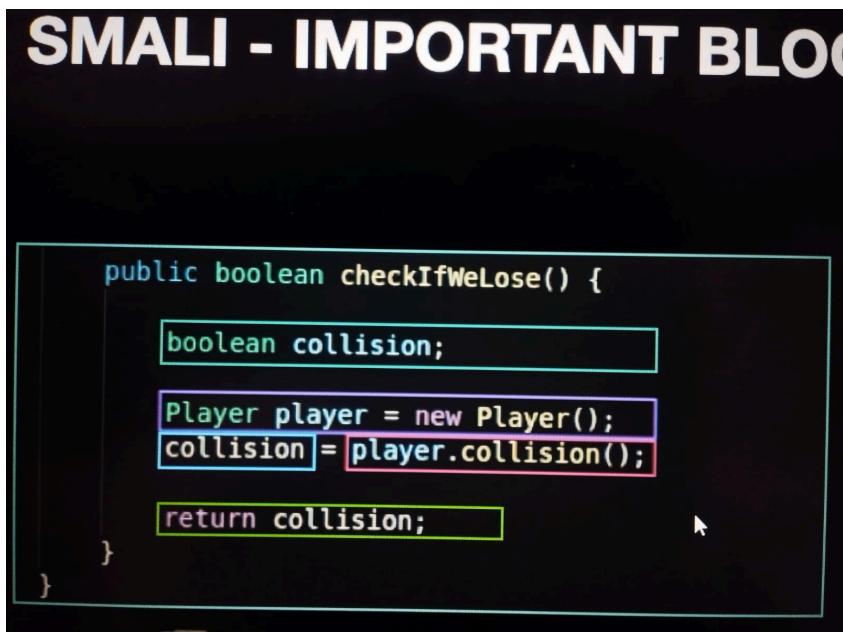
```
20     android:componentFactory=" androidx.core.app.CoreComponentFactory">
21 <activity android:name="com.apphacking.rootcheck.Login"/>
22 <activity android:name="com.apphacking.rootcheck.MainActivity">
23     <intent-filter>
24         <action android:name="android.intent.action.MAIN"/>
25         <category android:name="android.intent.category.LAUNCHER"/>
26     </intent-filter>
27 </activity>
28 </application>
29 </manifest>|
```

73-Object and Method

هو ان انتا بتعمل create object من class معين زي مثلا كده

هنا اه هو بيعمل create object player from class Player and assign value collision on object to collision

java code



smali code

```

4 # virtual methods
5 .method public checkIfWeLose()Z
6     .locals 2
7
8     .line 23
9     new-instance v0, Lcom/apphacking/smalione/Player;
10    invoke-direct {v0}, Lcom/apphacking/smalione/Player;:-><init>()V
11
12    .line 24
13    .local v0, "player":Lcom/apphacking/smalione/Player;
14    invoke-virtual {v0}, Lcom/apphacking/smalione/Player;:->collision()Z
15
16    move-result v1
17
18    .line 26
19    .local v1, "collision":Z
20    return v1
21
22 .end method

```

هنا اه استخدم new-instance علشان بعرف class اللي هو Player وبيتني في v0 وبعد كده استخدم invoke-direct وخلاله <init> ان هو initial حاجة خلي ان القيمة اللي انشئها اللي هي <init> يساويها لل player وبعد كده استخدم invoke-virtual علشان ستدعي قيمة من object اللي هي collision وبعد كده استخدم move-result v1 عيني الاحدجة اللي هتجبها من return v1 وبعد كده خد القيمة اللي في v1 وخلالها تبقى collision وتبقي هتبقي boolean عمل v1 اللي هو يعمل collision

22	new-instance vx,type	Instantiates an object type and puts the reference of the newly created instance into vx.	2200 1500 - new-instance v0, java.io.FileInputStream // type@0015 Instantiates type@0015 (entry #15H in the type table) and puts its reference into v0.

	invoke-virtual { parameters }, methodtocal	Invokes a virtual method with parameters.	6E53 0600 0421 - invoke-virtual { v4, v0, v1, v2, v3}, Test2.method5:(IIII)V // method@0006 Invokes the 6th method in the method table with the following arguments: v4 is the "this" instance, v0, v1, v2, and v3 are the method parameters. The method has 5 arguments (4 MSB bits of the second byte) ⁵ .

70	invoke-direct { parameters }, methodtocal	Invokes a method with parameters without the virtual method resolution.	7010 0800 0100 - invoke-direct {v1}, java.lang.Object.<init>():V // method@0008 Invokes the 8th method in the method table with just one parameter, v1 is the "this" instance ⁵ .

0A	move-result vx	Move the result value of the previous method invocation into vx.	0A00 - move-result v0 Move the return value of a previous method invocation into v0.

هنا بقى ده مثال تانى اه لو عندك invoke-virtual object فهور هنا
 <-- {v0,v1} كده يستدعى حاجة داخل object player.increasepower(900)

The screenshot shows a debugger interface with assembly code. Handwritten annotations explain the code:

- Player player = new Player();** is highlighted with a green box and a red arrow pointing to the variable declaration.
- v1 = 0x384 // 900** is written below the assembly line.
- player.increasePower(v1) // 300** is written in red, indicating the target method and its parameter value.
- This parameter** is written in red near the assembly line.

```

.line 15
new-instance v0, Lcom/appthacking/smalithree/Player;
invoke-direct {v0}, Lcom/appthacking/smalithree/Player;::<init>()
.line 16
Local v0, "player":Lcom/appthacking/smalithree/Player
const/16 v1, 0x384
invoke-virtual {v0, v1}, Lcom/appthacking/smalithree/Player;::increasePower(I)V
.line 18
return-void
end method

```

هنا بقى هنستخدمها في ايه هو ان نعمل بها للي عاوزين نعمله مثلاً ننشئ حاجة مش موجودة نتنطى
 خاصية بینا نعمل بها للي عاوزين نعمله مثلاً ننشئ حاجة مش موجودة نتنطى
 invoke-virtual to create public method

ex : in this app after install the app and show the logcat and after analyze the system we find he contain method for levelup

so how we declare this method on my smali code and execute it

The screenshot shows the JADX GUI with the Player class code:

```

package com.appthacking.smalithree;
/* loaded from: classes2.dex */
public class Player {
    int level = 1;
    int power = 50;

    Player() {
        System.out.println("A new player object has been created!");
    }

    public void levelUP() {
        this.level++;
        System.out.println("New level is " + this.level);
    }

    public void increasePower(int power) {
        this.power = power;
        System.out.println("New power level is " + this.power);
    }
}

```

A yellow bar highlights the `increasePower(int power)` method.

```

Apr 21 11:04
adb logcat | grep -color=auto -i system.out
[...]
04-21 11:04:08.638 13781 13781 I System.out: A new player object has been created!
04-21 11:04:08.638 13781 13781 I System.out: New power level is 900

```

so on file MainActivity.smali we will add this line to execute this method

```
invoke-virtual {v0}, Lcom/apphacking/smalthree/Player; ->levelUP() V
```

bulid the app and signing it and remove it and reinstall it

here we make the method of level and we increase the level to level two by using this public method

```

Apr 21 11:13
adb logcat | grep -color=auto -i system.out
[...]
04-21 11:13:45.795 14479 14479 I System.out: A new player object has been created!
04-21 11:13:45.795 14479 14479 I System.out: New power level is 900
04-21 11:13:45.795 14479 14479 I System.out: A new player object has been created!
04-21 11:13:45.795 14479 14479 I System.out: New power level is 900
04-21 11:13:45.795 14479 14479 I System.out: New level is 2

```

وده كان مثل زى نستخدم public method علشان نستدعيها او لو مش وجودة ننشئها وممكن نعمل بيه حاجات كتير

74-Static Method

لو عاوزين بقى نستدعى static method برضه هي نفسها زي invoke-virtual public method ولكن بدل invoke-static هتبقى

تُستخدم لاستدعاء دالة ثابتة (Static Method) داخل الكود

الدوال الثابتة يمكن استدعاوها مباشرة دون الحاجة إلى إنشاء كائن (Object) من الكلاس.

71	invoke-static {parameters}, methodtocall	Invokes a static method with parameters.	7110 3400 0400 - invoke-static {v4}, java.lang.Integer.parseInt(Ljava/lang/String;)I // method@0034 Invokes method@34 static method. The method is called with one parameter, v4 ⁵ .

Static Methods

SMALI - IMPORTANT BLOCKS

```
4 package com.apphacking.smalione;
5
6 public class Level {
7
8     public static int level = 0;
9
10    public static void nextLevel() {
11        level += 1;
12    }
13 }
```

```
1 package com.apphacking.smalione;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import android.os.Bundle;
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12
13         Level.nextLevel();
14     }
15 }
```

A diagram illustrating a call from the `onCreate` method of `MainActivity` to the `nextLevel` method of `Level`. A green arrow points from the `Level.nextLevel();` call in `MainActivity` to the `nextLevel()` method definition in `Level.java`. Another green arrow points from the `nextLevel()` method definition back to the call in `MainActivity`.

```

4     .method protected onCreate(Landroid/os/Bundle;)V
5         .locals 1
6         .param p1, "savedInstanceState"    # Landroid/os/Bundle;
7
8         .line 11
9         invoke-super {p0, p1}, Landroidx/appcompat/app/AppCompatActivity;.>onCreate(Landroid/os/Bundle;)V
10
11        .line 12
12        const v0, 0x7f0b001c
13
14        invoke-virtual {p0, v0}, Lcom/apphacking/smålione/MainActivity;.>setContentView(I)V
15
16        .line 14
17        invoke-static {}, Lcom/apphacking/smålione/Level;.>nextLevel()V
18
19        .line 16
20        return-void

```

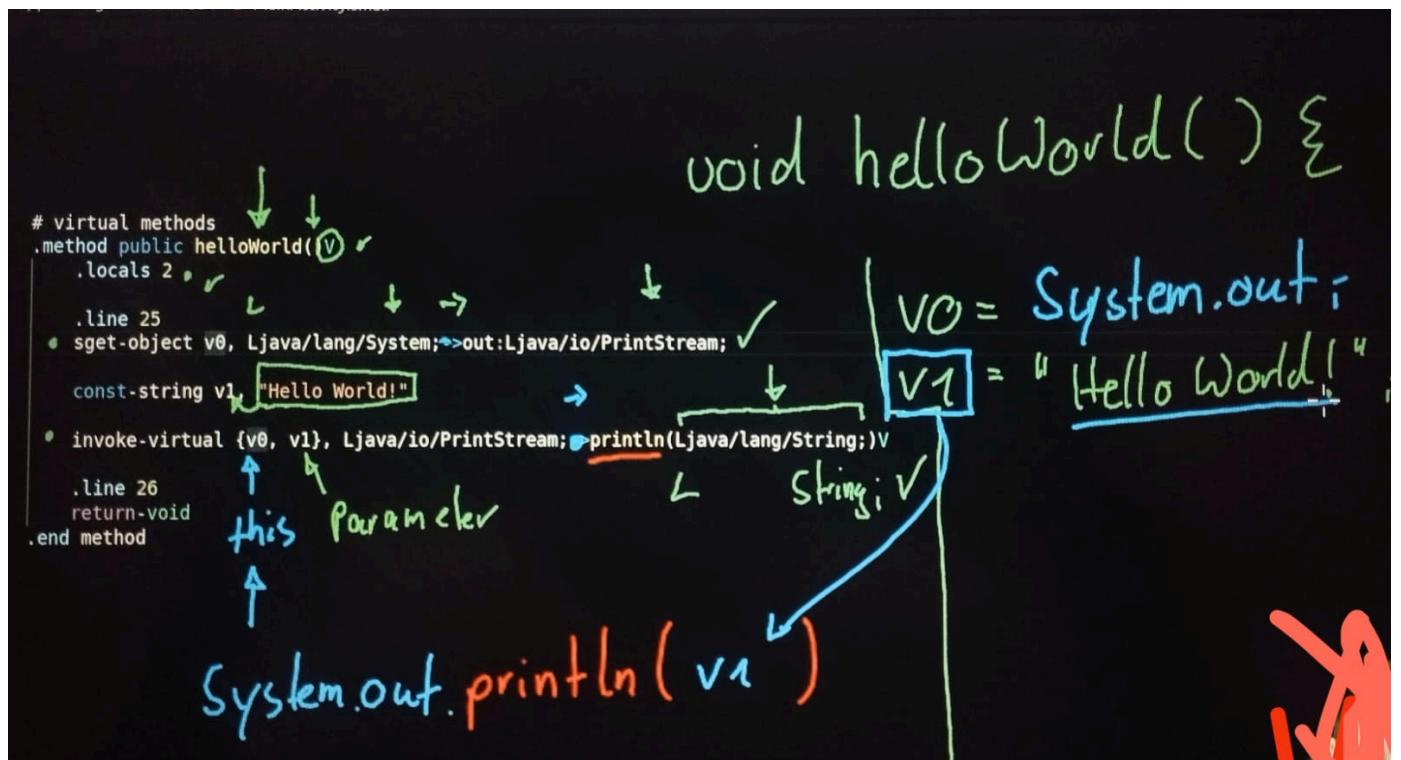
خلي بالك لو `system.out.println("this is static method")` ده هنا بيظهر في `logcat`

75- Hello world printed

ازاي بقى لو معملناش `class` وعاوزين نطبع حاجة زي مثلا زى ده `java`

هنا اه مش موجود ان هو ينشئ `object` ولكن لو حللنا الكود ده هنلاقى ان `sget-object` اللي هو `object` وبتستدعي `system` اللي جواه اللي هو `PrintStream` وتخزنها في `v0 --> v0=System.PrintStream`

بعد كده بخزن في `v1` "hello world" string المان `Println` method اللي هي `v1="hello world"`
علشان تطبع وبيأخذ 2 parmater this and `v1("hello world")`



another example

Objects and Methods

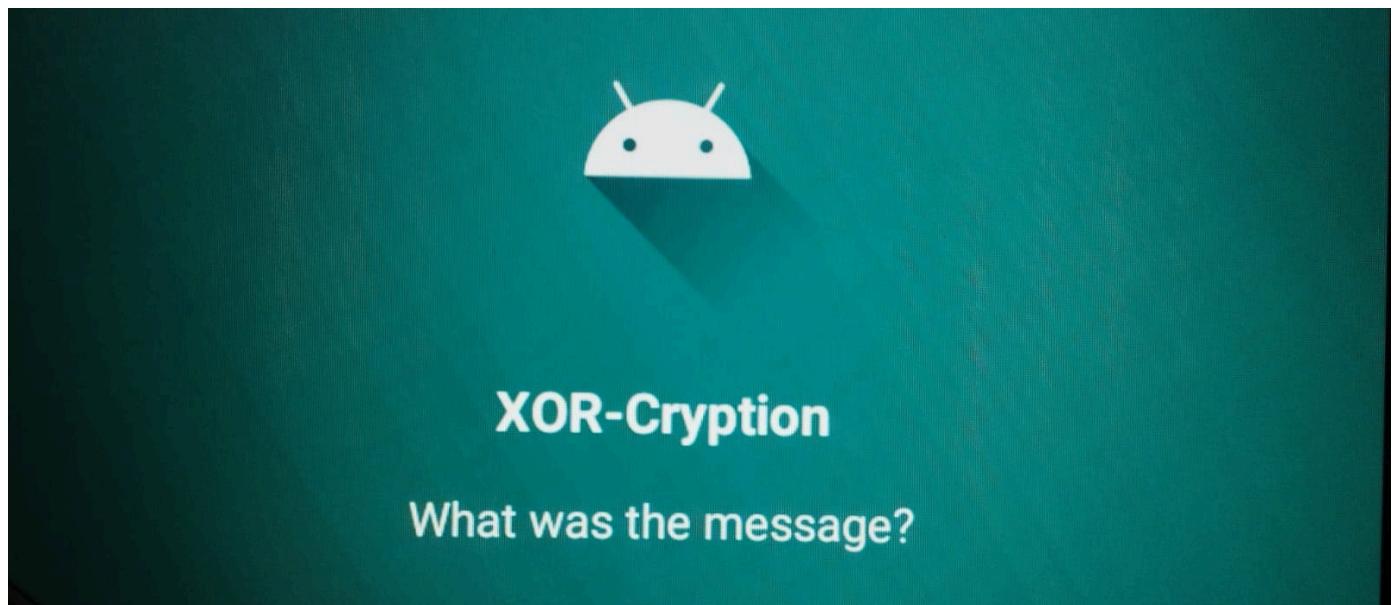
SMALI - IMPORTANT BLOCKS

```
3
4  public void helloWorld() {
5
6      String text = "Hello World!";
7      System.out.println(text);
8
9  }
```

```
3 # virtual methods
4 .method public helloWorld()V
5     .locals 2
6
7     .line 19
8     const-string v0, "Hello World!"
9
10    .line 20
11    .local v0, "text":Ljava/lang/String;
12    sget-object v1, Ljava/lang/System;->out:Ljava/io/PrintStream;
13
14    invoke-virtual {v1, v0}, Ljava/io/PrintStream;.>println(Ljava/lang/String;)V
15
16    .line 22
17    return-void
18 .end method
```

76-Printing out the Message Real World Example (Encryption) : Important

هنا اه في عبارة عن CTF وعاوزين نعرف ايه هي message



لو شوفنا بقى الكود دلوقتي بتاع اللي بتعمل function encryption : هنلاقي ان هي بتاخد string وبتعمل عملية XOR مع string دلوقتي هنشوف APPHACKING

AndroidManifest.xml XOREncryption MainActivity

```

package com.apphacking.encryption;

/* loaded from: classes.dex */
public class XOREncryption {
    public static String encryptDecrypt(String input) {
        char[] key = {'A', 'P', 'P', 'H', 'A', 'C', 'K', 'I', 'N', 'G'};
        StringBuilder output = new StringBuilder();
        for (int i = 0; i < input.length(); i++) {
            output.append((char) (input.charAt(i) ^ key[i % key.length]));
        }
        return output.toString();
    }

    this.textView = (TextView) findViewById(R.id.textViewMessage);
    XOREncryption.encryptDecrypt("589;a*8i#a#5+3&i#" + "2#1/$");
    this.textView.setText("What was the message?");
}

```

ایه هی اصلا XOR : هی عبارة عن operation بتقارن بین 2 bit ولو مختلفین بطلع 1 ولو متشابهین بطلع 0 زی کده مثلما

$$\begin{array}{ccccccc}
 01010111 & 01101001 & 01101011 & 01101001 \\
 \oplus & & & & & & \\
 11110011 & 11110011 & 11110011 & 11110011 \\
 = & 10100100 & 10011010 & 10011000 & 10011010
 \end{array}$$

زی بقی بنعمله مع string هو ان احنا بنشوف رقم char في ASCII ونحوله ل binary وبعد کده نعمل عملية XOR زی کده مثلما

1. Convert Characters to ASCII (Decimal):

Each character in the string is converted to its ASCII (decimal) value. For example:

- "h" → 104
- "w" → 119

2. Convert ASCII to Binary:

Each ASCII value is represented as an 8-bit binary number:

- 104 → 01101000
- 119 → 01110111

3. Apply XOR (Bit by Bit):

Perform the XOR operation between corresponding bits of the two binary numbers:

- 01101000 (104, "h")

- `01110111` (119, "w")

```

01101000
XOR 01110111
-----
00011111 (31 in decimal)

```

4. (Optional) Convert Back to ASCII:

You can convert the XOR result back to ASCII for storage or transmission, though it might result in non-printable characters.

5. Reverse the Process (Decryption):

To decrypt, you XOR the encrypted value with the same key. For example:

- `31` → `00011111` (binary)
- Key: `119` → `01110111` (binary)

```

00011111
XOR 01110111
-----
01101000 (104 in decimal, "h")

```

code python for this operation

```

#!/usr/bin/python3

def encrypt_decrypt(input_string, key):
    # Initialize the output as a list of characters
    output = []

    # Perform XOR operation
    for i in range(len(input_string)):
        # XOR each character in input_string with the corresponding
        # character in the key
        output.append(chr(ord(input_string[i]) ^ ord(key[i % len(key)])))

    # Join the list to form the resulting string
    return ''.join(output)

# Example Usage
input_string = input("Enter the string to encrypt/decrypt: ")
key = "APPHACKING" # Example key

# Encrypt/Decrypt
result = encrypt_decrypt(input_string, key)

```

```
# Display Results
print("Result (raw):", result)
print("Result (hex):", result.encode().hex())
```

he will convert the message to hex and we will convert from hex to ascii to know the secret message

بس دلوقتي تعالى نحل smali code

```
XOREncryption.encryptDecrypt("589;a*8i#>a#5+3&?i#\\"2#1/$");
this.textView.setText("What was the message?");
```

```
.line 21
const-string v0, "589;a*8i#>a#5+3&?i#\\"2#1/$"

invoke-static {v0}, Lcom/apphacking/encryption/XOREncryption;->encryptDecrypt(Ljava/lang/String;)Ljava/lang/String;
move-result-object v0

.line 22
.local v0, "encrypted":Ljava/lang/String;
iget-object v1, p0, Lcom/apphacking/encryption/MainActivity;->textView:Landroid/widget/TextView;

const-string v2, "What was the message?"

invoke-virtual {v1, v2}, Landroid/widget/TextView;->setText(Ljava/lang/CharSequence;)V

.line 24
return-void
.end method
```

هنا اه من 21 -- < هو هنا بيذخن string في v0 وبعج كده بيعمل invoke-static ان هو بيستدعي XOREncryption.encryptDecrypt(v0)

1- First Solution

وبع كده واه حاجو بيذخن القيمة اللي بترجع في v0--> move-result-object v0 بقى كده عرفنا ان ناتج هيقى في v0 من خال smali كود

```
.line 21
const-string v0, "589;a*8i#>a#5+3&?i#\\"2#1/$"

invoke-static {v0}, Lcom/apphacking/encryption/XOREncryption;-
>encryptDecrypt(Ljava/lang/String;)Ljava/lang/String;

move-result-object v0
```

كده بقى عرفنا ان القيمة هترجع في v0 عاوزين بقى ننسخدم smali علشان نحصل على القيمة منه

to print the value of v0 we need to use System.out.println(v0) so we need to create object and to create those objects from Delvik Operation Codes

:

1. **sget-object System.out**

2. **println() --> public method so we need to invoke-virtual**

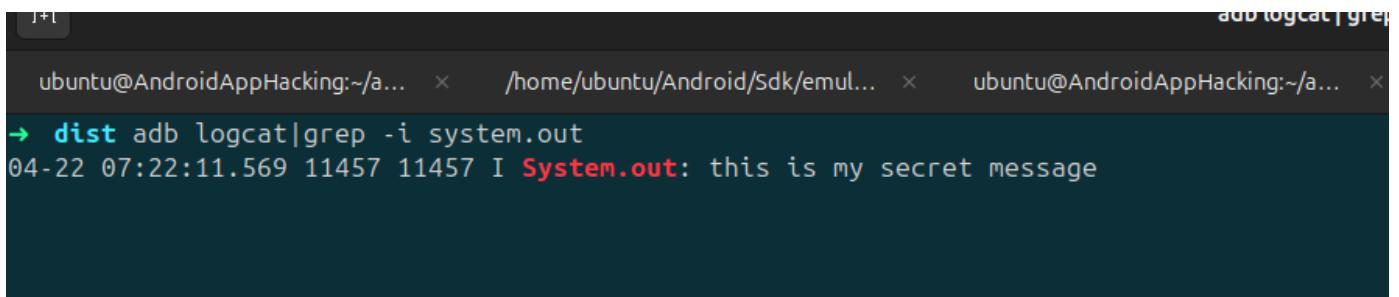
3. **System.out.println**

62	sget-object vx,field_id	Reads the object reference field identified by the field_id into vx.	6201 0C00 - sget-object v1, Test3.os1:Ljava/lang/Object; // field@000c Reads field@000c (entry #CH in the field id table) into v1.

```
sget-object v3,Ljava/lang/System; ->out: Ljava/io/PrintStream;  
invoke-virtual {v3,v0},Ljava/io/PrintStream; ->println(Ljava/lang/String;)V
```

دلوقي بقى لو علمنا build وشوفنا logcat

```
adb logcat | grep -i system.out
```

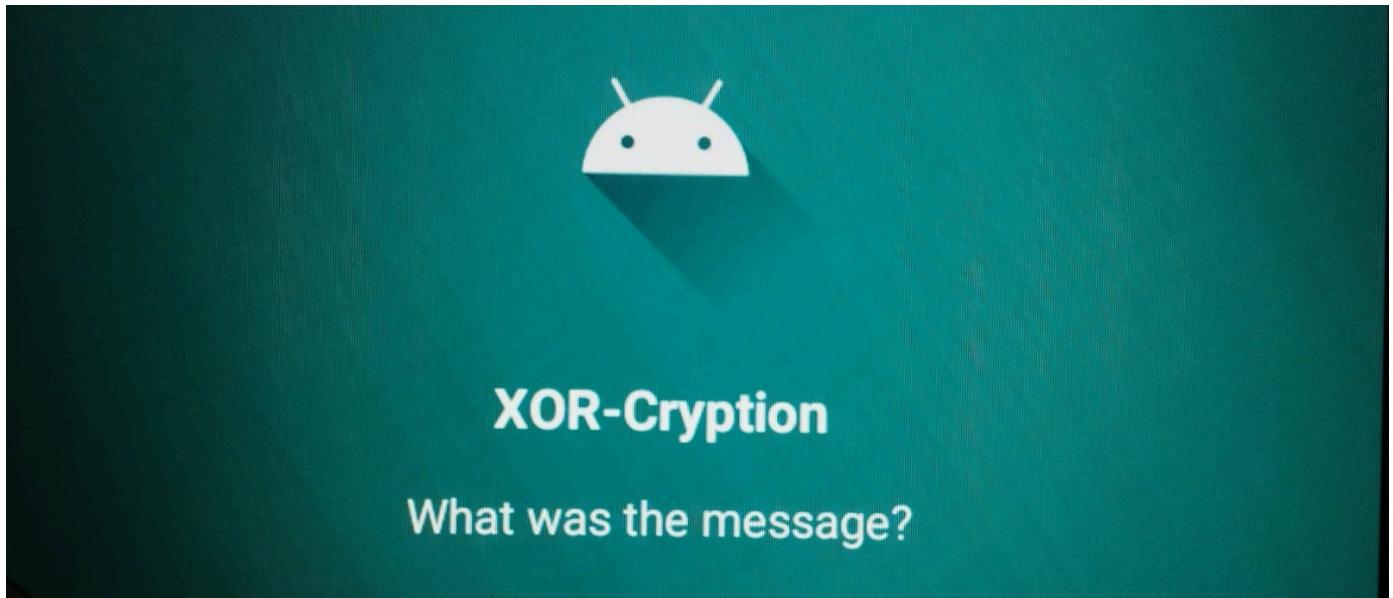


```
ubuntu@AndroidAppHacking:~/a... × /home/ubuntu/Android/Sdk/emul... × ubuntu@AndroidAppHacking:~/a... ×  
→ dist adb logcat|grep -i system.out  
04-22 07:22:11.569 11457 11457 I System.out: this is my secret message
```

the decrypt message : **this is my secret message**

2-Second Soulution

هنا بقى بدل ما نخليه يظهر في logcat نخليه يطبع في الشاشة بدل v2 : يعني هنا هو بيجب على الشاشة قيمة v2 اللي هي "message" احنا بدل دي بقى نحط القيمة اللي هتتيجي بعد ما يروح لل encryptDecrypt function اللي هي move-result-object v2 نخليها move-result-object v0 وبكده هي هيطبع الناتج



```
.line 21
const-string v0, "589;a*8i#>a#5+3&?i#\\"2#1/$"

invoke-static {v0}, Lcom/apphacking/encryption/XOREncryption;->encryptDecrypt(Ljava/lang/String;)Ljava/lang/String;

move-result-object v0

.line 22
.local v0, "encrypted":Ljava/lang/String;
iget-object v1, p0, Lcom/apphacking/encryption/MainActivity;->textView:Landroid/widget/TextView;

const-string v2, "What was the message?"

invoke-virtual {v1, v2}, Landroid/widget/TextView;->setText(Ljava/lang/CharSequence;)V

.line 24
return-void
.end method
```

هنا بقى الكود اللي هنعمله علشان نغير ل v2

```
.line 21
const-string v0, "589;a*8i#>a#5+3&?i#\\"2#1/$"

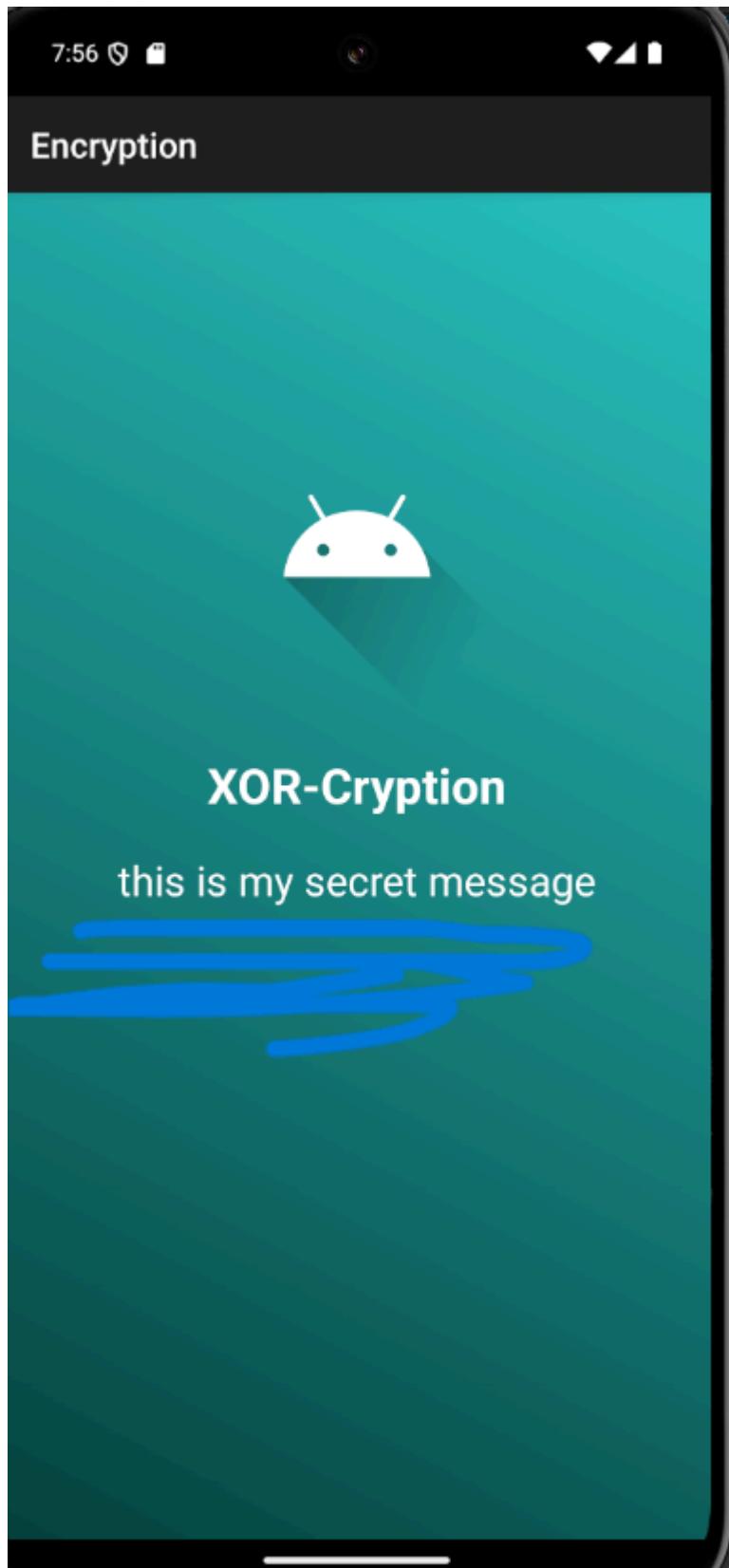
invoke-static {v0}, Lcom/apphacking/encryption/XOREncryption;->encryptDecrypt(Ljava/lang/String;)Ljava/lang/String;
move-result-object v2

.line 22
.local v0, "encrypted":Ljava/lang/String;
iget-object v1, p0, Lcom/apphacking/encryption/MainActivity;->textView:Landroid/widget/TextView;

invoke-virtual {v1, v2}, Landroid/widget/TextView;->setText(Ljava/lang/CharSequence;)V

.line 24
return-void
.end method
```

وبكده هيظهر على الشاشة



78- One Challenge To Recep All : part 1

try and catch on smali

1. **try Block:**

- o You specify the range of instructions that belong to the `try` block using `.try` and `.catch` directives.

2. `catch` Block:

- You define the exception type to catch and the label where the exception should be handled.

Example in Smali

```
.method public exampleTryCatch()V
.locals 1

:try_start
# Code in try block
const v0, 0x0          # Assign 0 to v0
div-int v0, v0, v0      # Divide by zero (causes ArithmeticException)
goto :try_end           # If no exception, skip the catch block

.catch Ljava/lang/ArithmeticException; {:try_start .. :try_end}
:catch_block

:try_end
return-void

:catch_block
# Code to handle exception
const-string v0, "Exception caught!"
invoke-static {v0}, Landroid/util/Log;->d(Ljava/lang/String;)I
return-void
.end method
```

Explanation

1. `try` Block (`:try_start .. :try_end`):

- The `try` block starts at `:try_start` and ends at `:try_end`.
- Any exception that occurs within this range will trigger the `catch` block.

2. `catch` Directive:

- The `.catch` directive specifies:
 - The type of exception to catch: `Ljava/lang/ArithmeticException;`.
 - The range of instructions to monitor for exceptions: `:try_start .. :try_end`.
 - The label to jump to if the exception is caught: `:catch_block`.

3. Exception Handling (`:catch_block`):

- This block is executed if the specified exception is thrown.

- In this example, it logs the message "Exception caught!".

4. Flow Control:

- If no exception occurs, the execution continues normally after `:try_end`.

دلوقي بقى فيالتطبيق ده هو بيستخدم AES encryption هنا في الكود ده لو لاقينا زي كده تشفير بيبقى اهم حاجة لازم نفهم ان نعرفه هو
seceret key and Initialization vector

وعلشان نجيب seceret key لازم نستدعى `getEncoded()` method اللي هي

`getEncoded`

```
public byte[] getEncoded ()
```

Returns the key material of this secret key. --> return the value of seceret key

AESCrypto

```
package com.apphacking.aes;

import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.KeyGenerator;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.SecretKey;

/* loaded from: classes2.dex */
15 public class AESCrypto {
16     public static byte[] encryptMessage(byte[] plaintext) {
17         try {
18             KeyGenerator keygen = KeyGenerator.getInstance("AES");
19             keygen.init(256);
20             SecretKey key = keygen.generateKey();
21             Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
22             cipher.init(1, key);
23             byte[] ciphertext = cipher.doFinal(plaintext);
24             cipher.getIV();
25             return ciphertext;
26         } catch (InvalidKeyException e) {
27             e.printStackTrace();
28             return null;
29         } catch (NoSuchAlgorithmException e2) {
30             e2.printStackTrace();
31             return null;
32         } catch (BadPaddingException e3) {
33             e3.printStackTrace();
34             return null;
35         } catch (IllegalBlockSizeException e4) {
36             e4.printStackTrace();
37             return null;
38         } catch (NoSuchPaddingException e5) {
39             e5.printStackTrace();
40             return null;
41         }
42     }
43 }
```

فلازم نضيفها بقى في الكود اما java او smali وتعمل print لها وبكده يبقى حصلنا على secert key : هنا بقى اللي بتحاجين نضيفه علشان
نجيب secert key

```
myArray=key.getEncoded();
secertkey=Base64.encodeToString(myArray,0);
System.out.println(secertkey);
```

smali

```
invoke-interface {v1},Ljava/crypto/SecertKey;->getEncoded();[B
move-result-object v2
.line 36
const/4 v3,0x0
invoke-static {v2,v3},Landroid/util/Base64;-
```

```

>encodeToString([BI)Ljava/lang/String;
move-result-object v3;
.line 38
sget-object v4,Ljava/lang/System;->out:Ljava/io/PrintStream;
invoke-virtual {v4,v3},Ljava/io/PrintStream;->println(Ljava/lang/String;)V
.line 42
return-void

```

Handwritten annotations on the assembly code:

- v0 = 0x0**
- v0 -> KeyGenerator**
- v1 = "AES"**
- v1 = KeyGenerator**
- v0 = KeyGenerator**
- v1 = 0x100**
- v1 = SecretKey**
- ! VO / V1**
- Do not over-ride!**
- v1 = SecretKey**
- v2 = [B** // key in bytes
- v3 = 0x0**
- v3 = Base64.encodeToString(v2,v3);**
- v4 = System.out**
- System.out.println(v3)**

```

16 method public static encryptMessage([B)B
17   .locals 5
18   .param p0, "plaintext" # [B
19
20   .line 17
21   const/4 v0, 0x0 ✓
22
23   .line 19
24   .local v0, "keygen":Ljavax/crypto/KeyGenerator; ✓
25   :try_start_0
26   const-string v1, "AES" ✓
27
28   invoke-static {v1}, Ljavax/crypto/KeyGenerator:->getInstance(Ljava/lang/String;)Ljavax/crypto/KeyGenerator;
29
30   move-result-object v1
31
32   move-object v0, v1
33
34   .line 20
35   const/16 v1, 0x100
36
37   invoke-virtual {v0, v1}, Ljavax/crypto/KeyGenerator:->init(I)V
38
39   .line 21
40   invoke-virtual {v0}, Ljavax/crypto/KeyGenerator:->generateKey()Ljavax/crypto/SecretKey;
41
42   move-result-object v1
43
44   ##### Beginning of our code #####
45
46   .line 35
47   invoke-interface {v1}, Ljavax/crypto/SecretKey:->getEncoded()[B
48
49   move-result-object v2 ✓
50
51   .line 36
52   const/4 v3, 0x0 <✓
53
54   invoke-static {v2,v3}, Landroid/util/Base64:->encodeToString([BI)Ljava/lang/String;
55
56   move-result-object v3 ✓
57
58   .line 38
59   sget-object v4, Ljava/lang/System;->out:Ljava/io/PrintStream;
60
61   invoke-virtual {v4,v3}, Ljava/io/PrintStream;->println(Ljava/lang/String;)V
62
63   ##### End of our code #####
64
65   .line 22
66   .local v1, "key":Ljavax/crypto/SecretKey;
67   const-string v2, "AES/CBC/PKCS5PADDING"
68
69   invoke-static {v2}, Ljavax/crypto/Cipher:->getInstance(Ljava/lang/String;)Ljavax/crypto/Cipher;
70
71   move-result-object v2

```

```

ls
out.apk out.apk.idsig
adb install out.apk
failed to install out.apk: Failure [INSTALL_FAILED_TEST_ONLY: installPackageLI]
adb install -t apk

adb logcat | grep -i system.out
21:21:42.903 13746 13746 I System.out: Encrypted Text
21:21:42.903 13746 13746 I System.out: 0nUF/yKT7x1roa4Wk2oSCZKaxzBhpVoC/kvm/Rv5ASw=

```



Read me if you can:

U3yKPS8Gp3oX6KMl3Z5tPxuo
gRswYeuvkORtp8jl6k=

وبهذه عرفنا نجيب Secert Key اللي بيتنشأ من AES encryption

part 2

هنا بقى هنشرح ازاي بدل ما نخلي مثلا الحاجة اللي عايزنها تظهر تكون على screen نفسه يعني متظاهرش في **MainActivity** وعلشان تظهر لازم تربطها ب **activity** لأن **page** هو **activity** لأن **logcat** هو **activity** موجود فقط

لو عاوزين ان اول ما نفتح التطبيق يظهر **text**

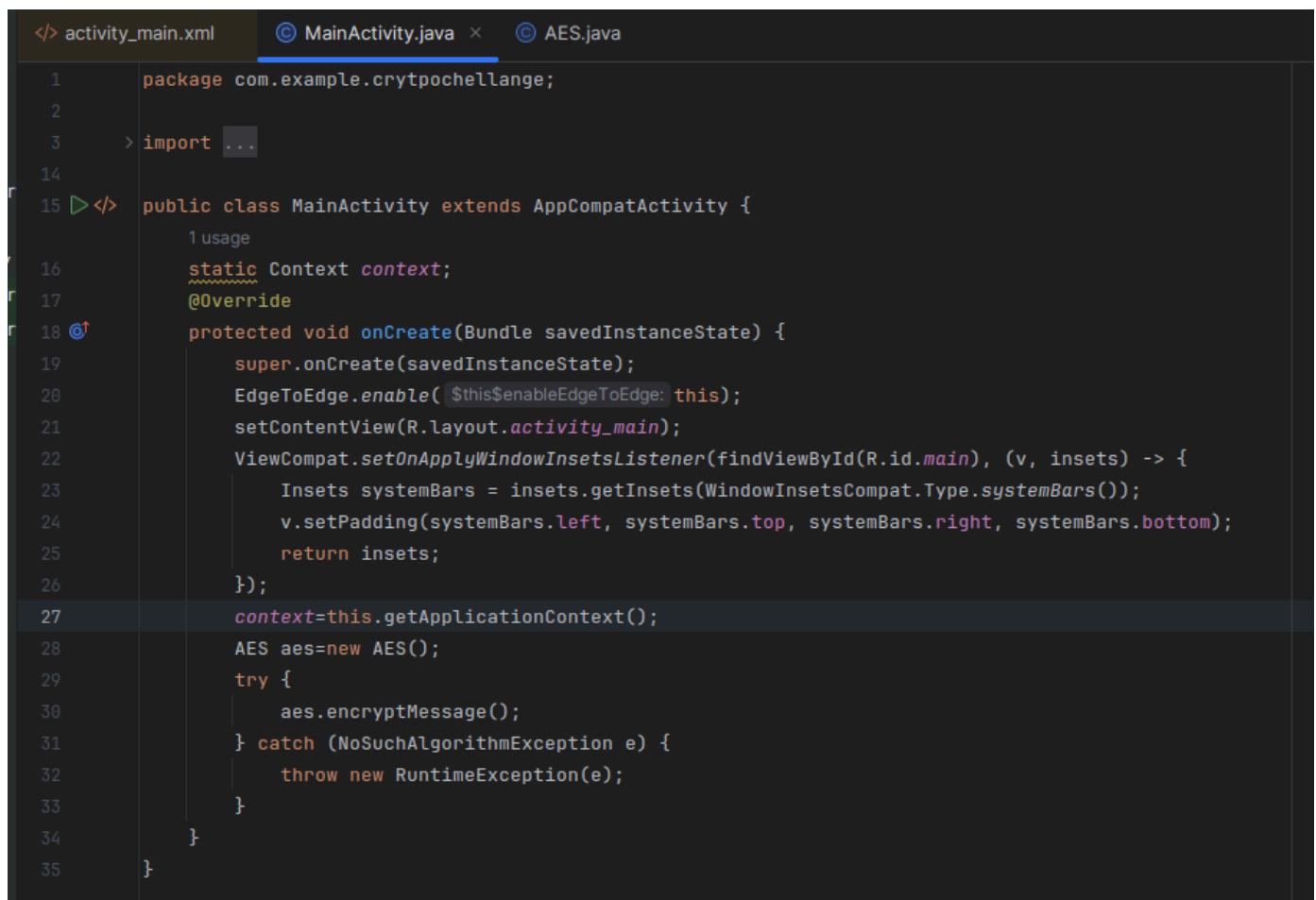
```
Toast.makeText(this.getApplicationContext(), "Hacking Your Application",
Toast.LENGTH_SHORT).show();
```

علشان نخلي **text** يظهر على الشاشة بنستخدم **Context**

```
in MainActivity
static Context context;
context=this.getApplicationContext();
in AES
Toast.makeText(MainActivity.context,"this pap",Toast.LENGTH_LONG).show();
```

هنا بقى مثل على اللي عملناه

1-MainActivity code

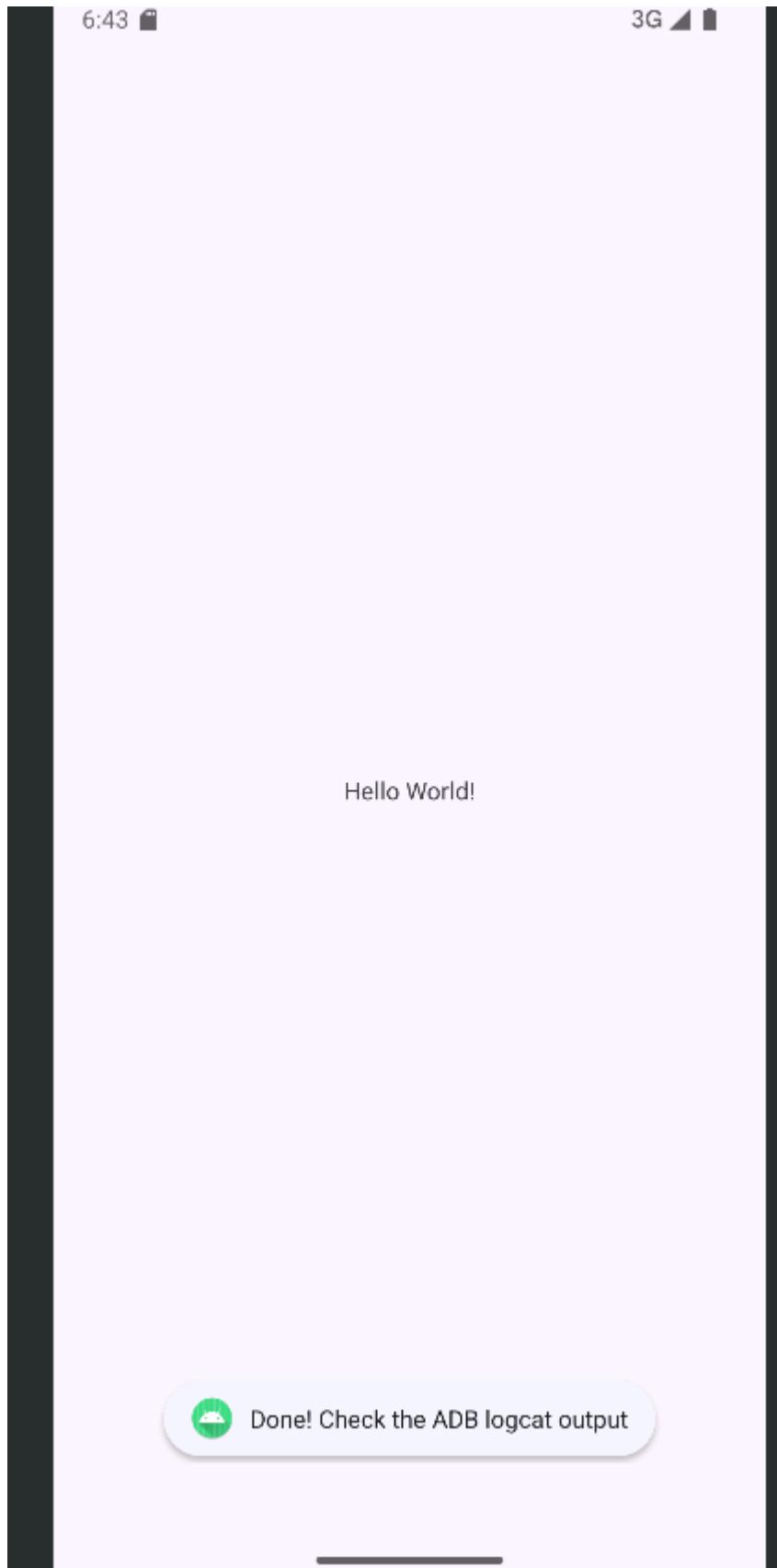


```
</> activity_main.xml   © MainActivity.java x   © AES.java
1 package com.example.cryptpochellange;
2
3 > import ...
4
5 D</> public class MainActivity extends AppCompatActivity {
6     1 usage
7     static Context context;
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         EdgeToEdge.enable( $this$enableEdgeToEdge: this);
12         setContentView(R.layout.activity_main);
13         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
14             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
15             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
16             return insets;
17         });
18         context=this.getApplicationContext();
19         AES aes=new AES();
20         try {
21             aes.encryptMessage();
22         } catch (NoSuchAlgorithmException e) {
23             throw new RuntimeException(e);
24         }
25     }
26 }
```

2-AES code

```
</> activity_main.xml ④ MainActivity.java ④ AES.java ×

1 package com.example.cryptpochellange;
2
3 import android.widget.Toast;
4
5 import java.security.NoSuchAlgorithmException;
6
7 import javax.crypto.KeyGenerator;
8 import javax.crypto.SecretKey;
9
10 2 usages
11 public class AES {
12     1 usage
13     public void encryptMessage() throws NoSuchAlgorithmException {
14         KeyGenerator keygen= KeyGenerator.getInstance( algorithm: "AES");
15         keygen.init( keysize: 256);
16         SecretKey key=keygen.generateKey();
17         Toast.makeText(MainActivity.context, text: "Hacking into your Application",Toast.LENGTH_LONG).show();
18         byte[] byteArray;
19         byteArray=key.getEncoded();
20         String keyString="0x";
21         for(int i=0;i<byteArray.length;i++){
22             keyString+=String.format("%02x",byteArray[i]);
23         }
24         System.out.println(keyString);
25         Toast.makeText(MainActivity.context, text: "Done! Check the ADB logcat output",Toast.LENGTH_LONG).show();
26     }
27 }
```



Logcat Logcat +

Medium Phone API 35 (emulator-5554) Android 15, A system.out

```
----- PROCESS STARTED (4468) for package com.example.cryptochallenge -----
2025-04-24 06:42:53.137 4468-4468 System.out com.example.cryptochallenge [I] this is the key0xa75879a7a6164ec75e2b3822d03464231b423c95f29b09e17a07b7a8d01cac3
----- PROCESS STARTED (4468) for package com.example.cryptochallenge -----
2025-04-24 06:42:53.137 4468-4468 System.out com.example.cryptochallenge [I] this is the key0xa75879a7a6164ec75e2b3822d03464231b423c95f29b09e17a07b7a8d01cac3
```

دلوقي بقى الكود اللي عملناه ده هنأخذ بقى كود smali ونحطه في التطبيق بتاعنا اللي هنعدل عليه: هنا اه java code for app دلوقي بقى

اللي محتاجين نعمل عليها من بعد سطر 21 هنأخذ كود smali من التطبيق اللي عملناه ونحطه في كود smali للتطبيق ده

```
package com.apphacking.aes;

import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.KeyGenerator;
import javax.crypto.NoPaddingException;
import javax.crypto.SecretKey;

/* loaded from: classes2.dex */
public class AESCrypto {
    public static byte[] encryptMessage(byte[] plaintext) {
        try {
            KeyGenerator keygen = KeyGenerator.getInstance("AES");
            keygen.init(256);
            SecretKey key = keygen.generateKey();
            Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
            cipher.init(1, key);
            byte[] ciphertext = cipher.doFinal(plaintext);
            cipher.getIV();
            return ciphertext;
        } catch (InvalidKeyException e) {
            e.printStackTrace();
            return null;
        } catch (NoSuchAlgorithmException e2) {
            e2.printStackTrace();
            return null;
        } catch (BadPaddingException e3) {
            e3.printStackTrace();
            return null;
        } catch (IllegalBlockSizeException e4) {
            e4.printStackTrace();
            return null;
        } catch (NoPaddingException e5) {
            e5.printStackTrace();
            return null;
        }
    }
}
```

هنا اه انا عامل للكود اللي المفروض هروح نحطه في app comment

```
##### Class com.example.crytpochellange.AES
(com.example.crytpochellange.AES)
.class public Lcom/example/crytpochellange/AES;
.super Ljava/lang/Object;
.source "AES.java"
```

```
# direct methods
.method public constructor <init>()V
    .registers 1
```

```
.line 10
invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    return-void
.end method

# virtual methods
.method public encryptMessage ()V
    .registers 10
    .annotation system Ldalvik/annotation/Throws;
        value = {
            Ljava/security/NoSuchAlgorithmException;
        }
.end annotation

.line 12
const-string v0, "AES"

    invoke-static {v0}, Ljavax/crypto/KeyGenerator;-
>getInstance(Ljava/lang/String;)Ljavax/crypto/KeyGenerator;

move-result-object v0

.line 13
.local v0, "keygen":Ljavax/crypto/KeyGenerator;
const/16 v1, 0x100

    invoke-virtual {v0, v1}, Ljavax/crypto/KeyGenerator;->init(I)V

.line 14
    invoke-virtual {v0}, Ljavax/crypto/KeyGenerator;->generateKey()Ljavax/crypto/SecretKey;

move-result-object v1
#####      form here begin of out code #####
.line 15
.local v1, "key":Ljavax/crypto/SecretKey;
sget-object v2, Lcom/example/crytpochellange/MainActivity;-
>context:Landroid/content/Context;

const-string v3, "Hacking into your Application"
```

```
const/4 v4, 0x1

    invoke-static {v2, v3, v4}, Landroid/widget/Toast;-
>makeText(Landroid/content/Context;Ljava/lang/CharSequence;I)Landroid/widget/
/Toast;

move-result-object v2

invoke-virtual {v2}, Landroid/widget/Toast;->show()V

.line 17
invoke-interface {v1}, Ljavax/crypto/SecretKey;->getEncoded() [B

move-result-object v2

.line 18
.local v2, "byteArray":[B
const-string v3, "0x"

.line 19
.local v3, "keyString":Ljava/lang/String;
const/4 v5, 0x0

.local v5, "i":I
:goto_22
array-length v6, v2

if-ge v5, v6, :cond_49

.line 20
new-instance v6, Ljava/lang/StringBuilder;

invoke-direct {v6}, Ljava/lang/StringBuilder;-><init>()V

invoke-virtual {v6, v3}, Ljava/lang/StringBuilder;-
>append(Ljava/lang/String;)Ljava/lang/StringBuilder;

move-result-object v6

aget-byte v7, v2, v5

invoke-static {v7}, Ljava/lang/Byte;->valueOf(B)Ljava/lang/Byte;
```

```
move-result-object v7

filled-new-array {v7}, [Ljava/lang/Object;

move-result-object v7

const-string v8, "%02x"

invoke-static {v8, v7}, Ljava/lang/String; ->format(Ljava/lang/String;
[Ljava/lang/Object;)Ljava/lang/String;

move-result-object v7

invoke-virtual {v6, v7}, Ljava/lang/StringBuilder; -
>append(Ljava/lang/String;)Ljava/lang/StringBuilder;

move-result-object v6

invoke-virtual {v6}, Ljava/lang/StringBuilder; -
>toString()Ljava/lang/String;

move-result-object v3

.line 19
add-int/lit8 v5, v5, 0x1

goto :goto_22

.line 22
.end local v5      # "i":I
:cond_49
sget-object v5, Ljava/lang/System; ->out:Ljava/io/PrintStream;

new-instance v6, Ljava/lang/StringBuilder;

invoke-direct {v6}, Ljava/lang/StringBuilder; -><init>()V

const-string v7, "this is the key"

invoke-virtual {v6, v7}, Ljava/lang/StringBuilder; -
>append(Ljava/lang/String;)Ljava/lang/StringBuilder;
```

```
move-result-object v6

    invoke-virtual {v6, v3}, Ljava/lang/StringBuilder;-
>append(Ljava/lang/String;)Ljava/lang/StringBuilder;

move-result-object v6

    invoke-virtual {v6}, Ljava/lang/StringBuilder;-
>toString()Ljava/lang/String;

move-result-object v6

    invoke-virtual {v5, v6}, Ljava/io/PrintStream;-
>println(Ljava/lang/String;)V

.line 23
sget-object v5, Lcom/example/cryptochellange/MainActivity;-
>context:Landroid/content/Context;

const-string v6, "Done! Check the ADB logcat output"

invoke-static {v5, v6, v4}, Landroid/widget/Toast;-
>makeText(Landroid/content/Context;Ljava/lang/CharSequence;I)Landroid/widget/
/Toast;

move-result-object v4

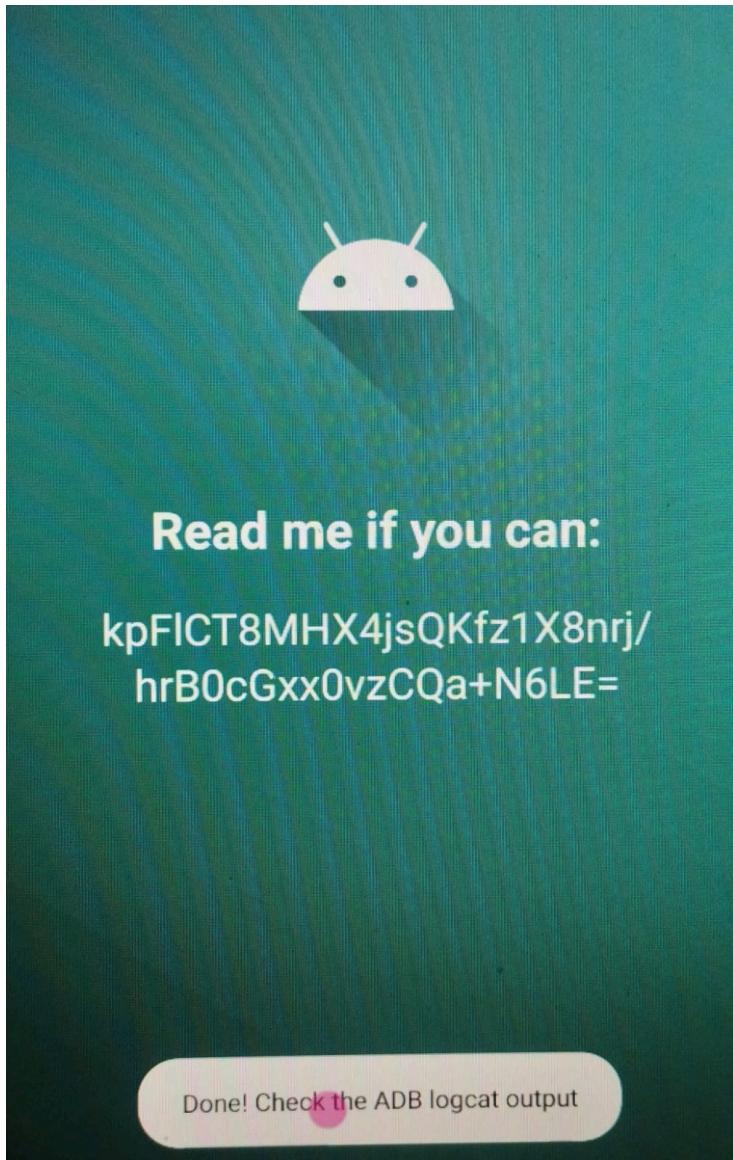
invoke-virtual {v4}, Landroid/widget/Toast;->show()V
##### to here end of our code #####
.line 24
return-void
.end method
```



Read me if you can:

**kpFlCT8MHX4jsQKfz1X8nrj/
hrB0cGxx0vzCQa+N6LE=**

Hacking into your application



```
▶ ~ adb logcat | grep -i Crypto
01-18 22:14:06.397 14916 14916 V NativeCrypto: Registering com/google/android/gms/org/conscrypt/Nativ
eCrypto's 294 native methods...
01-18 23:35:12.419 15240 15240 V NativeCrypto: Registering com/google/android/gms/org/conscrypt/Nativ
eCrypto's 294 native methods...
01-18 23:35:36.644 15916 15946 V NativeCrypto: Registering com/google/android/gms/org/conscrypt/Nativ
eCrypto's 294 native methods...
^C
→ ~ adb logcat | grep -i CryptoKey
01-18 23:36:29.959 16144 16144 I System.out: CryptoKey: 0x9d6f0ad3d7de6850004589274708deaec8747d2664
16ff1c094a95aa1fd66ff
```

part 3

هنا هنشوف ازاي هنجيب IV: اللي المفروض نعمله ان نعرف Cipher.iv ونحط فيه القيمة بتاعت ByteArray وبعد كده محتاجين نطبعه ب System.out.println

```
System.out.println("IV: " +
Base64.encodeToString(cipher.getIV(), 0));
System.out.println("Return Value: " +
Base64.encodeToString(ciphertext, 0));
```

Java code

```
AESEncryptor x
package com.apphacking.aes;

import android.util.Base64;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.KeyGenerator;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.SecretKey;

/* loaded from: classes2.dex */
15 public class AESEncryptor {
16     public static byte[] encryptMessage(byte[] plaintext) {
17         System.out.println("Param: " + Base64.encodeToString(plaintext, 0));
18         try {
19             KeyGenerator keygen = KeyGenerator.getInstance("AES");
20             keygen.init(256);
21             SecretKey key = keygen.generateKey();
22             byte[] myArray = key.getEncoded();
23             System.out.println("Key: " + Base64.encodeToString(myArray, 0));
24             Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
25             cipher.init(1, key);
26             byte[] ciphertext = cipher.doFinal(plaintext);
27             System.out.println("IV: " + Base64.encodeToString(cipher.getIV(), 0));
28             System.out.println("Return Value: " + Base64.encodeToString(ciphertext, 0));
29             return ciphertext;
30         } catch (InvalidKeyException e) {
31             e.printStackTrace();
32             return null;
33         } catch (NoSuchAlgorithmException e2) {
34             e2.printStackTrace();
35             return null;
36         } catch (BadPaddingException e3) {
37             e3.printStackTrace();
38             return null;
39         } catch (IllegalBlockSizeException e4) {
40             e4.printStackTrace();
41             return null;
42         } catch (NoSuchPaddingException e5) {
43             e5.printStackTrace();
44             return null;
45         }
46     }
47 }
```

Smali code

```

103    invoke-virtual {v4}, Ljava/lang/StringBuilder; ->toString()Ljava/lang/String;
104    move-result-object v4
105
106    invoke-virtual {v3, v4}, Ljava/io/PrintStream; ->println(Ljava/lang/String; )V
107
108    .line 22
109    .local v1, "key":Ljavax/crypto/SecretKey;
110    const-string v2, "AES/CBC/PKCS5PADDING"
111
112    invoke-static {v2}, Ljavax/crypto/Cipher; ->getInstance(Ljava/lang/String; )Ljavax/crypto/Cipher;
113
114    move-result-object v2
115
116    .line 23
117    .local v2, "cipher":Ljavax/crypto/Cipher;
118    const/4 v3, 0x1
119
120    invoke-virtual {v2, v3, v1}, Ljavax/crypto/Cipher; ->init(ILjava/security/Key; )V
121
122    .line 24
123    invoke-virtual {v2, p0}, Ljavax/crypto/Cipher; ->doFinal([B][B
124
125    move-result-object v3
126
127    .line 25
128    .local v3, "ciphertext":[B
129    invoke-virtual {v2}, Ljavax/crypto/Cipher; ->getIV()[B
130
131    move-result-object v2
132
133    sget-object v6, Ljava/lang/System; ->out:Ljava/io/PrintStream;
134
135    new-instance v4, Ljava/lang/StringBuilder;
136
137    invoke-direct {v4}, Ljava/lang/StringBuilder; -><init>()V
138
139    const-string v5, "IV: "
140
141    invoke-virtual {v4, v5}, Ljava/lang/StringBuilder; ->append(Ljava/lang/String; )Ljava/lang/StringBuilder;
142
143    move-result-object v4
144
145    const/4 v5, 0x0
146
147    invoke-static {v2, v5}, Landroid/util/Base64; ->encodeToString([BI)Ljava/lang/String;
148
149    move-result-object v5
150
151    invoke-virtual {v4, v5}, Ljava/lang/StringBuilder; ->append(Ljava/lang/String; )Ljava/lang/StringBuilder;
152
153    move-result-object v4
154
155    invoke-virtual {v4}, Ljava/lang/StringBuilder; ->toString()Ljava/lang/String;
156
157

```

the result we got IV and Secret key

```

→ Downloads adb logcat | grep -i system.out
→ Downloads adb logcat | grep -i system.out
04-24 10:12:42.901  4702  4702 I System.out: Encrypted Text
04-24 10:12:42.902  4702  4702 I System.out: Param: UzNjcjN0MG4zVDFTM00zU1NhZzM=
04-24 10:12:42.903  4702  4702 I System.out: Key: 9o/yopWyPzAh9pU6lg81rCF7C4+8b000I8QeyqypsKY=
04-24 10:12:42.910  4702  4702 I System.out: IV: dy99jpdssNYFA76mXAGvXg==
04-24 10:12:42.910  4702  4702 I System.out: Return Value: bLYZ0Hp6EraYJqczn7lDyahlChUV8DI8QL/P+FJpINc=

```

take those and go to Cipher Chef

Recipe

From Base64

Alphabet
A-Za-z0-9+=

Remove non-alphabet chars Strict mode

AES Decrypt

Key
9o/yopW... BASE64 ▾

IV
dy99jpdss... BASE64 ▾

Mode
CBC

Input
Raw

Output
Raw

Input

bLYZ0Hp6EraYJqczn7lDyahIChUV8DI8QL/P+FJpINC=

Output

S3cr3t0n3T1m3M3SSag3

