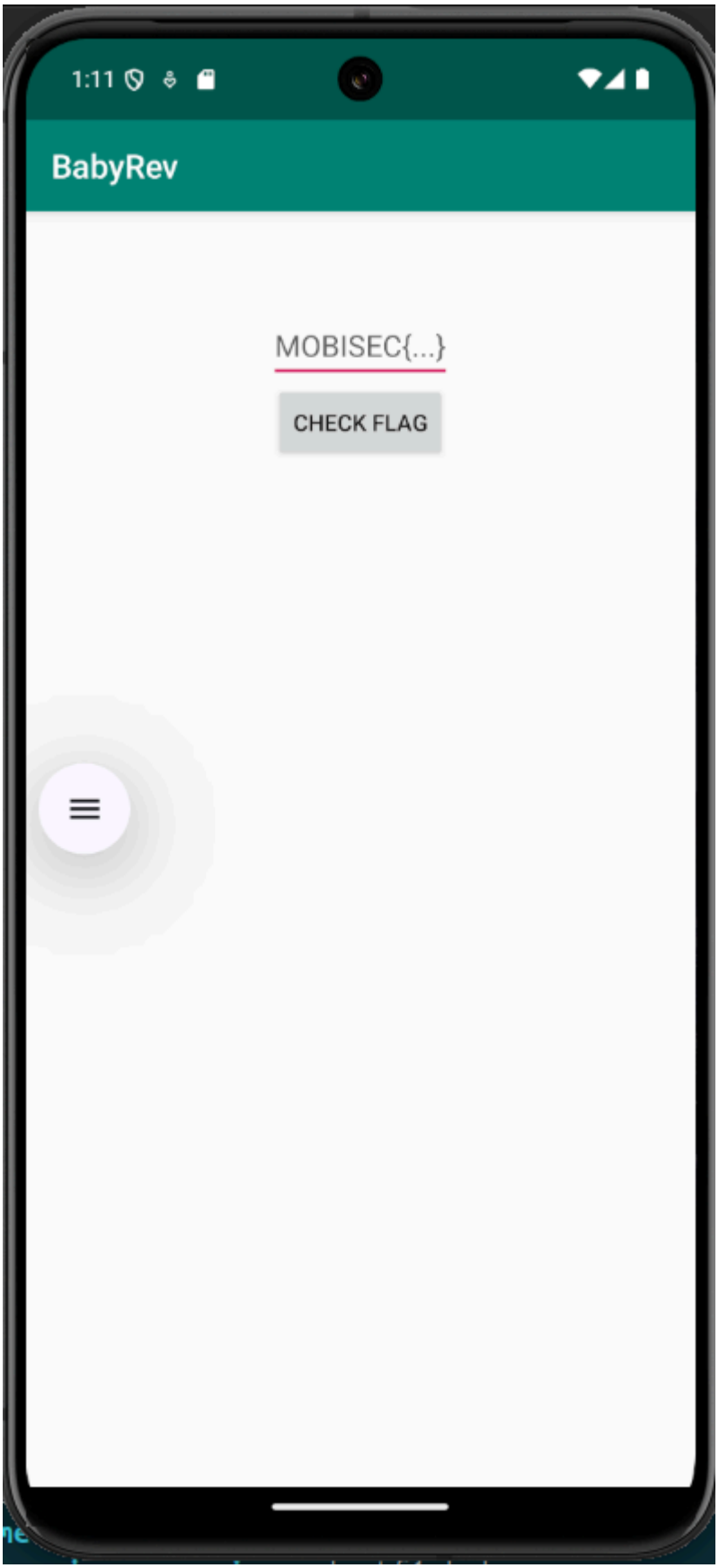
1-Babyrev - reverse engineering

1- reversing : babyrev

هنا ده اول تحدي وهنا هنحلل الكود وممكن يكون طويل شوية ودلوقتي هنحاول ننزل app ونفتحه ونشوفه

install app with adb

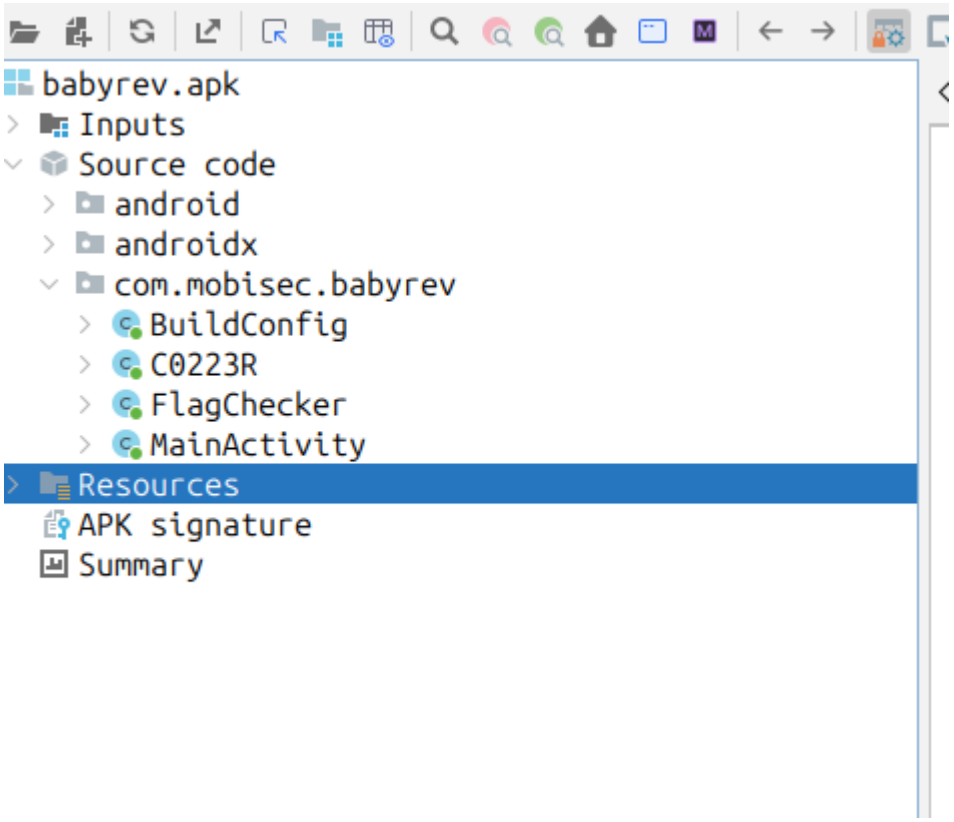
```
➜  ctf1_babyyrev adb install babyrev.apk
Performing Streamed Install
Success
```



هنا اه عاوزني ادخل flag ودلوقتي هنروح نشوف code for app هنستحه باستخدام jadx-gui ودي tool بنستخدمها لو عاوزين نعمل analyze ل apk app من غير ما نعمل له decompile

2- open app with jadx-gui

```
jadx-gui babyrev.apk
```



هنا اه التطبيق في كذا ملف بس دلوقتي هنروح نفتح الملف الخاص ب FlagChecker

3- open FlagChecker file

```
package com.mobisec.babyrev;

import android.content.Context;

/* loaded from: classes.dex */
public class FlagChecker {
    public static boolean checkFlag(Context ctx, String flag) {
        if (!flag.startsWith("MOBISEC{") || new
StringBuilder(flag).reverse().toString().charAt(0) != '}' || flag.length()
!= 35 || !flag.toLowerCase().substring(8).startsWith("this_is_") || !new
StringBuilder(flag).reverse().toString().toLowerCase().substring(1).startsWi
th(ctx.getString(C0223R.string.last_part)) || flag.charAt(17) != '_' ||
flag.charAt((int) (getY() * Math.pow(getX(), getY()))) != flag.charAt(((int)
Math.pow(Math.pow(2.0d, 2.0d), 2.0d)) + 1) ||
!bam(flag.toUpperCase().substring(getY() * getX() * getY(), (int)
(Math.pow(getZ(), getX()) - 1.0d))).equals("ERNYYL") ||
flag.toLowerCase().charAt(16) != 'a' || flag.charAt(16) != flag.charAt(26)
|| flag.toUpperCase().charAt(25) != flag.toUpperCase().charAt(26) + 1) {
            return false;
        }
        String r = getR();
        return flag.substring(8, flag.length() - 1).matches(r);
```

```
    }

    private static int getX() {
        return 2;
    }

    private static int getY() {
        return 3;
    }

    private static int getZ() {
        return 5;
    }

    private static String bam(String s) {
        String out = BuildConfig.FLAVOR;
        for (int i = 0; i < s.length(); i++) {
            char c = s.charAt(i);
            if (c >= 'a' && c <= 'm') {
                c = (char) (c + '\r');
            } else if (c >= 'A' && c <= 'M') {
                c = (char) (c + '\r');
            } else if (c >= 'n' && c <= 'z') {
                c = (char) (c - '\r');
            } else if (c >= 'N' && c <= 'Z') {
                c = (char) (c - '\r');
            }
            out = out + c;
        }
        return out;
    }

    public static String getR() {
        String r = BuildConfig.FLAVOR;
        boolean upper = true;
        for (int i = 0; i < 26; i++) {
            r = upper ? r + "[A-Z_]" : r + "[a-z_]";
            upper = !upper;
        }
        return r;
    }
}
```

دلوقتي ده الكود اللي من خلاله بنجيب قيمة FLAG ودلوقتي هنرو نحلله جزء جزء بس الاول خليني نتكلم عن function اللي فيه او اللي هنستخدمها نروح نحللها الاول

1- analyze bam function :

here --> bam function take string

- **char from a --> m and A --> M : add 13 char --> char += 13**
- **char from n --> z and N --> Z : sub 13 char --> char -= 13**

and return string after convert every char

```
for (int i = 0; i < s.length(); i++) {
            char c = s.charAt(i);
            if (c >= 'a' && c <= 'm') {
                c = (char) (c + '\r');
            } else if (c >= 'A' && c <= 'M') {
                c = (char) (c + '\r');
            } else if (c >= 'n' && c <= 'z') {
                c = (char) (c - '\r');
            } else if (c >= 'N' && c <= 'Z') {
                c = (char) (c - '\r');
            }
            out = out + c;
        }
        return out;
    }
```

دلوقتي بقي هنحلل condition اللي من خلاله بنجيب Flag

```
 if (!flag.startsWith("MOBISEC{") || new
StringBuilder(flag).reverse().toString().charAt(0) != '}' || flag.length()
!= 35 || !flag.toLowerCase().substring(8).startsWith("this_is_") || !new
StringBuilder(flag).reverse().toString().toLowerCase().substring(1).startsWi
th(ctx.getString(C0223R.string.last_part)) || flag.charAt(17) != '_' ||
flag.charAt((int) (getY() * Math.pow(getX(), getY()))) != flag.charAt(((int)
Math.pow(Math.pow(2.0d, 2.0d), 2.0d)) + 1) ||
!bam(flag.toUpperCase().substring(getY() * getX() * getY(), (int)
(Math.pow(getZ(), getX()) - 1.0d))).equals("ERNYYL") ||
flag.toLowerCase().charAt(16) != 'a' || flag.charAt(16) != flag.charAt(26)
|| flag.toUpperCase().charAt(25) != flag.toUpperCase().charAt(26) + 1) {
            return false;
        }
```

1- flag start with ---> MOBISEC{ : index from 0 -> 7

```
flag.startsWith("MOBISEC{")
```

2- the last char on the flag ---> } :index 34

```
 new StringBuilder(flag).reverse().toString().charAt(0) != '}'
```

3- the length of flag = 35

```
flag.length() != 35
```

4- flag from index 8 start with --> this_is_ : index from 8 -> 15

```
!flag.toLowerCase().substring(8).startsWith("this_is_")
```

5- ctx.getString(C0223R.string.last_part)

هنا العنصر اللي هو C0223R بيشاور علي عناصر تانية داخل التطبيق وعلشان نشوفه لازم نروح نعمل decompile

decompile with apktool

```
apktool d babyrev.apk
```

بيكون values اللي بيشاور عليها موجودة في babyrev/res/values دلوقتي بقي هنجي القيمة اللي بيقول عليها وهي ver_cis

```
➜  values cat strings.xml  |grep last_part
    <string name="last_part">ver_cis</string>
```

دلوقتي بقي بيقول ان القيمة اللي هيجبها هيعكسها يعني هتبقي sic_rev وهتبقي بعد { يعني هتبقي في اخر string قبل } ---> sic_rev

sic_rev} from index 27 -> 34

```
StringBuilder(flag).reverse().toString().toLowerCase().substring(1).startsWith(ctx.getString(C0223R.string.last_part))
```

now we have the string from index :

- MOBISEC{ : index from 0 -> 7
- this_is_ : index from 8 -> 15
- sic_rev} from index 27 -> 34

6- index 17 on flag = _

```
flag.charAt(17) != '_'
```

7- on file we have three funxtion : 1- getX() =2 , 2- getY=3 , getZ()=5

here : if we calc these operation we will find ---> index 24 = index 17

index 24 --> _

```
flag.charAt((int) (getY() * Math.pow(getX(), getY()))) !=
flag.charAt(((int) Math.pow(Math.pow(2.0d, 2.0d), 2.0d)) + 1)
```

8-here if we calc the se operation we will find index from 18 --> 23 == ERNYYL , but using bam function, so we will convert it

after convert --> index from 18 --> 23 == REALLY

```
!bam(flag.toUpperCase().substring(getY() * getX() * getY(), (int)
(Math.pow(getZ(), getX()) - 1.0d))).equals("ERNYYL")
```

9- index 16 = a

```
flag.toLowerCase().charAt(16) != 'a'
```

10 - index 16 == index 26 , so index 26 will equal a --> 26=a

```
flag.charAt(16) != flag.charAt(26)
```

11- index number 25 = index number 26 --> a ,but he add a+1 --> b

so index will equal --> 25 =b

```
flag.toUpperCase().charAt(25) != flag.toUpperCase().charAt(26) + 1)
```

finally after collect all index the final flag will be

MOBISEC{ThIs_iS_A_ReAlLy_bAsIc_rEv}