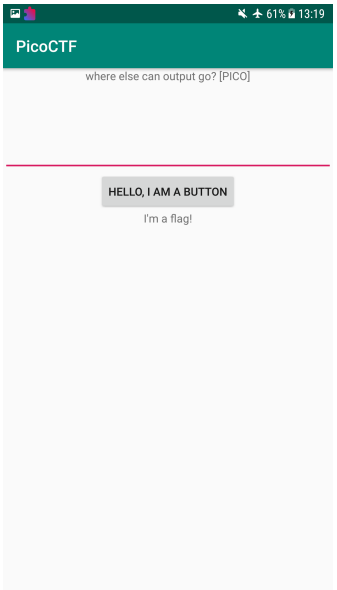
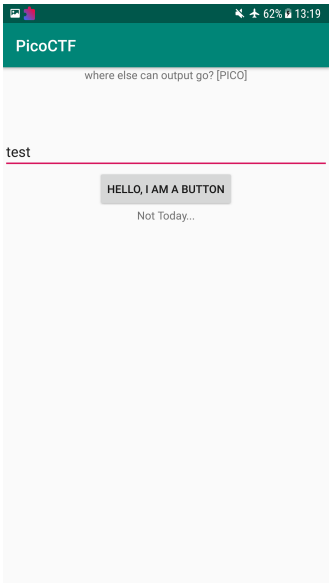
Mark Seliternikov
picoCTF - droids0 [300 points]

לאתגר זה מקבלים קובץ APK. כאשר מריצים אותו במכשיר אנדרואיד ככה נראה הפלט



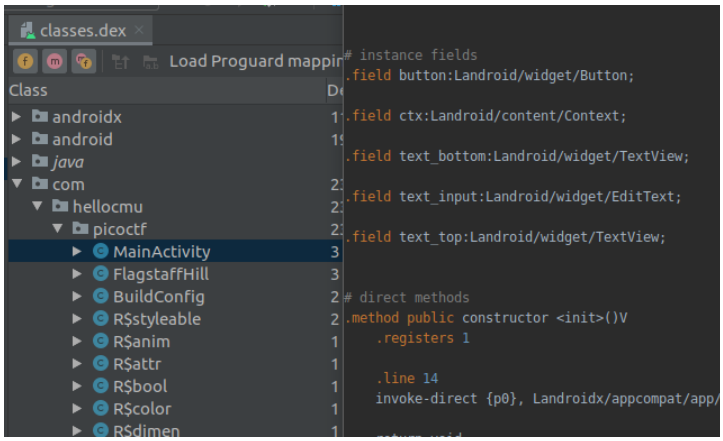
ככה נראה הפלט כאשר לחצתי על הכפתור



לאחר מכן עשיתי Extract לקובץ והתחלתי לחפש רמזים לדגל וזה מה שמצאתי

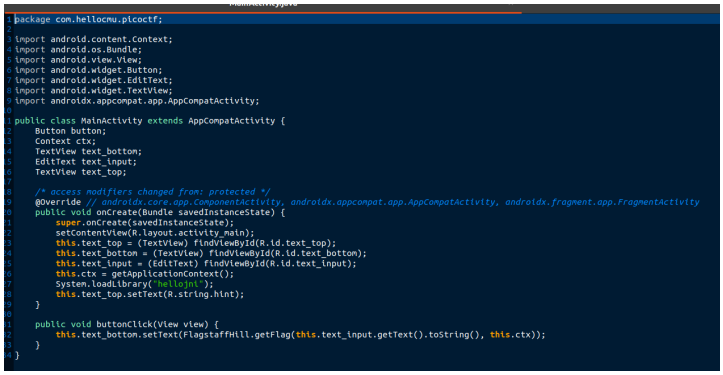
```
mark@workstation:~/Downloads/zero$ strings classes.dex | grep pico
"Lcom/hellocmu/picoctf/BuildConfig;
$Lcom/hellocmu/picoctf/FlagstaffHill;
#Lcom/hellocmu/picoctf/MainActivity;
Lcom/hellocmu/picoctf/R$anim;
Lcom/hellocmu/picoctf/R$attr;
Lcom/hellocmu/picoctf/R$bool;
Lcom/hellocmu/picoctf/R$color;
Lcom/hellocmu/picoctf/R$dimen;
!Lcom/hellocmu/picoctf/R$drawable;
Lcom/hellocmu/picoctf/R$id;
 Lcom/hellocmu/picoctf/R$integer;
Lcom/hellocmu/picoctf/R$layout;
Lcom/hellocmu/picoctf/R$mipmap;
Lcom/hellocmu/picoctf/R$string;
Lcom/hellocmu/picoctf/R$style;
"Lcom/hellocmu/picoctf/R$styleable;
Lcom/hellocmu/picoctf/R;
com.hellocmu.picoctf
mark@workstation:~/Downloads/zero$
```

לאחר מכן פתחתי את הקובץ ב-Android Studio כמו שהומלץ ב-Hints. חיפשתי את ה-Main.
כדי להבין את ההרכב של הקבצים למדתי קצת על אנדרואיד כדי להבין איך לגשת.
למדתי ש-APK זה בעצם קובץ מקומפל של אנרואיד שה-source code שלו הוא Java.
ה-Bytecode שראיתי בעצם זה בשפת smali.



הבנתי שיש לי 2 דרכים לפרש את איך שהקובץ עובד.
1) להבין איך הקובץ עובד דרך smali
2) להבין איך הקובץ עובד דרך Java
בחרתי ב-Java כי זה יותר ידידותי למשתמש, לכן כדי לעשות לזה דיקומפייל השתמשתי בתוכנה שנקראת Jadx שמצאתי ב-github.

התמקדתי ב-2 קבצי Java עיקריים



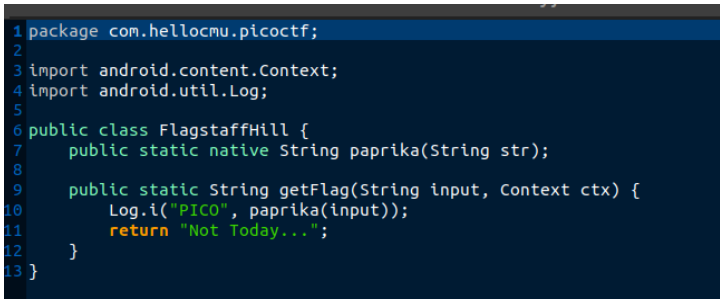
```
package com.hellocmu.picoctf;

import android.content.Context;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    Button button;
    Context ctx;
    TextView text_bottom;
    EditText text_input;
    TextView text_top;

    /* access modifiers changed from: protected */
    @Override // androidx.core.app.ComponentActivity, androidx.appcompat.app.AppCompatActivity, androidx.fragment.app.FragmentActivity
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        this.text_top = (TextView) findViewById(R.id.text_top);
        this.text_bottom = (TextView) findViewById(R.id.text_bottom);
        this.text_input = (EditText) findViewById(R.id.text_input);
        this.ctx = getApplicationContext();
        System.loadLibrary("hellojni");
        this.text_top.setText(R.string.hint);
    }

    public void buttonClick(View view) {
        this.text_bottom.setText(FlagstaffHill.getFlag(this.text_input.getText().toString(), this.ctx));
    }
}
```



```
package com.hellocmu.picoctf;

import android.content.Context;
import android.util.Log;

public class FlagstaffHill {
    public static native String paprika(String str);

    public static String getFlag(String input, Context ctx) {
        Log.i("PICO", paprika(input));
        return "Not Today...";
    }
}
```

ניתן לראות ב-MainActivity שיש קריאה לפונק' שנמצאת ב-FkagstaffHill, נשלחים לשם 2 פרמטרים (ה-input ו-ctx)

```
public void buttonClick(View view) {
    this.text_bottom.setText(FlagstaffHill.getFlag(this.text_input.getText().toString(), this.ctx));
}
```

לאחר בדיקה ראיתי שכל מה שהאפליקציה עושה בצד לקוח זה להחזיר סטרינג "Not Today..." בלי קשר למה התוכן של ה-input.

```
return "Not Today...";
```

אבל שמתי לב שיש קריאה ל-Log. באותו שלב לא ידעתי מהם Logs ומה השימוש שלהם. אז הדבר שנותר לי לעשות זה לחפש באינטרנט וללמוד.

```
Log.i("PICO", paprika(input));
```

(הייתה לי תחושה שזה לא נמצא שם סתם)

לאחר חיפוש מצאתי שיש דרך לראות Logs בעזרת תוכנה שנקראת adb אשר משמשת לדבג אנדרואיד.
השתמשתי בפונק' הנקראת logcat כדי לראות את ה-Logs.

שמתי לב כאשר לחצתי על הכפתור קפץ log של הדגל. ואז נעלם בשאר ה-logs.
כדי להקל על עצמי השתמשתי שוב ב-logcat אבל עם grep ומצאתי את הדגל :)

```
mark@XPS-13-9360:~/Tools/ghidra/ghidra_9.2.2_PUBLIC$ adb logcat | grep picoCTF
03-14 [redacted] 7511 7511 I PICO : picoCTF{a.moose.once.bit.my.sister}
03-14 [redacted] 7511 7511 I PICO : picoCTF{a.moose.once.bit.my.sister}
03-14 [redacted] 7511 7511 I PICO : picoCTF{a.moose.once.bit.my.sister}
```