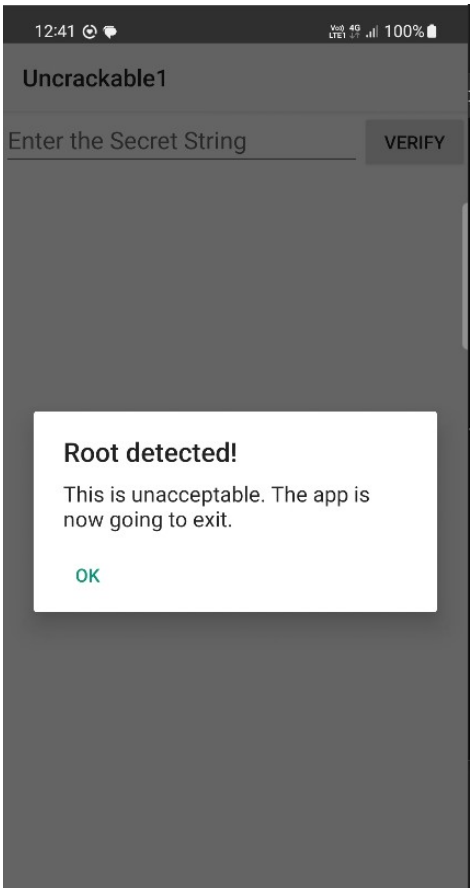
תרגיל Uncrackable1

בתרגיל זה ניישם טכניקות של ניתוח סטטי ודינמי כדי לפתור את האתגר Uncrackable1.apk.

מטרה: עליכם להבין מה האפליקציה עושה ולמצוא את הקוד הסודי.

פתרון:

1. ניתוח התנהגות: הריצו את האפליקציה:



2. קודם כל צריך לנטרל את בדיקת ה root. אפשר לדלג על שלב זה ולחפש דרך להגיע לסוד ישירות, אבל מומלץ לתרגל תרחיש זה.

2.1 ניטרול בדיקת root:

- נבצע ניתוח סטטי כדי למצוא את הבדיקה.
  - פתחו את ה apk ב jadx-gui
  - נגלה ב mainfest שיש activity אחד:

```
android:name="sg.vantagepoint.uncrackable1.MainActivity'
```

- הקוד הרלוונטי נמצא כאן:

```
protected void onCreate(Bundle bundle) {
    if (c.a() || c.b() || c.c()) {
        a("Root detected!");
    }
```

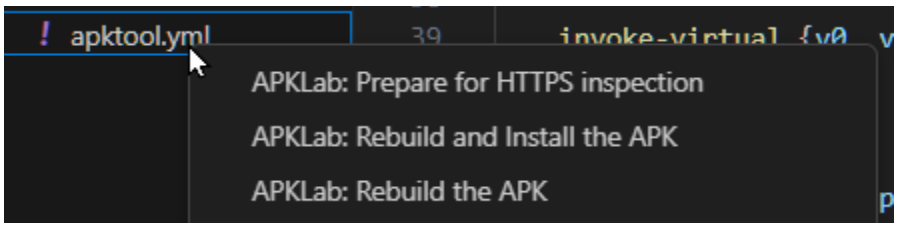
- צריך לשנות את התנאי כך שיחזיר False או לבטל אותו.
- אפשר לעשות patching ל apk או לשנות דינמית עם פרידה.

- **אפשרות 1**: patching:

  - פתחו את ה apk עם apklab. כל הקוד smali להלן מבצע את הבדיקות, אפשר להסיר אותו (שורות 55 עד 93 כולל):

```
.method protected onCreate(Landroid/os/Bundle;)V
    .locals 1

    invoke-static {}, Lsg/vantagepoint/a/c;->a()Z

    move-result v0

    if-nez v0, :cond_0

    invoke-static {}, Lsg/vantagepoint/a/c;->b()Z

    move-result v0

    if-nez v0, :cond_0

    invoke-static {}, Lsg/vantagepoint/a/c;->c()Z

    move-result v0

    if-eqz v0, :cond_1

    :cond_0
    const-string v0, "Root detected!"

    invoke-direct {p0, v0}, Lsg/vantagepoint/uncrackable1/MainActivity;->a(Ljava/lang/String;)V

    :cond_1
    invoke-virtual {p0}, Lsg/vantagepoint/uncrackable1/MainActivity;->getApplicationContext()Landroid

    move-result-object v0

    invoke-static {v0}, Lsg/vantagepoint/a/b;->a(Landroid/content/Context;)Z

    move-result v0

    if-eqz v0, :cond_2

    const-string v0, "App is debuggable!"

    invoke-direct {p0, v0}, Lsg/vantagepoint/uncrackable1/MainActivity;->a(Ljava/lang/String;)V
```

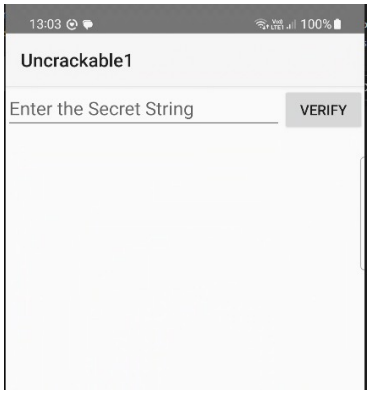
  - בנו מחדש את ה apk:



- הריצו מחדש את ה apk שבניתם:

```
\UnCrackable-Level11\dist\UnCrackable-Level1.apk
```

- בדיקת ה root בוטלה:



o **אפשרות 2**: frida

- וודאו ש frida-server רץ על המכשיר:

```
a53x:/data/local/tmp # ./frida-server-arm64
```

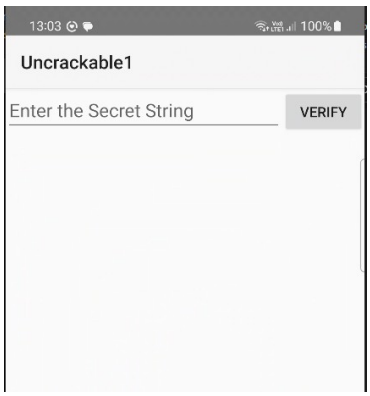
- כתבו את הסקריט הבא בתוך uncrackable1.js:

```
// root detection

Java.perform(function() {
  Java.use("sg.vantagepoint.uncrackable1.MainActivity").a.implementation = function(s) {
    console.log("Tamper detection suppressed, message was: " + s);
  }
});
```

- הריצו את פרידה עם הסקריפט:

```
frida -U -f owasp.mstg.uncrackable1 -l uncrackable1.js
```

- וודאו שהאפליקציה נפתחת בלי בדיקת root:



2.2 שליפת הסוד:

- הריצו trace על הפונקציה a שמחזירה את הסוד מפוענח מהצפנה:
- frida-trace -U -j "*a!a" Uncrackable1

```
C:\Users\tal>frida-trace -U -j "*a!a" Uncrackable1
Instrumenting...
a.a: Auto-generated handler at "C:\\Users\\tal\\__handlers__\\sg.vantagepoint.uncrackable1.a\\a.js"
a.a: Auto-generated handler at "C:\\Users\\tal\\__handlers__\\sg.vantagepoint.a.a\\a.js"
Started tracing 2 functions. Press Ctrl+C to stop.
           /* TID 0x1287 */
  2457 ms  a.a("122")
  2458 ms     | a.a([-115,18,118,-124,-53,-61,124,23,97,109,-128,108,-11,4,115,-52], [-27,66,98,21,-5
3,91,-102,6,-61,-96,-75,-26,-92,-67,118,-102,73,-24,-16,116,-8,46,-1,29,-107,-85,124,23,20,118,24,-25
])
  2461 ms     | <= [73,32,119,97,110,116,32,116,111,32,98,101,108,105,101,118,101]
  2463 ms  <= false
```

- תרגמו את ה ascii לתווים וקבלו את הסוד:

```
C:\Users\tal>python -c "print(bytes([73,32,119,97,110,116,32,116,111,32,98,101,108,105,101,118,101]))"
b'I want to believe'
```