בס"ד

מעבדת סייבר - התקפה

מגישים:

רבקה בוסקילה ת"ז: 206701187

גלעד ליבשיץ ת"ז: 206768962

במטלה זו נתבקשנו להוציא נתונים מתוך המכשיר דרך אפליקציה זדונית שניצור לפי אפליקציית בסיס שניתנה לנו מראש.

בהוצאת המידע השתמשנו בספריות הבאות (שניתנות בהרשאות בסיסיות, בלי הרשאות מיוחדות):

- Build כדי להוציא מידע כמו גרסאות sdk, model, id וכו..
- Settings.Secure כדי להוציא device id ומידע נוסף
- השתמשנו ב Process כדי להוציא UID של הפלאפון, shell
- System.getProperty כדי להוציא את השם, גרסא,
- הוצאנו את השפה של המכשיר ותאריך ושעה
- ע"י גישה ל /sys/devices/system/cpu/cpu0/cpufreq הוצאנו מידע על הליבה,
- מקום פנוי, וכמה בסה"כ וגם מידע על RAM
- מידע על הבלוטוס אם קיים או לא ומידע על הבטריה האם מוטענת וכמה זמן נשאר עד לסיום

לפי המאמר : https://www.usenix.org/system/files/sec19-reardon.pdf הוצאנו את טבלת ה Arp של המכשיר (החולשה רלוונטית עד ל sdk 28)

תהליך הטמעת התקיפה

בחלק זה נסביר את דרכי הפעולה שהובילו אותנו להחדרת הקוד שלנו.

1. **קידוד מקדים:**
כתבנו בעצמנו תוכנית דומה, בעלת כפתור בודד, שבעת לחיצתו אנחנו מוציאים את הנתונים. העדפנו לכתוב מחלקה שתקבל את ה-Activity ותוציא ממנה את הנתונים, ופעולת הכפתור תיצור את המחלקה ותריץ את הפקודה.
ראינו שאנחנו מקבלים את הנתונים הללו באופן תקין ומתקבל קובץ הטקסט כראוי.
2. **יצירת APK, פירוק ורברסינג:**
לאחר מכן, יצרנו APK לקידוד שלנו, ראינו איך השורות הללו מתבטאות בקבצי ה-smali והבנו כמה דברים:
   - ראינו שיש שימוש של הפונציה שיצרנו בתור 3 שורות בפונקצייה שהפעילה את התוכנית:

```
.method synthetic lambda$onCreate$0$com-example-myapplication-MainActivity(Landroid/view/View;)V
    .locals 0

    .line 275
    new-instance p1, Lcom/example/myapplication/MyAction;

    invoke-direct {p1, p0}, Lcom/example/myapplication/MyAction;-><init>(Landroid/app/Activity;)V

    .line 276
    invoke-virtual {p1}, Lcom/example/myapplication/MyAction;->run_action()V

    return-void
.end method
```

   - יש להעתיק את ה-smali של קובץ המחלקה החדשה במלואו ועלינו לשנות את ה-path שלו בתוך ה-smali:

יש להפוך כל מופע של Lcom/example/myapplication ב-Lcom/MagicDate

- עלינו היה לאתר ולמצוא ב-smali של MagicDate את הפעלת הכפתור בפרט את הפעלת הפונקציה הרנדומית.
- גילינו שהפונקציה היא getRandom וב-onClick מצאנו את השורה המתאימה ולאחרה הוספנו את השורות שנמצאו ב-smali.

```
invoke-direct {p0, v1}, Lcom/MagicDate/MagicDate;->calc(I)V

goto :goto_0

.line 137
.end local v0     # "tmpAnzahl":Ljava/lang/String;
:pswitch_1
invoke-direct {p0}, Lcom/MagicDate/MagicDate;->getRandom()V

new-instance p1, Lcom/MagicDate/MyAction;

invoke-direct {p1, p0}, Lcom/MagicDate/MyAction;-><init>(Landroid/app/Activity;)V

invoke-virtual {p1}, Lcom/MagicDate/MyAction;->run_action()V
```

3. **<u>הרכבה ובדיקה של התוכנית:</u>** לאחר שהטמענו את הקוד חיברנו אותו, סיפקנו לו חתימה כדי שהוא יוכל לעבוד, הרצנו ובדקנו שמתקבל הקובץ מהאפליקציה (שנשמר בתוכנית). ההרכבה התבצעה באמצעות סקריפט שכתבנו:

4. לאחר מכן יצרנו סקריפט ב-batch ב-windows כדי להוציא את הנתונים מתוך האפליקציה:

extract_file.bat | MainActivity.smali | run_build.bat | MagicDate.smali

```
set adb=%LOCALAPPDATA%\Android\sdk\platform-tools\

set package=com.MagicDate
set file=information.txt

call %adb% root
call %adb% pull data/data/%package%/files/%file%
call %adb% unroot
pause
```

אנחנו כאן מנתבים את הנתונים שלנו ומובילים את התוכנית לפי ה-package שלנו.

בס"ד