מטלת סוף מעבדת התקפה

רעיון ההתקפה הוא להוציא מידע מהנתקף בשביל סקירה של נתוני הטלפון והנקף לצורך קידום התקפות נוספות בעזרת בקשת כמה שפחות הרשאות מהמשתמש (READ_SMS).

הקוד, תחת הפונקציית ()hack, מוציא מן המשתמש את נתוני הטלפון שלו ומערכת ההפעלה של הטלפון. הנתונים כוללים מספר טלפון, שם המשתמש, סוג הטלפון, גרסאת מערכת הפעלה ועוד.

בנוסף לכך הקוד מציג את כל האפליקציות (שניתן לגשת אליהם ללא בקשת הרשאות) שמותקנות על מכשיר הנתקף, כגון whatsapp.

בהחלטתי להוציא את מספר הטלפון של הנתקף, בהיותו חשוב להמשיך את ההתקפה בהנחה שלתוקף אין ידע מי הוריד את האפליקציה שלו, נדרשתי לבקשת הרשאות גישה נוספות. מתוך ההרשאות שמאפשרות גישה למספר טלפון קריאת SMS הייתה הכי שימושית. לכן בנוסף למספר הטלפון האפליקציה מאפשרת להוציא רשימה של כל שיחות הSMS של הנתקף. רשימה זאת מאפשר לקבל מספרי טלפון של אנשי הקשר של הנתקף והנושאים שקרובים אליו. מכך ניתן להשתמש במידע זה על מנת לקדם התקפות phishing על הנתקף בקלות רבה.

כל המידע נכתב לתוך קובץ information.txt בתוך תיקיית המידע של האפליקציה.

אופן הבנייה של האפליקציה היה כתיבת הקוד הזדוני ALF, בדיקתו ומימושו כחבילת APK. לאחר מכך השתמשתי באפליקציה magicDate ככלי להחדרת הקוד בעזרת apktool.

החדרת הקוד לקבצי האפליקציה כלל כמה שלבים:

עדכון קובץ הmanifest לדרישת שימוש בהרשאת קריאת SMS.

החדרת ספריות הקוד לאפליקציה: בגלל שmagicDate נכתב בjava וALF בkotlin קיימים ספריות חסרות באפליקציה לכן העתקתי את קבצי המחלקות והנתונים של ALF לתוך magicDate.

החדרת הקוד: תחילה העתקתי את קוד הsmali של ()hack אל תוך המחלקה הראשית של magicDate בתור פונקציה פנימית של המחלקה (direct) ועדכנתי את הקריאות המחלקה הישנות של ALF לmagicDate. לאחר מכן איתרתי את פונקציית onCreate של magicDate והכנסתי לשם פונקציית בקשת הרשאות של גישה לSMS.

לבסוף אחרי חקירה של קובץ הlayout והR.id של magicDate איתרתי את מיקום, מבנה ושמו של כפתור הrandomize בנוסף לפעולתו בעת לחיצה. בפונקציית onClick של

הlayout שמכיל את הכפתורים החדרתי קריאה לפונקציית ()hack מתחת לפונקציית ()getRandom הרגילה של הכפתור.

לבסוף הצלחתי להתקין ולהריץ את האפליקציה באמולטורים של גרסאותת sdk 24 ו30 כאשר ההתקפה ויצרית קובץ המידע information.txt מתבצעים רק בלחיצת כפתור randomize.