פרויקט גמר – מעבדת סייבר התקפה

מגישים:

שם: אילון נעמת.

ת.ז: 315303529

שם: מיכאל מטבייב.

ת.ז: 315918557

בפרויקט הגמר התבקשנו לקבל אפליקציה תקינה ולהחדיר אליה תוכן זדוני אשר גונב מידע מהמשתמש/מכשיר אשר האפליקציה רצה על גביה.

תחילה, הורדנו את האפליקציה התקינה ובחנו אותה, תוך הסתכלות באימולטור ובהסתכלות בקבצי smali שלה. דבר ראשון ששמנו לב אליו הוא שישנה פונקציה שנקראת onclick אשר נקראית כאשר לוחצים על כפתור random באפליקציה, ולכן הבנו ששמה אנו צריכים לשתול את הקוד הזדוני שלנו.

מכיוון שכתיבה בsmali אינה אידיאלית החלטנו לייצר אפליקציה משלנו ובה יהיה תוכן הקוד הזדוני אשר אותו נייצא לקובץ apk. מקובץ הapk אנו נחלץ את קבצי smali ומהם ניקח את הקוד הזדוני שלנו וננסה להשתיל אותם בקובץ smali של האפליקציה התקינה.

את קבצי smali של האפליקציה התקינה ושל התוכן הזדוני שלנו הוצאנו על ידי פקודה – apktool d magicDate.apk וגם apktool d app-debug.apk .

הקוד הזדוני שלנו נמצא בפונקציה אחת הנקראת write_file לכן ידענו למצוא אותה בקבצים שלנו ולראות כיצד היא נקראת מפונקציה אחרת. לאחר שהבנו כיצד הפונקציה נקראית בsmali וגם כיצד היא נקראית, עברנו לsmali של האפליקציה התקינה ובפונקציית הonclick (שמופעלת על ידי לחיצה על כפתור random) הוספנו שורה שקוראת לפונקציה שלנו. בנוסף לכך הוספנו את הפונקציה הזדונית שלנו בvirtual methods של קובץ smali והתאמנו את הפונקציה ומשתניה לפורמט המתאים של magicDate.

לאחר שהוספנו את הקוד הזדוני שלנו לאפליקציה התקינה, הוספנו את ההרשאות הנדרשות בקובץ manifest, ולאחר מכן עשינו repacking על ידי הפקודה – apktool b magicDate. בנוסף הוספנו את האפשרות לבצע debug על האפליקציה על ידי הוספת השורה "android:debuggable="true בapplication (על מנת שיהיה יותר נוח לעבוד עם קבצי האפליקציה בandroid studio).

לאחר שביצענו את הפקודה שלעיל, נוצרה לנו תיקייה חדשה בשם dist אשר בה נמצא קובץ הapk החדש בעל התוכן הזדוני. על מנת לחתום את הקובץ תחילה יצרנו מפתח אשר איתו נחתום את הקובץ על ידי הפקודה - keytool -genkey -v -keystore debug.keystore -storepass android -alias androiddebugkey -keypass android -keyalg RSA -keysize 2048 -validity 10000 (יש לציין כי תחילה יצרנו alias למפתח debug.keystore בשם androiddebugkey).

לאחר שיצרנו את המפתח בתוך תיקיית dist עם קובץ הapk החדש הרצנו את הפקודה אשר חותמת את הקובץ - jarsigner -verbose -sigalg SHA256withRSA -digestalg SHA-256 -keystore debug.keystore magicDate.apk androiddebugkey -storepass android -keypass android.

ניתן לוודא כי אכן חתמנו את הקובץ באמצעות המפתח על ידיד הפקודה - jarsigner -verify -verbose -certs magicDate.apk.

ולאחר שחתמנו את הקובץ ביצענו zipalign על ידי הפקודה - zipalign -v 4 magicDate.apk 315303529_315918557.apk. (בסרטון ההסבר קראנו לקובץ החדש בשם outline.apk במקום 315303529_315918557.apk).

לאחר שהקובץ החדש נוצר התקנו אותו על האפליקציה על ידי גרירה (מחקנו את האפליקציה הקודמת), ולחצנו על כפתור הrandom אשר מריץ את הסקריפט הזדוני שלנו. לאחר מכן בדקנו כי אכן הקובץ information.txt נוצר ובו כל המידע אשר רצינו לגנוב (בתוך הDevice Files Explorer).

תוכן הפונקציה

הפונקציה אשר גונבת את המידע מהמשתמש ומהמכשיר ושומרת אותו בinformation.txt נקראת write_file.

כמות המידע אשר הצלחנו לגנוב מהמשתמש/מהמכשיר הוא כ50 פרטי מידע אשר קיטלגנו אותם לפי הקטגוריות הבאות:

המידע אשר גנבנו הוא מידע על המערכת (System Information), מידע על הרשת (Network Information), רשימת האפליקציות במכשיר (Application Information), מידע על אחסון (Storage Information), מידע על אנשי הקשר (Contacts Information), מידע על המשתמשים (Account Information).

על מנת לראות את המידע המפורט אשר הצלחנו לגנוב יש להסתכל בקובץ הinformation.txt.

סך הכל הוספנו 2 הרשאות שדורשות את אישור המשתמש והם הרשאת contacts והרשאת call logs (היו עוד 2 הרשאות, הקשורות לאותן ההרשאות הקודמות, שהוספנו בקובץ הmanifest אך הן אינן דורשות את אישור המשתמש ולכן אינן גורמות לאפליקציה להראות יותר זדונית).

קישור לסרטון הסבר

https://youtu.be/sKqFAoGADTk