**שאלה 2:**

|  |  |
| --- | --- |
| דו״ח ממצאים - echoutil | |
| איום | קריאה לפונקציה unreachable() בechoutil, שאינה נקראת משום מקום בקוד, בעזרת buffer overflow של הקלט לתוכנית. |
| רכיבים מושפעים | תוכנת echoutil, בפרט פיצ׳ר hex escaping. |
| פרטי מודול | Mmn02-q2.cpp, void handle\_escape(const char\*); |
| סוג חולשה | חשיפת מידע מוסתר של האפליקציה והדפסתו. |
| פרטים | כאשר מפעילים את הoptions עבור echoutil, (באמצעות הגדרת משתנה סביבה ECHOUTIL\_OPT\_ON=1), ומפעילים את התוכנה, לדוגמה, עם הארגומנטים:  -e "$(python3 -c "print('\\\\xAAAAAAAAAAAAAAA\x1c') ")"  הפונקציה unreachable() לפתע נקראת ומודפס למסך המסר ‘Cowabanga!’ – דבר שלא היה צריך להתאפשר.  פרצה זו מתאפשרת עקב העובדה שהתוכנה מניחה שקלט המשתמש לא עולה על 17 תווים – לכל ארגומנט להדפסה – שמתחיל בתו ׳\׳ ונכנס לביצוע ב-handle\_escape. כאן הכנסנו 18 תווים, עקב העובדה שאין בדיקה של אורך קלט המשתמש וישנה העתקה עד סופו לתוך l.buffer, נוצר מצב של פלישה לשדה הבא של הstruct – מופע של Handler, שמכיל בפרט רק ערך אחד - מצביע לvtable, שהרי יש פונקציות וירטואליות אשר מוגדרות בHandler.  במקרה לעיל, הכנסת התו האחרון העודף משנה את המצביע הנ״ל לאותה vtable – על המכונה שלי, ההכנסה מסיתה אותו 4 בתים בדיוק אחורה ממיקומו המקורי שנבדק בהרצה קודמת – ואז, כאשר יש קריאה ל-helper(), שמיקומו צפוי (מבחינת הקומפיילר) להיות מאוחסן בvtable+4, למעשה נקרא unreachable(), שנמצא בדיוק במיקום של:  \*(vtable + 4 - 4) = \*vtable  בגלל ההיסט. בקריאה לinterpret נקרא, לכאורה, helper, אך בפועל – unreachable, שמיקומו התקין הושם במקום שבו נמצא המצביע לעיל, עקב שינויו כנ״ל. |
| דרישות לניצול | גישה לקובץ ההרצה של התוכנה, וגישה למשתני הסביבה (להפעלת options) |
| הצעה לתיקון | בדיקת אורך מחרוזת הקלט והגבלת האורך תוך כדי ההעתקה שלה לתוך l.buffer – לגודל קבוע ומוגדר כלשהו, שיכול להישאר 16, ועצירת ההעתקה במקרה וישנה חריגה מגודל מוגבל זה שצוין. ניתן לחילופין להדפיס שגיאה ולצאת מהתוכנית או לדלג על הארגומנט. |