ממן 12  
אופיר רוזנר  
200637494  
  
**שאלה 1**החולשה- הרחבת טיפוסים

* **החולשה** – מבוצע ניצול לרעה של יכולת של השאווה סמוייה של מספרים מסומנים(signed) וכאלו שלא מסומנים(unsigned), כלומר קבלת השי יכולה להיות חיובית,עבור אנשים שהקרדיט שלהם שלילי, למרות שנידרש מינימום של 750 קרדיטים לשם קבלת השי. זאת ללא התרעה ובאופן בלתי צפוי.  
  ניתן לראות שהקבוע bound = 750 הוא מסוג "שלם לא מסומן" (uint), בעוד credit הוא מסוג "שלם עם סימן"(int), כלומת השפה, c++, נדרשת לבצע בין int ל unint ,כלומר תיהיה פה הרחבת הטיפוס int באופן בלתי מפורש ל unint.
* **ההתקפה** – אם לדומגא יש ליעל הסדטודנטית מינוס קרדיטים בחשבון, למשל יש לה מינוס שקל(1-) בחשבון הבבק שלה בגלל שהקרדיט הוא מסוג INT המחשב יעשה השלמה לשתיים כלומר המספר שנקבל זה הערך השלם הכי גדול שיש עבור INT שזה 2^32-1(או 4294967295), וברור שהמספר הנ"ל גדול מ750, ולכן הפונקציה תחזיר "אמת", כלומר שיש ליעל זכאות למתנה למרות שבפועל יש לה מינוס בחשבון. הערה הדבר יעבוד עבור (כמעט) על מספר שלילי אפשרי(חוץ מאלה שמשלימים כמו שהוסבר לעיל מתחת ל750)
* **התיקון** – במקרה שלנו התיקון פשוט   
   במקום int bound = 750 unsinged  
   נרשום int bound = 750

תיקון זה יגרום להשאווה להתבצע בין 2 מספרים שלמים מסומנים(int), במקרה זה, גם אם הסטודנטית שלנו בעלת קרדיט שלילי בחשבון – התשובה תחזור כפי שרצינו

**שאלה 2**

* **איום** – קריאה לפונקציה unreachable() ב " echoutil", אשר אינה נקראת משום מקום קוד, בערת הצפת הבאפר( buffer overflow) של קלט התוכנית.
* **רכיבים מושפעים** – “echoutil” ,בפרט בפיצ'ר\פונקציה handle\_escape עבור hex escaping
* **פרטי מודול -** mmn02-q2.cpp, void handle\_escape(const char\* str)
* **סוג חולשה -** חשיפת מידע מוסתר, אשר לא אמור היה להיות נגיש, של האפליקציה והסתרתו
* **פרטים –** כאשר מפעילים את ה options עבור " echoutil"( באמצעות הגדרת משתנה הסביבה ECHOUTIL\_OPT\_ON ל1) ומפעילים את התוכנה, עם הארגומנטים,נעזר בפייטון בשביל לשלוח כמות ארגומנטים כרצוננו  
  -e “$(python3 -c print(\\\\xAAAAAAAAAAAAAA**A**\xc4"))”  
  הפונקציה unreachable() , תקרא, למרות שאין לה שום קריאה מתוך התוכנית(באופן סדיר), ותודפס למסך ההודעה\הטקסט " Cowabunga", דבר אשר לא היה צריך להתאפשר.  
  פרצה זו מתאפשרת עקב העבודה שהתוכנית שלנו מניחה שקלט שהמשתמש מזין לא עולה על 17 תווים לכל ארגומנט להדפסה – כזה שמתחיל בתו '/' ונכנס לביצוע בפונקציה handle\_escape, בדוגמא שלנו הכנסנו 18 תווים, תו אחד יותר מהאורך המקסימלי מזה שהפונקציה מצפה לקבל,כלומר הפונקציה לא מבצעת בדיקת אורך הקלט מהמשתמש.  
  ישנה העתקה של הקלט עד סופו עבור l.buffer;, במצב זה נוצר מצב של גלישה לשדה הבא של ה struct-מופע של ה Handler, שמכיל בפרט רק ערך אחד, שהוא מבציע ל vtable, שהרי יש פונקציות וירטואליות, אשר מוגדרת ב Handler.  
  במקרה לעיל, הכנסת התו הנוסף(העודף) משנה את המבציע הנ"ל לאותו vtable, ההכנסה מסיתה אותו ב4 בתים(בדוגמא לעיל) בדיוק אחורה ממיקומו המקורי שנבדק בהרצה הקודמת-ואז, כאשר מתבצע הקריאה לפונקציה helper(), שמיקומו, הצפוי,לפי הקומפיילר, להיות מאוחסר vtable+4, ששם נמצת למעשה הפונקציה unreachable, שנמצאת בדיוק במקיום   
  \*(vtable +4 -4) = \*vtable

נסתכל בזיכרון המחסנית  
Text, chat or text message

Description automatically generated  
ניתן לראות שאפשר לעדכן את התאים בזיכרון,אשר באים לפני ערך הכתובת של המצביע הווירטואלי ולאחר מכן להקטין ב4 את הכתובת(כפי שהוסבר לעיל) וכך התוצאה תראה  
Text

Description automatically generated

בעזרת הפקודה שרשמו בהתחלה עלינו להוסיך תו מסויים לאחר התו הראשון של הארגומנוט '\\' של הארגומנט השני לפוקצייה הראשית(main), לאחר מכן את הריפוד של התויים ובסוף את ערך ההקסה החדשה() שבאצמעותו נדרוס את הקוד

בקריאה ל interpret, ציפינו שיקרא, helper, אך בפועל הפונקציה unreachable תופעל, שמיקומה התקין הושם במקום שבו נמצא המצביע לעיל, עקב השינוי שהוסבר לעיל.

* **דרישות לניצול –** גישה לקובץ ההרצה של התוכנית, וגישה למשתני הסביבה(להפעלת options)
* **הצעה לתיקון –** בדיקת אורך המחרוזת והגבלת האורך, והעתקה של המחרוזות לתוך ה l.buffer; - לגודל קבוע כלשהו, וכזה המגודר מראש, יכול להישאר 16 או יכול להיות כל מספר אחר, ועצירת ההעתקה במקרה וישנה חריגה מגודל הנ"ל, לחלופין ניתן לעצור ולהדפיס\לזרוק שגיאה, לצאת מהתוכנית או פשוט לדלג על הארגומנטים אשר נמצאים מעבר לגודל שהגודר לעיל.