**הנדסה לאחור 236496 – תרגיל בית 4**

מגישים:  
אלון פליסקוב 315468116  
שליו ריסין 211578794

חלק יבש

שאלה 1

מדובר ברצף הפקודות:  
  
או, מדובר ברצף הבתים 60 c3.  
את רצף הבתים הזה ניתן למצוא ב-DLL לגיטימי. למשל, עבור הפקודה הבאה:

add BYTE PTR [eax-0x3d], ah

הקידוד שלה הוא רצף הבתים:

00 60 c3

ניתן לראות כי חלק מהקידוד של הפקודה הוא רצף הבתים 60 c3. לכן, ניתן למצוא את הגאדג'ט הנ"ל ב-DLL שבו מופיעה הפקודה שציינו.

שאלה 2

שאלה 3

שרשרת ה-ROP לא עובדת במקרה ספציפי זה, כי strcpy בגרסתה של UNICODE, מפסיקה לקרוא מ-buffer כאשר היא נתקלת ברצף של שני בתים 0x0000. זאת מכיוון שב-UNICODE, הרצף 0x0000 הוא ה-null terminator שמסמן סיום של מחרוזת (bad byte).  
אנחנו חייבים שב-buffer יופיע, למשל, רצף הבתים 0x00000040 צריך להופיע כי הוא מהווה את הארגומנט של ההרשאות החדשות לדף ב-VirtualProtect.

שאלה 4

כעת מסופק לנו ה-gadget שנותן לנו לבצע negation על רצף של ארבעה בתים בזכרון.  
כשנרצה לשים ב-buffer רצף של 4 בתים שבהם יש 2 בתים שהם 0x0000, אז במקום זה נשים את אותם הבתים negated מלכתחילה. כשנצטרך אותם, נוכל לקרוא אותם מהמחסנית באמצעות הגאדג'ט החדש, ואם נצטרך אותם על המחסנית אז נוכל לקרוא אותם באמצעות הגאדג'ט החדש ואז לדחוף מחדש, באמצעות הגאד'גט השמיני (pushad).

שלב שני – התחברות מוצלחת

ניסינו להתחבר לכל 4 המשתמשים.  
A screenshot of a computer error

Description automatically generated with medium confidence  
A screenshot of a computer

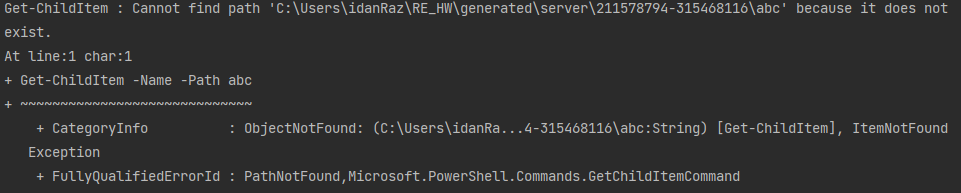
Description automatically generated with medium confidence  
A screenshot of a computer screen

Description automatically generated with low confidence  
A screenshot of a computer screen

Description automatically generated with low confidence  
גילינו כי ל-archer יש הרשאות של admin, ולאחרים אין.  
לכן, ל-archer יש את ההרשאות הגבוהות ביותר. האינדיקציה לכך היא הודעת ה-  
“Welcome archer (Admin)” ש-archer מקבל בכניסתו למערכת. בנוסף, ל-archer יש גישה לשתי פקודות נוספות, שלאחרים אין: PEEK, LOAD.  
כתבנו קובץ **users.py** כמו שהתבקשנו.

שלב שלישי – כתיבת shellcode

שלב רביעי – הרצת קוד על השרת

בחלק זה, רצינו לנסות לחקור ולגלות איך להריץ קוד shellכרצוננו על השרת.  
הרצנו את פקודת PEEK עם ארגומנט אקראי, למשל "abc", וראינו שעלתה שגיאת Get-ChildItem:  
  
לאחר גיגול זריז של השגיאה, ראינו שמדובר בפקודת PowerShell שמקבלת PATH של תיקיה\קובץ ומדפיסה פרטים על הקבצים ב-PATH.  
הרעיון הבא היה לתת ארגומנט '.' ל-PEEK. קיבלנו כתוצאה:  
A computer screen shot of a black screen

Description automatically generated

לכן, מה שפקודת PEEK עושה זה לקבל PATH argument והיא משתמשת ב-Get-ChildItem כדי להדפיס פרטים על ה-PATH, כאשר PATH הוא ביחס ל-current working directory.  
בנוסף, ה-current working directory משתנה ל-PATH אחרי PEEK.  
המטרה הבאה היא להבין איך להריץ קוד בהינתן ש-PEEK מריצה Get-ChildItem על הארגומנט.  
מכיוון שה-PATH בסוף הפקודה של Get-ChildItem, חשבנו לבדוק אופציה של שרשור פקודה נוספת לפקודה הנוכחית, כך שהפקודה הנוספת תהיה האמצעי שלנו להריץ מה שנרצה על השרת.  
חיפשנו, ומצאנו שעל ידי התו ';' ניתן לשרשר שתי פקודות באותה שורה.  
נרצה שהפקודה הראשונה לא תיכשל ותוביל לשגיאה, לכן בכל מקרה נבחר את הארגומנט שלה להיות '.' כי בכל directory יש '.' .  
השורה שנשלח ל-PEEK תהיה:  
.; <our\_command>  
כאשר our\_command היא קוד לבחירתנו. בפרט, our\_command יכולה להכיל semi-colons נוספים, כלומר הפקודה יכולה להכיל מספר פקודות נפרדות ל-shell.  
ראינו שמודפס ה-working directory כל פעם שאנחנו מריצים את PEEK על השורה שכתבנו, לכן ביצענו output redirection כדי "להשתיק" את ההדפסה הזו שאינה רצויה. השורה החדשה:  
. > $null ; <our\_command>  
שלב חמישי – כיבוי השריפות