שאלה 1

סעיף א

מנגנון DEP הוא מנגנון שמשתמש בדגל NX הנמצא במעבד כדי למנוע הרצת קוד מאיזורים המסומנים כאיזורי DATA . למשל, בעזרת DEP ניתן למנוע הרצת קוד על המחסנית, וכך להקשות על תוקף להשתמש בחולשת חריגה מחוצץ כדי לשתול קוד במחסנית ולהריץ אותו. ניתן להתגבר על המנגנון הזה באמצעות הפונקציה virtualprotect אשר משנה את ההרשאות עבור קטע קוד מסוים. נקרא לפונקציה זו ולאחר מכן נוכל לקפוץ לבאפר בו נמצא הקוד שלנו ולהריץ אותו.

סעיף ב

מנגנון SEH הוא מנגנון ניהול ה exception handlers -ב . windows -הוא מיושם ברשימה מקושרת על ה , stack - כאשר

האיבר הראשון ברשימה מוצבע ע"י . fs:[0] כל איבר ברשימה הוא מצביע ל handler -ומצביע לאיבר הבא ברשימה. כאשר

פותחים בלוק try חדש, הקומפיילר מוסיף איבר חדש ברשימה, וכאשר יוצאים מבלוק try הוא מסיר אותו. במקרה של חריגה מערכת ההפעלה ניגשת לאיבר הראשון ברשימה הנמצא ב fs:[0] -וקופצת ל handler -המוצבע ממנו. בנוסף קיים handlerדיפולטי הנמצא בסוף הרשימה, וה next -של האיבר האחרון ברשימה מצביע עלFFFFFFFF .

סעיף ג

התוקף יכול לראות את איברי הרשימה של SEH ובפרט לדעת איפה נמצא ת הכתובת של ה handler -באיבר הראשון ברשימה)שהוא יהיה גם ה handler -הראשון שיקרא במקרה של חריגה(. הוא יכול לזהות את האיבר הראשון ברשימה ע"י

כך שיהיה בו את המצביע לאיבר הבא ברשימה)נתון שהוגדרו שני ,) exception handlers כלומר יהיה בו כתובת לאיזור יותר גבוה במחסנית. והוא יכול לוודא שאכן שני אלה איברים ברשימה המקושרת של SEH ע"י המשך מהלך על הרשימה עד להגעה ל: FFFFFFFF -

סעיף ד

התוקף ימלא את ה buffer -תחילה ב shellcode -אותו הוא מעוניין להריץ. לאחר מכן יכניס NOPS עד להגעה למיקום בו נמצאת כתובת ה handler -הראשון ב . SEH -שם הוא יכניס את הכתובת ל , VirtualProtect -ומעליו)בכתובות גבוהות יותר)את הכתובת של ה(shellcode -כך שלאחר החזרה מ virtualProtect -אנחנו נקפוץ ל , shellcode -ולבסוף יכניס את

הארגומנטים ל virtualProtect -שיכילו את הכתובת של ה , shellcode -את גודל ה , shellcode -הרשאות הרצה, וכתובת

לאיזור מותר לכתיבה כלשהו על המחסנית. נשים לב שהתוקף יודע את כתובת ה shellcode -כיוון שהוא רואה את המחסנית)נתון(. מכיוון שדרסנו את הקנרית, בחזרה מהפונקציה תיזרק חריגה שתגרום להפעלת ה handler -הראשון. וכך נגרום ל virtualprotect -להתבצע ולאחר מכן ל shellcode -שלנו.

-		
	Shellcode	Start of buffer
	...	
	...	
	End of shellcode	
	Nops	
	...	
	...	
	End of nops	
	VirtualProtect	הכתובת בה היה המצביע ל- handler הראשון ב-SEH
	Shellcode address	
	Shellcode address	
	Size of shellcode	
	Execute permissions	
	Some area on the stack	
+		

שאלה 2

סעיף א

הפונקציה sub_401460:

```
int sub_401460(int num){
      int a, counter=0;
      for(int i=0; i<=31; i++){
            a=2^i;
            if (num & a != 0){
                  counter++;
            }
      }
      return counter;
}
```

במילים:

הפונקציה מקבלת כקלט מספר בן 32 ביטים, ומחזירה כפלט כמה פעמים מופיע המספר 1 ביצוג הבינארי של המספר.

היא עושה זאת ע"י ביצוע and לוגי עם חזקות של 2 (1-31) והמספר, וכל פעם שהתוצאה שונה מ0 (כלומר יש 1 במקום הi במספר) מעלים counter.

לבסוף הפונקציה מחזירה את הcounter.

סעיף ב

1. מבנה הנתונים המרכזי והיחיד של המשחק הוא הטבלה הבאה **כאשר השורה הראשונה היא לא חלק מהטבלה, אלא הגודל של כל שדה בעמודה שלה, בביטים**:

4	4	4	1	3
‘A’	‘8’	3	0	0
‘F’	‘6’	2	0	0
‘G’	‘2’	2	1	0
‘D’	‘3’	4	1	0

2. בכל תור קולטים שני תווים (char), נסמן את הראשון c1 והשני c2, תווים חוקיים מקיימים:
$'A' \leq c1 \leq' G' and '0' \leq c2 \leq '8'$
3. בהינתן חלק חוקי, עוברים על הטבלה שורה שורה ומבצעים את הפעולה הבאה:
אם השדה הרביעי בשורה הוא 0:

אזי אם c2 שווה לשדה השני בשורה, וגם c1 נמצא בטווח האותיות שבין השדה הראשון, והשדה הראשון ועוד השלישי, אז מדליקים את הביט בשדה החמישי, שהוא החיסור בין c1 לשדה הראשון.
דוגמא: **עבור השורה הראשונה, כאשר הוכנס הקלט** $\boldsymbol{c1 = 'B', c2 = '8'}$ אזי מתקיים $c2 = table[0][1]$ וגם $'A' \leq c1 <' A' + 3 = 'D'$, אזי נדליק בשדה החמישי באותה שורה את הביט מספר: $'B' - 'A' = 1$ כלומר אם הוא היה 0 הוא יהיה $10_2 = 2_{10}$.
אם השדה הרביעי בשורה הוא 1:
אזי אם c1 שווה לשדה הראשון בשורה, וגם c2 נמצא בטווח הchar שבין השדה השני, והשדה השני ועוד השלישי, אז מדליקים את הביט בשדה החמישי, שהוא החיסור בין c2 לשדה השני.
דוגמא: **עבור השורה השלישית, כאשר הוכנס הקלט** $\boldsymbol{c1 = 'G', c2 = '2'}$ אזי מתקיים $c1 = table[0][0]$ וגם $'2' \leq c2 <' 2' + 2 = '4'$, אזי נדליק בשדה החמישי באותה שורה את הביט מספר: $'2' - '2' = 0$ כלומר אם השדה היה 0 הוא יהיה $01_2 = 1_{10}$.

4. חוקי המשחק הם:
המשחק נמשך 11 סיבובים, כל סיבוב מכניסים c1,c2 חדשים, ש**חייבים** להיות חוקיים, ולאחר מכן עוברים על כל הטבלה ומשנים את השדה החמישי בכל שורה בהתאם לכללים שציינו בסעיף קודם. בסוף כל סיבוב נבדוק האם ערך החזרה של הפעולה שתיארנו בסעיף הראשון על השדה החמישי בשורה מסוימת שווה לשדה השלישי באותה שורה, אם זה נכון עבור כל השורות ניצחנו במשחק.

נשים לב אם כן, שכדי לנצח במשחק נצטרך להכניס זוגות של c1,c2 כך שבכל פעם ידלק ביט אחר בערך של השדה החמישי בשורה מסוימת (להדליק את אותו ביט פעמיים לא יעזור), כדי שמספר הביטים הדלוקים יהיה הערך של השדה השלישי באותה שורה. נשים לב שסכום כל העמודה השלישית בלוח הוא 11, ויש לנו בדיוק 11 סיבובים להדליק ביטים, ולכן זה בדיוק מסתדר. נכניס זוגות של c1,c2 כך שבכל סיבוב נדליק ביט אחר בשורה הראשונה, עד שידלקו בה שלושה ביטים (נשים לב שבגלל התנאים זה גם מקסימום הביטים שנצליח להדליק בלי לחזור על אותו ביט פעמיים), נחזור על זה בכל שאר השורות וננצח.

סעיף ג

הקלט הוא:

A

8

B

8

C

8

F

6

G

6

G

2

G

3

D

3

D

4

D

5

D

6