ארגון ותכנות המחשב

תרגיל בית 4 (יבש)

המתרגל האחראי על התרגיל: לירן רדל.

הנחיות:

- שאלות על התרגיל ב- **Piazza** בלבד.
- ההגשה **בזוגות**.
- על כל יום איחור או חלק ממנו, שאינו באישור מראש, יורדו 5 נקודות.
  - ניתן לאחר ב-3 ימים לכל היותר.
  - הגשות באיחור יתבצעו דרך אתר הקורס.
- לכל שאלה יש לרשום את התשובה במקום המיועד לכך.
- את התרגיל יש להגיש באתר הקורס כקובץ **pdf**.
- יש לענות בטופס התרגיל. ניתן לעשות זאת באחת מהאפשרויות הבאות:
  - להקליד את התשובות במסמך ה-WORD ולבסוף לשמור כ-pdf.
  - לכתוב אותן על גבי גרסת ה-pdf, בעזרת הטאבלט החביב עליכן בכתב ברור וקריא.
  - להדפיס את גרסת ה-pdf ולכתוב על הטופס המודפס את התשובות בכתב יד ברור וקריא.
- תיקונים לתרגיל, אם יהיו, יופיעו ממורקרים.

חלק ראשון – קידוד פקודות וקבצי ELF (25 נקודות)

ניק ועדן הצליחו להביס את זאקום בעזרת קריאות המערכת שיצרו במטלה הקודמת, ומצאו בין אבני המאגמה שבו מקדשו נמצא, דיסק-און-קי מרותח עם סמלים עתיקים עליו. כשניסו לקרוא את הקובץ במחשב, הוא הופיעה בשפה מאוד מוזרה, אך הם יודעים ששמה של התוכנית הינה ForbiddenDropsOfZakum. עזרו לניק ועדן לפענח את התוכנית המצורפת לתרגיל, בעזרת הכלים שנלמדו בקורס (כגון readelf, objdump וכדומה).

סעיף 1 (10 נקודות)

1. מה גודל ה-Section Header Table של התוכנית? Text_____

2. כמה Program Headers מוגדרים בקובץ? _____

3. מה גודל כל כניסה ב-Program Header Table? _____

4. עבור כל program header מסוג LOAD, הכניסו את נתוניו לטבלה הבאה (יתכנו שורות ריקות):

הרשאות (סמנו את ההרשאות)	כמה bytes ייקראו מהקובץ ל-segment	גודל שיתפוס בזיכרון הווירטואלי	כתובת טעינה לזיכרון הווירטואלי	מיקום בקובץ (offset בבתים)
R W E	0x_____	0x_____	0x_____	0x_____
R W E	0x_____	0x_____	0x_____	0x_____
R W E	0x_____	0x_____	0x_____	0x_____

5. באיזו כתובת יתחיל הקוד את ריצתו? 0x_____

6. ידוע כי קיים משתנה בשם findme. השלימו את ערך האתחול החסר, ואת כתובתו:

char findme[15] = _____.

כתובת findme הינה (בבסיס הקסהדצימלי): 0x_____.

המשך השאלה בעמוד הבא

סעיף 2 (15 נקודות)

בכל שורה בטבלה הבאה מופיע קוד קצר בפקודות אסמבלי, או קידוד שלו לשפת מכונה.

עבור כל שורה מלאו (כפי שמופיע בשורה הראשונה כדוגמה):

1. את קידוד הפקודות לפי סדר הופעתן, משמאל לימין, או את הפקודות לפי סדר תרגומן.
2. כתובת בזיכרון התוכנית שבו נמצא קידוד הפקודות. אם הקידוד מופיע בכמה אזורי זיכרון, בחרו באזור בעל הרשאות הרצה.

   רמז: הכתובת בה מופיע הקידוד יכולה להיות שילוב של חלקים מקידוד של פקודות אחרות. מומלץ לקרוא על הכלי grep ולהשתמש בכלים כגון objdump ו-hexdump, שנלמדו בתרגול 8.

פקודות	קידוד	כתובת
`movq %rcx,%rax`	**48 89 c8**	0x400758
`leave` `ret`	_____	0x_____
	48 3d 80 10 60 00 48 89 e5	0x_____
`test %al,(%rax)`	_____	0x_____
`movl $0x7271b848,%eax`	_____	0x_____
`leaq 0x200865(%rip),%rsi`	_____	0x_____

חלק שני – הנדסה לאחור (25 נקודות)

ניק ועדן הצליחו בעזרת לבישת קסדת הזאקום שהכינו מזאקום לאחר הבסתו, שנתנה להם הרבה חוכמה, להוציא חלק מקוד ה-C הכולל את פונקציית main של הקובץ ForbiddenDropsOfZakum.

להלן הקוד שהצליחו להוציא: (secret אינו מוגדר ב-main ותוכנו אינו ידוע)

```c
int main() {
    char password[16];

    printf("Enter the password in order to reveal the secret:\n");
    scanf("%s", password);

    if (brokenCheck(password) == 0) {
        printf("The secret is: %s\n", secret);
    } else {
        printf("You were wrong, and soon I will be revived!!\n");
    }

    return 0;
}
```

הם גילו כי הקובץ ישן וכתוצאה מהשנים הרבות שבילה במאגמה של זאקום, פונקציית הבדיקה של password, הנקראת brokenCheck, נהרסה.

ניק ועדן שמו לב, כי קוד מסוג זה בעל חולשה מסויימת, וכי ניתן לבצע התקפת **ROP** עלייה. על התקפת ROP ניתן לקרוא כאן: https://en.wikipedia.org/wiki/Return-oriented_programming

סעיף 1 (5 נקודות)

ניתן לראות כי לפי הקוד, המשתמש מצפה לקבל סיסמא באורך 16 תווים, ובעצם בעזרת הפונקציה scanf קורא את הסיסמא שהמשתמש מכניס אל תוך הבאפר password. מה הבעיה בשימוש ב-scanf בקוד זה?

_____

_____

המשך השאלה בעמוד הבא

סעיף 2 (5 נקודות)

ניק ועדן התנסו עם התוכנה, וניסו להכניס את הסיסמה: "maplestoryisthebestnostalgicgame!!".
לאיזו כתובת תקפוץ פקודת ret שמבצעת הפונקציה main בסיום ריצתה?
רמז: מה אורך הקלט שהכניסו ניק ועדן? על מה הדבר משפיע?

_____

_____

סעיף 3 (15 נקודות)

תנו דוגמא לקלט שיגרום לתוכנית להדפיס את secret, כלומר לבצע את שורה 8 בקוד ה-C. **יש להסביר בקצרה** את דרך פעולתכם כולל ניתוח הקלט שבחרתם. בתשובתכם השתמשו בפורמט \xHH כדי לציין קלט הקסהדצימלי.
לדוגמא אם הקלט הינו האות a ואחריה בית עם ערך 0x8F שאחריו 0x90, כתבו "a\x8F\x90".
יש לצרף צילום מסך של ההדפסה. ייתכן וכי יהיו הדפסות נוספות שיקרו, וכי התוכנית לא תסתיים כראוי לאחר ההדפסה.
מומלץ להשתמש בכלי echo כדי להזין את הקלט לתוך הקובץ ForbiddenDropsOfZakum.

קלט לתוכנית: _____

הסבר: _____

_____

_____

צילום מסך של ההדפסה:

חלק שלישי – לינקר סטטי (30 נקודות)

דניאל, סטודנטית בקורס את"מ רוצה להתנסות בקישורי האסמבלי שלה, ובידע החדש שרכשה לגביי לינקרים, וכתבה 2 קבצים – a.o, b.o. דניאל השתמשה במחשב ישן ותקול, שהלינקר הסטטי בו לא עובד כראוי, וכאשר הפעילה על הקבצים את הלינקר הסטטי, גילתה כי החורים שהלינקר אחראי למלא בזמן קישור סטטי כלל לא מולאו, וצריכה את עזרתכם בידע החדש שרכשתם בנושא הלינקרים כדי לתקן את קובץ הריצה שיצרה.

בדף הנספחים נמצאים הדפסות של כלים (כגון objdump, readelf) שהופעלו על הקבצים a.o ו-b.o, שני קבצי object files שאוחדו ביחד לקובץ ab.out יחיד.

סעיף 1 (3 נקודות)

מהי פעולת הקישור שהורצה על מנת ליצור את ab.out? יש לנמק בקצרה את מבנה הפקודה.

להלן פלט objdump של ab.out, עם חלקים חסרים וללא פיענוח הפקודות (בקובץ זה ולא בנספח):

```
Disassembly of section .text:
00000000004000b0 <_start>:
  4000b0:   48 c7 c6 ___ ___ ___ ___
  4000b7:   48 8b 14 25 ___ ___ ___
  4000be:   ___
  4000bf:   48 c7 c7 ___ ___ ___ ___
  4000c6:   e8 0b 00 00 00
  4000cb:   48 89 c3
  4000ce:   e8 ___ ___ ___ ___
  4000d3:   48 89 c1
00000000004000d6 <foo>:
  4000d6:   b8 06 00 00 00
  4000db:   c3
00000000004000dc <cat>:
  4000dc:   48 c7 c0 ___ ___ ___ ___
  4000e3:   c3

Disassembly of section .data:
00000000006000e4 <msg2>:
  6000e4:   08 07
  6000e6:   06
  6000e7:   05 04 03 02 01
00000000006000ec <msg1>:
  6000ec:   18 17
  6000ee:   16
  6000ef:   15 14 13 12 11
```

המשך השאלה בעמוד הבא

סעיף 2 (15 נקודות)

עליכם למלא את כל החלקים החסרים בפלט ה-objdump של ab.out לעיל, כך שהקוד שנוצר הינו הפלט התקין של הקשר הסטטי לאחר התיקון.

סעיף 3 (5 נקודות)

כאשר דניאל הסתכלה על הפקודה מתחילה בכתובת `0x4000c6`, תהתה מדוע האסמבלר לא השאיר חור ללינקר הסטטי למלא. מה קידוד הפקודה המתחילה בכתובת זו, ומדוע לא נותר חור ללינקר הסטטי למלא?

_____

_____

סעיף 4 (4 נקודות)

יש למלא עבור כל קובץ, אילו סמלים יש להגדיר כ-`extern` וכ-`global` **בקוד של הקובץ** על מנת שהקישור יתבצע כרצוי: (אם אין צורך למלא מיקום בטבלה, יש לסמן X או להשאיר ריק)

קובץ	סמלים שיש להגדירם כ-`global` בקובץ	סמלים שיש להגדירם כ-`extern` בקובץ
`a.asm`		
`b.asm`		

סעיף 5 (4 נקודות)

דניאל, ששונאת לינקרים, החליטה לכתוב כל הקוד בקובץ אחד ולא לחלק את הקוד שלה לכמה קבצים, כדי שלא תצטרך להשתמש בלינקר סטטי. תארו יתרון אחד וחסרון אחד על ההחלטה שלה לכך:

יתרון: _____

_____

_____

חסרון: _____

_____

_____

חלק רביעי – לינקר דינאמי (20 נקודות)

דניאל רוצה להריץ קוד שמשתמש בפונקציה printf מהספרייה הסטנדרטית של C, והשתכנעה כי יש צורך בלינקר סטטי, אך מתעקשת כי אין צורך בלינקר דינאמי, ולא מעוניינת להשתמש בו כדי להריץ את הקוד שכתבה.

סעיף 1 (4 נקודות)

מה על דניאל לעשות על מנת שתוכל להריץ את הפונקציה בקוד המקורי שלה, ללא שימוש בלינקר דינאמי? אין לציין דגל, אלא להסביר מה התוצאה הרצויה.

_____

_____

סעיף 2 (6 נקודות)

תנו 2 חסרונות עיקריים בחוסר שימוש בלינקר דינאמי במקרה הכללי (מה שימוש בלינקר דינאמי מייעל לנו?):

חסרון ראשון: _____

_____

_____

חסרון שני: _____

_____

_____

המשך השאלה בעמוד הבא

סעיף 3 (5 נקודות)

דניאל השתכנעה, וכעת החליטה להשתמש בלינקר דינאמי, כלומר תקשר באופן דינאמי את הספרייה הדינאמית glibc, ומשתמשת בהדפסות בקוד שלה ואך ורק בפונקציה printf מ-glibc. מה על דניאל לעשות כדי שזמן **טעינת** התוכנית (כל הזמן מרגע "ההפעלה" ועד תחילת ריצת הקוד מה-entry point שלו) שלה יקטן? יש להשתמש בידע הנלמד בקורס בלבד.

_____

_____

_____

_____

סעיף 4 (5 נקודות)

בהנחה שדניאל קימפלה את הקוד שלה עם הדגלים `-Wl,-zlazy`, מה יהיה ההבדל בין הקריאה לפונקציה printf בפעם הראשונה ובפעם השנייה?

_____

_____

_____

בהצלחה!