הטכניון – מכון טכנולוגי לישראל

ארגון ותכנות המחשב

# תרגיל 3 - חלק יבש

המתרגל האחראי על התרגיל: תומר כץ.

שאלותיכם במייל בעניינים מנהלתיים בלבד, יופנו רק אליו.

שאלות בעל-פה ייענו על ידי כל מתרגל.

הוראות הגשה:

- לכל שאלה יש לרשום את התשובה במקום המיועד לכך.
- יש לענות על גבי טופס התרגיל ולהגיש אותו באתר הקורס כקובץ PDF.
- על כל יום איחור או חלק ממנו, שאינו בתיאום עם המתרגל האחראי, יורדו 5 נקודות.
  - גם הגשות באיחור יש להגיש באתר במקום המתאים לכך.
  - שאלות הנוגעות לתרגיל יש לשאול דרך הפיאצה בלבד.
  - ההגשה בזוגות.

מבוא

בעודכם מסתובבים במסדרונות בניין טאוב, בשביל להגיע להרצאה באת"מ, מצאתם על הרצפה דיסק-און-קי חשוד. על הדיסק-און-קי מוטבע הלוגו של המוסד ובצידו השני חרוטה כתובת בשפה זרה. בתוך הדיסק-און-אי נמצא קובץ ההרצה verySecretProgram (המוצרף לכם לתרגיל). מטרתכם בתרגיל בית זה היא לפענח מה אותה תוכנה מסתורית עושה. מומלץ להיעזר בכלים עליהם למדנו בקורס (objdump, readelf וכו').

שימו לב: שני חלקי התרגיל מבוססים על אותו קובץ verySecretProgram המצורף לתרגיל. אל דאגה הקובץ לא באמת יהרוס לכם את המחשב :)

חלק א' – **Reverse Engenering** (35 נקודות- 5 כל סעיף)

בחלק זה נסתכל ונחקור את התוכנית המקומפלת וננסה להבין מה היא עושה.

1. מה גודל ה Section header table? _____

2. כמה program headers מוגדרים בקובץ? _____

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

| הרשאות (סמנו את ההרשאות) | גודל בזיכרון | גדול בקובץ | כתבות בזיכרון | מיקום בקובץ (**offset** בבתים) |
|---|---|---|---|---|
| R W X | | | | |
| R W X | | | | |
| R W X | | | | |
| R W X | | | | |

4. מהו ערך הבית שנמצא בכתובת `0x4015f8`? _____

5. להלן הגדרה של משתנה שנמצא בכתובת `0x603040` השלימו את ערך האתחול החסר:

```
unsigned long hash = 0x_______________
```

השאלה ממשיכה בעמוד הבא

6. לאחר שאספתם בסעיפים הקודמים מספר נתונים יבשים על קובץ ההרצה, אתם כעת מעוניינים להבין ממש מה התוכנית שממנה נוצר קובץ ההרצה.

לצורך כך חבר שלכם שבמקרה עובד במחלקה הסודית להגנת הטכניון, השתמש ב-Decompiler המשוכלל שלו, אך לרוע מזלכם חלקים מן התוכנית לא הצליחו להשתחזר מפאת סודיות יתר. מלאו את החלקים החסרים בקטע הקוד הבא.

<u>הערות :</u>

ניתן להשתמש בשם של המשתנה מהסעיף הקודם.

שימו לב שהקוד קומפל ע"י קומפיילר לכן נמצאות בו כל מיני אופטימיזציות, לדוגמא, במקום לקרוא לפונקציה checkPasswordAux, הקומפיילר עשה לה inline.

בנוסף הקומפיילר מוסיף קוד שאינו מופיע בקוד c, לדוגמא קוד שמגן מחריגת חוצץ, התעלמו ממנו בתרגיל.

```
1.  int checkPasswordAux(char* s){
2.      int sum = __________;
3.      while(__________ ){
4.          char c = *s;
5.          if(c-'a'> 25){
6.              return 100;
7.          }
8.          while(c){
9.              sum += c _______ ;
10.             c ____ 1;
11.         }
12.         s++;
13.         }
14.      return sum;
15.  }
16.  bool checkPassword(char* s){
17.     char* copy = s;
18.     if(checkPasswordAux(s) > ______){
19.         return 0;
20.     }
21.     s = copy;
22.     unsigned long y = 0;
23.     while(_______){
24.         unsigned long x = *s - ______;
25.         if(x>____){
26.             return 0;
27.         }
28.         if(y > _____){
29.             return 0;
30.         }
31.         y = __________;
32.         s++;
33.     }
34.     return ____________;
35. }
```

7. מהי הסיסמה הנכונה שתגרום לפונקציה checkPassword להחזיר true:

_____

חלק ב' – חלק לח. **Binary Exploitation** (65 נקודות)

בחלק זה ננצל חולשה (פרצת אבטחה) בתוכנית בכדי לגרום לה להריץ קוד לבחירתנו על המחשב של המשתמש. נשתמש בטכניקה לניצול חולשות מסוג **ROP**. להלן הגדרת פונקציית main:

```
int main(){

    char password[16];
    printf("enter your password\n");
    scanf("%s", password);
    if(checkPassword(password)){
        printf("Good to see you back agent R. As you know your next mission
will take place in %s. See you there. \n", password);
        return 0;
    }
    printf("wrong password! After 3 wrong passwords this program will destroy
the computer. Good luck. \n");
    return -1;
}
```

1) הסבירו בקצרה מה הבעיה בקריאה של התוכנית ל scanf? (5 נקודות)

_____

_____

2) משתמש הכניס את הקלט הבא:

`supercalifragilisticexpialidocious`

לאיזה כתובת תקפוץ פקודת ret שמבצעת הפונקציה main בסופה? (5 נקודת)
רמז: לפתרון הסעיף מומלץ להסתכל בקוד אסמבלי של main או להשתמש ב־gdb.

_____

השאלה ממשיכה בעמוד הבא

3) בכל שורה בטבלה הבאה מופיע קוד קצר בעמודה השמאלית. עבור כל קטע קוד מלאו:

a. את קידוד הפקודות לפי סדר הופעתן, משמאל לימין.

b. כתובת בזיכרון התוכנית שבו נמצא קידוד הפקדות. 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. לכן בסעיף זה מומלץ <u>שלא</u> להיעזר ב objdump.

ראו דוגמה בשורה הראשונה. (10 נקודות)

| פקודות | קידוד | כתובת |
|---|---|---|
| `pop %r13`<br>`pop %rdi`<br>`ret` | 41 5d 5f c3 | 0x401b6b |
| `syscall` | | |
| `pop %rax`<br>`ret` | | |
| `pop %rsi`<br>`pop %r15`<br>`ret` | | |
| `add %r15, %rdi`<br>`ret` | | |
| `push %rbp`<br>`mov %rsp, %rbp`<br>`call *%rax` | | |

4) תנו דוגמא לקלט שיגרום לתוכנית לצאת עם קוד יציאה 48x0. להצגת קוד היציאה של התוכנית האחרונה שהרצתם הריצו את הפקודה "`echo $?`". צרפו צילום מסך של ערך היציאה. לכתיבת ערכים בינארים בתשובה שלכם השתמשו בפורמט \xHH. לדוגמה, אם הקלט הוא האות a ואחריה בית עם ערך 80x0 שאחריו 90x0, כתבו "a\x80\x90". אין חשיבות לפלט שיודפס לגבי נכונות הסיסמא. (15 נקודות)

<u>השאלה ממשיכה בעמוד הבא</u>

5) תנו דוגמא לקלט שיגרום לתוכנית ליצור תיקייה בשם `my_first_rop` עם הרשאות 0755 (אוקטלי) תחת התיקייה הנוכחית. הניחו שלא קיים קובץ או תיקייה בשם זה תחת התיקייה הנוכחית ושיש הרשאות ליצור תיקייה זו. אין חשיבות לדרך היציאה מהתוכנית ואין חשיבות לפלט שמודפס לגבי נכונות הסיסמה. בפרט, זה בסדר שהתוכנית תסתיים כתוצאה מsegfault או סיגנל אחר לאחר יצירת התיקייה. (30 נקודות)