מבחן בהנדסה לאחור - 236496

סמסטר חורף תשפ"א

מועד א׳, 22.02.2021

מרצה: עומר קדמיאל

מתרגל: עידן רז

חלק ראשון

משך חלק זה של הבחינה: 3 וחצי שעות.

בחלק זה של הבחינה ישנן 3 שאלות. ענו על כולן.

מותר להשתמש בכל חומר עזר, אך אין לשוחח, להתייעץ, לקבל רמז, לתת רמז, או להיעזר בשום אדם אחר על המבחן, על תוכנו, והתשובות לשאלות. <u>יש לפתור את השאלות לבד.</u>

בכל מקרה של שימוש בחומר חיצוני יש לתת הפניה אליו.

תוכנות מותרות: התוכנות אשר נלמדו בקורס. בפרט, אין להשתמש בכלי Decompilation!

ענו תשובות ברורות ככל האפשר. **<u>נמקו את כל תשובותיכם</u>**.

במידה שניתן לענות על שאלה במספר דרכים, **<u>ניקוד מלא יינתן לפתרונות קצרים ופשוטים</u>**.

<u>השאלות והקבצים המצורפים דומים אך אינם זהים לכל הסטודנטים</u>.

הגשה אלקטרונית באתר המקצוע עד השעה 12:30. יש להגיש קובץ ZIP (ואך ורק ZIP) ובו כל קבצי ההגשה. ההגשה כוללת בדיוק את הקבצים הבאים:

1. קובץ בשם part1.pdf עם התשובות בכתב <u>לכל</u> השאלות.
2. קובץ בשם crackme.txt אותו תצרו בשאלה 2.
3. קובץ הi64/idb מIDA המתאים לcrackme.exe.

<u>הערה לכל המבחן: בתרגילים הרטובים, אם אינכם מצליחים להגיע לפתרון שרץ בפועל, תארו במילים מה ניסיתם לעשות - זה עשוי לזכות אתכם בניקוד חלקי!</u>

בהצלחה!

שאלה 0:

נא להעתיק את הקטע הבא בתחילת ה-PDF ולמלא את שמכם המלא ותעודת הזהות שלכם כאישור לכך שקראתם אותו.

אני _____, ת.ז. _____, מאשר/ת כי אני מכיר את מדיניות הקורס והפקולטה בנושא יושר אקדמי במהלך הבחינה. בפרט, אני מאשר/ת כי אני מודע/ת לכך שבמהלך הבחינה אסור לי לתקשר עם אחרים בכל אמצעי כל שהוא, למעט עם צוות הקורס.

שאלה 1 (20 נקודות) - יבש:

1. בעקבות החולשות שהתגלו במימוש heap בעזרת רשימות מקושרות, עלה הרעיון להשתמש בעץ בינארי על מנת לנהל את הheap.

כל איבר בעץ מתאר סגמנט רציף של זכרון. מבנה הנתונים HeapTreeNode מתאר איבר בעץ:

```
{
        Int         Type
        Int         Start
        Int         End
        HeapTreeNode *      Left
        HeapTreeNode *      Right
        HeapTreeNode *      Father
        HeapTreeNode *      Sibling
}
```

כאשר:

- Type יכול להיות
  - Free = 0
  - Used = 1
  - Father = 2

- Start ו-End הם התחלת וסיום הזכרון הרלוונטי
- Left ו-right הם מצביעים לבנים הנוצרים בתהליך הקצאת הזכרון.
- Father הוא מצביע לאב של הצומת (null בשורש).
- Sibling הוא מצביע לבן השני של אותו האב (null בשורש)

1. בתחילת התוכנית קיים עלה אחד שהוא Free שמכיל את כל כתובות הזכרון של ה-heap.
2. הקצאת זכרון מתרחשת בתהליך הבא:
   a. מוגדר (על ידי תהליך אחר שאינו רלוונטי לשאלה) היכן יש להקצות את הזכרון.
   b. נעשה חיפוש על העץ במטרה למצוא את ה-Free Node הרלוונטי. כתובת ההתחלה של הזכרון שהוקצה שווה לכתובת ההתחלה של ה-Free Node.
   c. אם גודל הזכרון שהוקצה שווה לכלל הזכרון ב-node הדבר היחיד שמשתנה הוא שה-node הופך ל-Used. אחרת, ה-Node הופך להיות Father, ומוקצים שני צמתים חדשים לעץ (בכתובות זכרון אקראיות) שמוגדרים כבנים של הצומת הנוכחי: שמאלי – הזכרון שהוקצה וימני – שעדיין פנוי.

3. שחרור זכרון מתבצע ידי שינוי הדגל של used ל-free ואז מתבצע תהליך pruning לעץ באופן הבא:

אם Sibling->Type == free אז :

Father->Type = Free

Father->Left = null

Father->Right = null

אחרת, לא מתבצע שום דבר נוסף.

בנוסף, קיים מנגנון garbage collection אשר מנקה את כל הצמתים אשר אין מצביעים מאבות אליהם.

א. האם ניתן ורצוי לבצע תהליך מקביל לתהליך המתואר בסעיף 3 לעיל גם במקרה של הקצאת זיכרון ושני אחים used? פרטו והסבירו
ב. הסבירו מהי חולשות use after free והאם המעבר לעץ מתמודד עם חולשות כאלו? נמקו
ג. בהנתן שקיימת יכולת לדרוס את אחד ה-node-ים (לצורך העניין באמצעות buffer overflow) – האם ניתן להגיע לחולשה שניתן לנצל אותה בדומה לחולשות שנסקרו בכיתה? איזה node צריך לדרוס? פרטו מה החולשה נותנת (ומה לא) ותנו דוגמאות לשימוש בה.
ד. איזה מההגנות שנסקרו בקורס יכולה להתמודד (הגנתית) עם האיום המתואר בסעיף ג'?

2. נתונה קופסא, המקבלת מספר קוד באורך 256 ביט ומשווה אותו לקוד שנמצא בתוכה. ההשוואה נעשית בית בית – 8 ביט בכל פעם. בכל אחד מן התסריטים הבאים, הסבירו האם וכיצד ניתן לנחש את הקוד. לא תתקבל תשובה ללא הסבר.
   1. נעשית השוואה בית בית, וכאשר מגיעים לבית לא נכון – הקופסא מפסיקה את ההשוואה ומדליקה נורה אדומה. אם היא מקבלת את הקוד הנכון – היא מדליקה נורה ירוקה.
   2. כנ"ל, אך נעשית בדיקה על כלל הבתים לפני שנדלקת הנורה.
   3. בדומה ל-2, אך בסוף המעבר על כל הקוד נשלחת הודעה לשרת חיצוני מה הבית הראשון אשר היה שגוי וממתינה עד לקבלת התשובה ממנו. ניתן להניח כי יש גישה להודעות הנשלחות לשרת החיצוני.
   4. בדומה ל-3, אך ההודעה מוצפנת סימטרית
   5. בדומה ל-4, אך נשלחת הודעה על כל בית שגוי
   6. בדומה ל-5, אך באמצעות bit field (הודעה אחת לכל הבתים השגויים)

שאלה 2 (30 נקודות) - CrackMe:

למבחן מצורף הקובץ crackme.exe. חקרו את הקובץ וענו על השאלות שלהלן. את התשובות המילוליות יש לרשום בקובץ part1.pdf ואילו את הקלט המבוקש בסעיף ג', **יש לרשום הן בקובץ part1.pdf והן בקובץ crackme.txt.**

א. תרגמו את הפונקציה **sub_ 40148A** לקוד דמוי C. אין להשתמש בכלי decompilation אלא לתרגם בעצמכם לקוד אשר המתכנת הסביר היה כותב. מה מטרת פונקציה זו בתכנית? מה היא מקבלת כקלט, מחזירה כפלט, וכיצד היא מבצעת את פעולותיה?

ג. התכנית שקיבלתם הינה משחק. תארו את פעולת המשחק על פי הסעיפים הבאים:

1. מהם מבני הנתונים במשחק? כיצד הם מאותחלים?
2. איזה קלט מהמשתמש נחשב חוקי?
3. בהינתן קלט חוקי, מהי השפעתו על המשחק? למשל, אם ישנו לוח, כיצד מושפע הלוח? איך משתנים מבני הנתונים כתוצאה מקלט זה?
4. מהם חוקי המשחק? בנוסף, מהם התנאים המובילים לניצחון במשחק?

ד. מצאו וקלט המוביל לניצחון במשחק. שימו לב שעל הקלט לעבוד, כלומר, להוציא פלט המעיד על הצלחה, עבור הרצת השורה הבאה :

```
crackme.exe < input.txt
```

יש לצטט את הקלט גם ב part1.pdf.