מבחן בהנדסה לאחור - 236496

סמסטר אביב תשפ"א

מועד א׳, 22.07.2021

מרצה: עומר קדמיאל

מתרגל: עידן רז

חלק שני

משך חלק זה של הבחינה: 3 שעות.

בחלק זה של הבחינה ישנן 2 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>

נא לכבות מכשירים סלולריים ושעונים חכמים.

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 17:00. יש להגיש קובץ ZIP (ואך ורק ZIP) ובו כל קבצי ההגשה. ההגשה כוללת בדיוק את הקבצים הבאים:

1. קובץ בשם ID_part2.pdf (כאשר ID זאת תעודת הזהות שלכם) עם התשובות בכתב <u>לכל</u> השאלות.
2. הקבצים אשר תצרו בשאלה 2 – Injector.cpp, hook.cpp, Injector.exe, hook.dll.

<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 (25 נקודות) - יבש:

אנחנו רוצים להריץ קוד שהצלחנו להחדיר למערכת באמצעות String Buffer Overflow (לדוגמה באמצעות strcpy).

הקוד באורך 4096B, נמצא ב-[edx] ואין לו הרשאות ריצה.

אין לנו גישה לפונקציות ה-winAPI. מאידך, בכתובת 0x41234 קיימת הפונקציה

_cdecl unsigned int SafeMem (int ActionType)

כאשר ActionType:

1 – מקצה זכרון בגודל 4096B עם הרשאות ReadWrite ומחזירה את הכתובת להתחלתו.

2 – מקצה זכרון בגודל 4096B עם הרשאות ReadExecute ומחזירה את הכתובת להתחלתו.

4 – מעבירה בין שני סוגי הזכרון (מחליפה בין ReadWrite לReadExcecute ולהפך) עבור ההקצאה האחרונה שבוצעה.

א. כיתבו קוד בשפת C המבצע את הפונקציונאליות הנדרשת באמצעות הפונקציה SafeMem ו-memcpy בלבד. ניתן להניח כי קיימים משתנים לפי רצונכם.
ב. הסבירו מה המשמעות של _cdecl, ואיך ישפיע על שרשרת הROP.
ג. בהנתן שאנחנו עובדים ב-ASCII, מה הבעיות העומדות בפנינו על מנת לייצר שרשרת ROP רלוונטית?
ד. לאור הסעיפים הקודמים, כתבו שרשרת ROP המבצעת את הפעולות הרלוונטיות. ניתן להניח קיומם של העתקה מרגיסטר לרגיסטר, העתקה מרגיסטר לזיכרון, פונקציות מתמטיות בסיסיות, פונקציות ביטים בסיסיות (כולל XOR) וגאדגטי pop, כמו גם הכתובת לmemcpy (0x71727374). יועדפו פתרונות קצרים ויעילים.
הערות:
- פתרונות שלא מכילים NULL bytes יקבלו בונוס.
- ניתן להניח כי הכתובות של כל הגאדגטים אינם מכילים NULL.
- אין להשתמש בגאדגטי popa , push או וריאציות rep movs כלשהם.
- שימו לב לכך שedx הוא volatile.

שאלה 2 (25 נקודות) – רטוב:

חלק א

נתונה הפונקציה הבאה:

```
char* foo(char* address) {
    if (address[0] != '\xE8') {
        return NULL;
    }
    int x = *(int*)(address + 1);
    return address + x + 5;
}
```

הסבירו בקצרה מה הפונקציה עושה, מה היא מקבלת ומחזירה?

חלק ב

למבחן מצורף הקובץ secret.exe. התוכנה secret.exe מדפיסה מחרוזת מוצפנת כלשהי.

תוכן המחרוזת שמודפסת מסתמך ע"י הפונקציה שמתחילה בכתובת 0x401460 ומסתיימת בכתובת 0x40150C.

פונקציה זו מקבלת int, מבצעת חישוב מסובך ומחזירה int אחר. בנוסף, לפונקציה זו יש מספר נקודות כניסה (כלומר ישנן מספר נקודות בקוד המבצעות call לכתובות שונות בגוף הפונקציה) ונקודת יציאה אחת. נקודת הכניסה יכולה להשפיע על החישוב המתבצע. לא ניתן להניח כי לאחר הret אין פקודות חשובות של פונקציה אחרת. ידוע כי החישוב שמבצעת הפונקציה משפיע רק על המשתנים הלוקאליים שעל המחסנית והרגיסטרים.

על מנת לפענח את המחרוזת, עליכם לכך שההפעלה הi של הפונקציה (כל נקודות הכניסה נספרות ביחד) עם הערך x תחזיר ערך כאילו קראו לפונקציה מאותה נקודת כניסה עם הערך x+i. כלומר: עבור ההפעלה הראשונה – f(x) -> f(x+1), עבור ההפעלה השנייה – f(x) -> f(x+2) וכך הלאה.

עליכם לבצע hooking באמצעות **DLL Injection** על התכנית secret.exe כך שהמחרוזת תודפס בצורה המפוענחת שלה. את ה Hook יש לבצע על הפונקציה הנ"ל בלבד! **אסור לבצע כל שינוי נוסף בתכנית.**

עליכם לבצע זאת באמצעות **hooking רגיל** בלבד, כפי שנלמד בקורס, **ללא שימוש בIAT hooking או Hot patching**.

מותר לבצע דריסה אחת בלבד בפונקציה – בכל מקום בין תחילתה (כתובת 0x401460) לסופה (כתובת 0x40150C).

שימו לב שהמחרוזת המפוענחת צריכה להכיל את הת"ז שלכם.

רמז: השתמשו בפונקציה מחלק א'.

עבור שאלה זו עליכם להגיש את הקבצים הבאים:

- injector.cpp - קוד המקור של ה-injector.
- injector.exe - ה-injector המקומפל.
- hook.cpp - מכיל את הקוד עבור האתחול של ה-hook, ואת פונקציית ה-hook עצמה.
- hook.dll - ה-dll המוזרק.
- צרפו צילום מסך של המתרחש לאחר הרצה מוצלחת של ההוק ב-part2.pdf.

על שמות הקבצים ב-ZIP להיות זהים לאלו המופיעים ברשימה לעיל.