הנדסה לאחור - מטלה 2

מיכאל בייגל 213764251 דניאל יהושע 215332362

Wizard

שלב 1

ניתחנו את הקוד סטטית ושמנו לב שעל הקלט להיות 12 מספרים שונים בין 0 ל11.
וכי הקלט עובר המרה סטטית ומודפס בצורת מגן דוד.
להלן ההמרה שעובר:

user input -> Magen-david output
10 --> 1
1 --> 2
11 --> 3
8 --> 4
5 --> 5
0 --> 6
4 --> 7
6 --> 8
9 --> 9
7 --> 10
3 --> 11
2 --> 12

כאשר הסדר הוא מלמעלה למטה ומשמאל לימין בהדפסת המגן דוד.
(כלומר קלט המתחיל ב 11 ואז 0 ידפיס קודקוד עליון 3 וקודקוד שמאלי עליון 6).

שלב 2

מכאן שמנו לב ל7 תנאים הבאים:
למגן דוד 2 משולשים:
סכום כל אחת מהצלעות במשולשים אלה צריך להיות שווה ל26(סה"כ 6 צלעות)
בנוסף על סכום 6 הקודקודים להיות גם 26.

מכאן שהקלט הראשוני הוא:
10 1 2 9 11 0 3 4 7 8 5 6

שלב 3

כעת עלינו להכניס קלט כך ש29 התווים הראשונים שלו לאחר המניפולציה הבאה יהיו שווים ל"PIAIAEDHMCCIKPHGGHJAAKMIFELMM"

המניפולציה הינה:

```
for i=0; i<=28; i++:
str[i] = str[i]-41x
str[i] = str[i] xor str[(i+1)%29]
str[i] = str[i]+41x
```

מכאן שעל המחרוזת להתחיל ב:"MDLLDDHEDPNPHNCFDFCLLLBNFAEPD"

שלב 4

נשים לב שבתחילת פונקציה זו(הפונקציה שלוקחת מחרוזת כקלט ועוד)
יש בהתחלה בדיקה האם זאת הקפיצה הגיע מתוך הפונקציה עצמה ועם כן יש הדפסות).
שמנו לב שמתבצע השרשור הבא:

str = str + function address

ובנוסף שכתובת החזרה של הפונקציה נמצאת במרחק 45 בתים בתחילת המחרוזת שנכניס כקלט.

מכאן שאם נרפד את "MDLLDDHEDPNPHNCFDFCLLLBNFAEPD" בעוד 16 בתים נוכל לשכתב את ערך החזרה של הפונקציה כך שתקרא לעצמה שוב(כעת עם דגל 1)

לכן נכניס "MDLLDDHEDPNPHNCFDFCLLLBNFAEPDAAAAAAAAAAAAAAAA"
כלומר נרפד ב16 בתים כלשהם ונקפוץ בפעם השניה לתחילת הפונקציה כעת עם דגל שמאפשר את הדפסת הסיסמא המוצפנת:

The hashed password is: 92e400a0E7FB96B8d5b160d965786160

Goblin

ראשית, שמנו לב שיש memset בקוד , לאחר הרצה ומיפוי של המערך לאחר אותם memsets , שמנו לב למטריצה הנראית כך:

```
Row,Col  0,  1,  2,  3,  4,  5,  6,  7

  0    .    .    2    .  'F'    7    .    .   ← start (0,7) Left diraction
  1    X    X    X    X    X    X    X    X
  2    .    .    .    .    .    .    4    .
  3    1    .    .    .    .    .    .    .
  4    .    .    .    .    .    .    .    .
  5    X    X    X    X    X    X    X    X
  6    .    .    .    .    .    .    .   'F'
  7    .    .    .    .    .    .    .    .
       ^
       |
       |
     end (7,0)
```

לאחר מכן שמנו לב לכך שיש מצביע שנע בהתאם לקלט שאנו מכניסים: הוא כמעט תמיד יתחיל עם C ולאחריו מספר כלשהו, כשהכנסנו קלט עם מספר שמנו לב שהמצביע זז לפי אותו מספר, וכאשר הוא הגיע לשורה הבאה (כשאין x) כיוון התנועה שלו התהפך . כעת שמנו לב לכך שכאשר הוא עומד על מספר כלשהו, יכולנו לקבל כקלט גם את U,D שהניעו אותו מעלה או מטה במספר צעדים הזהה לאותו מספר. בנוסף ראינו שכיוון התנועה לא משתנה כאשר נקפוץ, ולא נוכל לצאת מתחום המערך בקפיצה. לבסוף ראינו שאנו נצא מהקוד כאשר נגיע לX, וכאשר נגיע למשבצת עם F , משתנה מסוים שזיהינו כ"ציון" יעלה ב1. וכאשר נגיע למשבצת היעד, הציון שלנו ייבדק אם הוא שווה ל2. מכאן הסקנו,שנצטרך לאסוף את כל הFים בדרך, כך שנגיע למשבצת האחרונה. לכן הקלט שהכנסנו היה:

C5
U
C3
D
C6
U
C9

וזה הביא לנו את הפלט:

The encryption super secret key is ?I,cykJ![QWL*_6d

Giant

שלב 1

תחילה מכניסים כקלט מספר הקסדסימלי hex_input המקיים:

hex_input < 0 && not hex_input + 1 < 0

המספר היחיד שמקיים זאת הוא: hex_input = 0x80000000

שלב 2

נכניס עוד 3 מספרים דצימלים המקיימים:

x = 224

y < 0 && y - 1 > 0

-82z^2 -7380z - 166049 = 1

מכאן ש:

x=224

המספר היחיד המקיים את הדרישה y=2147483647(0x7FFFFFFF)

פשוט פתירת משוואה ריבועית z=-45

שלב 3

כעת נשים לב שיש מערך של 8 מספרים:

4 9 2 5 3 8 7 1

וכי נעשית בדיקה האם כל מספר גדול מקודמו על מנת להגיע אל הדפסה כלשהיא.
בנוסף יש קריאה רקורסיבית לפונקציה עם מצביע לחצי ממערך.

ראינו כי עלינו להכניס קלט של 10 ספרות הקסדצימליות וכי מבוצע שכתוב של הקוד.
במקום בו יש כבר nops שמוכנים לסוג של hotpatching הנמצא בידיוק אחרי הקריאה הרקורסיבית שהוזכרה.

כאן חשדנו כי מדובר שmerge sort שחסרה בו הקריאה השניה לחצי השני של המערך.
עלינו לחשב את הoffset מסוף 5 הnops עד לתחילת הפונקציה ולהכניס זאת כקלט.

calling address(0x00401772) - merge_sort address(0x004018A8) =

-0x136(0xFFFFFECA)

E8 = call opcode

כלומר נכניס כקלט: E8CAFEFFFF

שלב 4

ההדפסה היחידה שנותרה היא בפונקציה שהגדרנו ונקראית רק בעת שגיאה.
שמנו לב בקוד כי יש קריאה מזיכרון מכתובת:

4 תווים שהכנסנו כקלט וערבול עליהם - "NLUL"

נשים לב שעל הקלט שלנו להיות "NULL" כך שלאחר המניפולציה והחיסור בעצם נקרא מכתובת 0, מה שיעלה לנו שגיאה אשר תטופל ע"י הפונקציה שלנו כנדרש.

וזה הביא לנו את הפלט:

The encryption params are 8 (rounds) and 69629837 (delta)

Decrypt

לאחר שפתרנו את כלל הקבצים הקודמים, החלטנו להריץ את decrypt עם הפרמטרים שקיבלנו.
לאחר בדיקה קלה בקוד ראינו שהkey הוא האיבר האחרון, לפניו הסיסמא, ולבסוף הdelta , כך שהrounds זה הפרמטר הראשון.
כעת ניסינו את הסיסמא שקיבלנו ללא הצלחה, אך אז שמנו לב שאם הסרנו תו מסוים בסיום מהhashed password בקלט, הפלט נשאר זהה. הסרנו כך את 16 התווים האחרונים , וקיבלנו פלט מסוים, ולאחר מכן החלטנו להריץ רק עם 16 התווים האחרונים וקיבלנו את החצי השני של הסיסמא, כך שלבסוף הסיסמא לshared היא:

0SD5XPGWNSOPHO4N

כלומר התבצע פענוח של בלוקים בגודל 16 בתים.

Sheep

ראשית, פתחנו את sheep.jpg בעזרת analyzer.exe שניתן לנו. זה נתן לנו את הקובץ sheep.out. שלאחר פתיחה בIDA גילינו שהוא למעשה קובץ הרצה.

כעת, שמנו לב לכך שיש פונקציה בקוד הנ"ל שמאתחלת מערך דו מימדי בגודל 6x6. אותו מערך כולל רצפים של אותיות באורך של 2 או 3, ובניהם נקודות. מיד שמנו לב לכך שהמערך הדו מימדי הנ"ל הוא למעשה מפה של המשחק "שעת השיא" בו אנו נצטרך להביא את המכונית שמסומנת בx לקצה השני של הלוח (x זה הסימון הקונבנציונלי למכונית שאותה נצטרך להזיז). כעת שמנו לב שיש מקרה יחיד של קלט, המקבל 3 תווים , על התו השלישי יש בדיקה אם הוא מספר, על התו השני אם הוא האות L,D,U,R ולכן הסקנו כי התו הראשון הוא של האות המתאימה. כעת פתרנו את השלב שהוצג שם ושמנו לב שהקלט שלנו משנה את המחרוזת שנוציא בסיום, לאחר שגילינו שהקלט הראשון שהכנסנו הוביל למחרוזת שגויה. ניסינו מספר אפשרויות שונות:

קלט: ED3 GR1 FD3 AU1 QR2 PL1 RD1 OR3 XU3

פלט: You won txb#wfne!
Here is a single ucb#shge: UU5ZX29HWK. Use it0pjcboy.

קלט : FD3 AU1 ED3 QR2 GR1 PL1 RD1 OR3 XU3

Zou&pom the afme!

Here is a#sih`lf use ehde: UU5ZX29HWH. Ste#it wiubly.

FD3 ED3 GR1 QR2 AU1 PL1 RD1 OR3 XU3: קלט

Zou"amo6vhe game!
Here is a#silqnd6wse code: UU5ZX29HWH. Weg!v wisely

עד שהגענו לרצף הפתרון הבא:

ED3 GR1 FD3 QR2 AU1 PL1 RD1 OR3 XU3: קלט

שהחזיר לנו:

You won the game!

Here is a single use code: UU5ZX29HWK. Use it wisely.

לסיום נעזרנו בהפעלת analyzer.exe על התמונה
Qrcode.png ששני התווים הסופיים בתוצאה שם נתנו לנו את המיקום בלוח שנצטרך לשנות (A4),
ועם הסיסמא הנתונה לנו הצלחנו להחזיר את הלוח לסדרו.

<u>קבצי עזר:</u>

analyzed_qrcode.png:
תוצאת הפעלת analyzer.exe על התמונה qrcode.png שהופיעה בתיקייה הפרטית של wizard.

Sheep.exe
קובץ הout שקיבלנו מהכספת shared , לאחר שינוי חתימה מout לexe כדי שנוכל להריצו במערכת windows.

Decrypt_half1.args
הפרמטרים שנדרשנו להם כדי לפענח את החצי הראשון של הסיסמה

Decrypt_half2.args
הפרמטרים שנדרשנו להם כדי לפענח את החצי השני של הסיסמה

Codes.pdf:
קובץ הpdf שנמצא בתיקייה shared ומסביר על דרישות הקלט של הסיסמה.