עקרונות וכלים באבטחת מחשבים - תרגיל בית 5

חן גולדנברג – 312141609

יואב כהן - 203115373

שאלות 1.1 + 1.2 :

מצורפת תמונה המכיל את תוצאת ריצה של שני האלגוריתמים (קבלת זמני גישה לזכרון הראשי וזמני גישה למטמון). בריצה זו דרשנו מינימום של 100 איטרציות עד להתייצבות החציון.
ע"מ לדייק את התוצאה אף יותר ניתן לבצע מספר איטרציות גדול בכמה סדרי גודל.

```
cyber@cybercomp:~/HW$ ./basic
main_memory_access_latency results:
total number of iterations: 102
median memory access time recoreded: 292
minimum memory access time recoreded: 252
maximum memory access time recoreded: 1302

cache_access_latency results:
total number of iterations: 102
median memory access time recoreded: 39
minimum memory access time recoreded: 34
maximum memory access time recoreded: 65

median main memory access latency: 292
median cache access latency: 39
```

שאלה 1.3:

ניתן לבחור כל מספר הגיוני שמתאים להפרדת זמני הגישה לזכרון הראשי למטמון. ראינו בסעיפים קודמים שזמן הגישה הממוצע למטמון הוא בערך 39 מחזורי שעון, והגישה לזכרון הראשי היא בסביבות 292 מחזורי שעון.

לצורך בחירת ערך הסף אשר יבדיל האם הכתובת שאליה ניגשנו נמצאת במטמון או בזכרון הראשי, ניתן לבחור מספר שנמצא באמצע בין מקסימום מחזורים לגישה למטמון (65) למינימום מחזורים לגישה לזכרון הראשי (252).

נקבל כי ערך הסף הנבחר הוא: $\frac{65+252}{2} = 158$

שאלה 1.5:

לצורך חישוב קצב התעבורה לביט נעזרנו בתוכנית לצורך ספירת מספר התווים שעוברים והתקשל שכולל תו הEOF ישנם 24 תווים. כל תו הוא 8 ביט, ולכן סה"כ אנו מעבירים 192 ביטים.

ניתן לקראות בתמונה המצורפת שזמן הריצה של התוכנית הוא 0.002 שניות.

ומתקבל קצב התעבורה הבא : $\frac{192}{0.002} = 96000$

```
cyber@cybercomp:~/HW$ ./sender < demo.t
xt
num of chars + EOF = 24
cyber@cybercomp:~/HW$
```

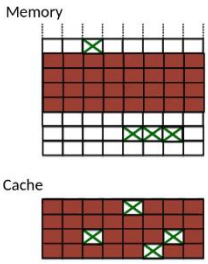
```
cyber@cybercomp:~/HW$ time ./receiver
Hello covert channel!

real    0m0.002s
user    0m0.002s
sys     0m0.000s
cyber@cybercomp:~/HW$
```

שאלה 2.1:

בקוד המצורף ניתן לראות שבמידה והMSB של תא מסויים במערך המידע מקיים שערכו הוא אפס – אנו ניכנס לתוך הIF בקוד, ובכך תוקף יכול להזליג את הביט הבודד הנ"ל בעזרת מתקפת תזמון המטמון. כלומר, עבור המערך הנתון, אשר בגודל 100 – דולפים 100 ביט.
ובאופן כללי, למערך סודות בגודל n דולפים n ביטים.

שאלה 2.2:

תקיפת Prime and Probe



מודל התקיפה -

- התוקף והנתקף רצים מעל אותה ליבה. כלומר המשאב המשותף הוא מטמון L1.
- התוקף בעל יכולות user space.
- התוקף מכיר את הקוד הנתקף, ואת המיפוי של כל שורה בקוד.

תיאור התקיפה -

- התוקף יוצר באפר שנשמר בזיכרון בגודל זיכרון המטמון.
- התוקף נכנס לכל הבאפר ובכך ממלא את המטמון במידע שלו.
- הקורבן מבצע את קטע הקוד שלו ובכך מפנה חלק מהבלוקים במטמון שהתוקף הכניס.
- התוקף שוב מבצע גישה לכל הבאפר ומודד את זמני הגישה עבור כל בלוק.
  הסטים אליהם הקורבן ניגש הם אלו עבורם זמן הגישה גדול יותר.
- התוקף שומר העתק של גישות הנתקף וע"י הנדסה לאחור יכול לגלות את הMSBים בעזרת התבוננות בשורה המטמון שמכילה את $Zero_msbs_counter$.

שאלה 2.3:

נתאר את השינוי בקוד שימנע את התקיפה שתארנו בסעיף קודם תוך שמירה על נכונות הקוד:

נציע את שני הפתרונות המצורפים, כאשר המחשבה מאחוריי המימוש נובעת מהרצון לבטל את יכולת התוקף להשיג מידע אודות תוכן הסוד שמתקבל בעקבות מעקב על שורת המטמון המתאימה.

בשני הפתרונות המוצעים דאגנו לעדכן את תוכן המשתנה $Zero_msbs_counter$ בכל ריצה של האיטרציה. תוכן ה$Zero_msbs_counter$ משתנה בהתאם לערך הMSB, אך מאחר ואנו מעדכנים את ערכו בכל איטרציה – התוקף תמיד רואה שיש גישה לשורת המטמון ואינו יכול להסיק מידע על הMSB.

```
int  zero_msbs_counter = 0 ;
void count_zero_msbs(char * data ,unsigned data_size)
    for (int i = 0; i < data_size ; i++){
        if ((data[i]>>7) == 0)
            zero_msbs_counter++;
        else
            zero_msbs_counter += 0;
    }
}
```

```
int  zero_msbs_counter = 0 ;
void count_zero_msbs(char * data ,unsigned data_size) {
    for (int i = 0; i < data_size ; i++)
        zero_msbs_counter += ((data[i]>>7) == 0);
}
```