**מעבדה RoP and stack attacks**

1. הורדתי את הקבצים הבאים מהmoodel :

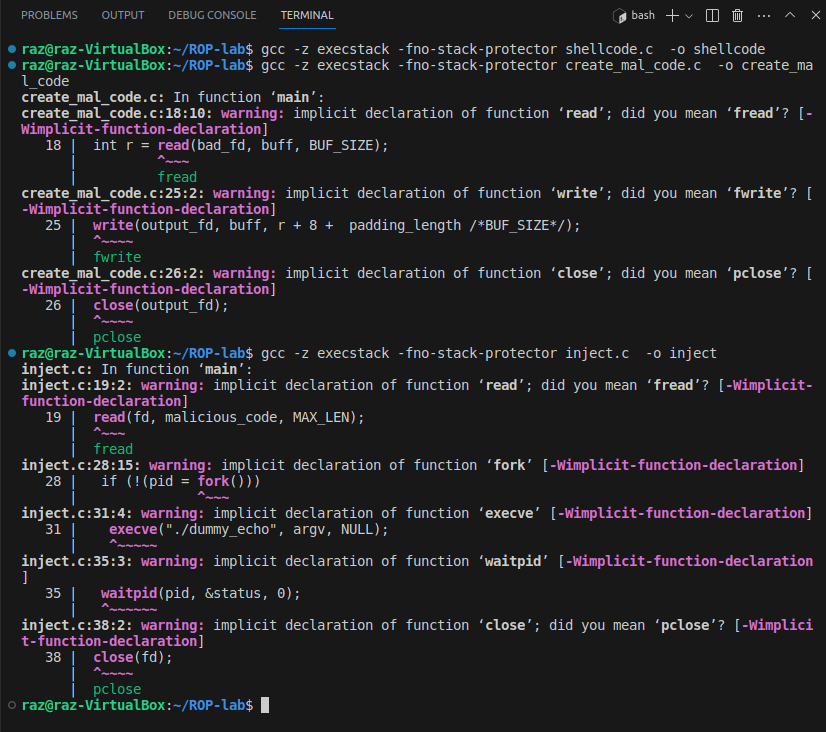
* dummy\_echo
* inject.c
* create\_mal\_code.c
* shellcode.c

קמפלתי את כל הקבצים כך:

gcc -z execstack -fno-stack-protector **shellcode.c** -o **shellcode**

gcc -z execstack -fno-stack-protector create\_mal\_code.c -o create\_mal\_code

gcc -z execstack -fno-stack-protector inject.c -o inject





1. הקלדנו:

hexedit shellcode  
והלכנו לפי הסרטון לחלק שמתחיל בUH. ראינו שהכתובת היא 0x112D תרגמנו למספר עשרוני וקיבלנו : **4397**

תמונה שמכילה טקסט, צילום מסך, חשמל, מחשב

התיאור נוצר באופן אוטומטי

1. dd if=shell skip=1632 count=36 bs=1 of=shellcode – **בסרטון**  
   **אצלנו:** לאחר שעשנו hexedit והוצאנו את הכתובת בסעיף (2)

dd if=shellcode skip=**4397** count=36 bs=1 of=shell\_code

36+0 records in

36+0 records out

36 bytes copied, 0.000368274 s, 97.8 kB/s

1. הקלדתי:  
   ls -l shell\_code.bin

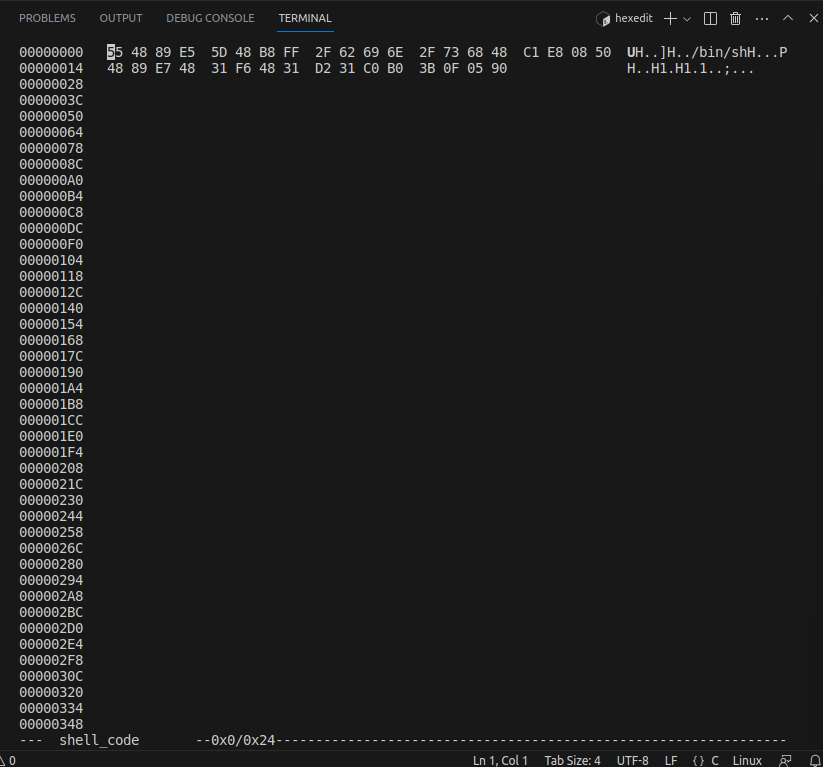
**Out:**

ls: cannot access 'shell\_code.bin': No such file or directory

1. hexedit shell\_code

יצר שגיאה לכן התקנתי את 6+7+8

1. sudo apt-get install python3-apt
2. sudo apt-get update --fix-missing
3. sudo apt install hexedit
4. hexedit shell\_code



Ctrl+x ליציאה וחזרה לטרמינל

1. הרצתי:

./create\_mal\_code

**Out:**

./prog assumed\_shellcode\_addr padding shellcode outputfile

1. הרצתי:

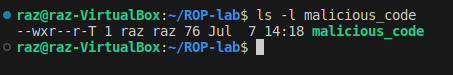
./create\_mal\_code 0 20 shell\_code malicious\_code

1. הרצתי:

ls -l malicious\_code

**Out:**

--wxr--r-T 1 raz raz 76 Jul 5 12:39 malicious\_code



14) chmod 666 malicious\_code

15) הרצתי:

ls -l malicious\_code

**Out:**

-rw-rw-rw- 1 raz raz 76 Jul 5 12:39 malicious\_code

תמונה שמכילה טקסט, גופן, צילום מסך

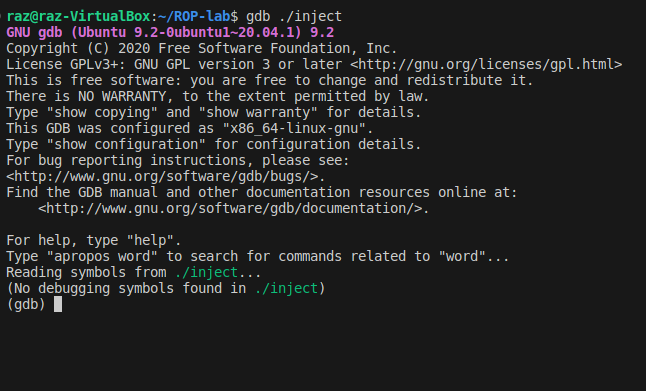
התיאור נוצר באופן אוטומטי

16) hexedit malicious\_code

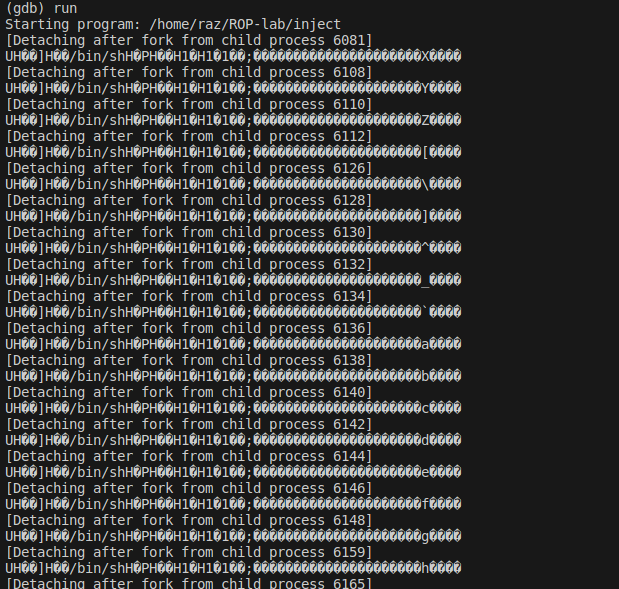
תמונה שמכילה צילום מסך, טקסט

התיאור נוצר באופן אוטומטי

18) gdb ./inject



19) רושמים run



**יש שגיאות לכן ניסינו לפתור כך:**

20) יצרנו קובץ victim\_code.c עם הקוד מהמצגת:

#include <stdio.h>

#include <string.h>

void copy\_and\_echo(const char\* p)

{

int length = strlen(p);

char buffer[length + 1]; // Add space for null terminator

int i, j = 0;

for (i = 0; i < length; i++)

{

if (p[i] >= 32 && p[i] <= 126)

{

buffer[j++] = p[i];

}

}

buffer[j] = '\0'; // Null terminate the buffer

printf("%s\n", buffer);

}

int main(int argc, const char\* argv[])

{

if (argc < 2)

return -1;

copy\_and\_echo(argv[1]);

return 0;

}

21) קימפלנו:

gcc -z execstack -fno-stack-protector victim\_code.c -o dummy\_echo

22)

./dummy\_echo

23) gdb ./inject

24) Run

25) רשמנו:

p $sp

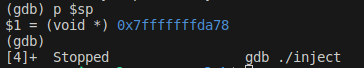
**Out:**

$1 = (void \*) 0x7fffffffda58

(gdb) Quit

(gdb)

[10]+ Stopped gdb ./inject



26) לקחנו את הכתובת, שמנו ב

Inject.c

27) קימפלנו שוב :

./i

28)

./inject

