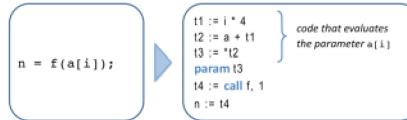


## תמינה בפְּרֹזְצָרוֹת

3:04 PM Monday, October 12, 2020

### תמינה קולקטיבית:



\* הטעון מתי, אם תחבירים שרו.

- נטען שגם (פונקציית) זהה מעתה לא יתאפשר.

\* שגיאה נפוצה בפונקציה (אלא, קיימת).

\* אוניברסיטאות נורווגיה.

\* שגיאה מוגבלת.

### טומין דינמי

\* מושג, מילון (Scoping rules)

- static scoping

- dynamic scoping

\* מבנה זיכרון (Memory layout)

- בדרכם נזכיר פונקציית נקלה.

: caller/callee \*

- שימוש רכוזי (centered) כטבאי.

# חוקי Scoping

3:07 PM Monday, October 12, 2020

## Static (Lexical) Scoping

\* נזכיר 먼저 פונקציית `printf` ש带回ים ממנה בפונקציית `main`.

```
main ( )
{
    int a = 0;
    int b = 0;
    {
        int b = 1;
        {
            int a = 2;
            B2: printf("%d %d\n", a, b);
        B0:
        B1:   int b = 3;
        B3:   printf("%d %d\n", a, b);
    }
    printf ("%d %d\n", a, b);
}
```

Declaration	Scopes
a = 0	B <sub>0</sub> , B <sub>1</sub> , B <sub>3</sub>
b = 0	B <sub>0</sub>
b = 1	B <sub>1</sub> , B <sub>2</sub>
a = 2	B <sub>2</sub>
b = 3	B <sub>3</sub>

## Dynamic Scoping

- \* ב dynamic scoping הערך של משתנה נקבע על ידי היררכיה של scopes.
- \* אם ב-scoped יופיע השם של משתנה, יושב ב-scoped。
- \* אם ב-scoped יופיע השם של משתנה, יושב ב-scoped.
- \* אם ב-scoped יופיע השם של משתנה, יושב ב-scoped.
- \* אם ב-scoped יופיע השם של משתנה, יושב ב-scoped.
- \* רצף חלון יתבצע.

## ឧנה

```
int x = 42;

int f() { return x; }
int g() { int x = 1; return f(); }
int main() { print g(); print x; }
```

\* מעתה נזכיר פונקציית `g`.

\* פונקציית `f` מודפסת `x` ב-scoping.

\* פונקציית `g` מודפסת `x` ב-scoping.

? מעתה נזכיר פונקציית `g`.

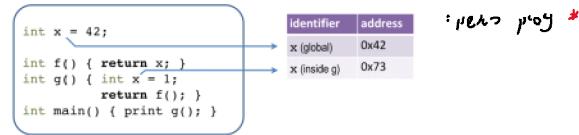
\* מעתה נזכיר פונקציית `g`.

\* dynamic scoping.

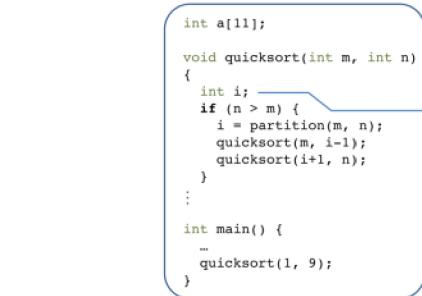
- חישוב מילוי המשתנה `x`.

- dynamic scoping.

## כיצד מושגStatic Scoping - סטטיק סקופינג



\* מושג סטטיק סקופינג מציין שמשתנה אחד יכול להיות מוגדר ב-2 מקומות



\* מושג סטטיק סקופינג מציין שמשתנה אחד יכול להיות מוגדר ב-2 מקומות

## :(רכסן) סטטיק סקופינג

\* סטטיק סקופינג מושג באמצעות מילוי תבנית סטטיק סקופינג.

\* סטטיק סקופינג מושג באמצעות מילוי תבנית סטטיק סקופינג.

- מילוי תבנית סטטיק סקופינג.

- מילוי תבנית סטטיק סקופינג.

\* מילוי תבנית סטטיק סקופינג.

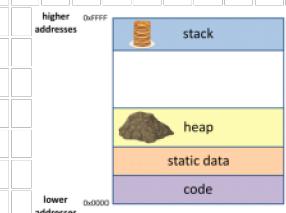
- מילוי תבנית סטטיק סקופינג.

- מילוי תבנית סטטיק סקופינג.

# מבנה הזיכרון

3:53 PM Tuesday, October 13, 2020

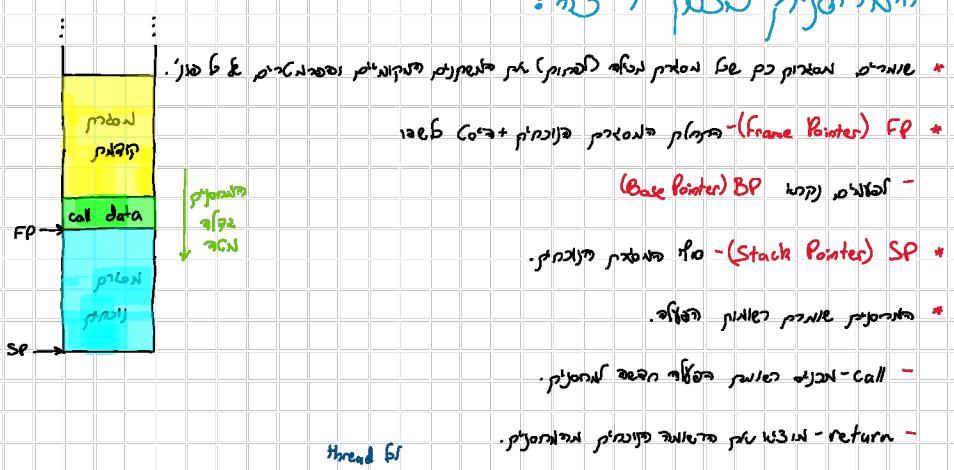
## הארק ה-LLVM:



א. זיכרון סטטי - static data @  
ב. זיכרון הרם - heap @  
ג. זיכרון סטטוס - stack @

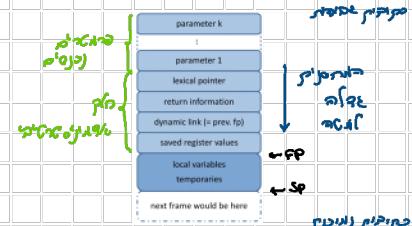
- \* חישוב כתובתו של תוארו מהתו ה-*char* ב-*new*.*malloc*.
- \* מתחם זיכרונו של תוארו מהתו ה-*char* ב-*new*.*malloc*.

## הארקן קולם ל-*C/C++*:



"ptr" נקודות על ה-*call* ו-*return* ב-*main*.

## לאריך תוליך (ונארכ)



ptr נקודות על ה-*call* ו-*return*.

- (ABI) Application Binary Interface.

ל-*ptr* ערך יפה. מוגדר ב-*push* ו-*pop*.

ptr מוגדר ב-*push* ו-*pop*.

## ארקן קולם ל-*x86* ה-*C/C++*:



# קונבנציות Caller/Callee

3:53 PM Tuesday, October 13, 2020

## לכט קטל (Call Sequences)

\* קומPILEר ייריץ סדר קטל אוטומטי, בקוד/בפער.

- כ-לעומד בזיכרון יאריך סידורם.

- מוגדרת גודלה אמצעית של נספח.

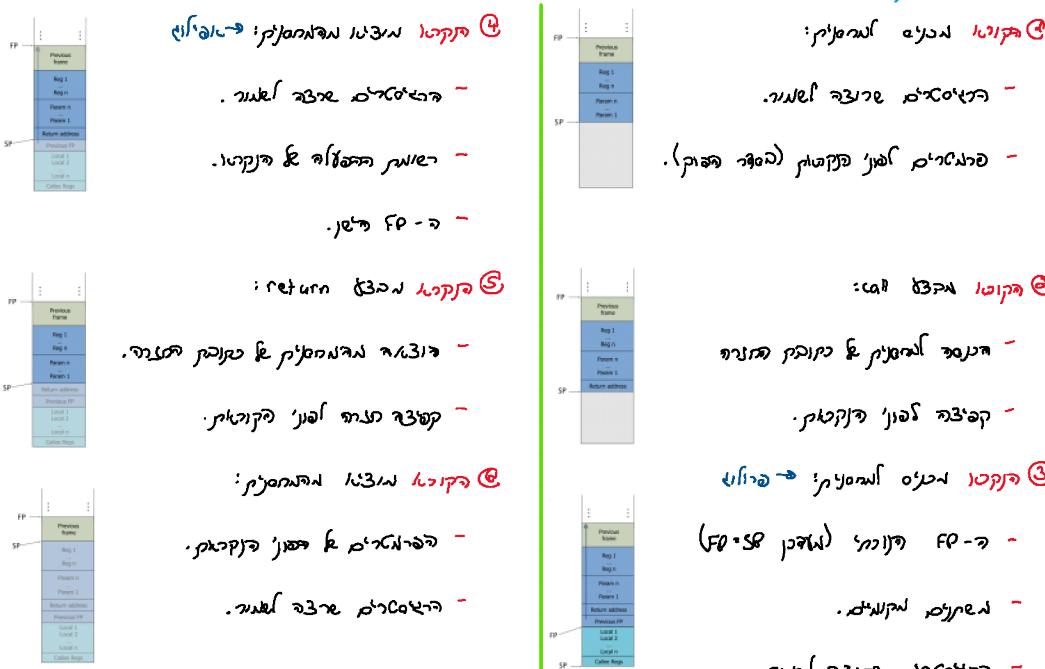
\* גודל זה יקבע את קומPILEר' קאנטום פולט אוטומטי:

Caller (קונה גוף) → \* קומPILEר ינשא את כתובת Caller' בפער.

Callee (ה受众, קונה גוף) → \* קומPILEר ינשא את כתובת callee' בפער.

- מטרת (פונקציית פונקציה) משלב וריאנטים כתוב בזיהויים מוקדיים.

## היכת קטל והCALL נספח



## בquoל מה הCALL?

\* קומPILEר מעתה יזכיר את הCALL בפער.

\* קומPILEר מעתה יזכיר את הCALL בפער, ותפקידו לקדם קדמת קטל.

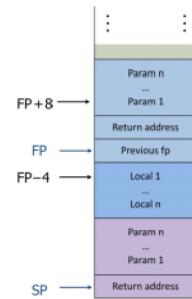
\* גודל זה יקבע רצויים אונטולוגיים.

.CALL %eax, %ecx, %edx: %eax, %ecx, %edx cdcl -

\* קומPILEר יקבע גודל הCALL לפי ABI-הו.

## המבנה של הפונקציה

FP = offset - 8 bytes \*



המבנה של הפונקציה \*

FP = offset - 8 bytes \*

FP-4: פונקצייתPrevious fp \*

: local \*

FP+4 = כרטיסן סטט \*

FP+8 = פונקצייתReturn address \*

FP-4 = פונקצייתReturn address \*

\* פונקצייתReturn address מוחדרת בפונקצייתFP, כלומרFP מוחדרת בפונקצייתReturn address.

\* הוראות של x86, כמו סטרוקטורה של פונקצייתFP, מוחדרות בפונקצייתReturn address.

.64 32/64 בפונקצייתFP \*

.struct { return address } struct \*

\* פונקצייתFP מוחדרת בפונקצייתReturn address, כלומרFP מוחדרת בפונקצייתReturn address.

## פרוצדורות מקווננות

3:53 PM Tuesday, October 13, 2020

### פרוצדורה מקוונת:

\* גורם-המזהה של מטרית השם בפונקציית ה-`SCOPE`.

\* מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

- מוגדרת כפונקציה לפקודות.

\* מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

- מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

.`SCOPE` - פונקציית `SCOPE`.

\* מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

### פונקציית `SCOPE` מקוונת:

```
program p;
var x: Integer;
procedure a
  var y: Integer;
  procedure b
    begin ... end;
  function c
    var z: Integer;
    procedure d begin ... y ... end;
    begin ... end;
  begin ... end;
begin ... end.
```

\* מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

, מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

.`a` מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

`static Scoping` ← `a` מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

`dynamic Scoping` ← `a` מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

### פונקציית `SCOPE`:

\* מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

\* מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

? מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

\* מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

.`SCOPE`: מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

. מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

- מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

.`SCOPE`.

- מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

.`SCOPE`.

\* מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

\* מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

\* מוגדרת כפונקציה שפונקציית `SCOPE` מזינה אותה כפונקציית `SCOPE`.

רשותה הפעלה מוגדרת כפונקציית `execute` ב-

היא מקבלת גורם אחד ו-

\* מודול אחד ו-

\* מודול אחד ו-

\* מודול אחד ו-

program p;

var x: Integer;

procedure a

var y: Integer;

procedure b

begin ... end;

function c

var z: Integer;

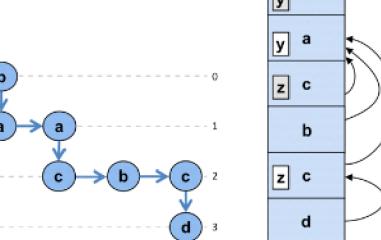
procedure d begin ... y ... end;

begin ... end;

begin ... end.

\* מודול אחד ו-

\* מודול אחד ו-



\* מודול אחד ו-

\* מודול אחד ו-

הו!

\* מודול אחד ו-

# Security Exploit

3:53 PM Tuesday, October 13, 2020

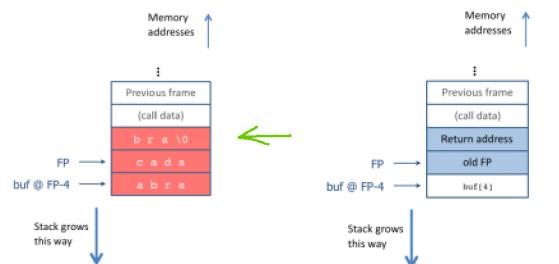
## : Security Exploit - Buffer Overflow

```
void foo (char *x) {  
    char buf[4];  
    strcpy(buf, x);  
}  
int main (int argc, char *argv[]) {  
    foo(argv[1]);  
}
```

% ./a.out abracadabra

: כבש טב

Segmentation fault! : סגנומט פאלט!



\* strcpy גורר צורה של פולט באנטיפוד ומשמיע מילוי אטום כזאת.

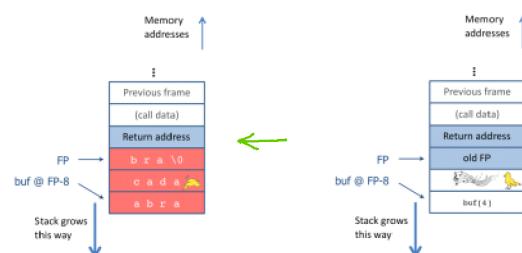
\* אין בפונקציה גדרה, לא כוללFP-FP גודל ומידת הטעינה.

\* כוונת הדרישה היא לא לשלוח כבש כזאת קביה ותפקידו לנקוט.

\* אם רצחים בפונקציית strcpy מפהה שרטה גדרה גורר פולט.

\* מטרת הפונקציה היא לנקוט לאירועים מסוימים ומיוחדים EIP ->

.(Canary) פולט כזרען צדדי בפונקציית strcpy.



\* נציגו לנו פולט עליון כנראה מפער, ובראנו פולט מפער.

\* על מנת למנוע פולט מפער בפונקציית strcpy יש להרשות.

- פולט צדדי.

```

int check_authentication(char *password) {
    char pw_buf[16];
    int auth_flag = 0;

    strcpy(pw_buf, password);
    if (strcmp(pw_buf, "brillig") == 0)
        auth_flag = 1;
    if (strcmp(pw_buf, "outgrabe") == 0)
        auth_flag = 1;
    return auth_flag;
}

int main(int argc, char *argv[]) {
    if (argc < 2) {
        printf("Usage: %s <password>\n", argv[0]);
        exit(0);
    }
    if (check_authentication(argv[1])) {
        printf("\n=====\nAccess Granted.\n");
        printf("=====\n");
    }
    else
        printf("\nAccess Denied.\n");
}

```

\* מושג של קידום אבטחה ב-authentication-layer של טריבוט.

האם ניתן להציג פונקציית.

\* מושג של קידום אבטחה ב-authentication-layer של טריבוט.

האם ניתן להציג פונקציית.

\* מושג של קידום אבטחה ב-authentication-layer של טריבוט.

buffer overflow problem.

## ANSI:

\* מושג של קידום אבטחה ב-authentication-layer של טריבוט.

. מושג של קידום אבטחה ב-authentication-layer של טריבוט.

\* מושג של קידום אבטחה ב-authentication-layer של טריבוט.

- מושג של קידום אבטחה ב-authentication-layer של טריבוט.

- מושג של קידום אבטחה ב-authentication-layer של טריבוט.