בס"ד

פתרון לתרגיל cheater – ישי לוטווק

אחרי שמצאתי את הפונקציה main ניסיתי להבין מה היא עושה

להלן הפונקציה main לאחר הוספת תוויות שעוזרות להבין מה הקוד פה עושה:

```
push    ebp
mov     ebp, esp
sub     esp, 8
mov     eax, ___security_cookie
xor     eax, ebp
mov     [ebp+var_4], eax

programer_code:
lea     eax, [ebp+var_8]
push    eax
push    offset unk_402100
call    scanf
add     esp, 8

assign_0_in_var_8:
push    eax
lea     eax, [ebp+var_8]
mov     dword ptr [eax], 0
pop     eax

compare_var_8_to_0Ah:
cmp     [ebp+var_8], 0Ah
mov     ecx, offset aBad ; "bad\n"
mov     eax, offset aGood ; "good\n"
cmovle  eax, ecx
push    eax
call    printf

compiler_code:
mov     ecx, [ebp+var_4]
add     esp, 4
xor     ecx, ebp
xor     eax, eax
call    @__security_check_cookie@4 ; __security_check_cookie(x)
mov     esp, ebp
```

מצאתי מה הפקודה cmove עושה:

Example; this copies `edx` to `ecx` if `eax` and `ebx` are equal:

```
cmp eax, ebx
cmove ecx, edx
```

This does the same as:

```
cmp eax, ebx
jne skip
  mov ecx, edx
skip:
```

נשים לב כי הפקודה אצלנו קצת שונה – cmovle ולא cmove. כלומר לא רק שווה אלא קטן שווה.

הבנתי שאם הערך המתקבל גדול מ0Ah (שהוא בעצם 10) אמור להתקבל good ואם הערך קטן או שווה ל0Ah אמור להתקבל bad.

אלא שבקטע שנמצא לאחר התווית assign_0_in_var_8 נדרס הערך שמשתמש מכניס באופן שלעולם לא נקבל good. זאת מכיוון שהקוד משווה בין הערך שנמצא בvar_8 לבין הערך 0Ah, וכיון ש0 לעולם לא יהיה גדול מ0Ah נקבל תמיד bad. לכן תיקנתי את הקוד באופן שהוא ידלג על הקטע הזה וכך קיבלתי את הקוד הבא:

```
; int __cdecl main(int argc, const char **argv, const char **envp)
main proc near

var_8= dword ptr -8
var_4= dword ptr -4
argc= dword ptr  8
argv= dword ptr  0Ch
envp= dword ptr  10h

push    ebp
mov     ebp, esp
sub     esp, 8
mov     eax, ___security_cookie
xor     eax, ebp
mov     [ebp+var_4], eax

programer_code:
lea     eax, [ebp+var_8]
push    eax
push    offset unk_402100
call    scanf
add     esp, 8

assign_0_in_var_8:
jmp     short compare_var_8_to_0Ah
```

```
compare_var_8_to_0Ah:
cmp     [ebp+var_8], 0Ah
mov     ecx, offset aBad ; "bad\n"
mov     eax, offset aGood ; "good\n"
cmovle  eax, ecx
push    eax
call    printf
```

דוגמת הרצה:

```
C:\Windows\System32\cmd.exe
6
bad

C:\assembly\staticAnalysis\Patching\cheaterEx>cheater_solution.exe
7
bad

C:\assembly\staticAnalysis\Patching\cheaterEx>cheater_solution.exe
8
bad

C:\assembly\staticAnalysis\Patching\cheaterEx>cheater_solution.exe
9
bad

C:\assembly\staticAnalysis\Patching\cheaterEx>cheater_solution.exe
0
bad

C:\assembly\staticAnalysis\Patching\cheaterEx>cheater_solution.exe
10
bad

C:\assembly\staticAnalysis\Patching\cheaterEx>cheater_solution.exe
11
good

C:\assembly\staticAnalysis\Patching\cheaterEx>cheater_solution.exe
12
good
```