Computer Security lab 1:

Name: Prince Choudhary
SID: 862254827

1) Ans: 250382

Steps:

- Run crackme0x00, enter any password, it would say invalid.
- Run in gdb mode go to disas and examine the lines.
- The output would be like the screenshot below. You can see the password is stored in the file which is “250382”
- Receive input via “%s”
- Input is not processed
- Password is compared at line 85
- Password is stored at PTR[esp+0x4]

```
Legend: code, data, rodata, value

Temporary breakpoint 1, 0x0804845b in main ()
gdb-peda$ disas
Dump of assembler code for function main:
   0x08048414 <+0>:     push   ebp
   0x08048415 <+1>:     mov    ebp,esp
   0x08048417 <+3>:     sub    esp,0x28
   0x0804841a <+6>:     and    esp,0xfffffff0
   0x0804841d <+9>:     mov    eax,0x0
   0x08048422 <+14>:    add    eax,0xf
   0x08048425 <+17>:    add    eax,0xf
   0x08048428 <+20>:    shr    eax,0x4
   0x0804842b <+23>:    shl    eax,0x4
   0x0804842e <+26>:    sub    esp,eax
   0x08048430 <+28>:    mov    DWORD PTR [esp],0x8048568
   0x08048437 <+35>:    call   0x8048340 <printf@plt>
   0x0804843c <+40>:    mov    DWORD PTR [esp],0x8048581
   0x08048443 <+47>:    call   0x8048340 <printf@plt>
   0x08048448 <+52>:    lea    eax,[ebp-0x18]
   0x0804844b <+55>:    mov    DWORD PTR [esp+0x4],eax
   0x0804844f <+59>:    mov    DWORD PTR [esp],0x804858c
   0x08048456 <+66>:    call   0x8048330 <scanf@plt>
=> 0x0804845b <+71>:    lea    eax,[ebp-0x18]
   0x0804845e <+74>:    mov    DWORD PTR [esp+0x4],0x804858f
   0x08048466 <+82>:    mov    DWORD PTR [esp],eax
   0x08048469 <+85>:    call   0x8048350 <strcmp@plt>
   0x0804846e <+90>:    test   eax,eax
   0x08048470 <+92>:    je     0x8048480 <main+108>
   0x08048472 <+94>:    mov    DWORD PTR [esp],0x8048596
   0x08048479 <+101>:   call   0x8048340 <printf@plt>
   0x0804847e <+106>:   jmp    0x804848c <main+120>
   0x08048480 <+108>:   mov    DWORD PTR [esp],0x80485a9
   0x08048487 <+115>:   call   0x8048340 <printf@plt>
   0x0804848c <+120>:   mov    eax,0x0
   0x08048491 <+125>:   leave
   0x08048492 <+126>:   ret
End of assembler dump.
gdb-peda$ x /s 0x804858c
0x804858c:      "%s"
gdb-peda$ x /s $ebp-0x18
0xffffd1c0:     'a' <repeats 24 times>
gdb-peda$ x /s 0x804858f
0x804858f:      "250382"
gdb-peda$ x /s 0x8048596
0x8048596:      "Invalid Password!\n"
gdb-peda$ x /s 0x80485a9
0x80485a9:      "Password OK :)\n"
gdb-peda$
```

```
_init
[01/15/22]seed@VM:~/.../lab1$ ./crackme0x00
IOLI Crackme Level 0x00
Password: 250382
Password OK :)
```

2) Ans: 5274

Steps:

- Run in gdb mode go to disas and examine the lines.
- Look for string corresponding to Password Ok, and trace back
- The hexadecimal value compared with is 0x149a
- Convert 149a from hexadecimal to decimal
- We have the value “5274”

- Try this value after running the crackme file.
- It works, hence we got the password.
- It takes input “%d”
- Compared at line 71

```
Legend: code, data, rodata, value

Temporary breakpoint 1, 0x0804842b in main ()
gdb-peda$ disas
Dump of assembler code for function main:
   0x080483e4 <+0>:     push   ebp
   0x080483e5 <+1>:     mov    ebp,esp
   0x080483e7 <+3>:     sub    esp,0x18
   0x080483ea <+6>:     and    esp,0xfffffff0
   0x080483ed <+9>:     mov    eax,0x0
   0x080483f2 <+14>:    add    eax,0xf
   0x080483f5 <+17>:    add    eax,0xf
   0x080483f8 <+20>:    shr    eax,0x4
   0x080483fb <+23>:    shl    eax,0x4
   0x080483fe <+26>:    sub    esp,eax
   0x08048400 <+28>:    mov    DWORD PTR [esp],0x8048528
   0x08048407 <+35>:    call   0x804831c <printf@plt>
   0x0804840c <+40>:    mov    DWORD PTR [esp],0x8048541
   0x08048413 <+47>:    call   0x804831c <printf@plt>
   0x08048418 <+52>:    lea    eax,[ebp-0x4]
   0x0804841b <+55>:    mov    DWORD PTR [esp+0x4],eax
   0x0804841f <+59>:    mov    DWORD PTR [esp],0x804854c
   0x08048426 <+66>:    call   0x804830c <scanf@plt>
=> 0x0804842b <+71>:    cmp    DWORD PTR [ebp-0x4],0x149a
   0x08048432 <+78>:    je     0x8048442 <main+94>
   0x08048434 <+80>:    mov    DWORD PTR [esp],0x804854f
   0x0804843b <+87>:    call   0x804831c <printf@plt>
   0x08048440 <+92>:    jmp    0x804844e <main+106>
   0x08048442 <+94>:    mov    DWORD PTR [esp],0x8048562
   0x08048449 <+101>:   call   0x804831c <printf@plt>
   0x0804844e <+106>:   mov    eax,0x0
   0x08048453 <+111>:   leave
   0x08048454 <+112>:   ret
End of assembler dump.
gdb-peda$ x /s 0x804854c
```

```
[01/16/22]seed@VM:~/.../lab1$ ./crackme0x01
IOLI Crackme Level 0x01
Password: 5274
Password OK :)
[01/16/22]seed@VM:~/.../lab1$
```

3) Ans: 338724

Steps:

- Run in gdb mode go to disas and examine the lines.
- Look for string corresponding to Password Ok, and traceback
- In this case (attached screenshot) we can see
- We can see we are comparing the eax value for the password
- If we go up and trace eax = eax + edx = 90 + 492 = 582
- Then we are doing eax * PTR [ebp-0x8]= 582 * 582 = 338724
- Finally value “338724” is getting compared.
- Receive input at “%d”
- Compared at line 106

```
Temporary breakpoint 1, 0x0804842b in main ()
gdb-peda$ disas
Dump of assembler code for function main:
   0x080483e4 <+0>:     push   ebp
   0x080483e5 <+1>:     mov    ebp,esp
   0x080483e7 <+3>:     sub    esp,0x18
   0x080483ea <+6>:     and    esp,0xfffffff0
   0x080483ed <+9>:     mov    eax,0x0
   0x080483f2 <+14>:    add    eax,0xf
   0x080483f5 <+17>:    add    eax,0xf
   0x080483f8 <+20>:    shr    eax,0x4
   0x080483fb <+23>:    shl    eax,0x4
   0x080483fe <+26>:    sub    esp,eax
   0x08048400 <+28>:    mov    DWORD PTR [esp],0x8048548
   0x08048407 <+35>:    call   0x804831c <printf@plt>
   0x0804840c <+40>:    mov    DWORD PTR [esp],0x8048561
   0x08048413 <+47>:    call   0x804831c <printf@plt>
   0x08048418 <+52>:    lea    eax,[ebp-0x4]
   0x0804841b <+55>:    mov    DWORD PTR [esp+0x4],eax
   0x0804841f <+59>:    mov    DWORD PTR [esp],0x804856c
   0x08048426 <+66>:    call   0x804830c <scanf@plt>
=> 0x0804842b <+71>:    mov    DWORD PTR [ebp-0x8],0x5a
   0x08048432 <+78>:    mov    DWORD PTR [ebp-0xc],0x1ec
   0x08048439 <+85>:    mov    edx,DWORD PTR [ebp-0xc]
   0x0804843c <+88>:    lea    eax,[ebp-0x8]
   0x0804843f <+91>:    add    DWORD PTR [eax],edx
   0x08048441 <+93>:    mov    eax,DWORD PTR [ebp-0x8]
   0x08048444 <+96>:    imul   eax,DWORD PTR [ebp-0x8]
   0x08048448 <+100>:   mov    DWORD PTR [ebp-0xc],eax
   0x0804844b <+103>:   mov    eax,DWORD PTR [ebp-0x4]
   0x0804844e <+106>:   cmp    eax,DWORD PTR [ebp-0xc]
   0x08048451 <+109>:   jne    0x8048461 <main+125>
   0x08048453 <+111>:   mov    DWORD PTR [esp],0x804856f
   0x0804845a <+118>:   call   0x804831c <printf@plt>
   0x0804845f <+123>:   jmp    0x804846d <main+137>
   0x08048461 <+125>:   mov    DWORD PTR [esp],0x804857f
   0x08048468 <+132>:   call   0x804831c <printf@plt>
   0x0804846d <+137>:   mov    eax,0x0
   0x08048472 <+142>:   leave
   0x08048473 <+143>:   ret
End of assembler dump.
gdb-peda$ x /s 0x804830c
0x804830c <scanf@plt>:  "\377%\004\240\004\bh\b"
gdb-peda$ x /s 0x804830c 0x804856c
A syntax error in expression, near `0x804856c'.
gdb-peda$ x /s  0x804856c
0x804856c:      "%d"
```

```
[01/16/22]seed@VM:~/.../lab1$ ./crackme0x02
IOLI Crackme Level 0x02
Password: 338724
Password OK :)
[01/16/22]seed@VM:~/.../lab1$
```

4) Ans: 338724

Steps:

- Run in gdb mode go to disas and examine the lines.
- Here also we are doing eax = eax + edx = 90 + 492 = 582
- Then we are doing eax * PTR [ebp-0x8] = 582 * 582 = 338724
- Hence finally value “338724” is getting compared.
- Taking input in “%d”
- Compared at line 9 of test()

```
Temporary breakpoint 1, 0x080484df in main ()
gdb-peda$ disas
Dump of assembler code for function main:
   0x08048498 <+0>:     push   ebp
   0x08048499 <+1>:     mov    ebp,esp
   0x0804849b <+3>:     sub    esp,0x18
   0x0804849e <+6>:     and    esp,0xfffffff0
   0x080484a1 <+9>:     mov    eax,0x0
   0x080484a6 <+14>:    add    eax,0xf
   0x080484a9 <+17>:    add    eax,0xf
   0x080484ac <+20>:    shr    eax,0x4
   0x080484af <+23>:    shl    eax,0x4
   0x080484b2 <+26>:    sub    esp,eax
   0x080484b4 <+28>:    mov    DWORD PTR [esp],0x8048610
   0x080484bb <+35>:    call   0x8048350 <printf@plt>
   0x080484c0 <+40>:    mov    DWORD PTR [esp],0x8048629
   0x080484c7 <+47>:    call   0x8048350 <printf@plt>
   0x080484cc <+52>:    lea    eax,[ebp-0x4]
   0x080484cf <+55>:    mov    DWORD PTR [esp+0x4],eax
   0x080484d3 <+59>:    mov    DWORD PTR [esp],0x8048634
   0x080484da <+66>:    call   0x8048330 <scanf@plt>
=> 0x080484df <+71>:    mov    DWORD PTR [ebp-0x8],0x5a
   0x080484e6 <+78>:    mov    DWORD PTR [ebp-0xc],0x1ec
   0x080484ed <+85>:    mov    edx,DWORD PTR [ebp-0xc]
   0x080484f0 <+88>:    lea    eax,[ebp-0x8]
   0x080484f3 <+91>:    add    DWORD PTR [eax],edx
   0x080484f5 <+93>:    mov    eax,DWORD PTR [ebp-0x8]
   0x080484f8 <+96>:    imul   eax,DWORD PTR [ebp-0x8]
   0x080484fc <+100>:   mov    DWORD PTR [ebp-0xc],eax
   0x080484ff <+103>:   mov    eax,DWORD PTR [ebp-0xc]
   0x08048502 <+106>:   mov    DWORD PTR [esp+0x4],eax
   0x08048506 <+110>:   mov    eax,DWORD PTR [ebp-0x4]
   0x08048509 <+113>:   mov    DWORD PTR [esp],eax
   0x0804850c <+116>:   call   0x804846e <test>
   0x08048511 <+121>:   mov    eax,0x0
   0x08048516 <+126>:   leave
   0x08048517 <+127>:   ret
End of assembler dump.
gdb-peda$ x /s 0x8048330
0x8048330 <scanf@plt>:  "\377%\004\240\004\bh\b"
gdb-peda$ x /s 0x8048634
0x8048634:      "%d"
```

```
Temporary breakpoint 2, 0x0804846e in test ()
gdb-peda$ disas
Dump of assembler code for function test:
=> 0x0804846e <+0>:     push   ebp
   0x0804846f <+1>:     mov    ebp,esp
   0x08048471 <+3>:     sub    esp,0x8
   0x08048474 <+6>:     mov    eax,DWORD PTR [ebp+0x8]
   0x08048477 <+9>:     cmp    eax,DWORD PTR [ebp+0xc]
   0x0804847a <+12>:    je     0x804848a <test+28>
   0x0804847c <+14>:    mov    DWORD PTR [esp],0x80485ec
   0x08048483 <+21>:    call   0x8048414 <shift>
   0x08048488 <+26>:    jmp    0x8048496 <test+40>
   0x0804848a <+28>:    mov    DWORD PTR [esp],0x80485fe
   0x08048491 <+35>:    call   0x8048414 <shift>
   0x08048496 <+40>:    leave
   0x08048497 <+41>:    ret
End of assembler dump.
```

```
gdb-peda$ q
[01/16/22]seed@VM:~/.../lab1$ ./crackme0x03
IOLI Crackme Level 0x03
Password: 338724
Password OK!!! :)
[01/16/22]seed@VM:~/.../lab1$
```

5) Ans: Digit sum = 15 eg 78

Steps:

- Run in gdb mode go to disas and examine the lines.
- In this the main function print a message and takes the password.
- It loops through the password and adds the numbers and compares whether its 15 or not.
- It take input as “%s”
- Comparison is made at line 82 of check()

```
Breakpoint 1, 0x08048553 in main ()
gdb-peda$ disas
Dump of assembler code for function main:
   0x08048509 <+0>:     push   ebp
   0x0804850a <+1>:     mov    ebp,esp
   0x0804850c <+3>:     sub    esp,0x88
   0x08048512 <+9>:     and    esp,0xfffffff0
   0x08048515 <+12>:    mov    eax,0x0
   0x0804851a <+17>:    add    eax,0xf
   0x0804851d <+20>:    add    eax,0xf
   0x08048520 <+23>:    shr    eax,0x4
   0x08048523 <+26>:    shl    eax,0x4
   0x08048526 <+29>:    sub    esp,eax
   0x08048528 <+31>:    mov    DWORD PTR [esp],0x804865e
   0x0804852f <+38>:    call   0x8048394 <printf@plt>
   0x08048534 <+43>:    mov    DWORD PTR [esp],0x8048677
   0x0804853b <+50>:    call   0x8048394 <printf@plt>
   0x08048540 <+55>:    lea    eax,[ebp-0x78]
   0x08048543 <+58>:    mov    DWORD PTR [esp+0x4],eax
   0x08048547 <+62>:    mov    DWORD PTR [esp],0x8048682
   0x0804854e <+69>:    call   0x8048374 <scanf@plt>
=> 0x08048553 <+74>:    lea    eax,[ebp-0x78]
   0x08048556 <+77>:    mov    DWORD PTR [esp],eax
   0x08048559 <+80>:    call   0x8048484 <check>
   0x0804855e <+85>:    mov    eax,0x0
   0x08048563 <+90>:    leave
   0x08048564 <+91>:    ret
End of assembler dump.
gdb-peda$ x /s 0x8048374
0x8048374 <scanf@plt>:  "\377%\004\240\004\bh\b"
gdb-peda$ x /s 0x8048682
0x8048682:      "%s"
gdb-peda$
```

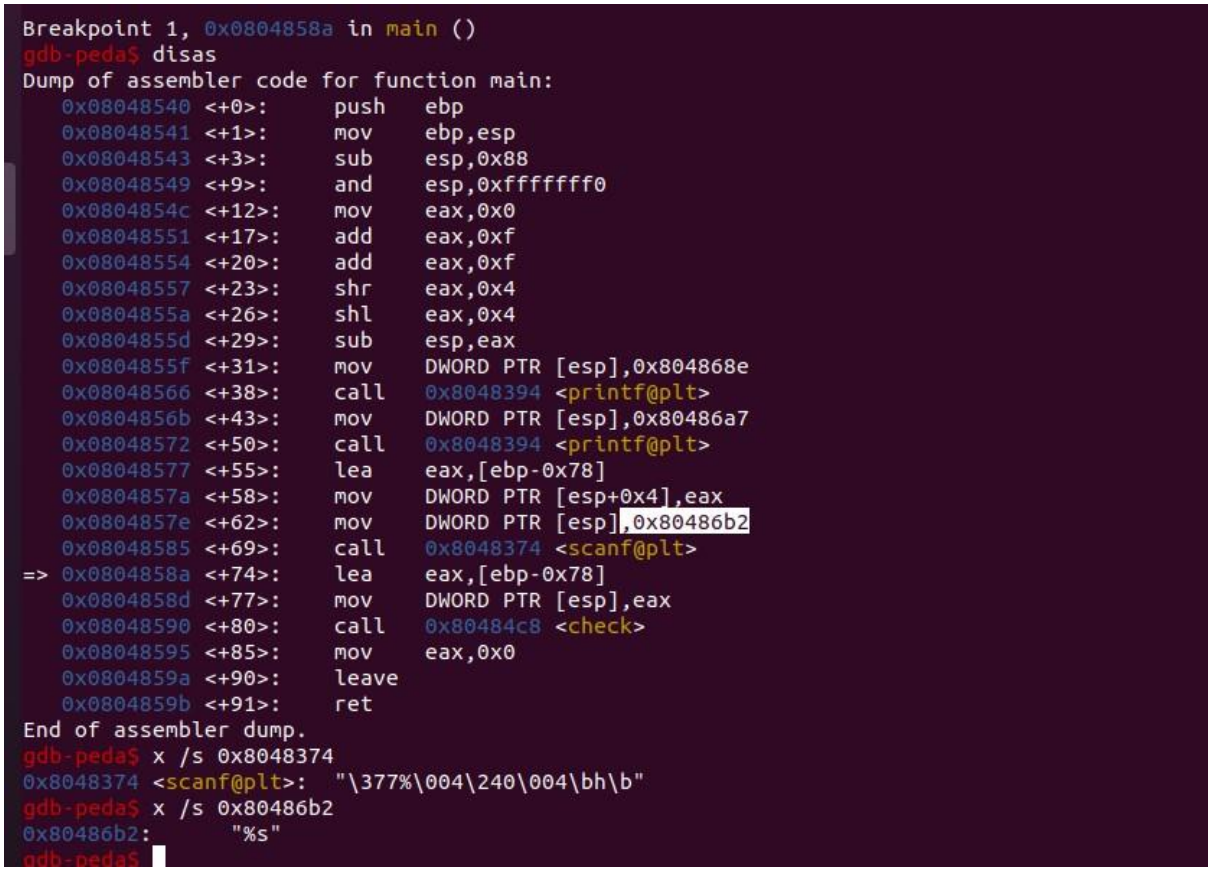
```
gdb-peda$ disas
Dump of assembler code for function check:
=> 0x08048484 <+0>:     push   ebp
   0x08048485 <+1>:     mov    ebp,esp
   0x08048487 <+3>:     sub    esp,0x28
   0x0804848a <+6>:     mov    DWORD PTR [ebp-0x8],0x0
   0x08048491 <+13>:    mov    DWORD PTR [ebp-0xc],0x0
   0x08048498 <+20>:    mov    eax,DWORD PTR [ebp+0x8]
   0x0804849b <+23>:    mov    DWORD PTR [esp],eax
   0x0804849e <+26>:    call   0x8048384 <strlen@plt>
   0x080484a3 <+31>:    cmp    DWORD PTR [ebp-0xc],eax
   0x080484a6 <+34>:    jae    0x80484fb <check+119>
   0x080484a8 <+36>:    mov    eax,DWORD PTR [ebp-0xc]
   0x080484ab <+39>:    add    eax,DWORD PTR [ebp+0x8]
   0x080484ae <+42>:    movzx  eax,BYTE PTR [eax]
   0x080484b1 <+45>:    mov    BYTE PTR [ebp-0xd],al
   0x080484b4 <+48>:    lea    eax,[ebp-0x4]
   0x080484b7 <+51>:    mov    DWORD PTR [esp+0x8],eax
   0x080484bb <+55>:    mov    DWORD PTR [esp+0x4],0x8048638
   0x080484c3 <+63>:    lea    eax,[ebp-0xd]
   0x080484c6 <+66>:    mov    DWORD PTR [esp],eax
   0x080484c9 <+69>:    call   0x80483a4 <sscanf@plt>
   0x080484ce <+74>:    mov    edx,DWORD PTR [ebp-0x4]
   0x080484d1 <+77>:    lea    eax,[ebp-0x8]
   0x080484d4 <+80>:    add    DWORD PTR [eax],edx
   0x080484d6 <+82>:    cmp    DWORD PTR [ebp-0x8],0xf
   0x080484da <+86>:    jne    0x80484f4 <check+112>
   0x080484dc <+88>:    mov    DWORD PTR [esp],0x804863b
   0x080484e3 <+95>:    call   0x8048394 <printf@plt>
   0x080484e8 <+100>:   mov    DWORD PTR [esp],0x0
   0x080484ef <+107>:   call   0x80483b4 <exit@plt>
   0x080484f4 <+112>:   lea    eax,[ebp-0xc]
   0x080484f7 <+115>:   inc    DWORD PTR [eax]
   0x080484f9 <+117>:   jmp    0x8048498 <check+20>
   0x080484fb <+119>:   mov    DWORD PTR [esp],0x8048649
   0x08048502 <+126>:   call   0x8048394 <printf@plt>
   0x08048507 <+131>:   leave
   0x08048508 <+132>:   ret
End of assembler dump.
gdb-peda$
```

```
gdb-peda$ q
[01/16/22]seed@VM:~/.../lab1$ ./crackme0x04
IOLI Crackme Level 0x04
Password: 96
Password OK!
[01/16/22]seed@VM:~/.../lab1$
```

6) Ans: Digit sum = 16 and even integers eg 88

Steps:

- Run in gdb mode go to disas and examine the lines.
- This is similar to the previous one, it takes password input from the user
- Compares the sum of it with 0x10 or 16
- Disassembly parallel is also called
- It would also consider to take even integers only
- If sum is 16 and is even integers then password is correct.
- Input is taken in “%s”
- Comparison is made at line 82 of check()


Breakpoint 1, 0x0804858a in main ()
gdb-peda$ disas
Dump of assembler code for function main:
0x08048540 <+0>: push ebp
0x08048541 <+1>: mov ebp,esp
0x08048543 <+3>: sub esp,0x88
0x08048549 <+9>: and esp,0xfffffff0
0x0804854c <+12>: mov eax,0x0
0x08048551 <+17>: add eax,0xf
0x08048554 <+20>: add eax,0xf
0x08048557 <+23>: shr eax,0x4
0x0804855a <+26>: shl eax,0x4
0x0804855d <+29>: sub esp,eax
0x0804855f <+31>: mov DWORD PTR [esp],0x804868e
0x08048566 <+38>: call 0x8048394 <printf@plt>
0x0804856b <+43>: mov DWORD PTR [esp],0x80486a7
0x08048572 <+50>: call 0x8048394 <printf@plt>
0x08048577 <+55>: lea eax,[ebp-0x78]
0x0804857a <+58>: mov DWORD PTR [esp+0x4],eax
0x0804857e <+62>: mov DWORD PTR [esp],0x80486b2
0x08048585 <+69>: call 0x8048374 <scanf@plt>
=> 0x0804858a <+74>: lea eax,[ebp-0x78]
0x0804858d <+77>: mov DWORD PTR [esp],eax
0x08048590 <+80>: call 0x80484c8 <check>
0x08048595 <+85>: mov eax,0x0
0x0804859a <+90>: leave
0x0804859b <+91>: ret
End of assembler dump.
gdb-peda$ x /s 0x8048374
0x8048374 <scanf@plt>: "\377%\004\240\004\bh\b"
gdb-peda$ x /s 0x80486b2
0x80486b2: "%s"
gdb-peda$

```
Breakpoint 2, 0x080484c8 in check ()
gdb-peda$ disas
Dump of assembler code for function check:
=> 0x080484c8 <+0>:     push   ebp
   0x080484c9 <+1>:     mov    ebp,esp
   0x080484cb <+3>:     sub    esp,0x28
   0x080484ce <+6>:     mov    DWORD PTR [ebp-0x8],0x0
   0x080484d5 <+13>:    mov    DWORD PTR [ebp-0xc],0x0
   0x080484dc <+20>:    mov    eax,DWORD PTR [ebp+0x8]
   0x080484df <+23>:    mov    DWORD PTR [esp],eax
   0x080484e2 <+26>:    call   0x8048384 <strlen@plt>
   0x080484e7 <+31>:    cmp    DWORD PTR [ebp-0xc],eax
   0x080484ea <+34>:    jae    0x8048532 <check+106>
   0x080484ec <+36>:    mov    eax,DWORD PTR [ebp-0xc]
   0x080484ef <+39>:    add    eax,DWORD PTR [ebp+0x8]
   0x080484f2 <+42>:    movzx  eax,BYTE PTR [eax]
   0x080484f5 <+45>:    mov    BYTE PTR [ebp-0xd],al
   0x080484f8 <+48>:    lea    eax,[ebp-0x4]
   0x080484fb <+51>:    mov    DWORD PTR [esp+0x8],eax
   0x080484ff <+55>:    mov    DWORD PTR [esp+0x4],0x8048668
   0x08048507 <+63>:    lea    eax,[ebp-0xd]
   0x0804850a <+66>:    mov    DWORD PTR [esp],eax
   0x0804850d <+69>:    call   0x80483a4 <sscanf@plt>
   0x08048512 <+74>:    mov    edx,DWORD PTR [ebp-0x4]
   0x08048515 <+77>:    lea    eax,[ebp-0x8]
   0x08048518 <+80>:    add    DWORD PTR [eax],edx
   0x0804851a <+82>:    cmp    DWORD PTR [ebp-0x8],0x10
   0x0804851e <+86>:    jne    0x804852b <check+99>
   0x08048520 <+88>:    mov    eax,DWORD PTR [ebp+0x8]
   0x08048523 <+91>:    mov    DWORD PTR [esp],eax
   0x08048526 <+94>:    call   0x8048484 <parell>
   0x0804852b <+99>:    lea    eax,[ebp-0xc]
   0x0804852e <+102>:   inc    DWORD PTR [eax]
   0x08048530 <+104>:   jmp    0x80484dc <check+20>
   0x08048532 <+106>:   mov    DWORD PTR [esp],0x8048679
   0x08048539 <+113>:   call   0x8048394 <printf@plt>
   0x0804853e <+118>:   leave
   0x0804853f <+119>:   ret
End of assembler dump.
gdb-peda$
```

```
[01/16/22]seed@VM:~/.../lab1$ ./crackme0x05
IOLI Crackme Level 0x05
Password: 88
Password OK!
[01/16/22]seed@VM:~/.../lab1$
```