March 27 Lab

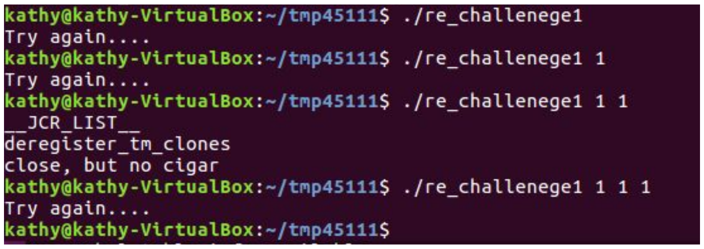
< Level 1 >

Goal

Print “You did it !”

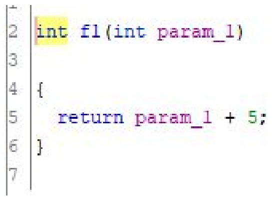
Steps

Run the file

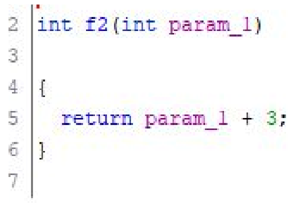


```
kathy@kathy-VirtualBox:~/tmp45111$ ./re_challenege1
Try again....
kathy@kathy-VirtualBox:~/tmp45111$ ./re_challenege1 1
Try again....
kathy@kathy-VirtualBox:~/tmp45111$ ./re_challenege1 1 1
__JCR_LIST__
deregister_tm_clones
close, but no cigar
kathy@kathy-VirtualBox:~/tmp45111$ ./re_challenege1 1 1 1
Try again....
kathy@kathy-VirtualBox:~/tmp45111$
```

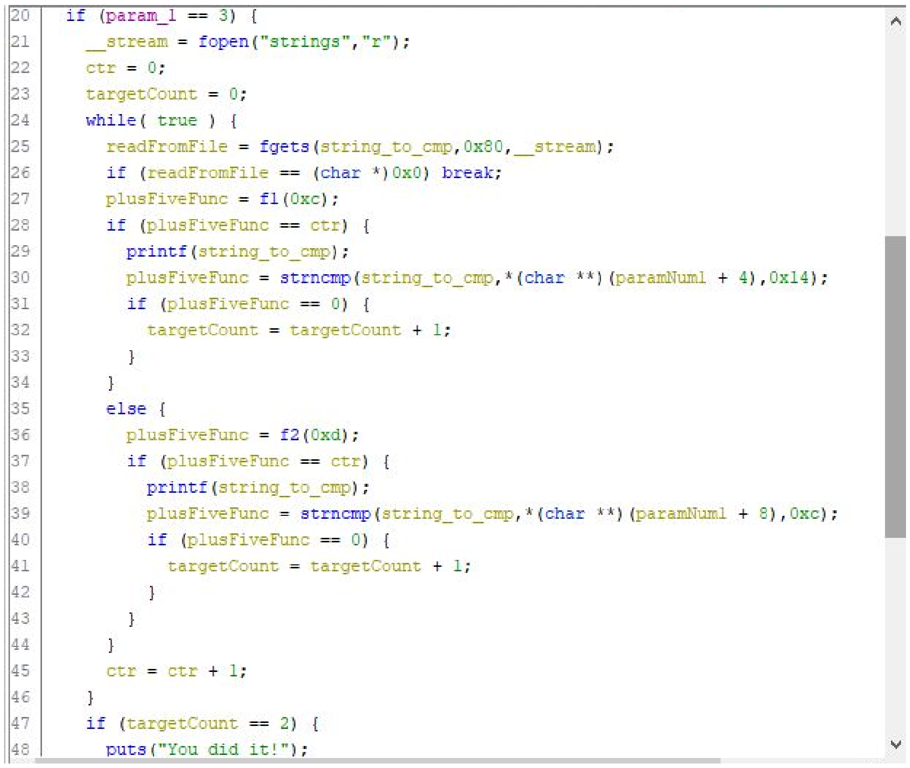
Open Ghidra and navigate to main, also check the function used in main. Change variable name to help analyze.



```
int f1(int param_1)

{
  return param_1 + 5;
}
```

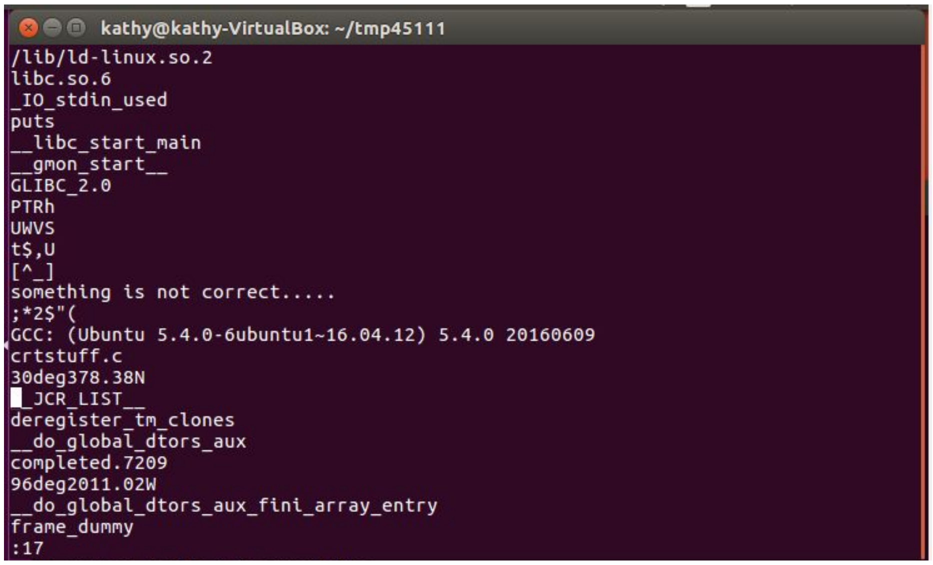


```
int f2(int param_1)

{
  return param_1 + 3;
}
```



```
  if (param_1 == 3) {
    __stream = fopen("strings","r");
    ctr = 0;
    targetCount = 0;
    while( true ) {
      readFromFile = fgets(string_to_cmp,0x80,__stream);
      if (readFromFile == (char *)0x0) break;
      plusFiveFunc = f1(0xc);
      if (plusFiveFunc == ctr) {
        printf(string_to_cmp);
        plusFiveFunc = strncmp(string_to_cmp,*(char **)(paramNum1 + 4),0x14);
        if (plusFiveFunc == 0) {
          targetCount = targetCount + 1;
        }
      }
      else {
        plusFiveFunc = f2(0xd);
        if (plusFiveFunc == ctr) {
          printf(string_to_cmp);
          plusFiveFunc = strncmp(string_to_cmp,*(char **)(paramNum1 + 8),0xc);
          if (plusFiveFunc == 0) {
            targetCount = targetCount + 1;
          }
        }
      }
      ctr = ctr + 1;
    }
    if (targetCount == 2) {
      puts("You did it!");
```

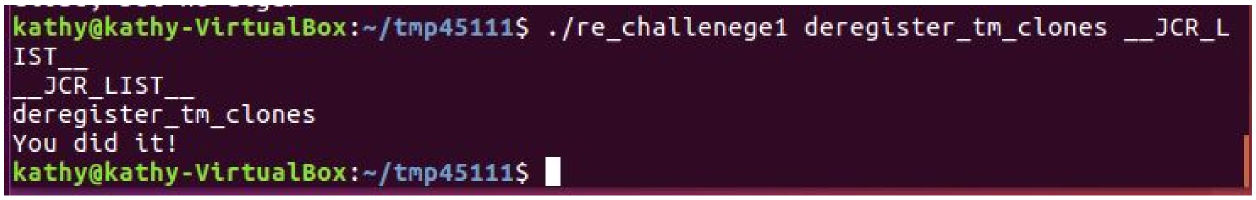
So we can see that there is a loop iterating the lines from the “strings” file, f1 function will result in 17 (adding 5 to 12) and f2 function will result in 16 (adding 3 to 13). When the loop counter hits these to value, strncmp will be called and targetCount will be incremented if user input matches the string to compare.



```
kathy@kathy-VirtualBox: ~/tmp45111
/lib/ld-linux.so.2
libc.so.6
_IO_stdin_used
puts
__libc_start_main
__gmon_start__
GLIBC_2.0
PTRh
UWVS
t$,U
[^_]
something is not correct.....
;*2$"(
GCC: (Ubuntu 5.4.0-6ubuntu1~16.04.12) 5.4.0 20160609
crtstuff.c
30deg378.38N
█_JCR_LIST__
deregister_tm_clones
__do_global_dtors_aux
completed.7209
96deg2011.02W
__do_global_dtors_aux_fini_array_entry
frame_dummy
:17
```

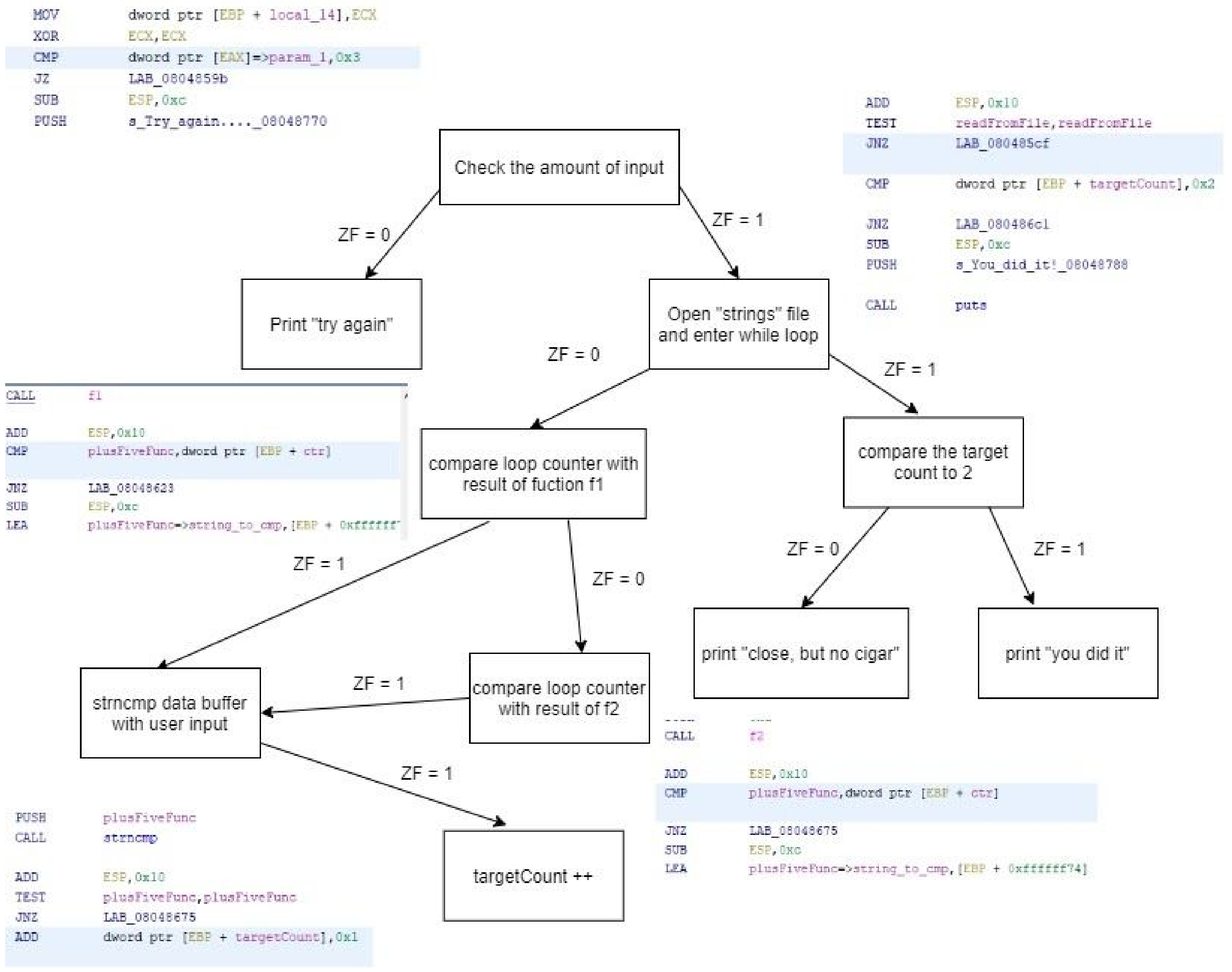
We will then open the “strings” file, and find line 17 and 16 and use them as input. Because of the right to left argument convention, The first input will need to be line 17 and then second input is line 16. (assume first line of file starting at index 0)

Answer



```
kathy@kathy-VirtualBox:~/tmp45111$ ./re_challenege1 deregister_tm_clones __JCR_L
IST__
__JCR_LIST__
deregister_tm_clones
You did it!
kathy@kathy-VirtualBox:~/tmp45111$ █
```

CFD


Check the amount of input
ZF = 0
ZF = 1
Print "try again"
Open "strings" file and enter while loop
ZF = 0
ZF = 1
compare loop counter with result of fuction f1
compare the target count to 2
ZF = 1
ZF = 0
ZF = 0
ZF = 1
strncmp data buffer with user input
ZF = 1
compare loop counter with result of f2
print "close, but no cigar"
print "you did it"
ZF = 1
targetCount ++

```
MOV         dword ptr [EBP + local_14],ECX
XOR         ECX,ECX
CMP         dword ptr [EAX]=>param_1,0x3
JZ          LAB_0804859b
SUB         ESP,0xc
PUSH        s_Try_again...._08048770
```

```
ADD         ESP,0x10
TEST        readFromFile,readFromFile
JNZ         LAB_080485cf

CMP         dword ptr [EBP + targetCount],0x2

JNZ         LAB_080486c1
SUB         ESP,0xc
PUSH        s_You_did_it!_08048788

CALL        puts
```

```
CALL        f1

ADD         ESP,0x10
CMP         plusFiveFunc,dword ptr [EBP + ctr]

JNZ         LAB_08048623
SUB         ESP,0xc
LEA         plusFiveFunc=>string_to_cmp,[EBP + 0xffffff
```

```
----        ----
CALL        f2

ADD         ESP,0x10
CMP         plusFiveFunc,dword ptr [EBP + ctr]

JNZ         LAB_08048675
SUB         ESP,0xc
LEA         plusFiveFunc=>string_to_cmp,[EBP + 0xffffff74]
```

```
PUSH        plusFiveFunc
CALL        strncmp

ADD         ESP,0x10
TEST        plusFiveFunc,plusFiveFunc
JNZ         LAB_08048675
ADD         dword ptr [EBP + targetCount],0x1
```

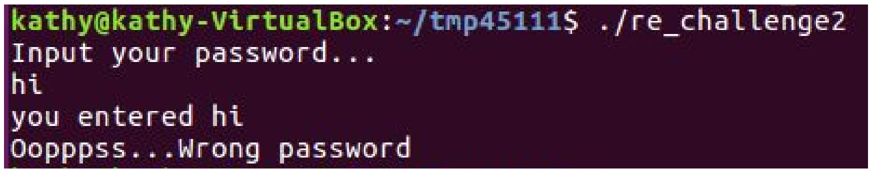
< Level 2 >

Goal

Print “You are great at this :) ”

Steps

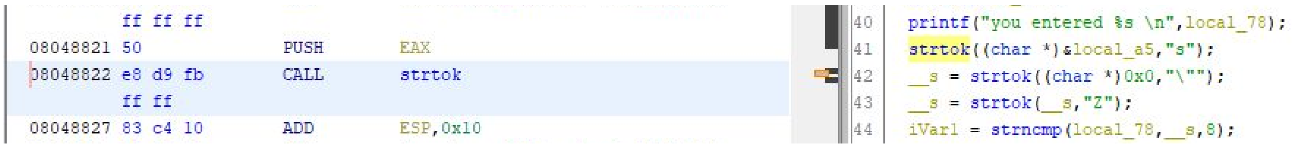
Run the file



```
kathy@kathy-VirtualBox:~/tmp45111$ ./re_challenge2
Input your password...
hi
you entered hi
Oopppss...Wrong password
```

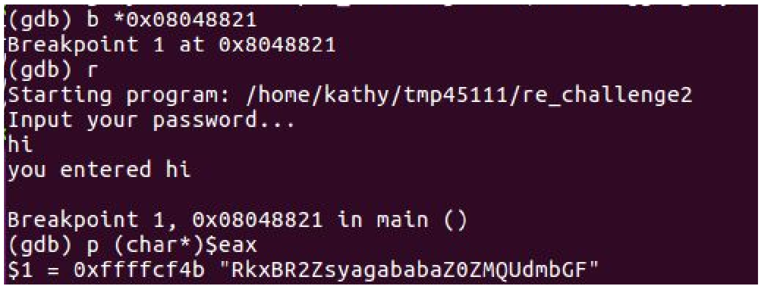
Find places to set breakpoints.

The first one is the address before the first strtok, so we can get local_a5 at EAX.



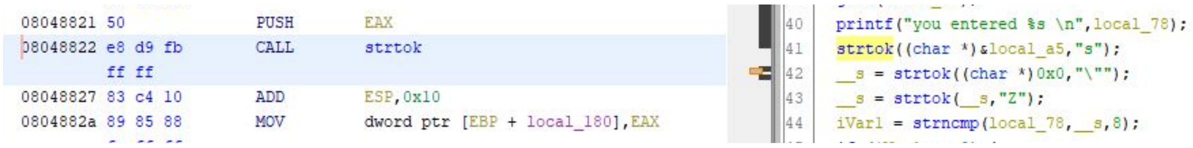
```
          ff ff ff
08048821 50           PUSH      EAX
08048822 e8 d9 fb     CALL      strtok
          ff ff
08048827 83 c4 10     ADD       ESP,0x10
```

```
printf("you entered %s \n",local_78);
strtok((char *)&local_a5,"s");
__s = strtok((char *)0x0,"\"");
__s = strtok(__s,"Z");
iVar1 = strncmp(local_78,__s,8);
```



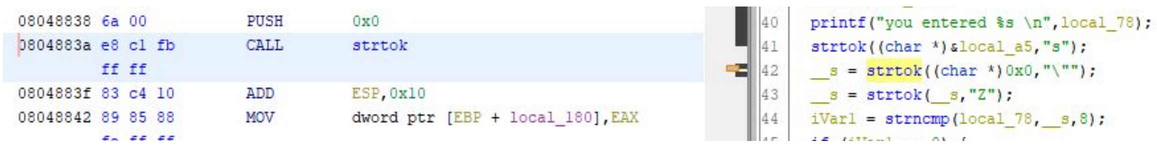
```
(gdb) b *0x08048821
Breakpoint 1 at 0x8048821
(gdb) r
Starting program: /home/kathy/tmp45111/re_challenge2
Input your password...
hi
you entered hi

Breakpoint 1, 0x08048821 in main ()
(gdb) p (char*)$eax
$1 = 0xffffcf4b "RkxBR2ZsyagababaZ0ZMQUdmbGF"
```

Then find the result string after the next 3 strtok. We will use all the addresses right after calling strtok, because the result of strtok is stored in EAX. (address ending in 27, 3f, 5b)



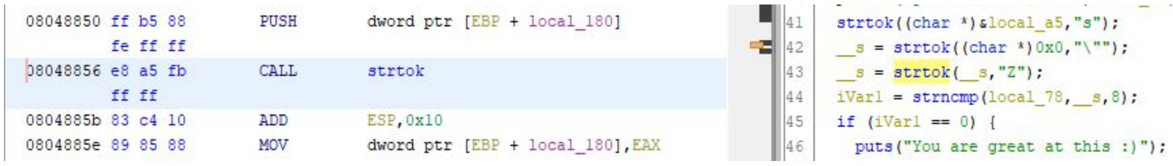
```
08048821 50           PUSH      EAX
08048822 e8 d9 fb     CALL      strtok
          ff ff
08048827 83 c4 10     ADD       ESP,0x10
0804882a 89 85 88     MOV       dword ptr [EBP + local_180],EAX
```

```
printf("you entered %s \n",local_78);
strtok((char *)&local_a5,"s");
__s = strtok((char *)0x0,"\"");
__s = strtok(__s,"Z");
iVar1 = strncmp(local_78,__s,8);
```



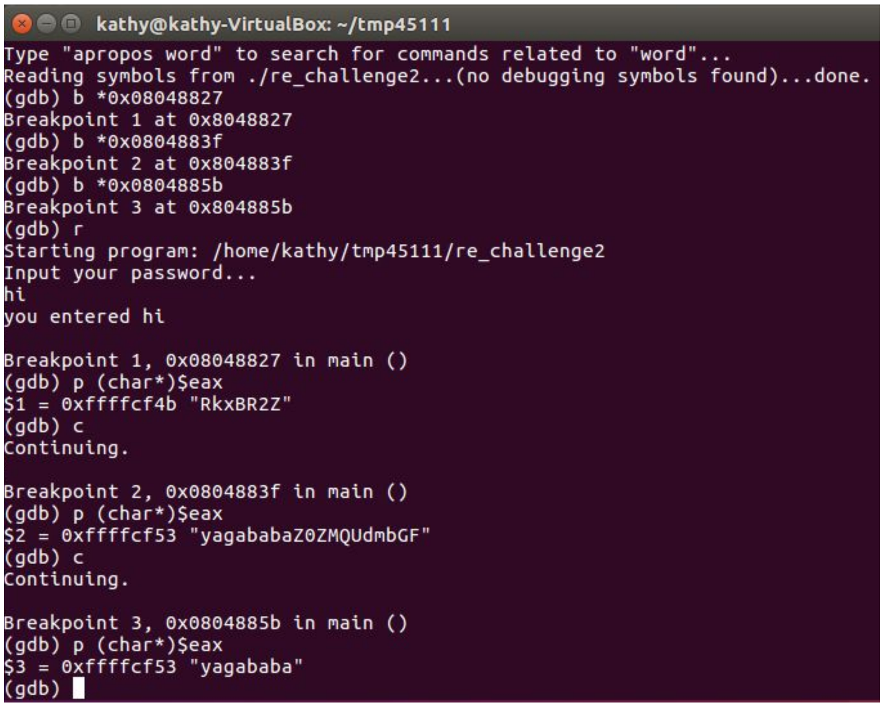
```
08048838 6a 00        PUSH      0x0
0804883a e8 c1 fb     CALL      strtok
          ff ff
0804883f 83 c4 10     ADD       ESP,0x10
08048842 89 85 88     MOV       dword ptr [EBP + local_180],EAX
```

```
printf("you entered %s \n",local_78);
strtok((char *)&local_a5,"s");
__s = strtok((char *)0x0,"\"");
__s = strtok(__s,"Z");
iVar1 = strncmp(local_78,__s,8);
```



```
08048850 ff b5 88     PUSH      dword ptr [EBP + local_180]
          fe ff ff
08048856 e8 a5 fb     CALL      strtok
          ff ff
0804885b 83 c4 10     ADD       ESP,0x10
0804885e 89 85 88     MOV       dword ptr [EBP + local_180],EAX
```

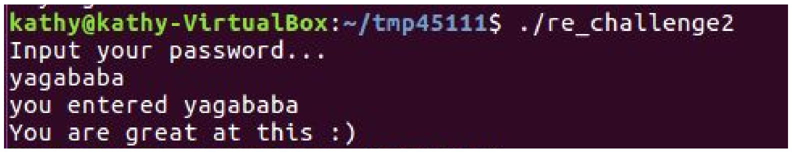
```
strtok((char *)&local_a5,"s");
__s = strtok((char *)0x0,"\"");
__s = strtok(__s,"Z");
iVar1 = strncmp(local_78,__s,8);
if (iVar1 == 0) {
  puts("You are great at this :)");
```



```
kathy@kathy-VirtualBox: ~/tmp45111
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./re_challenge2...(no debugging symbols found)...done.
(gdb) b *0x08048827
Breakpoint 1 at 0x8048827
(gdb) b *0x0804883f
Breakpoint 2 at 0x804883f
(gdb) b *0x0804885b
Breakpoint 3 at 0x804885b
(gdb) r
Starting program: /home/kathy/tmp45111/re_challenge2
Input your password...
hi
you entered hi

Breakpoint 1, 0x08048827 in main ()
(gdb) p (char*)$eax
$1 = 0xffffcf4b "RkxBR2Z"
(gdb) c
Continuing.

Breakpoint 2, 0x0804883f in main ()
(gdb) p (char*)$eax
$2 = 0xffffcf53 "yagababaZ0ZMQUdmbGF"
(gdb) c
Continuing.

Breakpoint 3, 0x0804885b in main ()
(gdb) p (char*)$eax
$3 = 0xffffcf53 "yagababa"
(gdb)
```

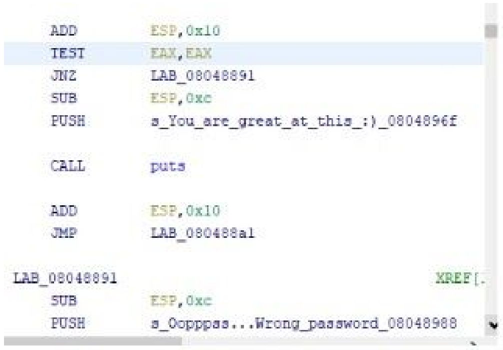
So from gdb, we can see that the first strtok cut the string at “s”. The second one is interesting, it gets the second half of the string after “s”. The third one cut the string at “Z”. Then we can see that this final string is used to compare with the user input.

Answer

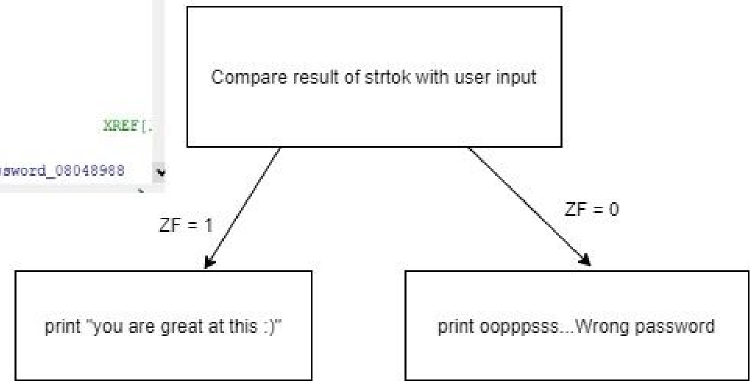


```
kathy@kathy-VirtualBox:~/tmp45111$ ./re_challenge2
Input your password...
yagababa
you entered yagababa
You are great at this :)
```

CFD



```
    ADD         ESP,0x10
    TEST        EAX,EAX
    JNZ         LAB_08048891
    SUB         ESP,0xc
    PUSH        s_You_are_great_at_this_:)_0804896f

    CALL        puts

    ADD         ESP,0x10
    JMP         LAB_080488a1

LAB_08048891                                          XREF[.
    SUB         ESP,0xc
    PUSH        s_Oopppss...Wrong_password_08048988
```


Compare result of strtok with user input
ZF = 1
ZF = 0
print "you are great at this :)"
print oopppsss...Wrong password

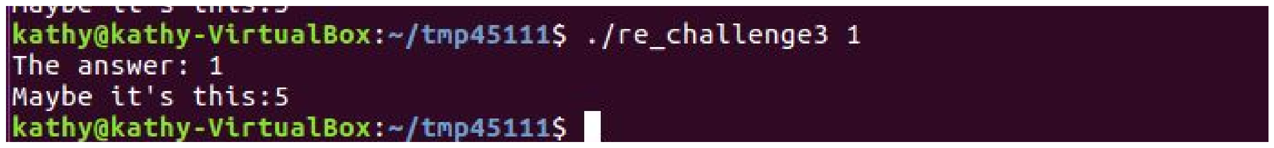
< Level 3 >

Goal

Print “You are amazing!!”

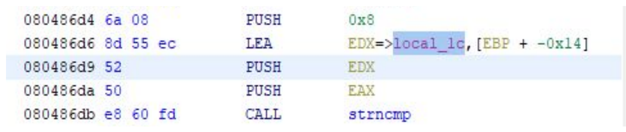
Steps

Run the file

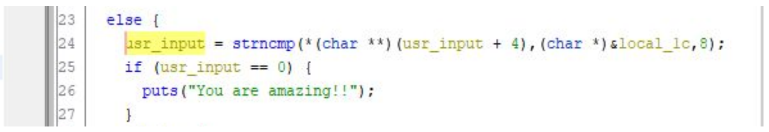


```
kathy@kathy-VirtualBox:~/tmp45111$ ./re_challenge3 1
The answer: 1
Maybe it's this:5
kathy@kathy-VirtualBox:~/tmp45111$ 
```

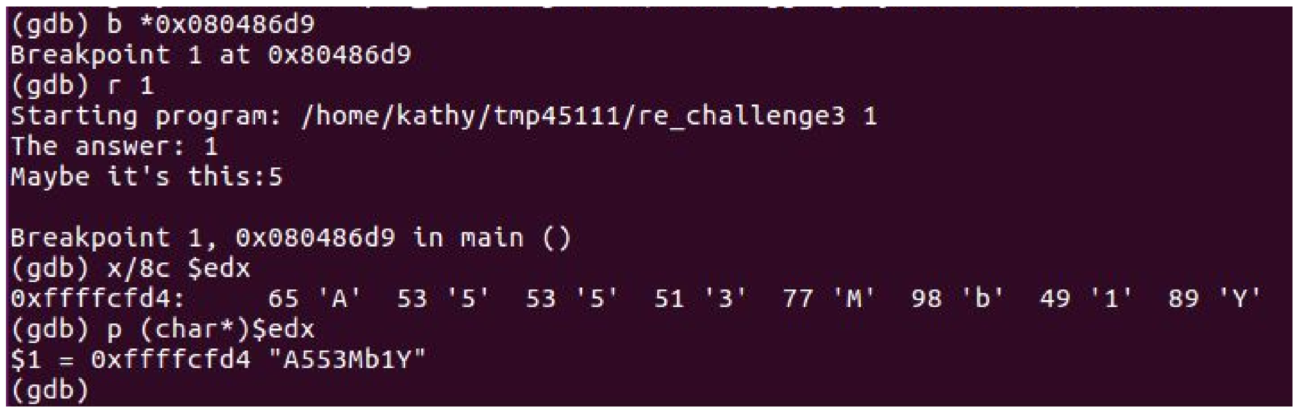
Find the place to set breakpoint, which is after local_1c gets save into EDX



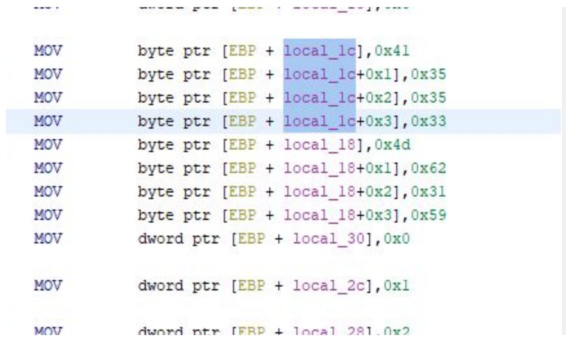
```
080486d4 6a 08          PUSH     0x8
080486d6 8d 55 ec       LEA      EDX=>local_1c,[EBP + -0x14]
080486d9 52             PUSH     EDX
080486da 50             PUSH     EAX
080486db e8 60 fd       CALL     strncmp
```



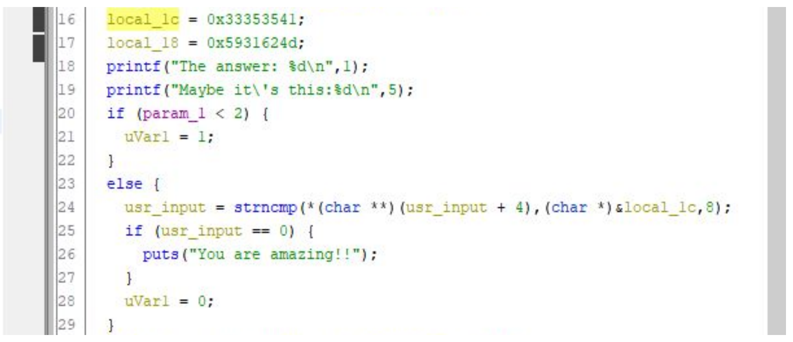
```
else {
  usr_input = strncmp(*(char **)(usr_input + 4),(char *)&local_1c,8);
  if (usr_input == 0) {
    puts("You are amazing!!");
  }
```



```
(gdb) b *0x080486d9
Breakpoint 1 at 0x80486d9
(gdb) r 1
Starting program: /home/kathy/tmp45111/re_challenge3 1
The answer: 1
Maybe it's this:5

Breakpoint 1, 0x080486d9 in main ()
(gdb) x/8c $edx
0xffffcfd4:     65 'A'  53 '5'  53 '5'  51 '3'  77 'M'  98 'b'  49 '1'  89 'Y'
(gdb) p (char*)$edx
$1 = 0xffffcfd4 "A553Mb1Y"
(gdb)
```



```
MOV       byte ptr [EBP + local_1c],0x41
MOV       byte ptr [EBP + local_1c+0x1],0x35
MOV       byte ptr [EBP + local_1c+0x2],0x35
MOV       byte ptr [EBP + local_1c+0x3],0x33
MOV       byte ptr [EBP + local_18],0x4d
MOV       byte ptr [EBP + local_18+0x1],0x62
MOV       byte ptr [EBP + local_18+0x2],0x31
MOV       byte ptr [EBP + local_18+0x3],0x59
MOV       dword ptr [EBP + local_30],0x0

MOV       dword ptr [EBP + local_2c],0x1
```

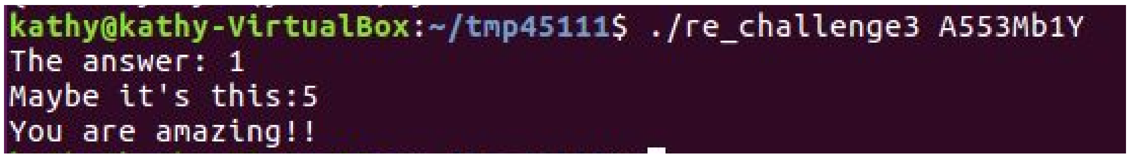


```
local_1c = 0x33353541;
local_18 = 0x5931624d;
printf("The answer: %d\n",1);
printf("Maybe it\'s this:%d\n",5);
if (param_1 < 2) {
  uVar1 = 1;
}
else {
  usr_input = strncmp(*(char **)(usr_input + 4),(char *)&local_1c,8);
  if (usr_input == 0) {
    puts("You are amazing!!");
  }
  uVar1 = 0;
}
```

0x41 = 65, 0x35 = 53, 0x33 = 51, 0x4d = 77, 0x62 = 98, 0x31 = 49, 0x59 = 89

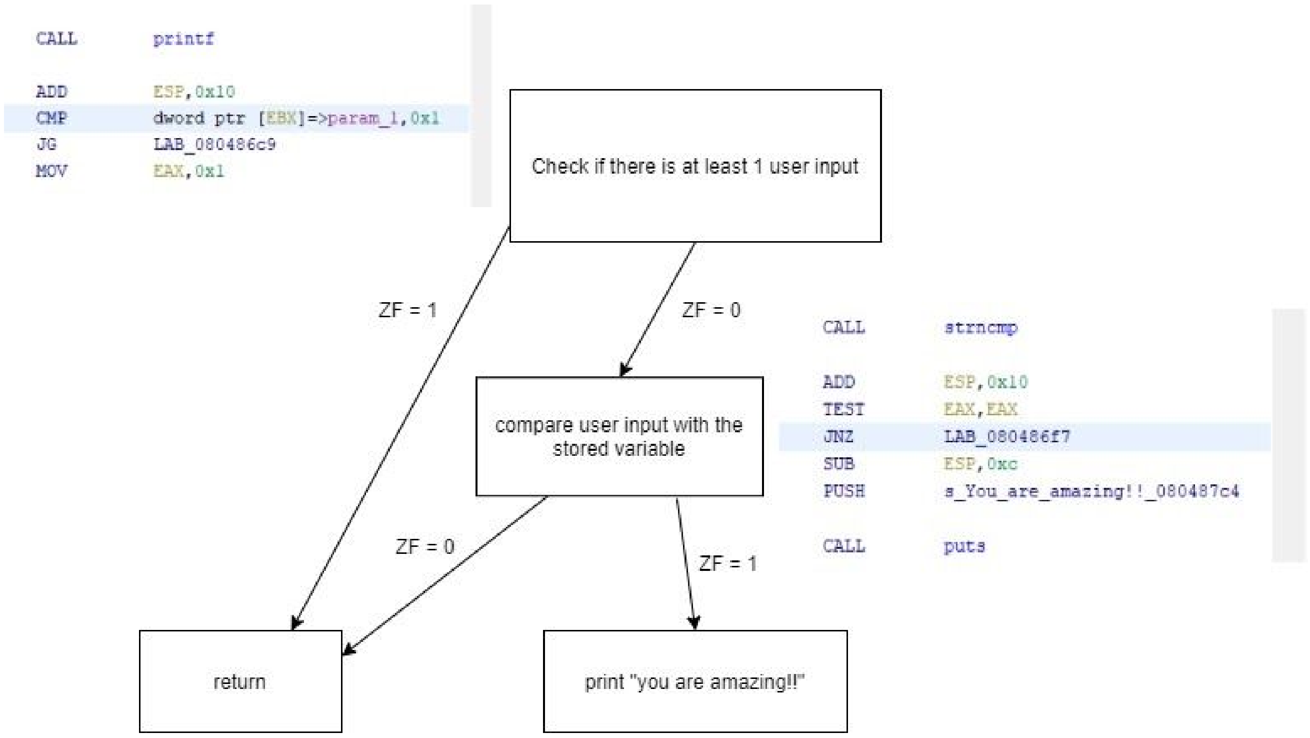
Because this is a 32 bit file, displacement should be 4 bytes. However, it is only use to store 1 byte (+0x1, +0x2, +0x3), so the bytes store in local_18 (also storing 1 byte at a time) get written into the last 4 bytes of local_1c.

Answer



```
kathy@kathy-VirtualBox:~/tmp45111$ ./re_challenge3 A553Mb1Y
The answer: 1
Maybe it's this:5
You are amazing!!
```

CFD



```
CALL        printf

ADD         ESP,0x10
CMP         dword ptr [EBX]=>param_1,0x1
JG          LAB_080486c9
MOV         EAX,0x1
```

```
CALL        strncmp

ADD         ESP,0x10
TEST        EAX,EAX
JNZ         LAB_080486f7
SUB         ESP,0xc
PUSH        s_You_are_amazing!!_080487c4

CALL        puts
```

Check if there is at least 1 user input
ZF = 1
ZF = 0
compare user input with the stored variable
ZF = 0
ZF = 1
return
print "you are amazing!!"