

# PSP0201

## Week 5

## Writeup

Group Name: Undecided

Members

ID	Name	Role
1211101390	Aslamia Najwa Binti Ahmad Khadri	Leader
1211100431	Mohammad Omar Torofder	Member
1211103388	Vishnu Karmegam	Member
1211103092	Farryn Aisha binti Muhd Firdaus	Member

## Day 17: Reverse Engineering – ReverseELFneering

**Tools used:** Attackbox, Firefox, Virtualbox

**Solution/walkthrough:**

### Question 1

View the pdf@main and print the code that is in the sim.main

```

Applications Places System
Sun 21 Feb, 02:00THM IP: 10.10.113.219
elfmceager@tbfc-day-17: ~
File Edit View Search Terminal Help

afl list functions
aflc count of functions
aflj list functions in json
afll list functions in verbose mode
aflq list functions in quiet mode
aflqj list functions in json quiet mode
afls print sum of sizes of all functions

[0x00400a30]> afl | grep main
0x00400b4d 1 68 syn.main
0x00400e10 10 1007 -> 219 syn._libc_start_main
0x00403870 39 661 -> 629 syn._nl_find_domain
0x00403b10 308 5366 -> 5301 syn._nl_load_domain
0x00415fe0 1 43 syn._IO_switch_to_main_get_area
0x0044cf00 1 8 syn._dl_get_dl_main_map
0x00470520 1 49 syn._IO_switch_to_main_wget_area
0x0048fae0 7 73 -> 69 syn._nl_finddomain_subfreeres
0x0048fb30 16 247 -> 237 syn._nl_unload_domain

[0x00400a30]> pdf @main
;-- main:
; (RIP) main: 68
; @p@main():
; var int local_ch @ rbp-0xc
; var int local_8h @ rbp-0x8
; var int local_4h @ rbp-0x4
; mov eax, 0
0x00400b4d 55 push rbp
0x00400b4e 48b9e5 mov rbp, rsp
0x00400b51 48b3ec10 sub rsp, 0x10
0x00400b55 c745f4040000 mov dword [local_ch], 4
0x00400b5c c745f8050000 mov dword [local_8h], 5
0x00400b63 8b55f4 mov edx, dword [local_ch]
0x00400b66 8b45f8 mov eax, dword [local_8h]
0x00400b69 01d0 add eax, edx
0x00400b6b 8945fc mov dword [local_4h], eax
0x00400b6e 8b4dfc mov ecx, dword [local_ch]
0x00400b71 8b55f8 mov edx, dword [local_8h]
0x00400b74 8b45f4 mov esx, dword [local_ch]
0x00400b77 b9c6 mov esi, eax
0x00400b79 488d3d881409 lea rdi, qword str.the_value_of_a_is_d_the_value_of_b_is_d
and_the_value_of_c_is_d
0x00400b80 b800000000 mov eax, 0
0x00400b84 e8f0ea0000 call syn._printf
0x00400b88 b800000000 mov eax, 0
0x00400b8f c9 leave
0x00400b90 c3 ret

[0x00400a30]>

```

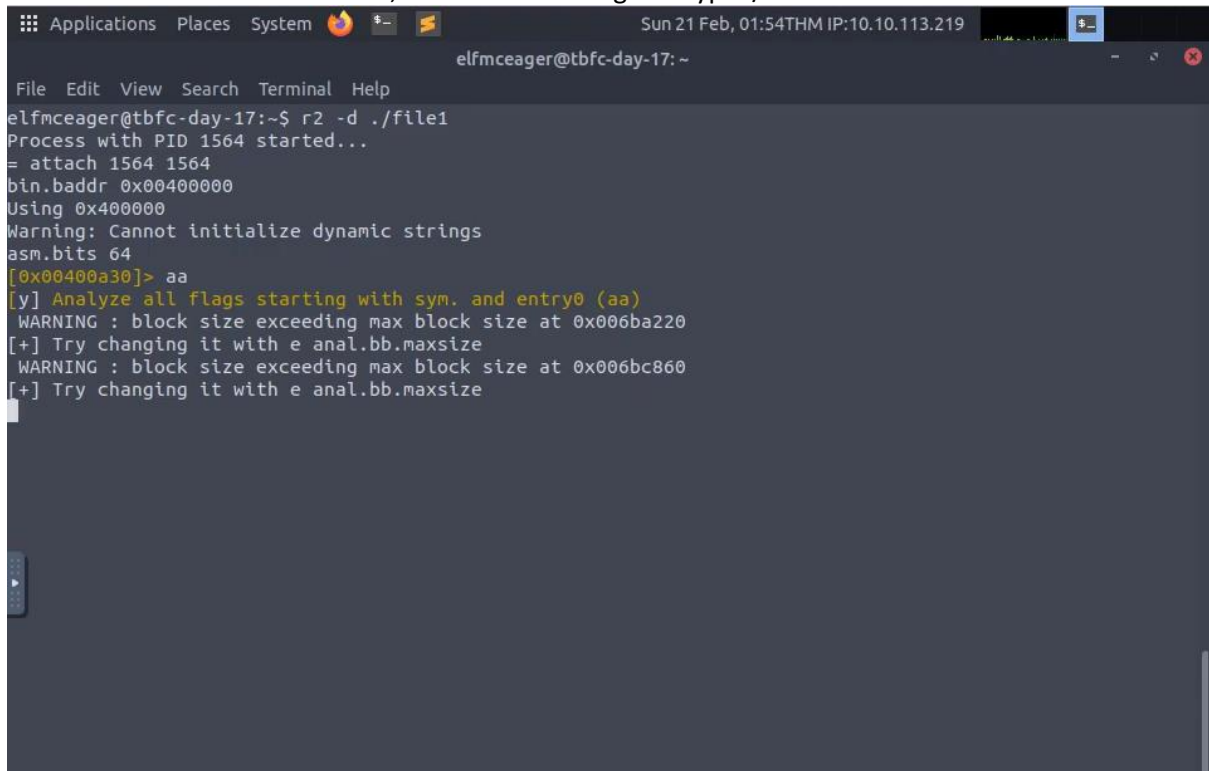
A register is a workbench where we can work on task one at a time where we can restore its value. We will have a register that will hold the value of local ch and another folder that hold the local 8h.




We have to perform mathematical functions on the two registers and it will overwrite the source register with the result.

Therefore, the data type that were matched with the size in bytes is byte = 1, Word = 2, Double Word = 4, Quad = 8, Single Precision = 4, Double Precision = 8

## Question 2

We have to run file 1 and attach, r2 -d means debug and type ./file1.

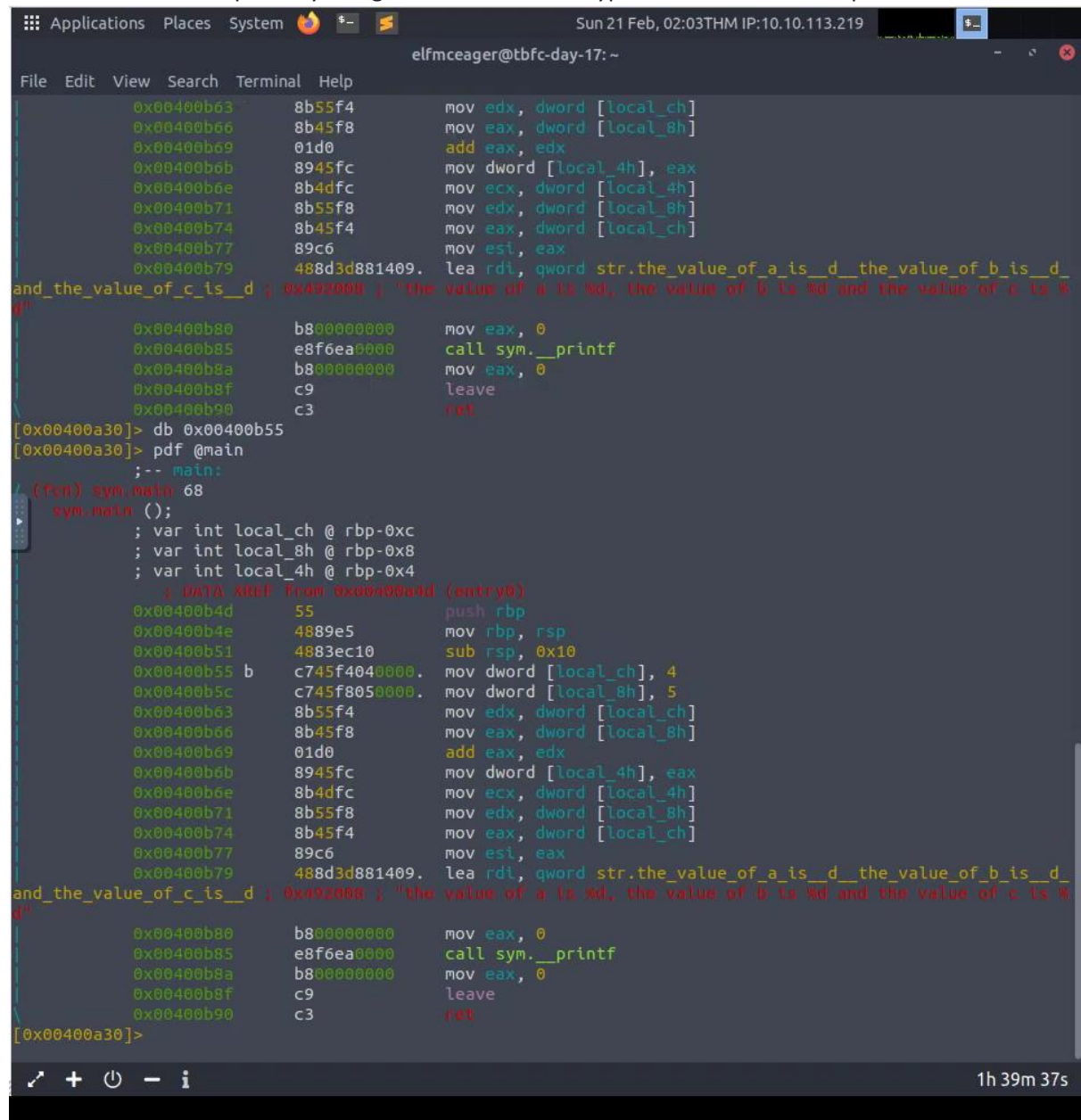


```
Applications Places System   Sun 21 Feb, 01:54THM IP:10.10.113.219 
elfmceager@tbfc-day-17: ~
File Edit View Search Terminal Help
elfmceager@tbfc-day-17:~$ r2 -d ./file1
Process with PID 1564 started...
= attach 1564 1564
bin.baddr 0x00400000
Using 0x400000
Warning: Cannot initialize dynamic strings
asm.bits 64
[0x00400a30]> aa
[y] Analyze all flags starting with sym. and entry0 (aa)
WARNING : block size exceeding max block size at 0x006ba220
[+] Try changing it with e anal.bb.maxsize
WARNING : block size exceeding max block size at 0x006bc860
[+] Try changing it with e anal.bb.maxsize
```

Thus, aa is the command that we entered to start analysing.

### Question 3

We can set the breakpoint by using the db command. Type db0x00400b55 and pdf@main.



```
elfmceager@tbfc-day-17: ~
Sun 21 Feb, 02:03THM IP:10.10.113.219

File Edit View Search Terminal Help

0x00400b63 8b55f4 mov edx, dword [local_ch]
0x00400b66 8b45f8 mov eax, dword [local_8h]
0x00400b69 01d0 add eax, edx
0x00400b6b 8945fc mov dword [local_4h], eax
0x00400b6e 8b4dfc mov ecx, dword [local_4h]
0x00400b71 8b55f8 mov edx, dword [local_8h]
0x00400b74 8b45f4 mov eax, dword [local_ch]
0x00400b77 89c6 mov esi, eax
0x00400b79 488d3d881409. lea rdi, qword str.the_value_of_a_is__d_the_value_of_b_is__d_
and_the_value_of_c_is__d ; 0x402000 ; "the value of a is %d, the value of b is %d and the value of c is %
0"
0x00400b80 b800000000 mov eax, 0
0x00400b85 e8f6ea0000 call sym.__printf
0x00400b8a b800000000 mov eax, 0
0x00400b8f c9 leave
0x00400b90 c3 ret

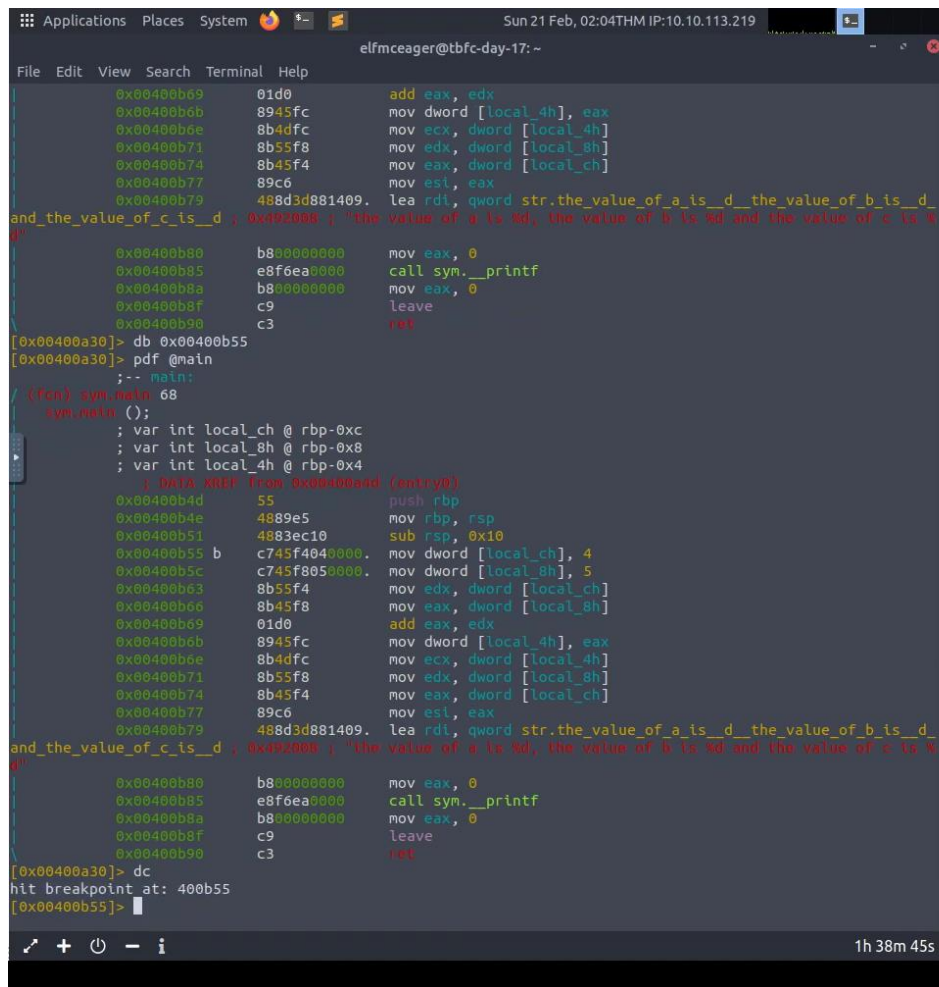
[0x00400a30]> db 0x00400b55
[0x00400a30]> pdf @main
;-- main:
(fcn) sym.main 68
sym.main ();
; var int local_ch @ rbp-0xc
; var int local_8h @ rbp-0x8
; var int local_4h @ rbp-0x4
; DATA ABX: From 0x00400add (entry0)
0x00400b4d 55 push rbp
0x00400b4e 4889e5 mov rbp, rsp
0x00400b51 4883ec10 sub rsp, 0x10
0x00400b55 b c745f4040000. mov dword [local_ch], 4
0x00400b5c c745f8050000. mov dword [local_8h], 5
0x00400b63 8b55f4 mov edx, dword [local_ch]
0x00400b66 8b45f8 mov eax, dword [local_8h]
0x00400b69 01d0 add eax, edx
0x00400b6b 8945fc mov dword [local_4h], eax
0x00400b6e 8b4dfc mov ecx, dword [local_4h]
0x00400b71 8b55f8 mov edx, dword [local_8h]
0x00400b74 8b45f4 mov eax, dword [local_ch]
0x00400b77 89c6 mov esi, eax
0x00400b79 488d3d881409. lea rdi, qword str.the_value_of_a_is__d_the_value_of_b_is__d_
and_the_value_of_c_is__d ; 0x402000 ; "the value of a is %d, the value of b is %d and the value of c is %
0"
0x00400b80 b800000000 mov eax, 0
0x00400b85 e8f6ea0000 call sym.__printf
0x00400b8a b800000000 mov eax, 0
0x00400b8f c9 leave
0x00400b90 c3 ret

[0x00400a30]>
```

We will see a lowercase b next to it which is telling us that we have a breakpoint.

## Question 4

We can type in dc which starts sunning the code and it will run up the first breakpoint.



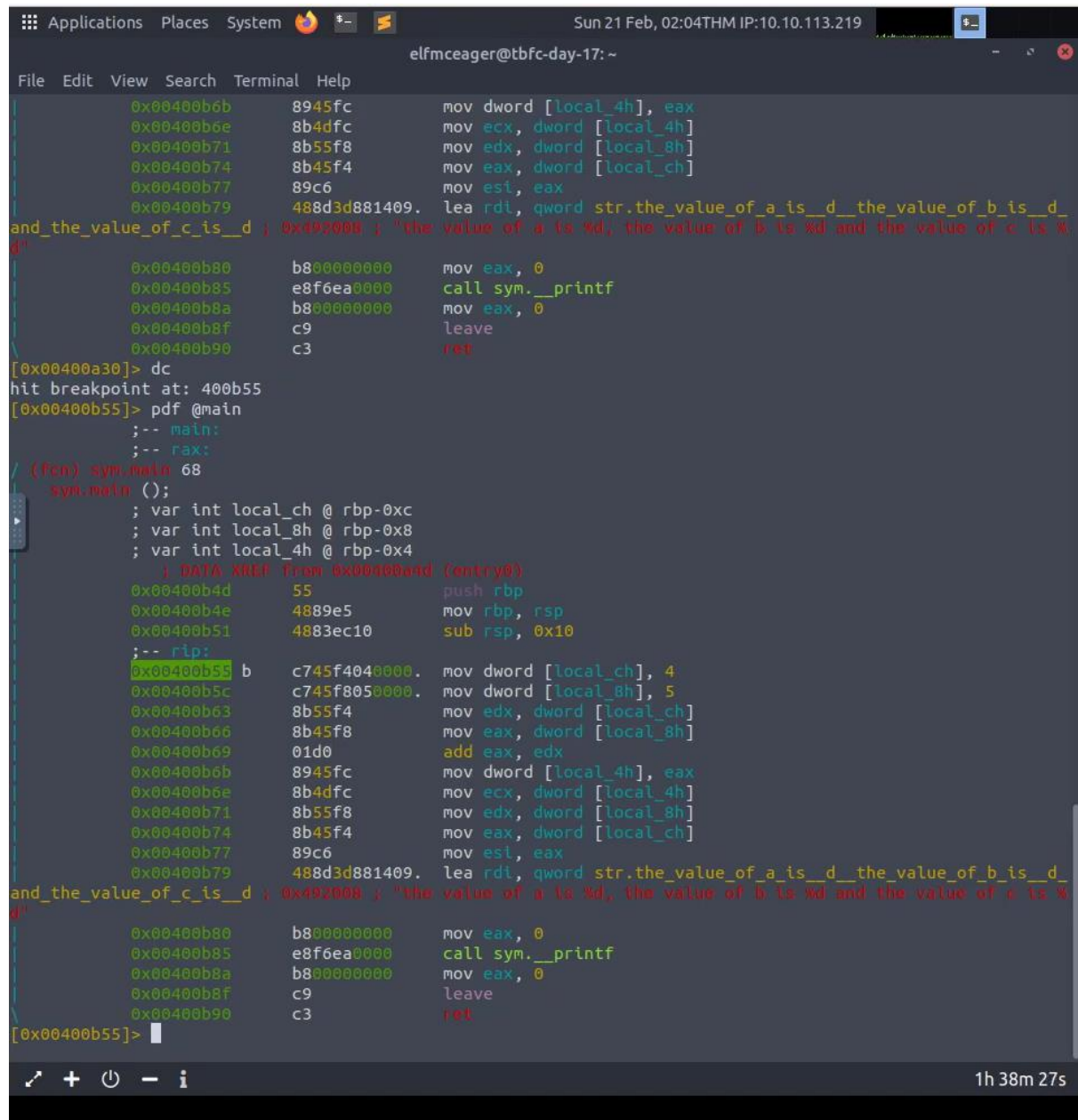
```
elfmceager@tbfc-day-17: ~
File Edit View Search Terminal Help
Sun 21 Feb, 02:04THM IP:10.10.113.219

0x00400b69 01d0 add eax, edx
0x00400b6b 8945fc mov dword [local_4h], eax
0x00400b6e 8b4dfc mov ecx, dword [local_4h]
0x00400b71 8b55f8 mov edx, dword [local_8h]
0x00400b74 8b45f4 mov eax, dword [local_ch]
0x00400b77 89c6 mov esi, eax
0x00400b79 488d3d881409 lea rdi, qword str.the_value_of_a_is_d_the_value_of_b_is_d
and_the_value_of_c_is_d ; 0x402000 ; "the value of a is %d, the value of b is %d and the value of c is %
0"
0x00400b80 b800000000 mov eax, 0
0x00400b85 e8f6ea0000 call sym.__printf
0x00400b8a b800000000 mov eax, 0
0x00400b8f c9 leave
0x00400b90 c3 ret

[0x00400a30]> db 0x00400b55
[0x00400a30]> pdf @main
;-- main:
(fcn) sym.main 68
sym.main():
; var int local_ch @ rbp-0xc
; var int local_8h @ rbp-0x8
; var int local_4h @ rbp-0x4
; break with fcn 0x00400b55 (entry0)
0x00400b4d 55 push rbp
0x00400b4e 4889e5 mov rbp, rsp
0x00400b51 4883ec10 sub rsp, 0x10
0x00400b55 b c745f4040000 mov dword [local_ch], 4
0x00400b5c c745f8050000 mov dword [local_8h], 5
0x00400b63 8b55f4 mov edx, dword [local_ch]
0x00400b66 8b45f8 mov eax, dword [local_8h]
0x00400b69 01d0 add eax, edx
0x00400b6b 8945fc mov dword [local_4h], eax
0x00400b6e 8b4dfc mov ecx, dword [local_4h]
0x00400b71 8b55f8 mov edx, dword [local_8h]
0x00400b74 8b45f4 mov eax, dword [local_ch]
0x00400b77 89c6 mov esi, eax
0x00400b79 488d3d881409 lea rdi, qword str.the_value_of_a_is_d_the_value_of_b_is_d
and_the_value_of_c_is_d ; 0x402000 ; "the value of a is %d, the value of b is %d and the value of c is %
0"
0x00400b80 b800000000 mov eax, 0
0x00400b85 e8f6ea0000 call sym.__printf
0x00400b8a b800000000 mov eax, 0
0x00400b8f c9 leave
0x00400b90 c3 ret

[0x00400a30]> dc
hit breakpoint at: 400b55
[0x00400b55]>
```

We have set a memory location 400b55 and that is where it stopped. When we do pdf@main again, we will see it is highlighted and there is a rip comment where we are currently sitting.

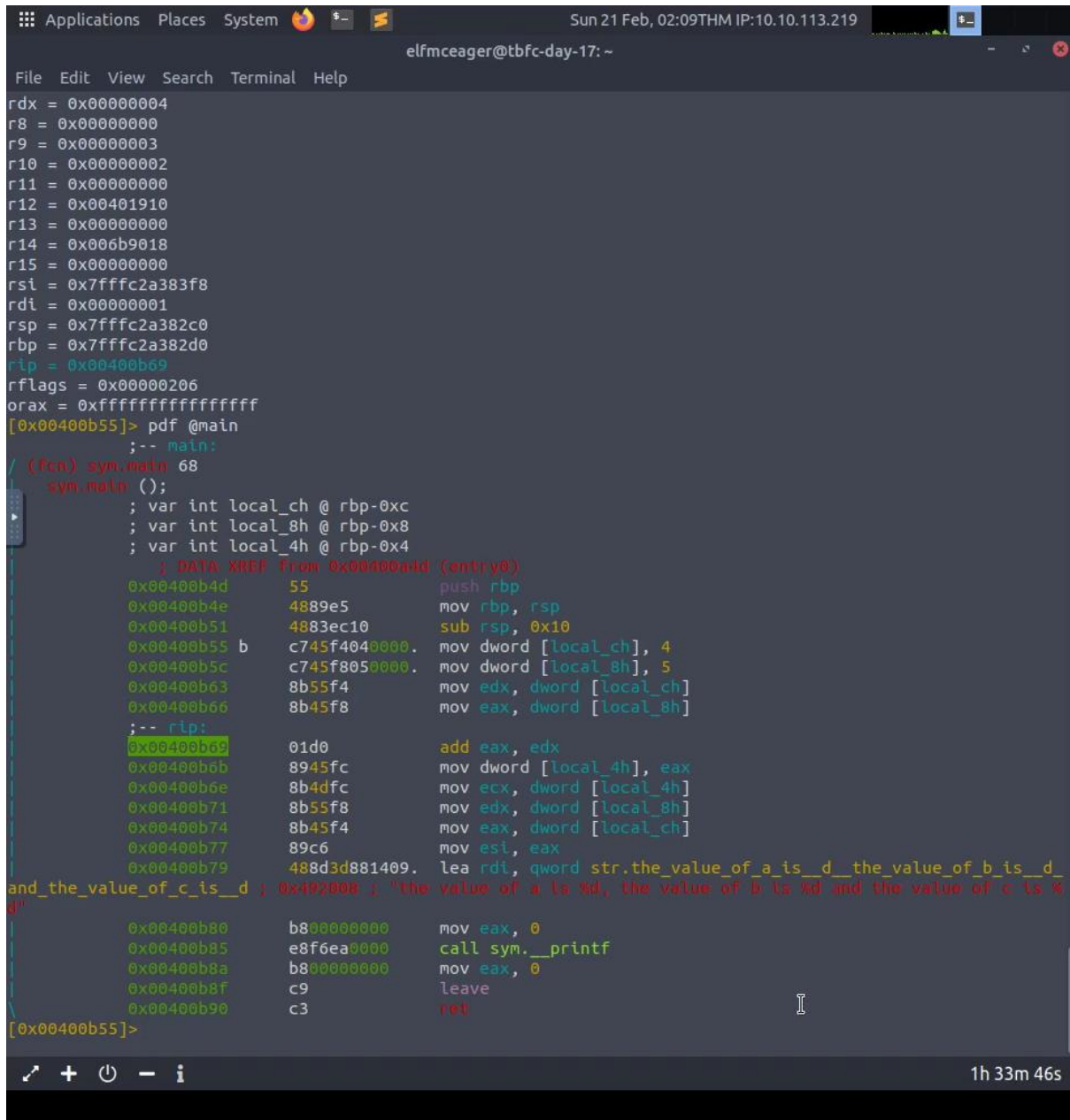


```
elfmceager@tbfc-day-17: ~
File Edit View Search Terminal Help
0x00400b6b 8945fc mov dword [local_4h], eax
0x00400b6e 8b4dfc mov ecx, dword [local_4h]
0x00400b71 8b55f8 mov edx, dword [local_8h]
0x00400b74 8b45f4 mov eax, dword [local_ch]
0x00400b77 89c6 mov esi, eax
0x00400b79 488d3d881409. lea rdi, qword str.the_value_of_a_is__d_the_value_of_b_is__d
and_the_value_of_c_is__d ; 0x400b79 ; "the value of a is %d, the value of b is %d and the value of c is %
0"
0x00400b80 b800000000 mov eax, 0
0x00400b85 e8f6ea0000 call sym.__printf
0x00400b8a b800000000 mov eax, 0
0x00400b8f c9 leave
0x00400b90 c3 ret
[0x00400a30]> dc
hit breakpoint at: 400b55
[0x00400b55]> pdf @main
;-- main:
;-- rip:
/ (fcn) sym.main 68
sym.main ();
; var int local_ch @ rbp-0xc
; var int local_8h @ rbp-0x8
; var int local_4h @ rbp-0x4
; DATA XREF from 0x00400b4d (entry0)
0x00400b4d 55 push rbp
0x00400b4e 4889e5 mov rbp, rsp
0x00400b51 4883ec10 sub rsp, 0x10
;-- rip:
0x00400b55 b c745f4040000. mov dword [local_ch], 4
0x00400b5c c745f8050000. mov dword [local_8h], 5
0x00400b63 8b55f4 mov edx, dword [local_ch]
0x00400b66 8b45f8 mov eax, dword [local_8h]
0x00400b69 01d0 add eax, edx
0x00400b6b 8945fc mov dword [local_4h], eax
0x00400b6e 8b4dfc mov ecx, dword [local_4h]
0x00400b71 8b55f8 mov edx, dword [local_8h]
0x00400b74 8b45f4 mov eax, dword [local_ch]
0x00400b77 89c6 mov esi, eax
0x00400b79 488d3d881409. lea rdi, qword str.the_value_of_a_is__d_the_value_of_b_is__d
and_the_value_of_c_is__d ; 0x400b79 ; "the value of a is %d, the value of b is %d and the value of c is %
0"
0x00400b80 b800000000 mov eax, 0
0x00400b85 e8f6ea0000 call sym.__printf
0x00400b8a b800000000 mov eax, 0
0x00400b8f c9 leave
0x00400b90 c3 ret
[0x00400b55]>
```



## Question 5,6 & 7

When we look at the code again, we will see that we are right at the line where we do the addition of `eax + edx`



```
Applications  Places  System  Sun 21 Feb, 02:09THM IP:10.10.113.219
elfmceager@tbfc-day-17: ~

File Edit View Search Terminal Help

rdx = 0x00000004
r8 = 0x00000000
r9 = 0x00000003
r10 = 0x00000002
r11 = 0x00000000
r12 = 0x00401910
r13 = 0x00000000
r14 = 0x006b9018
r15 = 0x00000000
rsi = 0x7fffc2a383f8
rdi = 0x00000001
rsp = 0x7fffc2a382c0
rbp = 0x7fffc2a382d0
rip = 0x00400b69
rflags = 0x00000206
orax = 0xffffffffffffff
[0x00400b55]> pdf @main
;-- main:
/ (fcn) sym.main 68
sym.main ();
; var int local_ch @ rbp-0xc
; var int local_8h @ rbp-0x8
; var int local_4h @ rbp-0x4
; DATA XREF From 0x00400a4d (entry0)
0x00400b4d 55 push rbp
0x00400b4e 4889e5 mov rbp, rsp
0x00400b51 4883ec10 sub rsp, 0x10
0x00400b55 b c745f4040000. mov dword [local_ch], 4
0x00400b5c c745f8050000. mov dword [local_8h], 5
0x00400b63 8b55f4 mov edx, dword [local_ch]
0x00400b66 8b45f8 mov eax, dword [local_8h]
;-- rip:
0x00400b69 01d0 add eax, edx
0x00400b6b 8945fc mov dword [local_4h], eax
0x00400b6e 8b4dfc mov ecx, dword [local_4h]
0x00400b71 8b55f8 mov edx, dword [local_8h]
0x00400b74 8b45f4 mov eax, dword [local_ch]
0x00400b77 89c6 mov esi, eax
0x00400b79 488d3d881409. lea rdi, qword str.the_value_of_a_is_d_the_value_of_b_is_d_
and_the_value_of_c_is_d ; 0x402008 ; "the value of a is %d, the value of b is %d and the value of c is %
d"
0x00400b80 b800000000 mov eax, 0
0x00400b85 e8f6ea0000 call sym.__printf
0x00400b8a b800000000 mov eax, 0
0x00400b8f c9 leave
0x00400b90 c3 ret
[0x00400b55]>
```

We have to look at the registers, `rax = 5`, `rdx = 4`, when we step, we will perform the addition and one of our registers will be modified.

```

Applications  Places  System  Sun 21 Feb, 02:09THM IP:10.10.113.219
elfmceager@tbfc-day-17: ~

File Edit View Search Terminal Help
[0x00400b55]> pdf @main
;-- main:
/ (fcn) sym.main 68
sym.main ();
; var int local_ch @ rbp-0xc
; var int local_8h @ rbp-0x8
; var int local_4h @ rbp-0x4
; DATA XREF from 0x00400aad (entry0)
0x00400b4d 55 push rbp
0x00400b4e 4889e5 mov rbp, rsp
0x00400b51 4883ec10 sub rsp, 0x10
0x00400b55 b c745f4040000. mov dword [local_ch], 4
0x00400b5c c745f8050000. mov dword [local_8h], 5
0x00400b63 8b55f4 mov edx, dword [local_ch]
0x00400b66 8b45f8 mov eax, dword [local_8h]
;-- rip:
0x00400b69 01d0 add eax, edx
0x00400b6b 8945fc mov dword [local_4h], eax
0x00400b6e 8b4dfc mov ecx, dword [local_4h]
0x00400b71 8b55f8 mov edx, dword [local_8h]
0x00400b74 8b45f4 mov eax, dword [local_ch]
0x00400b77 89c6 mov esi, eax
0x00400b79 488d3d881409. lea rdi, qword str.the_value_of_a_is__d_the_value_of_b_is__d_
and_the_value_of_c_is__d ; 0x492000 ; "the value of a is %d, the value of b is %d and the value of c is %
d"
0x00400b80 b800000000 mov eax, 0
0x00400b85 e8f6ea0000 call sym.__printf
0x00400b8a b800000000 mov eax, 0
0x00400b8f c9 leave
0x00400b90 c3 ret
[0x00400b55]> dr
rax = 0x00000005
rbx = 0x00400400
rcx = 0x0044ba90
rdx = 0x00000004
r8 = 0x00000000
r9 = 0x00000003
r10 = 0x00000002
r11 = 0x00000000
r12 = 0x00401910
r13 = 0x00000000
r14 = 0x006b9018
r15 = 0x00000000
rsi = 0x7fffc2a383f8
rdi = 0x00000001
rsp = 0x7fffc2a382c0
rbp = 0x7fffc2a382d0
rip = 0x00400b69
1h 33m 12s

```

Thus, 1 is the value of local\_ch when its corresponding movl instruction is called, 6 is the value of eax when the imull instruction is called, and 6 is the value of local\_4h before eax is set to 0.

### Thought Process/Methodology:

Print the code found in sim.main and view the pdf@main. A register serves as a work surface where we can do tasks one at a time in order to recover their value. The local ch value will be kept in a register, while the local 8h value will be kept in another folder. The two registers must be subjected to mathematical operations, and the results will be written over the source register. As a result, byte = 1, word = 2, double word = 4, quad = 8, single precision = 4, double precision = 8, and the data type were matched with the size in bytes. To execute file 1 and attach, type ./file1. Type r2 -d to debug. Thus, we entered the command aa to begin our analysis. Using the db command, we can set the breakpoint. Enter pdf@main and db0x00400b55. The fact that there is a lowercase b next to it indicates that we have reached a breakpoint. We may start the code by typing dc, and it will run up



the first breakpoint. It halted at a memory location that we placed at 400b55. It will be underlined and have a rip comment where we are sitting if we run a pdf@main once more. Rereading the code will reveal that we are currently on the line where the addition of eax and edx is performed. When we step, we will conduct the addition, and one of our registers will be changed. The registers are: rax = 5, rdx = 4, etc. Thus, 1 represents the value of local ch when the movl instruction corresponding to it is called, 6 represents the value of eax when the imull instruction is executed, and 6 represents the value of local 4h prior to eax being set to 0.