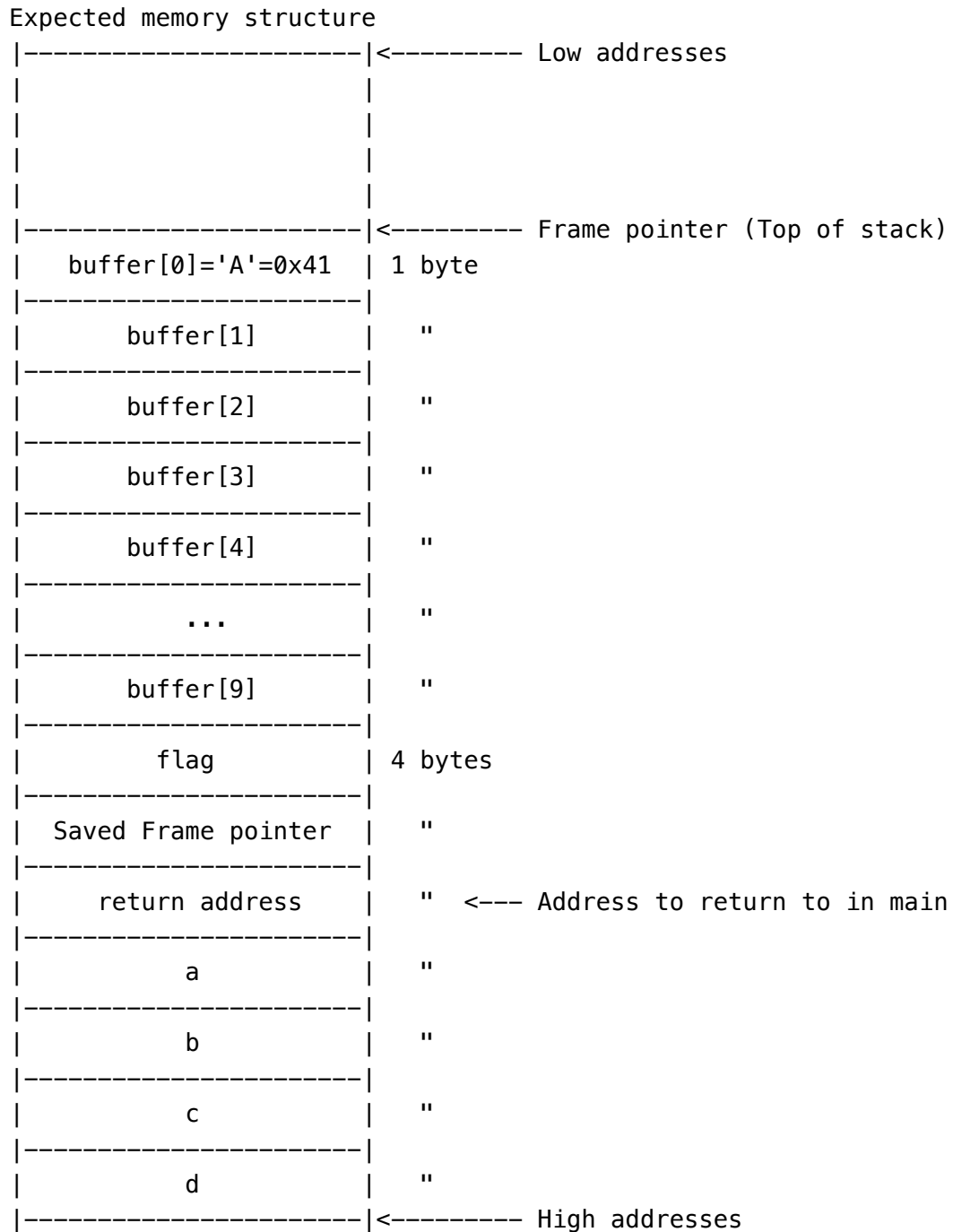


Stack Example walkthrough

- Picture of a stack frame for `test_function(int a, int b, int c, int d)`



- Open up `radare2` to confirm the above model

```

$ r2 -d stack_example
[>] aaa
[>] db main
[>] dc
[>] V

```

In Visual mode, hit `p` about twice to switch to debugger view.

- o `aaa` : Analyse all functions
- o `db main` : Set breakpoint on the main function
- o `dc` : Continue execution till next break point
- o `V` : Switch to Visual mode

You should see something similar below

```

student@csec-s: ~/code.csec-s/HTAOE
student@csec-s: ~/code.csec-s/HTAOE x
student@csec-s: ~/code.csec-s/HTAOE x
[0x080483db 115 /home/student/code.csec-s/HTAOE/stack_example]> f tmp;sr s.. @ s
0xffa94c24 0x080483ff 0x00000001 0x00000002 0x00000003 .....
0xffa94c34 0x00000004 0x00000000 0xf7532637 0x00000001 .....7&S.....
0xffa94c44 0xffa94cd4 0xffa94cdc 0x00000000 0x00000000 .L...L.....
0xffa94c54 0x00000000 0xf76ca000 0xf7715c04 0xf7715000 .....l..q..Pq.
eip 0x080483db    oeax 0xffffffff    eax 0xf76cbdbc    ebx 0x00000000
ecx 0x5bbab68e    edx 0xffa94c64    esp 0xffa94c24    ebp 0xffa94c38
esi 0xf76ca000    edi 0xf76ca000    eflags = 1ASI
; CODE (CALL) XREF from 0x080483fa (unk)
/ (fcn) sym.test_function 20
;-- eip:
0x080483db      55          push ebp
0x080483dc      89e5        mov ebp, esp
0x080483de      83ec10      sub esp, 0x10
0x080483e1      c745fc697a0. mov dword [ebp-0x4], 0x7a69
0x080483e8      c645f241   mov byte [ebp-0xe], 0x41
0x080483ec      90          nop
0x080483ed      c9          leave
0x080483ee      c3          ret
; CODE (CALL) XREF from 0x080483ef (unk)
/ (fcn) main 26
0x080483ef b 55          push ebp
0x080483f0      89e5        mov ebp, esp

```

- o Notice the order of values `0x00000001` , `0x00000002` , `0x00000003` and `0x00000004` .
- o Notice the green coloured column below the first golden line: This is the stack. Higher addr below and top is lower address.
- o Below is registers: Something we will cover in Reverse Engineering
- o Notice the highlighted address in while `0x080483ff` . Search below for the corresponding address.

```

stack_example.dump (~/.code.csec-s/HTAOE) - gedit
Open [+] Save

80483d5:      c9          leave
80483d6:      e9 75 ff ff jmp      8048350 <register_tm_clones>

080483db <test_function>:
80483db:      55          push    ebp
80483dc:      89 e5       mov     ebp,esp
80483de:      83 ec 10    sub     esp,0x10
80483e1:      c7 45 fc 69 7a 00 00 mov     DWORD PTR [ebp-0x4],0x7a69
80483e8:      c6 45 f2 41 mov     BYTE PTR [ebp-0xe],0x41
80483ec:      90          nop
80483ed:      c9          leave
80483ee:      c3          ret

080483ef <main>:
80483ef:      55          push    ebp
80483f0:      89 e5       mov     ebp,esp
80483f2:      6a 04       push   0x4
80483f4:      6a 03       push   0x3
80483f6:      6a 02       push   0x2
80483f8:      6a 01       push   0x1
80483fa:      e8 dc ff ff call    80483db <test_function>
80483ff:      83 c4 10    add     esp,0x10
8048402:      b8 00 00 00 mov     eax,0x0
8048407:      c9          leave
8048408:      c3          ret
8048409:      66 90       xchg   ax,ax
804840b:      66 90       xchg   ax,ax
804840d:      66 90       xchg   ax,ax

```

Observe the following image

```

student@csec-s: ~/code.csec-s/HTAOE
student@csec-s: ~/code.csec-s/HTAOE x student@csec-s: ~/code.csec-s/HTAOE x +
[0x080483db 115 /home/student/code.csec-s/HTAOE/stack_example]> f tmp;sr s.. @ s
0xffa94c10 0x00410001 0xffa94cd4 0xffa94cdc 0x00007a69 ..A..L..L..iz..
0xffa94c20 0xffa94c38 0x080483ff 0x00000001 0x00000002 8L.....
0xffa94c30 0x00000003 0x00000004 0x00000000 0xf7532637 .....7&S.
0xffa94c40 0x00000001 0xffa94cd4 0xffa94cdc 0x00000000 .....L..L.....
eip 0x080483ec  oeax 0xffffffff  eax 0xf76cbdbc  ebx 0x00000000
ecx 0x5bbab68e  edx 0xffa94c64  esp 0xffa94c10  ebp 0xffa94c20
esi 0xf76ca000  edi 0xf76ca000  eflags = 1SI
; CODE (CALL) XREF from 0x080483fa (unk)
/ (fcn) sym.test_function 20
| 0x080483db 55          push    ebp
| 0x080483dc 89e5       mov     ebp, esp
| 0x080483de 83ec10    sub     esp, 0x10
| 0x080483e1 c745fc697a0. mov     dword [ebp-0x4], 0x7a69
| 0x080483e8 c645f241  mov     byte [ebp-0xe], 0x41
| ;-- eip:
| 0x080483ec 90          nop
| 0x080483ed c9          leave
| 0x080483ee c3          ret
| ; CODE (CALL) XREF from 0x080483ef (unk)
/ (fcn) main 26
| 0x080483ef b 55          push    ebp
| 0x080483f0 89e5       mov     ebp, esp

```

- o Notice that according to our model, we expect to find the flag variable right after the buffer towards higher addresses

- o The highlighted value is 31337 in hex

```
student@csec-s:~/code.csec-s/HTA0E$ gdb -q
(gdb) p 0x00007a69
$1 = 31337
(gdb) █
```