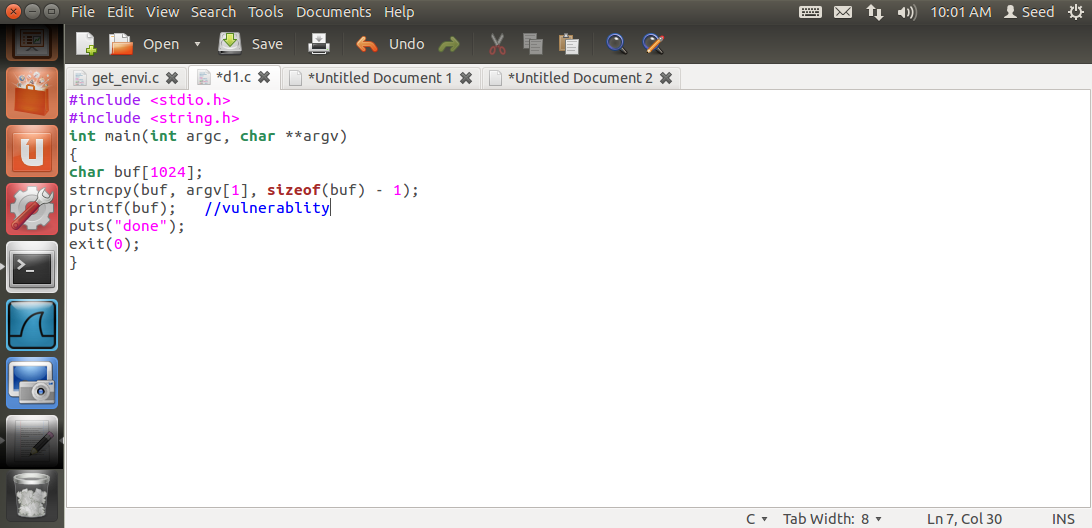
**STEP -1**

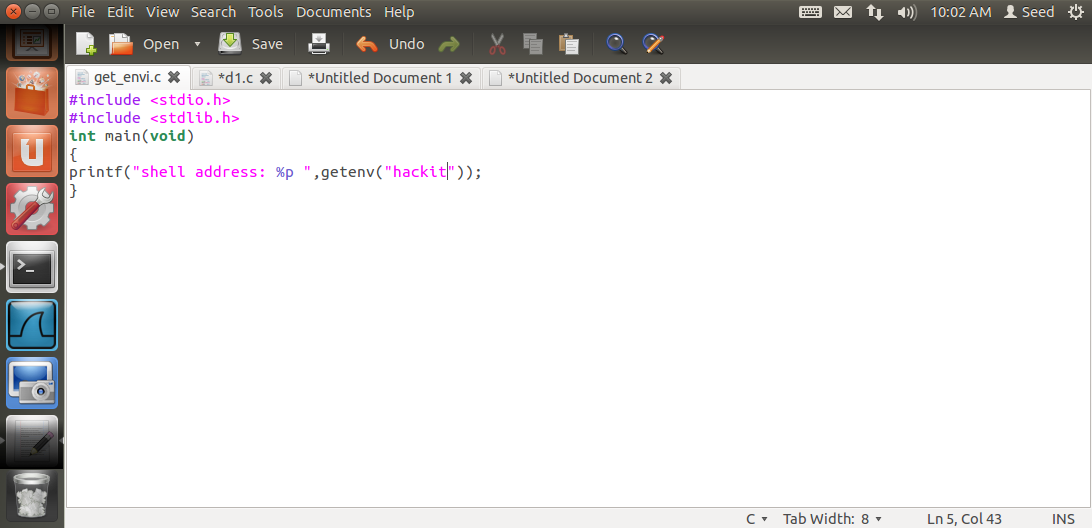
Vulnerable program: d1.c



Observing the above code, our idea is to write the shellcode into the environment variable, then get the address of the environment variable, and then override the address of the exit function, so that when the program executes to exit(0), it will execute our Shellcode.

**STEP -2**

The get.c given below is used to get the environment variable address:



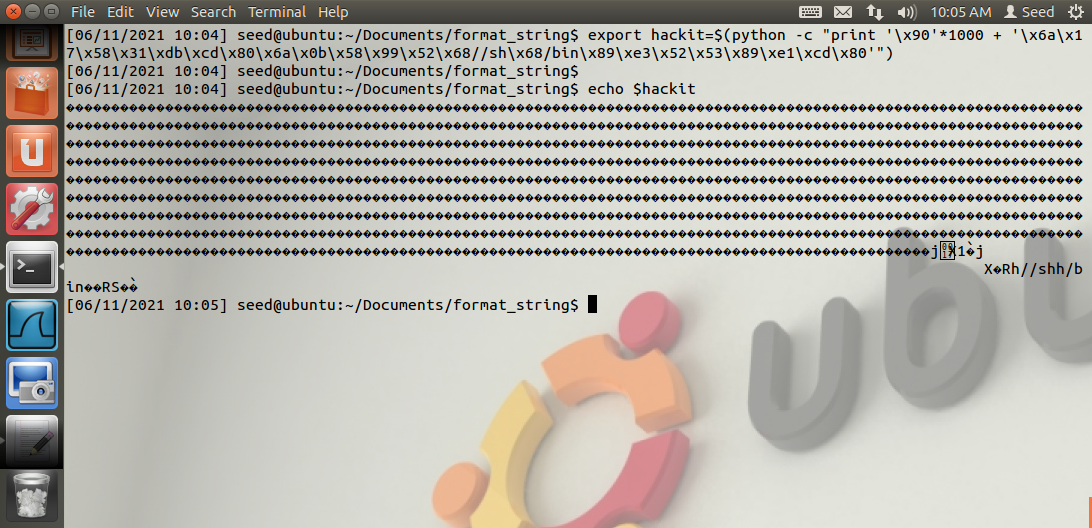
**STEP -3**

Compile get.c.

compile vul.c and make the stack executable



**STEP -4**

Define an environment variable ‘hackit’ to store the shellcode, and then execute get to get its address.

**STEP -5**

Get the hackit address: 0xbfffed17



**STEP -6**

Make root the owner of d1 vulnerable executable and set the setuid bit on, so that it can run with the privileges of the owner (root).



**STEP -7**

Query the location of aaaa in memory:



Aaaa (0x61616161) is in the 11th position.

**STEP -8**

Find the address of exit: objdump -R vul

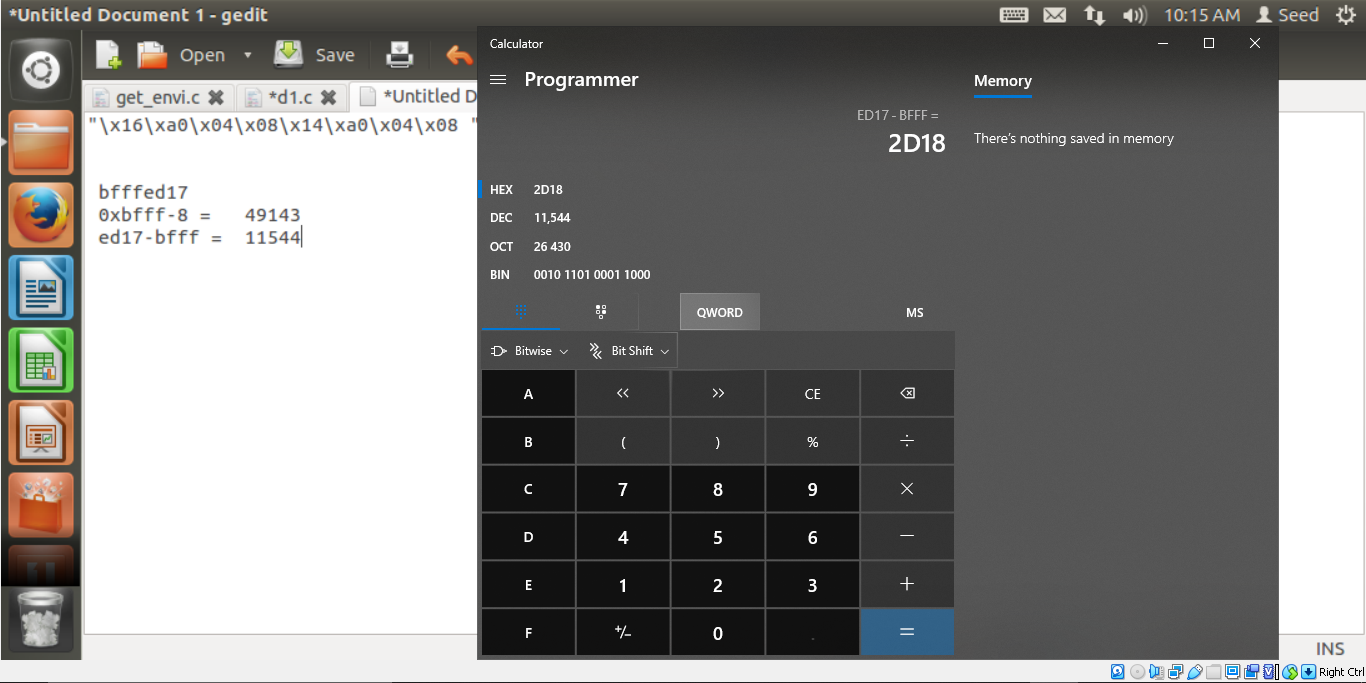
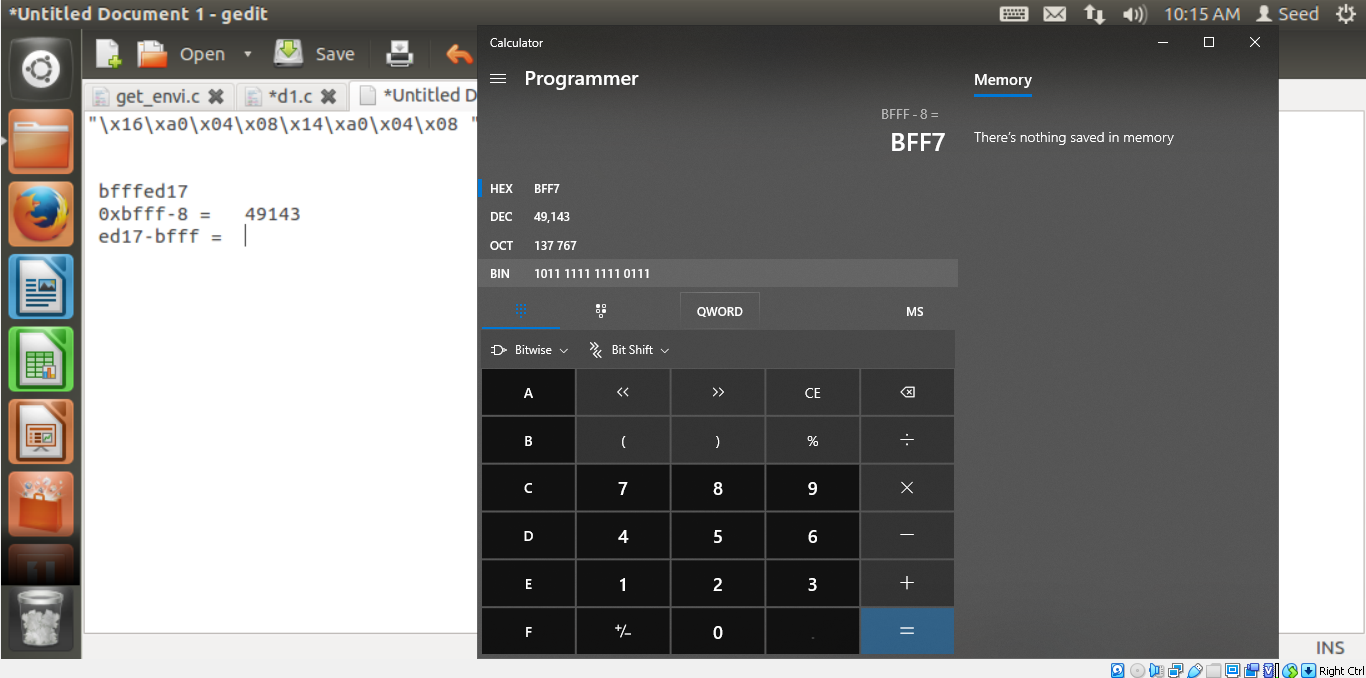
**STEP -9**

 Implement the attack, the key to implementing the attack is to calculate the shellcode address 0xbfffed17.

The required offset in the memory word pointed to by the return address of the constructor (ie 0x 0804a014).  
Calculate the offset:

bfff-8 = 49143 (dec)

ed17 – bfff = 11544 (dec)





**STEP-10**

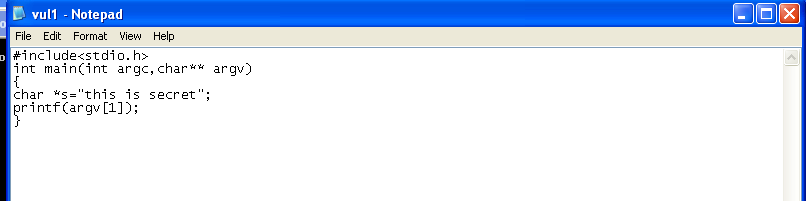
The attack was successful... successfully obtained root privileges #





**On windows XP:**

Below is the vul.c program used in this experiment.

****

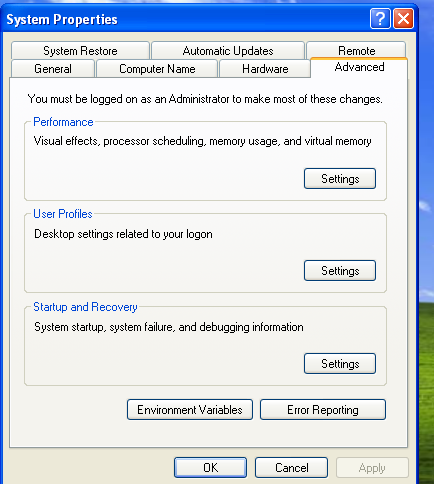
There is a countermeasure installed by Microsoft on the windows version call Data Execution Protection (DEP).

It is required to be turned off.

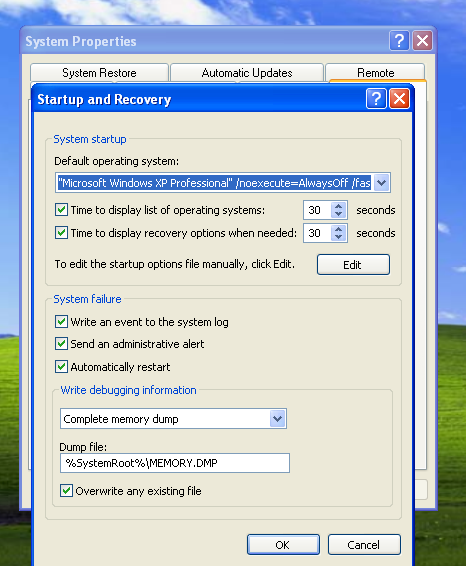
Click on Start->Right click on My computer->Properties



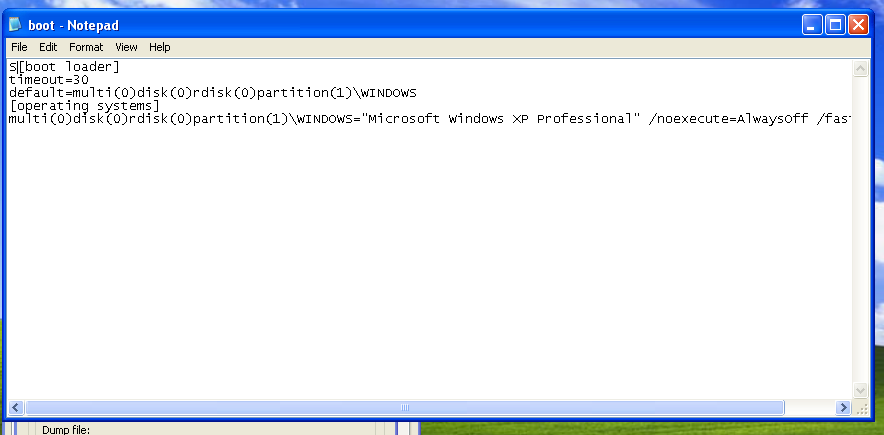
Select Advance tab->Under Startup and Recovery column select Settings.



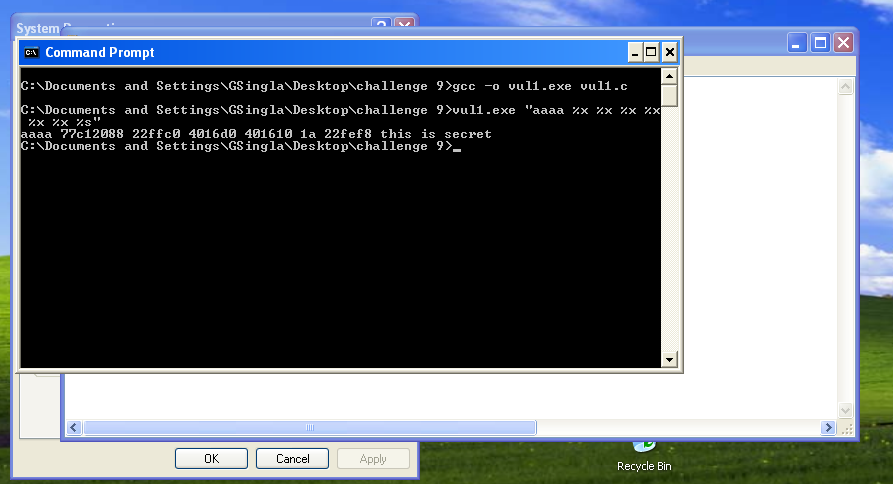
Under system startup column select edit button



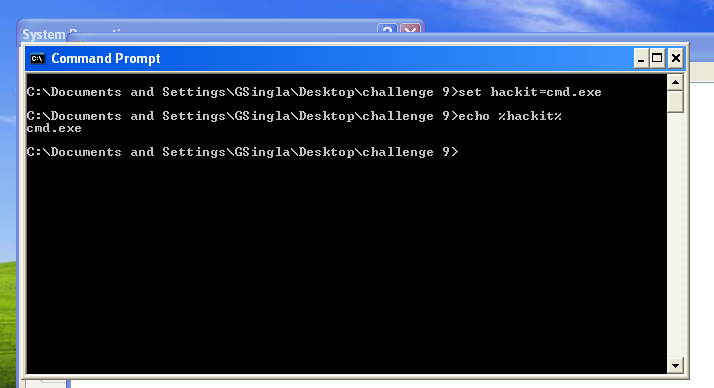
Under /noexecute=AlwaysOff



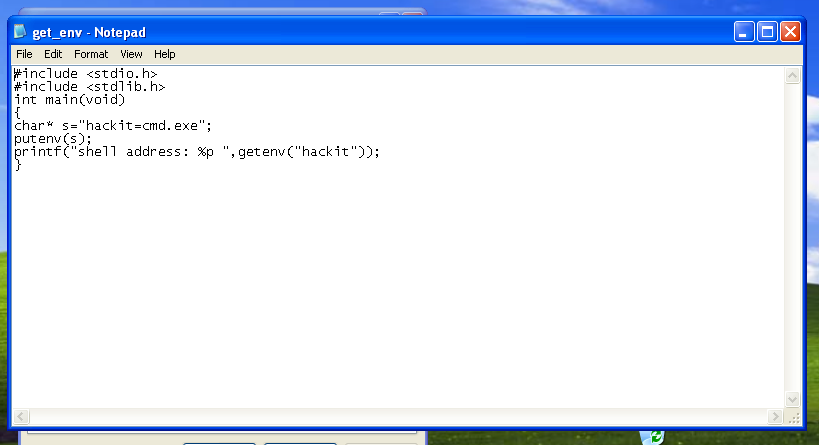
Format string vulnerability



Set the environment variable to execute cmd.exe

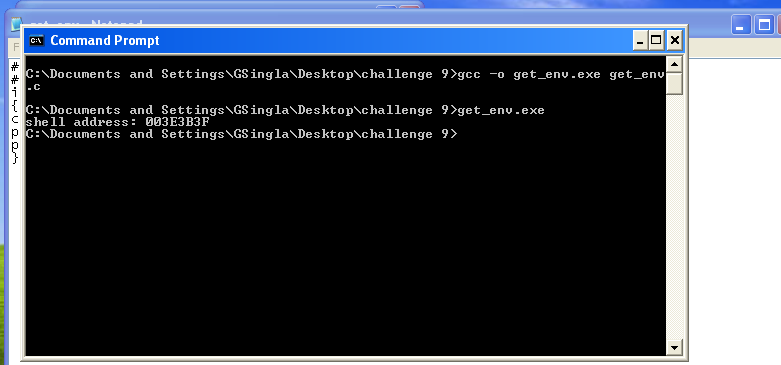


Program to set the environment variable and it’s address

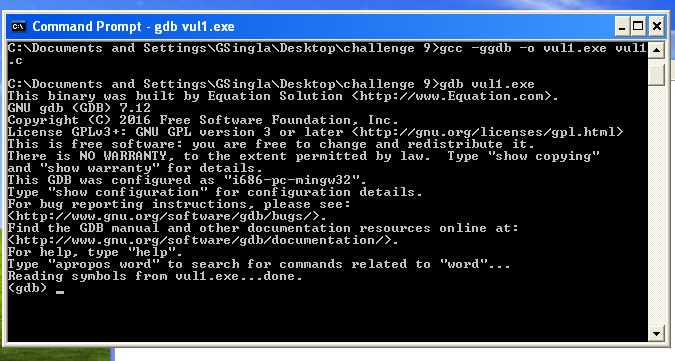


Compile the program.

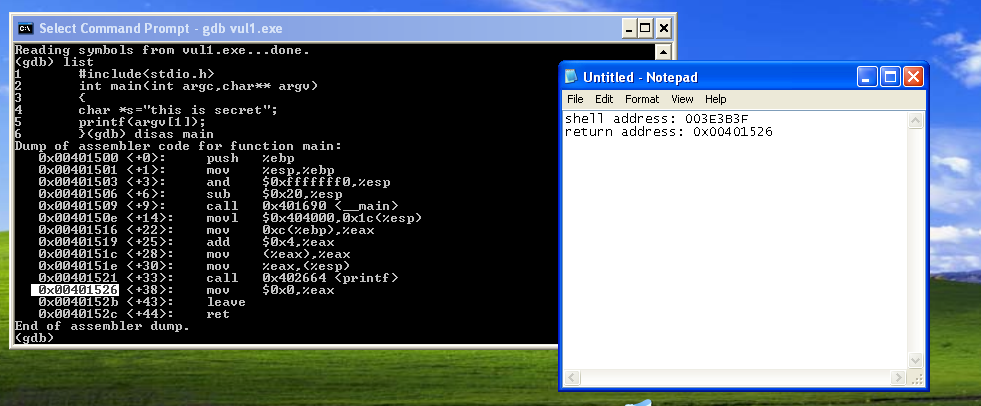
After executing get\_env.exe, address of hackit environment variable is received.



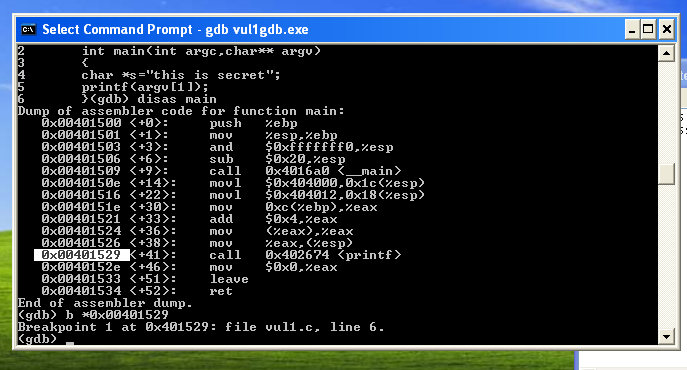
Get the debug version of vul1.exe to get the return address.



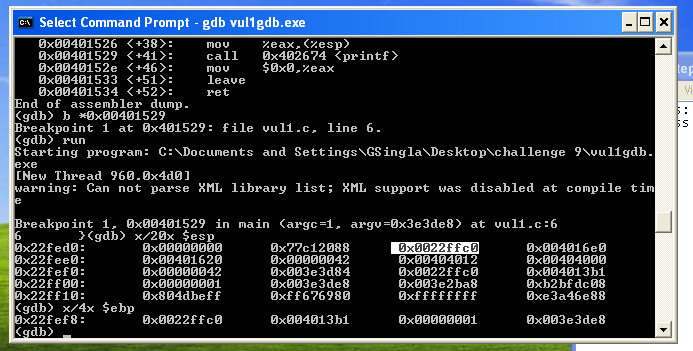
Where 0x0 is moved into %eax register, is taken as return address that we want our formatted string to return to after execution of desired exploit.



Set a break point just before printf was called.

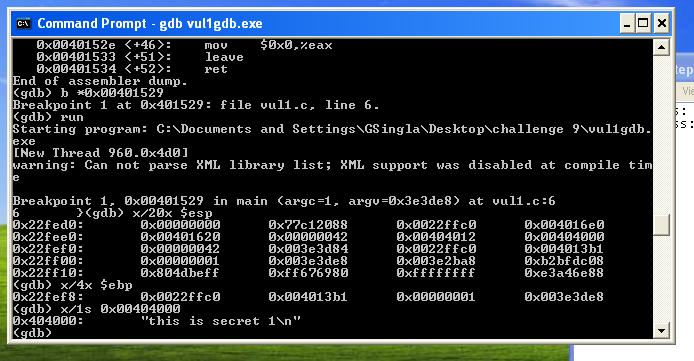


Here lets try to examine the memory/stack.



As we can see here, $ebp is placed at third address wrt $esp, it is frame pointer, and next address is original return address.

And our local variable is stored at 0x00404000



To run our exploit we need to manipulate original frame pointer and return address.

And replace the local variable address 0x00404000 with address of our set environment variable so that it can jump over to environment variable and execute it.

Format string manipulation:

