# Buffer Overflow Problem

The source code of a particular program we wrote is as follows:

```
#include <stdio.h>

int main() {
    int leet=0;
    char buf[80];

    puts("Hello, What's your name?");
    gets(buf);

    printf("Hello, %s!\n", buf);

    if(leet == 31337) {
        printf("Congratulations! If this did something nasty, like change grades, you would have won!\n");
    }

    return 0;
}
```

Actually, it turns out we are really, incredibly careless. You point out to us that **gets** is inherently unsafe. We mutter something[1] in response, then walk away. Because we won't listen to you, you decide to resort to more drastic measures to get our attention. Overflowing our program should do the trick...

(a) In fact, **gets** is so incredibly unsafe that gcc will throw a warning if you try to use it. Explain why the **gets** function is so unsafe.

(b) Execute a buffer overflow attack to overwrite the local variable **leet** with the value 31337, thus printing an otherwise unprintable message. Include in your solution the string you used in your attack.

(c) Explain how you would craft a string that would execute a shell, thus allowing you to write whatever you want across the system. (*Note: you don't have to actually execute the attack for this part.*)

(d) Suggest several ways that the code author or the compiler could prevent the attack in part (c).

(e) **Optional:** For bonus kudos, execute the attack in part (c) and let us know how you did it. Include your code.

You should compile this code on an x86 machine (Athena works fine), and compile it with gcc. We have tried this with gcc 4.1 and it worked well. For a nice overview of buffer overflow attacks, read "Smashing the Stack for Fun and Profit", by Aleph One, available on the course website.

[1] http://www.crypto.com/bingo/pr