Program #1: Buffer Overflow

Due: April 15, 2015 **Points:** 100

This problem asks you to implement a buffer overflow attack on a program. In the Resources area of SmartSite (or the Homework area of the nob.cs.ucdavis.edu class web site) is a program *bad.c* (also see below). This program contains a buffer overflow vulnerability; see the call to *gets*(3) at line 13. Your job is to exploit the overflow by providing input to the running process that will cause the program to invoke the function *trap* (which, you may notice, is not called anywhere else). You will know you've succeeded when you run the program, give it your input, and it prints "Gotcha!"

The following questions will help guide you. Please turn in your answers to them, a hex dump of the input you use to call *trap*, and a typescript or screen shot of you running the program *bad*, giving it your input, and showing its output.

1. What is the address of the function *trap*()? How did you determine this?
2. What is the address on the stack that your input must overwrite (please give both the address of the memory location(s), and their contents)? How did you locate this address?
3. What is the address of `buf`?
4. The *sled* is the input you give to alter the return address stored on the stack. What is the minimum length your sled must be?

bad.c

This is a listing of *bad.c*.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int trap(void)
5  {
6          printf(”Gotcha!\n”);
7          exit(0);
8  }
9
10 int getstr(void)
11 {
12         char buf[12];
13         gets(buf);
14         return(1);
15 }
16
17 int main(void)
18 {
19         getstr();
20         printf(”Overflow␣failed\n”);
21         return(1);
22 }
```