CS 491/591
Software Security

Assignment 2

Due by 11:59PM, Friday, **April 20**

NOTE: Assignment is to be done in your assigned teams. Do not look at another team's answer and do not allow another team to look at your answer. Doing so will result in charges of academic misconduct.

This assignment is the natural follow-up to assignment 1.
Overview: In this second assignment, you are to use the provided simple application that contains a vulnerability, identify the vulnerability, and then write an application that will exploit the vulnerability.

Description:
The assignment was heavily borrowed from Steven Dancewicz.
The accuse service is a simple system for assigning accusations. The program accepts on standard input the name of a victim and prints on standard output a message asserting that person's universal culpability. For example:

```
$ echo "Les Miles" | ./accuse
        It's all Les Miles's fault.
```

Source code for accuse.c is attached below.

Simple-minded string processing aside (it would be more correct, after all, to say that "It's all Les Miles' fault"), there is a serious problem with the accuse program. Despite our best efforts, a bug allows anyone who can provide input to this program to run arbitrary code on the target machine. (What might happen if it is run as a network service under inetd?)

Your job is to create input that will cause the accuse service to print out the helpful message "Now I pwn your computer" before it terminates. For example:

```
$ cat exploit_file | ./accuse
...
Now I pwn your computer
```

Here, the "..." may be additional output caused as a side-effect of your attack.

Deliverables: 1) your exploit application, and 2) a document discussing what the vulnerability is, how you were able to exploit it, and what your recommendation is for fixing the vulnerability.

Blame source code (accuse.c):

```
/*
 * Usage: accuse
 *    (reads one line from standard input)
 *
 * To compile:
 *    cc accuse.c -o accuse
 *
 * Install under inetd as follows:
 *  accuse  stream      tcp   nowait      root  /path/to/accuse   accuse
 *
 * Copyright 2003 by Bob T. Coder, PhD.
 */

#include <stdio.h>
#include <string.h>
#define INPUT_BUFFER 256  /* maximum name size */

/*
 * read input, copy into s
*/
void getline(char *s)
{
      int c;

      while ((c=getchar()) != EOF)
            *s++ = c;
      *s = '\0';
}

/*
 * convert newlines to nulls in place
 */
void removenewlines(char *s)
{
      int l;

      l = strlen(s);

      while (l--)
            if (s[l] == '\n')
                  s[l] = '\0';
}


int main()
{
      char victim[INPUT_BUFFER];

      getline(victim);
      /* this check ensures there's no buffer overflow */
      if (strlen(victim) < INPUT_BUFFER) {
            removenewlines(victim);
            printf("It's all %s's fault.\n", victim);
      }
      return 0;
}
```