Vulnerable Code

Stack 1

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>


char buffer[50];

int copy_buffer(char *input_buffer) {
    strcpy(buffer, argv[1]);

    printf("DEBUG: strcpy() executed...\n");
}


int main(int argc, char *argv[]) {




   if (argc < 2) {
      printf("Syntax error\n");
      printf("Syntax: %s <characters>\n", argv[0]);
      exit(0);
   }

    copy_buffer(argv[1]);

    printf("buffer content= %s\n", buffer);
    return 0;
}
```

Heap 2

```
# https://infosecwriteups.com/stack-based-buffer-overflow-practical-for-windows-vulnserver-8d2be7321af5

#include<stdio.h>
#include<string.h>
```

```
int main(void)
{
    char buff[15];
    int pass = 0;
    printf("\n Enter the password : \n");
    gets(buff);
    if(strcmp(buff, "P@ssw0rd")) {
        printf("\n Wrong Password \n");
    }
    else {
        printf("\n Correct Password \n");
        pass = 1;
    }
    if(pass) {
        printf("\n Execute guarded command \n");
        char command[50];
        strcpy(command, "ls -l" );
        system(command);
    }
return 0;
}
```