# Homework 3

Casey Bates

19 October 2020

## 1 Vulnerable to Buffer Overflow?

The given `readinput()` function *is* vulnerable to a buffer overflow attack. The `gets(str)` will read a line from the standard input and store it in the str buffer without checking the size of the input. In this case, str is only allotted 30 bytes, so an input of more than that will overflow into the rest of the stack frame. It exploit this vulnerability by overwriting the return address with the address of some malicious code.

```
1 void readinput() {
2     char str[30];
3     gets(str);
4     printf("%s", str);
5     return;
6 }
```

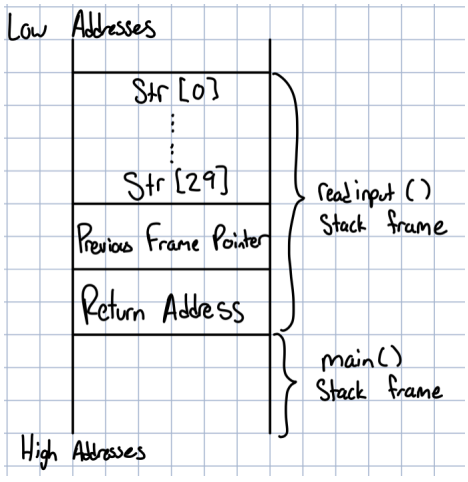


Figure 1: The stack frame of readinput()

# 2 Identify the Vulnerability

This function reads an input from the network into the variable `len`. `Len + 10` is then used to allocate memory for `buf` without validating the size of `len`, so it is vulnerable to integer overflow. If the integer read from the network is `UINT_MAX - 9` or higher, `len + 10` will overflow to 0-9. With 9 or less bytes allocated to `buf` and `len` being a very large integer, reading into `buf` will cause a buffer overflow that could potentially be exploited.

```
1 void copyinfo(w) {
2     size_t len;
3     char *buf;
4
5     len = read_int_from_network();
6     buf = malloc(len+10);
7     read(info, buf, len);
8     ...
9 }
```

# 3 Show at Least Two Vulnerabilities

## 3.1 Off by One Error

In the for loop on line 15, `i` is initialized to 0, and iterates until `i <= n`. However this will iterate one time more than the size of the shopping list. For example, if `n = 16`, `i` will iterate from 0 to 16. This is 17 times total, one more than the size of the shopping list.

## 3.2 Integer Overflow

On line 21, variable `len` of type `size_t` is added to `size_exp` of type `int`. This could cause an integer overflow, allowing `snprintf()` to write to an unintended memory location.

## 3.3 Format String Vulnerability

In the If statement on line 27, the program will run a command on the system with the names and prices of the expensive items. An attacker could name an item after a system command, (for example "cat /ect/passwd") and that command will be run.