Lab Exercise free

This is a lab exercise on developing secure software. For more information, see the introduction to the labs.

1. Task

Please fix the code below to fix a simple use-after-free bug.

2. Background

Practically all programming languages allow developers to quickly allocate memory and store data in that memory region. Once the program is finished using that memory, most programming languages automatically reclaim it.

However, the programming languages C and C++ require *manual* memory management. That is, developers using C and C++ must *manually* tell the system to release a memory region (using `free` and `delete` respectively). Manual memory management can have performance benefits, and it's conceptually simple. However, manual memory management can also lead to a variety of common types of bugs:

1. Double-free: Release the same memory region more than once.
2. Use-after-free: Use the memory (for reading or writing) after it's been released.
3. Missing release: Fail to release memory after it's no longer used.

These bugs often happen because it's difficult to be perfect, all the time, as software becomes larger and more complex. Many vulnerabilities have stemmed from manual memory management bugs. Not *all* such bugs are vulnerabilities, but many are.

3. Task Information

Please change the C code below to fix a simple use-after-free bug. This code for the function `tweak` accepts a string named `s`. It must call the function `asprintf` to create a new string that contains the text `pre_`, the input text (`s`), and the text `_post`. The function `tweak` must eventually return this new result. Unfortunately the current code makes a call to `free` to release a memory region *before* the last use of that memory. This can lead to a "use-after-free" bug. Whether or not this bug can cause a problem depends on many implementation details, but we don't want it to ever cause a problem.

Please fix this code! Use the “hint” and “give up” buttons if necessary.

4. Interactive Lab (COMPLETE!)

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

// Return tweaked version of string s. Frees s.
char *tweak(char *s) {
  char *result; // Put result here

  asprintf(&result, "pre_%s_post", s);
  free(s);
  return result;
}
```

Hint Reset Give up

This lab was developed by David A. Wheeler at The Linux Foundation.

Completed 2025-02-13T21:54:46.377Z 0cb11786-5c54-4fa2-906a-0707d2234212 125tsc31pv884t