Universität des Saarlandes
Lehrstuhl für Softwaretechnik
Prof. Andreas Zeller, Stephan Neuhaus
Design of Secure Software Systems
Exercise 5

Total number of points obtainable: 19, plus 0 bonus points.

1 Your Own Exploit

(9 points) The following code contains a buffer overflow vulnerability that makes it vulnerable to a stack-smashing attack. Use the techniques shown in the lecture to create an exploit.

```
#include <string.h>

void exploit_me(const char *buffer) {
  char copy[80];

  strcpy(copy, buffer);
}

int main(int argc, const char* argv[]) {
  exploit_me(argv[1]);
  return 1;
}
```

2 Non-Executable Stacks

(10 points) One possible defense against stack smashing attacks is to make stacks non-executable. That means that data on the stack cannot be executed by the processor. This would certainly invalidate the approach laid out in the lecture. However, it's not a silver bullet.

Look at the following code and write an exploit that lets you execute a shell, despite the fact that the function *shell*() is never called from *main*() or *exploit_me*().

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

static int shell() {
  char *const filename = "/bin/sh";
  char *const argv[] = { "/bin/sh", 0 };
  char *const envp[] = { 0 };

  return execve(filename, argv, envp);
}

static void exploit_me() {
  char buffer[300];

  gets(buffer);
}

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

  return 1;
}
```