Secure and Dependable Systems Module: CO-566
Constructor University Date: 2024-03-04
Dr. Jürgen Schönwälder **Due: 2024-03-11**

Problem Sheet #4

Problem 4.1: *word count* (4+2 = 6 points)

An ambitious first year student, receives the task to write an implementation of the Unix/Linux command `wc` to print newline, word, and byte counts for each file. She tries to avoid a longer somewhat boring coding session and instead she writes a program that simply calls the existing `wc` command to do the work (and she believes that copying a solution from a generative AI system is below her standards). This is the code she hands in.

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

int main(int argc, char *argv[])
{
    char command[256];

    strcpy(command, "wc");
    for (int i = 1; i < argc; i++) {
        strcat(command, " ");
        strcat(command, argv[i]);
    }

    (void) system(command);
    return 0;
}
```

a) Describe four distinct security problems or problems that may lead to security problems (depending on the usage context). Demonstrate the problem by providing a command that causes the program to misbehave.

b) Write a program that implements the solution of calling the existing program `wc` correctly.

Problem 4.2: *compiler hardening options* (1+1 = 2 points)

Compilers often provide special options to mitigate some of the problems we discussed in class. For each of the following `clang` options, briefly describe what they do and demonstrate how they help to prevent some of the problems;

a) `-D_FORTIFY_SOURCE=2`

b) `-fsanitize=safe-stack`

Problem 4.3: *SQL injection* (1+1 = 2 points)

A course registration system uses the following database table to track the students registered for their exams. The table is defined as follows:

```
CREATE TABLE Exams (
     StudentName    VARCHAR(255) NOT NULL,
     ModuleName     VARCHAR(255) NOT NULL,
     ExamDate       DATE
)
```

The registrar's office hired a student to implement a web frontend listing all exams a student is registered for. The backend is written in Python and has the following code fragment:

```
c.execute("SELECT * FROM Exams WHERE StudentName = '{}'".format(name))
```

The variable `name` holds the student name passed to the backend. The `format()` function replaces the curly braces of the format string with the arguments passed to the `format()` function.

a) Jane Doe did not show up for the SADS exam on 2023-05-23, which now counts as a failed attempt. Jane tries to fix this problem by removing herself from the exam registration table. What is the SQL injection Jane enters into the web frontend as her name?

b) The instructor of record received an exam participation list before Jane unregistered herself via the SQL injection attack. The instructor discovered an inconsistency during the exam. Jane panics and she likes to clear all traces by dropping the entire table. What is the SQL injection to achieve this?