HW4

Aung Kyaw Tun

April 2024

1 Problem 4.1

a) There's a potential security vulnerability in this code due to the use of `strcat()`. If the total length of command-line argument exceeds 256 characters, it could result in a buffer overflow, causing undefined behavior. For example the command below would achieve buffer overflow

```
./wordcount `perl -e 'print "A" x 300'`
```

Also since the program uses user input to construct a command, and due to lack of sanitization and proper validation, it's susceptible to malicious code injection. For example

```
./wordcount arg1; rm -rf /
```

could potentially lead to deletion of important files, or users can inject malicious files or code in its stead. Since the program also uses `system()`, it's vulnerable to arbitrary code execution leading to unauthorized access or system compromise.

b) The following program calls the existing program wc correctly with the use of `unistd.h` file.

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

int main(int argc, char *argv[]) {
    char *args[argc + 1];
    args[0] = "wc";

    for (int i = 1; i < argc; i++) {
        args[i] = argv[i];
    }
    args[argc] = NULL;

    execvp("wc", args);
    perror("execvp");
    return 1;
}
```

2 Problem 4.2

Macro `-D_FORTIFY_SOURCE=2`, specific to `gcc`, enables additional security checks at compile time to detect some buffer overflow errors when using various string and memory manipulation functions like `strcpy, memcpy, strcat, memsets, gets`, etc. When enabled, it performs additional checks for buffer overflows and other vulnerabilities by fortifying certain standard library functions.

`-fsanitize=safe-stack`, used with `clang`, enables the SafeStack feature, designed to protect against stack based buffer overflows. It works by separating the program stack into two parts, safe stack and unsafe stack. Safe stack stores return address, register spills, and local variables that are always accessed in a safe manner and unsafe stack stores everything else. This ensures that buffer overflow on the unsafe stack cannot overwrite anything on the safe stack.

3 Problem 4.3

a) Jane Doe can remove her name from the Exams table by entering the following code:

```
'; DELETE FROM Exams WHERE StudentName='Jane Doe' --
```

b) Jane can delete the entire table by the following code injection:

```
'; DROP TABLE Exams --
```