Computer Security (COM-301)

Software Security

Interactive Exercises

Carmela Troncoso

SPRING Lab

carmela.troncoso@epfl.ch

MCQ Final 2022

Which of the following statements are true:

a) The WˆX policy, when strictly enforced, prevents code injection attacks.

b) Canaries aim to protect from both code injection and control flow hijacking attacks.

c) Control flow hijacking attacks are possible even when the adversary does not know the addresses of the system functions.

d) A fuzzing testing framework must choose between attempting to test all control flows or test all data flows.

Side Channels

```c
int calculate(char* input) {
char secretkey[128] = [0, 1, 0, 1, 1, ...];
int output = 0;
for(int i = 0; i < 128; i++) {
    if (secretkey[i] == 0) {
        output += fast_function(input);
    }
    else if(secretkey[i] == 1) {
        output += slow_function(input);
    }
}
return output;}
```

Describe a **testing methodology** based on two techniques seen in the class: fuzzing and sanitization to test whether `calculate`() is vulnerable to timing attacks or not.

If the testing technique cannot be used to assess the vulnerability, explain why.

Mission failed.. successfully !

Gru has written the code below to manage the bonuses of minions that participate in evil missions. Gru asks for your help debugging the function. Identify two lines that contain unsafe code that may lead to a memory safety error. For those two lines i) explain the vulnerability, and ii) explain whether a Minion can exploit this vulnerability to increase their bonus even when their mission did not succeed. Assume that Minions are good teammates and will never steal a bonus from another Minion by providing others' successful mission names.

```
int bonus[100] = { 0 }; /* array of 100 integers initialized to 0 */
char successDB(char* mission) { /* function that returns 0 for failed
missions and 1 for successful missions. If the mission does not exist, it
crashes */ }

1: int AddMission(int minionID) {
2:     char success;
3:     char mission[30];
4:     gets(mission); /* minion provides name of mission */
5:     success = successDB(mission);
6:     if (success == 1) { bonus[minionID] += 1 };
7:     return 0; }
```