Raluca Popa
Spring 2018

CS 161
Computer Security

Homework 1

Due: Monday, January 29th, at 11:59pm

Instructions. This homework is due **Monday, January 29th, at 11:59pm**. No late homeworks will be accepted unless you have prior accomodations from us. This assignment must be done on your own.

Create an EECS instructional class account if you have not already. To do so, visit https://inst.eecs.berkeley.edu/webacct/, click "Login using your Berkeley CalNet ID," then find the cs161 row and click "Get a new account." Be sure to take note of the account login and password, and log in to your instructional account.

Make sure you have a Gradescope account and are joined in this course. The homework *must* be submitted electronically via Gradescope (not by any other method). Your answer for each question, when submitted on Gradescope, should be a single file with each question's answer on a separate page.

Problem 1 ***Policy*** **(10 points)**

The aim of this exercise is to ensure that you read the course policies, as well as to make sure that you are registered in the class and have a working EECS instructional class account.

Open the course website http://www-inst.eecs.berkeley.edu/~cs161/sp18/.

Please read and check that you understand the course policies. If you have any questions, please ask for clarification on Piazza. Then, determine the answer to this question, and submit it for Q1 on Gradescope.

How many project "slip days" do you get?

Problem 2 ***Collaboration*** **(10 points)**

You're working on a course project. Your code isn't working, and you can't figure out why not. Is it OK to show another student (who is not your project partner) your draft code and ask them if they have any idea why your code is broken or any suggestions for how to debug it?

Read the course policies carefully. Based on them, determine the answer to this question, and submit it for Q2 on Gradescope. A one-word answer is fine: we do not need a detailed explanation (though you may provide an explanation if you choose).

Problem 3 ***Vulnerable code*** **(40 points)**

Consider the following C code:

```
1  void meow(char *arg)
2  {
3    char randomCharacterArray[16];
4    printf("Hi I am Laurel. What is your name?\n");
5    scanf("%s", randomCharacterArray);
6    printf("Glad to have you in the class, %s\n", randomCharacterArray);
7  }
8
9  int main(int argc, char *argv[])
10 {
11   char beg[37];
12   strcpy(beg, ``Good Morning! '');
13   char end[22] = ``How was your weekend?'';
14   strncat(beg, end, 22);
15   meow(argv[1]);
16   return 0;
17 }
```

1. What is the line number that has a memory vulnerability and what is this vulnerability called?

2. Just before the program executes the line in part 1, the registers are:

   ```
   %esp:  0xBFFFFB20
   %ebp:  0xBFFFFB48
   ```

   Given this information, describe in detail how an attacker would take advantage of the vulnerability. Also make sure to include the address that the attacker needs to over-write. (Maximum 5 sentences)

3. What would you change to fix the problem in part 1?

4. Given the code as is, would stack canaries prevent exploitation of this vulnerability? Why or why not?

Problem 4 ***Reasoning about memory safety*** **(40 points)**

The following code finds the index of the first zero in the array and returns -1 if one does not exist. For the following questions, please be exhaustive in your answer.

```
1  int findFirstZero(uint8_t *arr, int length)
2  {
3      for (int i = 0; i < length; i++)
4      {
5          if (arr[i] == 0)
6          {
7              return i;
8          }
9      }
10     return -1;
11 }
```

1. Please identify the **preconditions** that must hold true for the following code to be memory safe.
2. Please identify the **postconditions** that must hold true.
3. Please identify the loop **invariants** that must hold true.

Problem 5 ***Feedback*** **(0 points)**

Optionally, feel free to include feedback. What's the single thing we could do to make the class better? Or, what did you find most difficult or confusing from lectures or the rest of class, and what would you like to see explained better? If you have feedback, submit your comments as your answer to Q5.