Nick Weaver
Fall 2018

CS 161 Computer Security

Homework 1

Due: Wednesday, 19 September 2018, at 11:59pm

Instructions. This homework is due **Wednesday, 19 September 2018, 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/fa18/`.

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 ***Security Principles*** **(20 points)**

Considering the following security principles, identify which ones Bob violates and how: **Security is economics**, **least privilege**, **fail-safe defaults**, **separation of responsibility**, **defense in depth**, **don't rely on security by obscurity**, **design security in from the start**, **consider human factors**, **complete mediation**, **detect if you can't protect**, **proactively study attacks**.

Getting on the cryptocurrency hype, one day Bob decides to set up his own exchange. He sets up all the infrastructure, but worries about forgetting the password, so Bob hides his login credentials in an HTML comment on the login page.

Eventually, Bob manages to gather a large user-base and realizes his site looks like a back-end developer trying to learn CSS, so he contracts out front-end work to Mallory's Do-No-Evil design firm (for an incredible price too!). He gives them an account with access to his front-end and back-end codebase, and databases of user information as well.

Finally, his site looks amazing but now Bob is worried about his users' security, so he sets up emails to be sent on *every* user event (logins, web pages visited, transactions, messages, password changes, etc). Bob now rests, assured that his web app is one of the most secure to ever exist.

Unfortunately for him, one day he wakes up to his website being featured on a well-known news site after a data leak. Pressured by an internet mob, he hires a contractor to find all the issues with his site. However, fixing the website ended up being a different story, as much of the code was written (uncommented) in a late-night coffee-fuled frenzy, and Bob finds that he can't change any aspect of the website without breaking it in its entirety. In a panic, Bob announced the closure of his site and goes into hiding.

Problem 4 ***Vulnerable code*** **(40 points)**

Consider the following C code:

```
 1 void greet(char *arg)
 2 {
 3   char buffer[16];
 4   printf(‘‘I am the Senate. What is your name?\n”);
 5   scanf(‘‘%s”, buffer);
 6   printf(‘‘It's treason then, %s\n”, buffer);
 7 }
 8
 9 int main(int argc, char *argv[])
10 {
11   char beg[3] = ‘Obi';
12   char end[11] = ‘Wan Kenobi?';
13   strncat(beg, end, 5);
14   greet(argv[1]);
15   return 0;
16 }
```

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 5 *Reasoning about memory safety* (35 points)

Consider the following C code.

```
/* (a) Precondition: ____________________ */
void dectohex(uint32_t decimal, char* hex) {
    char tmp[9];
    int digit, j = 0, k = 0;
    do {
        digit = decimal % 16;
        if (digit < 10) {
            digit += '0';
        } else {
            digit += 'A' - 10;
        }
        /* (b) Invariant: _______________ */
        tmp[j++] = digit;
        decimal /= 16;
    } while (decimal > 0);
    while (j > 0) {
        hex[k++] = tmp[--j];
        /* (c) Invariant: _______________ */
    }
    hex[k] = '\0';
}
```

1. Please identify the **preconditions** that must hold true for the following code to be memory safe. In addition, the precondition must be as conservative as possible (e.g. `decimal` cannot be required to be solely zero). Justify why your given precondition cannot be any less strict.
2. Please identify the loop **invariants** (b, c) that must hold true and justify them as well.

Problem 6 *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 Q6.