**CS 3750 [Fall 2017]**

**Assignment 5**

**Due date: Monday, 23 October 2017, by 11:00 p.m.**

***Print your name:* Rick Boles**

**Answer the following questions:**

1. *BitLocker* is a program that supports encryption of file and directories using EFS encryption and full-disk encryption using AES encryption standard. It is suggested that when using BitLocker on a laptop, the laptop should not use standby (sleep) mode, rather should be in hibernate mode. Explain why. [5 points]

Hibernation requires you to re-authenticate making use of the encryption. Standby does not, which would circumvent BitLocker.

1. Consider an access control policy consisting of the following two rule:

Rule1: *A subject S can write to an object O only if security\_level(O) ≤ security\_level(S)*

Rule 2: *If a subject S reads an object O, then the security level of S is updated after the read as MIN{security\_level(S), security\_level(O)}*

Can the above policy prevent integrity violations in a system consisting of some subjects and objects? Explain. It is assumed that all subjects and objects have initial security level assigned. [5 points]

Yes, because Rule #1 requires the level to be at least *security\_level(S)* so when we look at Rule #2 it essentially takes the route of least privilege those that have more than or equal to *security\_level(S)* get *security\_level(S)* and those that don’t have the permission from Rule #1 keep their original permission. Which was not enough to read Object O.

1. Consider the following C-code fragment:

int main(int argc, char \*argv[]){

char passwd[8];

int continue = 0;

strcpy(passwd, argv[1]);

if(strcmp(passwd, “CS3750”) == 0)

continue = 1;

if(continue)

login(); /\* a method that logs in the user \*/

}

**Note**: *In C, \* indicates a pointer, which behaves similarly to a reference in Java. Therefore, char \*argv[] indicates argv as an array of pointers (references to memory addresses) to characters.*

1. Explain how an attacker can achieve buffer-overflow attack with reference to the variables passwd[] and continue.

Since the strcpy continues to store even when the memory allotted is full it will overwrite in this case “continue”. This can be done by entering “CS3750111” making continue = 1 by taking advantage of passwd only being able to store 8 we pass it 9 with the last digit being a 1 allowing anyone to login.

[5 points]

(ii) Explain the ideal ordering of the memory cells (assuming memory addresses increases from left to right) that correspond to these two variables so that this attack can be avoided. [5 points]

Swapping:

char passwd[8];

int continue = 0;

So that they are:

int continue = 0;

char passwd[8];

This will keep the overflow from affecting continue. It would be better altogether to use strncpy to avoid any segmentation faults / overflow;

**Submission instructions:** write your name at the top and include answer to each question on this document preferably after each question. Please do not write your answers on a separate document or file. Submit the file through BlazeVIEW dropbox.