CSE 378: 09 WI – Ceze

Homework 2: Buffer Overflow

Due: Monday, February 2, 2009 at 5:00 PM.

The goal of this assignment is to learn about MIPS functions, calling conventions, and stack layout, and exploit this knowledge to "take over" a program. You will write a function in the MIPS assembly language that smashes the program's stack frame and verify it in SPIM.

Exploiting a Vulnerable Program's Stack

This simple assembly program reads in a student's NetID, verifies that the NetID is in the correct format, and returns the class grade of the student. The problem with this program is that the local, stack-allocated buffer used for holding the NetID (inside the `string_sanitize()` function) is of fixed size, and the `replace_spaces()` function does not check that the destination buffer is large enough to hold what's copied into it. If the NetID is too big for the destination buffer, `replace_spaces()` will just keep writing past the end and overwrite whatever happens to be adjacent in memory.

You will write a function, `attack_string()`, to modify the `netid_buffer` so that a student is automatically awarded with a grade of A+ whenever the professor attempts to reduce his/her grade to a C-. To accomplish this, the `netid_buffer` must be changed such that when the program executes, the stack frame of `string_sanitize()` gets smashed and the return address of `string_sanitize()` gets overwritten. You need to inject the address of the `automatic_a_plus()` function on the stack so that when `string_sanitize()` returns, it transfers the flow of control to the `automatic_a_plus()` function and runs the code in `automatic_a_plus()`.

There are 2 places where you need to alter the source code. These places are marked with the word "TODO" in a comment. All other source code should remain intact.

Sample Run

The output of this unmodified code looks like this:

```
Accessing student record…
jnelson
Student grade: C-
-- Connection to Student Records terminated --
```

After you have implemented the `attack_string()` function, the program output should look like this:

```
Accessing student record…
jnelson
Student grade: A+
-- Connection to Student Records terminated --
```

Some Advice

- It may be helpful to sketch out the stack layout and size at the time that the `string_sanitize()` function is called, so that you know what value on the stack you have to smash.
- You will have to consider endian-ness issues when crafting your value for input_buffer

What you should do:

1. Start by downloading the grade_change.s file:
   http://www.cs.washington.edu/education/courses/cse378/CurrentQtr/homeworks/grade_change.s

2. Modify the parts of the code marked "TODO".

3. Load your program into SPIM and execute it. If your code works, you will see output indicating that your program correctly displays a grade of A+.

4. When you are satisfied with your solutions, you will turn in the grade_change.s file containing your modifications.

5. Submit your assignment via the Catalyst WebTools at:
   https://catalysttools.washington.edu/collectit/dropbox/iannacci/4564
   Please include your name at the top of this file in a comment.