CSE 5382-001: Secure Programming
Spring 2019, © TJ, UTA 2019

Programming Assignment 3 (Part 1 and 2) – Rev B
Due: Part 1 – March 22, 2018, 23:59 UTA time, Part 2 – April 10, 2018, 23:59 UTA time
Buffer Overflow

The goal of this assignment is to write and exploit buffer overflows.

This will be an individual assignment (no teams).

Part 1 (Due March 22):

Develop a useful program with one intentional buffer overflow. By a "useful" program, I mean that the program should simulate some real-world process or program (e.g. ATM, movie rental, classified system authentication, grade book, etc.). You will not be graded on the complexity of the program, but on the exploitability of the buffer overflow. The following requirements are mandatory:

- The program shall be written in C or C++.
- The buffer overflow vulnerability shall be exploitable to give access to a value or resource that is otherwise protected by the program or operating system (e.g. free money, unauthorized login/capability). Crashing the program or getting a shell prompt should not be the only exploit.
- The program shall be built and executed in a Linux environment. The Linux environment must be configured to disable non-executable stacks and address space layout randomization which will make buffer overflow exploits easier to accomplish. Students will be expected to perform all parts of this assignment in this environment. This can be a bare metal installed operating system or within a virtual machine. See the separate attachment on disabling countermeasures for information on how to disable NX and ASLR.
- Another thing to keep in mind is that, depending on the version of gcc, the Stack Protector feature may be turned on by default. Make sure to add the -fno-stack-protector flag during compilation to turn it off (if supported by the compiler).
- The program shall have no more than 10 inputs from the user (this is to prevent students from trying to hide the overflow in excessive inputs).

Submission:

For Part 1 of the assignment, you will be expected to submit two deliverables via Blackboard as described below:

1. Submit program to be exploited per the following instructions:

- All vulnerable programs must be submitted by end of day on March 22; this is so that we can begin exploiting them after March 22 (once Part 2 is posted), and everyone has the same number of days to attempt exploits.
- Each student shall create a zip file named <your name>.zip where <your name> is your first initial, middle initial, then last name.
- The zip file shall contain the source code for your program along with a pre-built executable (built in the Linux environment described above).
- Your source code shall not contain your name or student ID (it should be anonymous so other students don't know whose code it is).

2. Submit a simple "how-to" on EXACTLY how to exploit your buffer overflow… screen shots are appreciated. This is also due by end of day on October 21. **DO NOT** share this how-to with any classmates.

After March 22, the names of the student submitted zip files for Part 1 will be changed to random numbers and attached as a zip file to Part 2. Once this is done, students will be notified by e-mail.

Part 2 (Due April 10):

Once you receive the e-mail notifying you that the zip files have been posted to Blackboard, you can begin attempting to exploit each other's buffer overflow programs by downloading the zip files from Part 2 of the assignment into your Linux virtual machine, unzipping them into unique folders, and running the included binaries. Remember that you are exploiting other students' programs, so some may not work properly. If a program does not work or you believe it is not exploitable (after you've studied the source code), simply explain that in your write-up below. You must exploit 7 of the programs to receive full credit for this portion. Please work alone. Questions can be e-mailed to the GTA and/or the professor.

Submission:

Use Blackboard to turn in a document that explains your overall approach to exploiting buffer overflows and the specific methods for each successful exploit. The explanations should be brief but complete (i.e. "at the Enter Name prompt input 15 A's). In addition, explain the output/value gained by the buffer overflow.

Bonus Points:

In addition to the 7 programs that must be exploited, if you successfully exploit an additional 5 programs, you can receive **30** bonus points. No partial credit will be given (this is a step function).