CS 727/827: Computer Security
Lab 3: Buffer Overflow

Dongpeng Xu

Due May 13

1 Goal

The goal of this lab is for students to have a practical experience of buffer overflow attack. Students will learn to exploit a buffer overflow vulnerability and use it to escape the serial number checking routine in the program.

2 Description

1. Initial Setup.
   - You need a local Linux environment. You cannot do it on Agate because ASLR need to be disabled. If you do not have a native Linux environment, please install an Ubuntu using a virtual machine like VirtualBox or VMware.
   - Disable address space randomization. It might be different depending on your Linux system. Here are some possible ways.
     `echo 0 | sudo tee /proc/sys/kernel/randomize_va_space`
     `sudo sysctl kernel.randomize_va_space=0`
   - Disable StackGuard protection scheme.
     Add `-fno-stack-protector` when using `gcc`.
2. Download and read the c source file `bof.c` from Canvas. It reads a serial number from a file and check whether it is valid. If it is valid, the program will execute the full version function. Otherwise, it executes the trial version function. Describe where the buffer overflow vulnerability is and why it is a buffer overflow.
3. Create different input files to test whether it passes the serial number checking. (Hint: You need a hex editor.)
4. Use `objdump` to show the disassembly code of every function (main, chkserial, fullversion, trialversion).

5. Use `gdb` to debug the program. Show the stack contents (local variables, stack pointer, return address, parameters) in every stack frame when the program enters the function `fullversion`.

6. Create a file to trigger a buffer overflow and escape from the serial number checking. The program executes the fullversion function without a correct serial number.

3 Submission

Please pack the following files into one compressed file (.zip) and submit to Canvas.

1. A lab report (.pdf) including the description and screenshot of every step.

2. A make file for compiling your program.

3. The input file you are using for the buffer overflow attack.

4 Resource

- `objdump` is a disassembler installed in most Linux distributions. `objdump -help` will give you enough information for this lab.

- `gdb` is a popular debugger in Linux. You may find its documentation via the following link.
  `https://www.gnu.org/software/gdb/documentation/`.

- You will need to compile your program using `-g` option to create the debug version so that `gdb` can debug it.