CS 577 Cybersecurity Lab

Lab 3 – due 10/1/15 11:59pm

Subject: Hijack control-flow and inject code

You are given a set of toy programs that include various buffer overflows. Your goal is to write an exploit that makes use of the vulnerability to hijack control flow and inject your shellcode in the victim process.

Assumptions

For this assignment we will assume that certain defenses like stack protection, ASLR, and non-executable stacks are not in place. This is achieved using the following:

Disabled stack protection: the option *-fno-stack-protector* was passed to GCC when compiling the victim programs.

Executable stack: the option *-z execstack* was passed to GCC when linking the victim programs.

Disabled ASLR: the gdb debugger can disable ASLR to assist in debugging, so we are going to run the victim program through gdb to disable ASLR. gdb on *gump* exhibits this behavior

Note that the heap is **not** executable!

Vulnerable programs and utilities

You are given the lab3_files.tar.gz archive which includes two vulnerable programs and the s-proc utility to help you print the shellcode. You should gain control of the instruction pointer using a stack overflow and a heap overflow with the respectively named binaries. After you gain control you should execute shellcode that prints the string "Hello, world\n" to standard output and then exit by invoking the *exit(0)* system call. You can use the same shellcode in both cases.

Both vulnerable programs read input from standard input. You can generate and store your exploitation strings in a file, which you can then provide to the vulnerable applications. E.g., *./stack_overflow < exploit.txt* or in gdb *run < exploit.txt* .

Deliverables

1. Deliver the source assembly used for generating the shellcode for your exploit. You should be able to demonstrate how you created your shellcode from this file during your examination, if requested. You should also submit an exploit that includes the data/text that given to the vulnerable program will exploit it

and run your shellcode, as well as any utilities you developed to help you create the exploit. [80%]

2. Include a report.txt file explaining your choices when developing the exploit, the tools you had to create to help you in developing it, and the manual investigation that you had to do. [20%]

Submission information

The code you submit for grading must build and run on the linux-lab, even if you use a different machine/environment for developing. You should use a particular host in the linux-lab for this assignment, which we have tested and will more likely result into consistently working exploits.

Submit all your files as a tar.gz archive through Canvas.