Lab: Attack Lab

The Attack Lab: Understanding Buffer Overflow Bugs

This assignment involves generating a total of five attacks on two programs having different security vulnerabilities. The directions for this lab are detailed but not difficult to follow. We give a brief overview of the lab below, but you should:

1. View the AttackLab tutorial video
2. Download the attached PDF and use it as you step through the lab

Outcomes you will gain from this lab include:

- You will learn different ways that attackers can exploit security vulnerabilities when programs do not safeguard themselves well enough against buffer overflows.
- Through this, you will get a better understanding of how to write programs that are more secure, as well as some of the features provided by compilers and operating systems to make programs less vulnerable.
- You will gain a deeper understanding of the stack and parameter-passing mechanisms of x86-64 machine code.
- You will gain a deeper understanding of how x86-64 instructions are encoded.
- You will gain more experience with debugging tools such as GDB and OBJDUMP.

You will want to study Sections 3.10.3 and 3.10.4 of the CS:APP3e book as reference material for this lab.

Note: In this lab, you will gain firsthand experience with methods used to exploit security weaknesses in operating systems and network servers. Our purpose is to help you learn about the runtime operation of programs and to understand the nature of these security weaknesses so that you can avoid them when you write system code. We do not condone the use of any other form of attack to gain unauthorized access to any system resources.

2 Logistics

As usual, this is an individual project. You will generate attacks for target programs that are custom gener- ated for you.

2.1 Getting Files

You can obtain your files by pointing your Web browser at:

http://cs2400.cs.colorado.edu:15513/

The server will build your files and return them to your browser in a tar file called targetk.tar, where k is the unique number of your target programs. **Note:** It takes a few seconds to build and download your target, so please be patient.

As in other labs, you'll be able to see the scoreboard for the lab at

```
http://cs2400.cs.colorado.edu:15513/scoreboard
```

Save the targetk.tar file in a (protected) Linux directory in which you plan to do your work. Then give the command: **tar -xvf targetk.tar**. This will extract a directory targetk containing the files described below.

You should only download one set of files. If for some reason you download multiple targets, choose one target to work on and delete the rest.

Warning: If you expand your targetk.tar on a PC, by using a utility such as Winzip, or letting your browser do the extraction, you'll risk resetting permission bits on the executable files.

The files in targetk include:

- **README.txt:** A file describing the contents of the directory
- **ctarget**: An executable program vulnerable to code-injection attacks
- **rtarget**: An executable program vulnerable to return-oriented-programming attacks
- **cookie.txt**: An 8-digit hex code that you will use as a unique identifier in your attacks.
- **farm.c**: The source code of your target's "gadget farm," which you will use in generating return-oriented programming attacks.
- **hex2raw**: A utility to generate attack strings.

In the following instructions, we will assume that you have copied the files to a protected local directory, and that you are executing the programs in that local directory.

What To Hand-in / Upload

You should upload a TAR file that contains at least your targets (ctarget, rtarget) and your attack strings.

attacklab.pdf

Submission status

Submission status	No attempt
Grading status	Not graded
Due date	Wednesday, 10 October 2018, 6:00 PM
Time remaining	Assignment is overdue by: 67 days 5 hours

Last modified	-
Submission comments	Comments (0)