CSE 484 / M 584 Autumn 2021: Lab #3

Vulnerability Writeup Lab

Due: Wednesday, December 8, 11:59pm
Turn in: Canvas assignment
Individual or group: Groups of up to 3 (as in prior labs)
Points: 20 (+2 bonus available)

Before you start

- Sign up on Canvas for a Lab 3 group ASAP!
- Read the background below, it will contextualize things

Goal

This lab is designed to give you some experience with performing *root-cause analysis* (RCA) on exploits. Conceptually, this is similar to the process that might happen if your company discovered an exploit in-the-wild being used against an application you make. You can see examples of this in Google Project Zero's (P0) writeups of exploits found by TAG (Google's Threat Analysis Group.) We strongly encourage reading a few of these as examples for your writeup in this lab.

What this practically means is that you will need to take a 'working' but unexplained exploit, determine what bugs in the application are used by this exploit, and propose an appropriate set of fixes for the application.

The application is a small HTTP webserver written in C. It ~~probably~~ has vulnerabilities beyond the ones you need to explain. You will be given 3 things as part of this assignment:

- The C HTTP server
- Two working exploits against that server (sploit1 and sploit2)
- A normal connection example to the server (nonsploit)

For each exploit you will need to turn in a writeup that contains the following:

- An explanation of what the exploit achieves (what goal did it accomplish?)
- An explanation of what set of specific bugs allowed the exploit to work. This should cite specific lines of code and explain the bugs in detail.
- A proposed set of fixes to the application that solve the root cause of the exploit, but preserve all application functionality

We provide a standardized template (borrowed and modified from Google's P0) that you'll fill for each exploit you analyze. The template is included as a docx and a PDF.

Background

You work on an open-source project: tinyserv. It is not the best tiny HTTP server, but it works. Most of your users use it to serve small static webpages from their own private servers, and it has a fancy admin page that shows all previous visitor's requests! Handily, the admin page is password protected by a completely random password each time the server starts, so only the owner of the server can possibly access it.

But today you got 2 reports of exploits being used against tinyserv in the wild. Your users are in danger! It is time to quickly perform an analysis of the exploits, determine the bugs underlying them, and propose patches!

Exploit 1 is incomplete, clearly. The copy you've recovered appears to be missing the payload (no shellcode!) but it certainly looks exploitable. You'll need to at least identify the crash.

Exploit 2 doesn't seem to do anything bad to the server, no crash, no changes to any files. But it looks like someone has found a way to read the admin logs without logging in!

Notes

- You can build+run this on attu.cs.washington.edu.
- You cannot build+run this on codered or on OSX
- You do not need to modify the sploit files, or write any exploit code
- You will need to choose a different port number if you are running at the same time as another group member on the same machine. (Or if some other group is using that port number on the same machine)
- Don't leave tinyserv running when you aren't using it; it is quite vulnerable!
- You can access the admin login page by visiting tinyserv's `/login.html` page (e.g. 127.0.0.1/login.html in a browser if running tinyserv locally)