Lab I - Source Code Analysis

Software Security, DV2546

Edgar Lopez-Rojas*
Blekinge Institute of Technology
School of Computing

November 1, 2015

Contents

1 Introduction

This lab deals with real vulnerabilities in a real, although small, server. Your task is to analyse the server to find vulnerabilities, exploit these vulnerabilities and finally most important to learn how to prevent and fix them. This lab is divided into several sub-tasks.

1.1 Time Approximation

We estimate that a student on average will use 60 hours to solve this exercise. Much of this time will be spent in the security laboratory. A virtual machine of an OPEN BSD OS installation named DV2546 lab1 is available with all the necessary software in the security lab under the image prepared for the course (user: seclab pass: seclab). You can copy the appliance and feel free to use it on your own virtual machine.

2 Recommended Reading Before the Lab

We recommend that you read:

- The source code for bugpop3 and examine what the external functions used by bugpop3 do. The external functions (libc) that bugpop3 uses are:

  `atexit atoi close crypt endpwent exit fclose fgetln fopen fprintf getpeername getpwent getpwnam gettimeofday malloc memcpy open poll printf recv rename sleep snprintf strcasecmp strcmp strcpy strncasecmp strncmp strncpy strsep unlink vsnprintf` and `write`

- Post Office Protocol version 3 (POP3) http://www.ietf.org/rfc/rfc1939.txt

3 Background

The POP3 server you will examine is written in the C programming language and designed to work on UNIX-like operating systems. The environment used in the Security Lab for this exercise is OpenBSD 3.1. On the lab machine you will find three valid user accounts; alice, bob and admin. You are given local access to the machine as the unprivileged user alice (password: alice). You do not know the password to any of the other accounts. *Please note that it takes a few minutes before starting up the system until you can send email messages.*

*Originally created by: Per Mellstrand and Martin Boldt

Bugpop3

The pop3 server used on the lab machines is specifically developed for this exercise and contains several security vulnerabilities. Your main task is to find these vulnerabilities and exploit them. The server will always run with super-user (root) privileges and is only accessible from the local machine (i.e. you cannot contact the pop3 server on another machine). The source code to Bugpop3d is provided as a printed material by the teacher. You will find the source code useful when looking for vulnerabilities. In the recommended reading you can find the specification for the POP3 protocol.

4 Tasks

This lab is divided into four sub-tasks roughly ordered by difficulty where task 1 is the least difficult and task 5 the most difficult. To pass this exercise you must successfully solve all 4 tasks. For a higher grade (A and B) you must successfully solve task 5 or make a strong effort to solve task 5.

4.1 Task 1: Vulnerability repository (Before the lab)

There are several repositories for software vulnerabilities, a few of them are listed here:

1. Security Focus (`http://www.securityfocus.com/`)
2. Open source vulnerability database (`http://osvdb.org/`)
3. National vulnerability database (`http://nvd.nist.gov/`)

Your task is to search for a recent vulnerability (no more than 90 days) related with buffer overflow. Get the information from the web source and be able to analyse countermeasures, discuss the risk and motivate your choice. This vulnerability should be posted also in the labs forum with the corresponding URL in order to avoid duplicated solutions and share your analysis with other students. **OBS!** Duplicated cases will not be accepted.

4.2 Task 2: Read Bob's mail

The first task is to read Bob's email. Bob has at least one email message.

4.3 Task 3: Prevent Bob from reading his mail

The second task is to make it impossible for Bob to read his email though Bugpop3d. You do not need to prevent him from reading his mail file with a text editor. The attack must be permanent (i.e. it is not sufficient to log in as Bob and perform dummy commands) and persist even after you log out of the system account.

4.4 Task 4: Prevent valid user logins

The third task is to prevent the users alice, bob and admin from logging in on the machine from the physical console (i.e. the attached screen and keyboard). The attack must be permanent (persist a logout) and must not be based on resource exhaustion (such as trying to fork-bomb the machine or filling up the entire file system) or on physically destroying or damaging the computer.

4.5 Task 5 (Optional): Obtain root access

This task is possibly the most difficult task of the lab and it is to obtain `root` access on the machine. To successfully pass this task you must exploit a vulnerability in Bugpop3d to obtain this privilege.

5 Examination

You are encourage to work in groups of two (2) students or alone after approval from the lab staff. *Please note that the time estimations done for this exercise are based on the assumption that students work pair-wise in groups.* The group will be examined as a whole (i.e. both students must be able to explain the methods and techniques used by the group to pass examination). *You only need to write one report for the whole group, not one per student.*

5.1 Report

This exercise is examined with both a written and an oral exam. You should write a short report (typically about one or two pages per task) where you (for each task) describe;

1. Briefly explain the *methods, techniques and tools* you used to solve (or attempted to solve) the task (max 1 page). Please include references to any tool you have used that you have found yourself (i.e not provided by us in any image) and please also include source code to any tools you have developed.
2. A detailed description of *BUGS* found, *VULNERABILITIES* discovered and *EXPLOITS*.
3. Source code of exploit and proper reference if taken from another source. Screen shots or output of the exploit to show completion of the task.
4. A discussion of *COUNTERMEASURES* for these vulnerabilities, one or several suggestions of *how to fix* the vulnerable system if it was in use. Notice that in most of the cases there is an obvious solution, but a good report should contain more elaborated solutions that address different aspects of how to protect against an specific or similar vulnerabilities. We expect a discussion of methods tools and techniques that you feel could have prevented, detected or in any other way helped in the development of this software. This is important factor for the quality of the report and will affect directly the grade.
5. A general *REFLEXION* of what you learnt during this lab (between 100-200 words) explaining How long time (approximation in whole hours) you spent on each task and How important, hard or easy you found each of the task.

This report should be sent to the examiner and if the written report is of acceptable quality you can continue with the oral examination. Several time slots for the oral exam will be provided by the examiner. The report should be in clear and understandable English.

5.2 Plagiarism

With exception of your lab partner you are not allowed to cooperate with other students or groups in any way that compromise the specific solution of each part of the lab. We check for plagiarism with a platform that compare your report with other web resources and other students reports from this term and past terms. We will not accept any report that contains plagiarism. In certain cases we will be force to inform the students that incurred in plagiarism.

Saying this, we encourage all the students to properly reference any source of information used to write the report and also to use their own words in the reflections and explanations of the steps to obtain the results.

5.3 Oral Examination

During the oral exam the group of students should explain and defend their report. After the oral exam you will be notified by the examiner if you passed or failed the examination. If you fail you must use another examination time and send in the report for that as if you were submitting the report for the first time.

During the oral exam you might also be asked to explain the source code of your solution, i.e. go through the source code and explain for the examiner what each line does. It is therefore very important that you *store all* source code, byte-code or binary code that you have produced or used when solving the task on the secured server. Note: you *must* be able to describe all code you have used to solve a lab or part of a lab.

5.3.1 Important issues regarding oral examination

The oral examination is an important part of the Software Security course. Please read this section carefully and be sure to follow the rules to avoid any problems during the examination.

- We are running oral examinations on a very tight schedule and cannot accept students arriving late for their examination. Please try to arrive at least ten minutes before your examination time.
- If you are working in a group both student must be present at the examination. Please remember that the group is examined as a whole and that the examiner freely can choose which student to direct a question to.
- You might be asked to demonstrate a part of, or the complete, solution to the exercise. *You must be prepared for this and have tools (scripts) developed to automate any step required.* Such tools must be available on your *seclab* account and should typically execute in less than 1 (one) minute.

5.4 Grading

For this lab you can get grade A-F, where:

1. **Grade A** requires that you finish the optional task, AND have a really good report AND can describe it well during the oral examination.
2. **Grade B** requires that you have a really good report AND attempt with a strong theoretical explanation of the optional task AND can describe everything very clear during the oral examination.
3. **Grade C** requires (no optional task) that you have a really good report AND can describe it well during the oral examination.
4. **Grade D** requires (no optional task) that you have either a good report OR can describe it well during the oral examination.
5. **Grade E** requires (no optional task) that you have a sufficient report AND can describe it during the oral examination.
6. **Grade F** when you do not fulfill the tasks required at least for grade E.

In all the cases you must fulfil tasks 1-4.

... And Finally

Good Luck!