Lab III - Web Security

Software Security, DV2546

Edgar Lopez-Rojas
Blekinge Institute of Technology
School of Computing

November 1, 2015

Contents

1 Introduction

This lab is based on the project OWASP Broken Web Applications (OWASPBWA https://www.owasp.org/index.php/OWASP_Broken_Web_Applications_Project). OWASPBWA contains 3 types of applications: first training applications, second realistic intentionally vulnerable applications and finally old version of real vulnerable applications. In order to get access to OWASP a virtual machine should be set up by downloading the exportable image from their website and using an application such as VM Ware or VirtualBox (recommended) virtualbox.org.

1.1 Time Approximation

We estimate that a student on average will use 40 hours to solve this exercise. Much of this time will be spent in the tutorials. A virtual machine of OWASPBWA named DV2546 lab3 is available with all the necessary software in the security lab under the image prepared for the course (user: seclab pass: seclab). You can also download the appliance and feel free to use it on your own virtual machine.

2 Recommended Reading Before the Lab

We recommend that you read:

- Chapter 17 Gray Hat Hacking. Chapter 9 and 12 Web Applications Hacker's Handbook.
- OWASP Top 10 Vulnerabilities 2013 https://www.owasp.org

3 Background

The Broken Web Applications (BWA) Project produces a Virtual Machine running a variety of applications with known vulnerabilities for those interested in:

- learning about web application security
- testing manual assessment techniques
- testing automated tools
- testing source code analysis tools
- observing web attacks
- testing WAFs and similar code technologies

4 Tasks

This lab is divided into four tasks roughly ordered by difficulty where task 1 is the least difficult and task 4 the most difficult.

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 injection or top 3 vulnerabilities from OWASP. Get the information from the web source and be able to analyze 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. Duplicated solutions will not be accepted.

4.2 Task 2: OWASP Broken Web Applications - WebGoat

Enter the application Webgoat with username: guest password: guest. Get familiar with the environment. Describe briefly what is the WebGoat project. Play around and successfully complete 3 tasks. You can use the hints and the tools provided to "cheat" but remember to put some real effort on understanding what vulnerabilities and what the exploits do. This will be asked during the oral examination.

4.3 Task 3: OWASP Broken Web Applications - Mutillidae

Mutillidae is a collection of Deliberately Vulnerable Web Pen-Testing Application. Your task here is to go through the OWASP Top 10 and perform the 2 of each of the examples for the Top 3 vulnerabilities and 4 more of your choice (10 forms in total). Explain clear the purpose of the form, the exploits used to perform the tasks and countermeasures required to fix them.

4.4 Task 4 (Optional and will bonus first lab): Old vulnerable applications

Select one of the applications from the section "OLD (VULNERABLE) VERSIONS OF REAL APPLICATIONS. Find at least 2 vulnerabilities on those and explain them in detail. HINT: You can make use of repositories and search for the bugs of the version you are using.

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

There is only a Pass (G) or Fail (U) grade for this lab. To pass this exercise you must successfully solve the first 3 tasks. For a bonus grade on one of the previous labs you must successfully solve task 4. In order to pass you must complete both the report and the oral examination.

... And Finally

Good Luck!