Homework 2: Exploits!

Due date & time: Due at **22:00 on November 17, 2018**. This is a firm deadline. This project MUST be finished independently.

1 Overview

It is time to show your hacking skills! In this assignment, you will have a bunch of web pages to attack. Your mission is to find and exploit 8 bugs in total.

More specifically, you are asked to locate bugs in our self-crafted web applications according to the categories of vulnerabilities listed in Table A and Table B. You are expected to find as many bugs as possible. There are 3 parts:

- In part A, there are 6 cases, including XSS, remote code execution, SQL injection, Local File Inclusion (LFI only), CSRF predictable tokens, and mixed content.
- In part B, you are asked to choose 2 out of 3 cases to finish. They are about session management and authentication, authentication flaws, and HTTP Requests.
- Part C is an optional part. Optional cases are only for further understanding, and they **will not** be graded. There are 23 cases in part C. You need to find the bugs in test cases and category them.

Some of the cases may have more than one bug. Just choose either one for mentioned bug category to exploit. If you claim and exploit more than one bug in a case, only one will be graded, and the rest will be discarded.

This assignment counts 15 marks (15%) towards your final grade.

2 Environment

The VM can be downloaded from the following link:

- A2.zip: https://goo.gl/ID6EAD

2.1 VM information

The information for the VM is as follows:

- MD5 checksum: (see a2.vdi.md5 in the zip file)
- OS Account:
    - username: student
    - password: student
- Homepage: http://www.wsb.com/Assignment2
- Source codes: /var/www/html/Assignment2/
- Database: http://www.wsb.com/phpmyadmin
- Management account:
    - username: root
    - password: student
- Developing account:
    - username: cs5331
    - password: v29AcujVSDKWadxe

If you need to access the database, you are recommended to use root and access via phpMyadmin (http://www.phpmyadmin.net/) or other MySQL clients.

3 What to exploit

3.1 Part A

In part A, you are asked to exploit 6 cases: case04, case06, case09, case24, case25, and case31. In Table A, we show the bug category and your target for each case in part A. A "flag" is a secret we embedded in the vulnerable application, which you need to find and output it using the exploit.

Case No.	Bug Category	Target
04	Persistent XSS	flag in cookie
06	Mixed Content	flag in cookie
09	CSRF Predictable tokens	prepare a script with a simple form to submit a message to case09.php
24	SQL Injection	flag in db
25	Local File Inclusion (LFI only)	flag in lfi.txt.php at Assignment2 directory
31	Remote Code Execution	cat /etc/passwd

Table A. Part A

3.2 Part B

In part B, you are asked to choose 2 out of 3 cases: case14, case23, and case32. If you submit all of 3 bugs, only the first 2 bugs will be considered for grading. In Table B, we show the bug category and your target for each case in part B.

Case No.	Bug Category	Target
14	Parameter pollution	flag on webpage when polluted
23	Logic authentication flaws	escalate your privilege to get the flag
32	Execution after redirect	flag output by server

Table B. Part B

3.3 Part C

Part C is an optional part. This part will not be graded. In Part A and Part B, we give

you the bug category for each test case. Now it is time for you to find and category bugs in the rest 23 test case.

4 Grading criteria

In this assignment, you have to:

- Identify the bug found and report. (0.5 point for each case)
- Explain how to exploit the bug found and document what you have observed. (1 point for each case in Part A; 1.5 points for each case in Part B).
- Except case 06, show that the bug is exploitable by providing an attack script (a bash script) which will automatically exploit the bug. We already specify the target in each category in Table A and Table B, e.g., alert the flag, echo the etc/passwd, etc. (2 points). For case 06, you only need to describe how the attacker can get the flag in cookie.

A sample of the attacking script is in the folder /var/www/html/Assignment2/sample. Please take a look to see how to structure yours. You may use third-party libraries in the scripts.

For every bug reported, the TAs will execute your corresponding bash script and check if it achieves the target stated in Table A and Table B. When you create the script for attacks, you should not make any assumption about the TA's prior knowledge. We just blindly click on your bash script. If all scripts in part A and part B works, you will get these 2 points. We will check the validity of the script as well to see if you craft your attack correctly, e.g., you cannot manually copy the cookies from the browser and put it in an alert box.

You are only allowed to find bug and create exploit for the mentioned categories, without utilizing any other vulnerabilities in the system. For example, bugs pertaining to VM's user password guessing/cracking or using pre-configured secrets in the VM (e.g., SSL private keys, root passwords) are not considered this assignment. Apache infrastructure exploits, browser compromise, etc., are all out-of-scope. Please focus on web application vulnerabilities instead of the underlying server-browser infrastructure.

5 Submission instruction

Create a folder called *code*, and place setup script (e.g., *setup.sh*), exploit scripts (e.g., *exploit04.sh*, *exploit06.sh*, *exploit09.sh*, etc.) and any other scripts needed by the exploit script. Create a folder called *report*, and place your report in it (report must be in pdf format). The report must contain your matric number (including the last alphabet), and your full name at the top.

Create a folder with your matric number, put folder *code* and folder *report* inside it, and zip the folder, and upload it to IVLE workbin's Homework 2 submission folder.

For example, if my matriculation number is A1234567Z, the zipped file should be *A1234567Z.zip*, and should contain two folders inside it, named *code* and *report*.

Your submission deadline is **17th November 10:00 pm SGT**. This is a firm deadline. Start early, submit early!