Assignment 2: Web Application Security Programming: SQL Injection and XSS

FIT5003 Software Security S2 2020

Faculty of Information Technology, Monash University

Submission Guidelines

- **Deadline:** Assignment 2 Report submission is due on **Sunday 8 th November 2020, 23:00**.
- **Demonstration Deadline:** The demonstration via Panopto platform or the interview must be shared/-conducted before Sunday 8 th November 2020, 23:00.
- **Submission Files:**

  1. A report in PDF file format of maximum 6 pages as a reference. Appendices and References are excluded from the page count.
  2. Appropriate code as an answer to relevant tasks

  Notes:

  1. A handwritten document is **not** acceptable and will **not** be marked even if converted and submitted electronically.

- **Individual Assignment:** This is an individual assignment so each student should work on the assignment tasks alone.
- **Submission Platform:**
  - Electronic submission via Moodle for the report.
  - Electronic submission via Moodle for the assignment task code and apks
  - Share of video interview files with your tutor via the Monash panopto service
- **Filename Format:** Name your files for different assignment tasks as follows:

  1. Submission via moodle: report_SID.pdf
  2. Sharing via Panopto: A2_interview_SID as title, in case of an interview. **Please follow the provided structure for avoiding mark reduction**
  3. Submission via moodle of assignment task files: use A2_task_number_SID and the appropriate file extension for the relevant task

- **Late Submission Policy:** Submit a special consideration form to formally request a late submission. For this semester, special consideration requests should be send directly to the faculty and not just the tutor team. However, do inform the teaching team of your request.
- **Late Submission Penalty:** A late submitted assignment without prior approval will receive a late penalty of 20% deduction per day (including Saturday and Sunday) or part thereof, after the due date and time.

- **Plagiarism:** It is an academic requirement that your submitted work be original. Zero marks will be awarded for the whole submission if there is any evidence of copying, collaboration, pasting from websites, or copying from textbooks.
  Note: Plagiarism policy applies to all assessments.
- **Grading Procedure:**
  - To receive a grade for the assignment you must demonstrate and explain your work by creating a video recording of **maximum 15 minutes** using **Panopto platform** and share it with your tutor.
  - You must only demonstrate what you have submitted via report.
  - If you have any privacy concern regarding the Panopto platform then you need to raise it with your tutor by **Wednesday 28 th October 2020**. Requests for interviews after this date will not be accepted.
  - You can use the report and any other notes you have prepared beforehand to help you explain and demonstrate your work.
- **IT Use Policy**: Your submission must comply with Monash University's IT Use Policy.

Marks

- This assignment is worth **20%** of the total unit marks.
- The assignment is marked out of **20** nominal marks.

For all the following tasks you must demonstrate the attack on the provided VM in a recorded (Panopto or Zoom) where you explain the attack while performing it.
For all tasks of the assignment, the VM can be found in the url appearing on Moodle

In this VM, the mutillidae web application server has been setup (it is a variation of the one used in the labs).
The password to be used in order to access the VM is
`qBsv+A6md$uwP3qM`

1 Task 1: Blind SQL Injection (8 Marks)

Extending the program that was provided in the last task of Week 9 lab, write a standalone python program to perform SQL injection in the Login page of the Mutilladae web application server of the provided VM (OWASP 2017 -> A1 Injection (SQL) -> SQLi - Bypass Authentication -> Login).
Your Goal is **to extract the password for the individualized username that corresponds to your studentID** using a blind SQL injection.
To find the individualized username use the link that exists in the unit's Moodle. **Hints**:

- You can use Burpsuite to notice that the application responds differently for a valid and invalid SQL query
- Observer the type of HTTP message that is sent from the client to the server and back and try to reproduce it in your python script.
- Consider using the `substring` sql function. e.g. `'or substr(@@version,1,1)=1` will bypass authentication, think how it can be leveraged to extract the password

2 Task 2: XSS Injection (8 Marks)

Consider that you are an authenticated user (use your studentID as username and as password the one that you've found on the first task). Use your credentials to login to the Mutilladae web application server. Exploit reflected XSS (OWASP 2017 -> A7 (XSS) -> Reflected (First Order) -> DNS Lookup) and change password for your username (student ID).
The answer to this task should contain a JavaScript file and a URL containing the initial script. **Hints**

- As a starting point you can use the file and approach of the task 2.2 in the week 10 lab
- Observer the type of HTTP message that is sent from the client to the server and back and try to reproduce it in your javascript file.

3 Task 3: SQL Injection Countermeasures (4 Marks)

Given the previous SQL injection, modify the Mutillidae PHP code, so that it is no longer vulnerable to SQL injection at Login. There are various countermeasures to be infused, to get the full mark you need to implement the most potent of such countermeasure.