Project 1 - eHealth Corp

Description

This assignment will focus on the existence of vulnerabilities in software projects, their exploration and avoidance. The objective is for students to develop a simple web page for a health clinic. The frontend should show basic information regarding the services, the possibility to contact the clinic, request appointments with doctors and download test results by providing a code. A backend should also be implemented with any functionality relevant to this purpose. Students may also add other functionality such as chat bots/helpdesk, videos, notifications, registration process, etc… Take in consideration that the web page should not show errors to the users, and should provide some basic functionality.

However, this application should also suffer from a specific set of weaknesses, which are not obvious to the casual user, but may be used to compromise the application, or the system.

Students should provide a both a flawed and correct version of the application, together with a report demonstrating how those vulnerabilities are explored and their impact. The project must include vulnerabilities associated with CWE-79 and CWE-89. An additional set of weaknesses must be considered, so that the sum of the CVSS of the vulnerabilities is at least 35.

For all vulnerabilities:

- Vulnerabilities should be distinct and have distinct CWEs;
- The CWE must be identified;
- The implementation must follow the logic and purpose of the application. That is, no page with the single purpose of showing the vulnerability;
- Students should be able to demonstrate the vulnerability in a report with scripts/screenshots;
- It is preferred to have vulnerabilities that result from bad patterns instead of those resulting from something that is missing. Avoid things like absence of brute force protection/access control/encryption/logging

A bonus of 10% can be provided if the vulnerability can be attributed to a bug (developer can repudiate having authored the vulnerability).

It is expected that a user can fully understand the purpose of the application, and use it. Implementation can be simple and some functions may be missing (e.g. backend may be only a webpage representing a restricted intranet). After reading the report, a reader should be able to understand the application, the vulnerabilities, their exploration and impact, and how they can be avoided.

The project is expected to be implemented by **a group of 4 students**, and **MUST** reside in a private repository in the github/detiuaveiro organization, using the Github Classroom functionality (this is mandatory).

Project delivery

Delivery should consist of a git repository with at least three folders and a file:

- `app`: contains the insecure application, including instructions to run it (Docker is recommended);
- `app_sec`: contains the secure application, including instructions to run it (Docker is recommended);
- `report`: contains a document (PDF, MD) describing the project, the vulnerabilities with their score and fix;
- `analysis`: contains scripts/textual descriptions/logs/screen captures demonstrating the exploration of each vulnerability;
- `README.md`: contains the project description, authors, enumerates vulnerabilities implemented.

Projects will be graded according the implementation and exploration of the flawed code, the implementation of the secure code, and the documentation produced. The participation in the Github repository may be taken in consideration, and individual students may have different grades.

The use of automated tools to scan the application is not forbidden. However, grading will mostly consider your contributions (code, effort, analysis) not on the findings (as they are deliberate).

This project is expected to be authored by the students enrolled in the course. The use of existing code snippets, applications, or any other external functional element without proper acknowledgement is strictly forbidden. Themes and python/php/javascript libraries can be used, as long as the main page and vulnerabilities are created by the students. Actually, we recommend students to get a free website template and use it.

If any content lacking proper acknowledgment is found in other sources, the assignment will not be graded and current rules regarding plagiarism will be followed.

References

- OWASP Top 10
- CWE@MITRE
- SQLMap
- Nikto
- OWASP ZAP

2022

PREVIOUS
Lab - Linux Secure Storage

Last updated on 22 Sep 2022