Assignment #2

Access github library and download the code

https://github.com/HCL-TECH-SOFTWARE/AltoroJ

Follow the prerequisites and Importing AltoroJ into Eclipse from GitHub instructions.

After successful running of the application, which should be almost similar to the following website: https://demo.testfire.net/ you will need to perform the following tasks as penetration tester and software engineer:

1. Spot 18 vulnerability (0.5 mark per vulnerability) = Total 9 Marks
2. Fix the found vulnerabilities into the code.
3. Do the retest to make sure that those vulnerabilities no longer exist.
4. (FIX + Retest 0.5 Mark per vulnerability) = Total 9 Marks
   + 2 Marks for environments setup (upon first vulnerability fix)

https://portswigger.net/burp

Deliverables:

Report contains the following:

1. Cover Page.
2. Findings List with severity, CVSS risk score and risk categorized
   Example

ID	Name	Original Severity	After Retest Severity
ID01	Cross site request forgery (CSRF) in Money Transfer	High	Resolved
ID02			

3. Finding details (test and retest), including the description, CVSS risk score, CVSS risk string, actual impact, and tailored recommendations
   This section details the findings identified during the engagement and lists the potential impact, risk rating and recommendations.

   Example:

   Name: ID01: Cross site request forgery (CSRF) in XXX
   Test Severity: High
   Test Score: use the calculator to get the score https://www.first.org/cvss/calculator/4.0
   Retest Severity: High
   Description: What did you find in the application: + Definition of CSRF

   Impact:

   The impact of this vulnerability is severe; successful exploitation leads the attacker to transfer from user account to any account

   Recommendations:

   The remediation recommendations are listed below in details:

   - **Implement CSRF Protestation**

   https://www.first.org/cvss/calculator/4.0

4. Finding scenarios section (test. fix and retest).

    a. **Finding Test Steps:**
    ID01

        - Open the following page + screenshot
        - Used Burp Suiteto intercept the request + screenshot
        - Added the following payload + screenshot
        - The script has run, and therefore, there is XSS vuln here + screenshot

    b. Fixing Steps
    ID01

        - I spot the input in the code
        - I made input validation and/or output encoding as following + screenshot

    c. Finding RE-Test Steps:

    ID01

        - Open the following page + screenshot
        - Used Burp Suite to intercept the request + screenshot
        - Added the following payload + screenshot (like the original test)
        - The script has NOT run, and therefore, there is NO XSS vuln here. And the issue is fixed + screenshot

-

Submission Guidelines:

- You should work in teams of 5 (minimum 4)
- ONLY the team leader should submit a Zip file under the name: ***<G#_TeamLeaderName_TeamLeaderID>***
- Team members must all be from the same lab (or have the same TA)
- Cheating is NOT tolerated by any means

-

Deadline:

- Date: Tuesday, 7th of May, 2024
- Time: 11:59 PM