University of East Anglia
COMPUTING SCIENCES

Module: CMP-7038B DEVELOPING SECURE SOFTWARE
Coursework 1: 002 Secure Development Project and Presentation (Group)

Set by:	Debbie Taylor: Debbie.taylor@uea.ac.uk
Checked by:	Dr Jeannette Chin
Date set:	21 January 2022
Value:	70%

Date due:	18 May 2022, 15:00, Week 12
Returned by:	15 June 2022
Submission:	Blackboard and demonstration/presentation

Learning outcomes

- Understand how ethics should play an essential part of software design
- Develop a secure and usable website that meets the needs of the user
- Analyse the effectiveness of a range of security methods and tools
- Analyse the evolving range of threats associated with the internet

Specification

Overview

The aim of this assignment is for your group to code a secure usable web-based blog system that mitigates, at minimum, the 5 most common security vulnerabilities of account enumeration, session hijacking, SQL Injection, Cross-site scripting and Cross-site request forgery.

You will ***work in the same groups of 3 as you did for assignment 001,*** to code and secure the blog using ***JavaScript and Node.js***. At minimum the blog will require registration and login authentication, search functionality, the ability to add, edit and delete posts. You can use pre-built security libraries, but you must *clearly explain* these during your demonstration and client report.

There will be regular formative SCRUM meetings with your Client, to discuss your ongoing progress, with the Lab Leaders playing the part of the Client. Therefore, everyone in your group needs to attend all lab sessions.

Description

In your groups, you are required to develop a small, secure and usable, web-based blog site that mitigates various security vulnerabilities. At minimum you should defend against the 5 most common vulnerabilities of:

- Account enumeration
- Session hijacking
- SQL Injection
- Cross-site scripting
- Cross-site request forgery

Your group needs to concentrate on the *security aspects* of a web-based blog and not web development, as you only need to produce a basic front end. This will be used to evidence your security processes and mitigations, therefore the front-end does not have to be pretty, just functional.

To secure your blog you must include hashing/salting, encryption and code an additional authentication method to username and password. This could include a graphical password, CAPTCHA, or 2FA, etc. The more complex and usable the additional functionality, the more marks available, however you must not sacrifice usability for complexity.

You also need to include ethical considerations and discuss these in your client report.

Additional marks will be available for groups that mitigate other potential vulnerabilities and/or attacks, alongside ensuring the blog remains usable throughout.

You must code your website using JavaScript and Node.js. Any Node framework (such as Express) is acceptable. You can also use pre-built security libraries but these must be clearly and concisely explained as to, how they work, what they secure against and exactly how they provide security protection. *If you cannot explain your library use, your group will not attain any marks for that mitigation.*

Each mitigation *must also be valid across the whole website*, e.g., you cannot mitigate SQL injection and then break it later when mitigating another vulnerability.

Trello will be used to monitor development progress and evidence/support any engagement issues.

Maximum 4-page Client Report: This must be written for your client, a Company Director who has limited cyber and development knowledge. You should include code snippets, diagrams, screenshots, algorithms, etc. to help the client visualise and understand your discussion points.
At minimum you must clearly and concisely discuss:

- How ethics are essential to secure web development.
- What mitigations were coded for each security vulnerability, and should include discussions of pre-built libraries, hashing, encryption, usability etc.
- What Authentication method/s were coded, and evidence of how they increase both usability and security.
- Discussion and evidence of user testing (should include these on a test plan).
- Discussion and evidence of system unit testing (including a completed test plan).

You must use the UEAcmpstyle LaTeX template and Harvard/APA referencing and not rely on technical jargon, as your client will not understand this. All discussions must be supported with good quality, recent and relevant, referencing. References are excluded from the maximum 4-pages and the only appendix allowed for this report is for the test plan.

(MSC only) Maximum 2-page Testing Report: This should discuss a theoretical critical analysis of different testing techniques and strategies and answer:

- Testing strategies implemented for your project, supported by academic evidence of why you chose your specific strategy
- What different approaches can software and web development companies employ to test the cyber security of their product(s)?
- Out of these approaches is there an ideal/common method used in Industry?

Again, you must use the UEAcmpstyle LaTeX template and Harvard/APA referencing to evidence your discussions and decisions. References are excluded from the maximum 2-pages and there are no appendices allowed for this report.

Relationship to formative and summative assessment

This assignment builds on the formative work you complete during your weekly lab sessions. There will also be regular lab-based SCRUM meetings, run by lab leaders acting as the Client, therefore, all group members are expected to attend all lab sessions.

Deliverables – Week 12

Blackboard upload – 3pm Wednesday 18th May

Each group member must upload a single zipped folder, including the following, to blackboard by 3pm Wednesday 18 May 2022, using the format of `studentID-GroupNumber.zip` e.g. *`10000000-PgtGroup01.zip`*.:

- **System Code and README document** – README should explain how to load and run the code.
- **Client Report** – PDF of the maximum 4-page Client Report with each group member's ID shown on page 1.
- **1 page Engagement report** – Stating scores out of 10 for each group member and a short paragraph for each group member, supporting the figures stated. (Blank form uploaded to Blackboard)
- **(MSc Only) Testing Report and Critical analysis** – PDF of the maximum 2-page report, Again, with each group member's ID shown on page 1.

Please Note: *Any upload after 14.59.59 will automatically be classed as a late submission*, when the score is added to EVision. Faculty members cannot waive late submissions therefore please ensure you leave yourself enough time to submit, or you will automatically accrue a late submission penalty.

Engagement Scoring:

- Final scores will be allocated based on engagement and this will be evidenced via the regular SCRUM meetings, individual engagement report and Trello.
- *If you have problems with any group member's engagement, do not leave it until the submission date, you **must share your concerns with the lab leaders during your normal lab sessions**, so the MO can chase for engagement and the issue can be logged.*
- During submission, each group member should include a single page PDF titled ***Engagement*** and this needs to show each member's contribution to the assignment (including your own) and a short paragraph explaining why.
- This should show engagement out of 10 e.g., if everyone provides equal work and effort then each student should show as 10/10 on the PDF and everyone will be allocated the full group score.
- If, however, a student does not complete the same amount of work as the others, you should rate them out of 10 accordingly (ie about half as much work, 5/10; barely engaging 2/10 etc). The scores across the 3 group members will be averaged and the relevant student will achieve a lower score.
- Example: One group member achieves an average of 3/10 and this is supported
- by both Trello and information provided during lab sessions. That student will be allocated 30% of the overall final group score e.g.,
    - 60% overall score
    - 1 student scored 10/10 from all members – scores 60%
    - 1 student scored 3/10 from all members – scores (60*0.3)% - scores 18%
    - 1 student scored 3/10 from 1 member, 4/10 from another, and 5/10 from themselves, averaging at 4/10 - scores (60*0.4) - scores 24%
- If there are any significant disputes, the engagement scores will be allocated by the markers based on the engagement reports, trello, and lab participation.

Demo marking – during lab sessions in week 12:

- Each group will be allocated a maximum 15-minute timeslot (12-minutes for the demo and 3-minutes for questions), during the lab sessions in week 12, to demo the system functionality.
- The demo must be structured to show the system working for both front and back-end security mitigations, evidence of authentication, encryption, hashing/salting, usability, etc.
    - The front end must show evidence of the security issues you have protected against, and system usability, from a user's viewpoint.
    - The back end must show and discuss the code required to mitigate each

vulnerability e.g., algorithms, functions, encryption, hashing/salting, etc., and include discussions of pre-built libraries used.

- You also need to show front and backend authentication methods used.
    - The front end must evidence the authentication, and usability, from the user's viewpoint
    - The backend must show and discuss the authentication processes coded.
    - You should also briefly cover any ethical considerations required during your development process.

- You must also show your Trello board, when discussing relevant information during the demo.
- ***Please note:*** The demo presentation has a maximum of 12-minutes, and nothing after this time will be marked. Everyone in the group must take part in the demo.

Resources

Lecture notes and previous lab sessions are highly relevant to this work. Lab leaders will be available to help with general queries.

An example UEAcmpstyle LaTeX document has been uploaded to Blackboard

Erickson J., (2008) *Hacking: The art of exploitation*. 2nd edition San Francisco: No Start Press

- OWASP Top 10 – 2020. Available at: https://owasp.org/www-project-top-ten/
- Anderson R. (2008) Security Engineering: A Guide to Building Dependable Distributed Systems, Wiley, and also available on the author's website (https://www.cl.cam.ac.uk/~rja14/book.html)
- The DBLP bibliography server (https://dblp.org/) is an excellent resource for most topics (it holds bibliographic data and links to 4.8M computing publications

A note on use of additional sources and plagiarism

Please note that while use of texts, or online sources, is encouraged in order to learn design and programming principles, use of functions, lists, etc., are not a substitute for completing the work yourself. It is not appropriate to find solutions or part solutions to assignments and submit them as your own work. Neither is it allowed to post questions on online forums requesting help or solutions to specific assignment tasks. To do either (copying/requesting) would be in breach of the university's regulations on plagiarism and collusion (General Regulation 18).

In the instances where you do use code (or any other work) copied from any source, you must acknowledge the source (e.g. including a comment with the URL and author alongside the copied sections). If in doubt, approach the coursework setter to discuss what is appropriate

Marking scheme – Secure Development Project : 70/100

Marker Name:	**Group Number:**

Marking Details	**Mark %**	**Comments**
Demo: Vulnerabilities and Usability Clear explanation/demonstration of front and back-end mitigations: • Account enumeration – 3 • Session hijacking – 6 • SQL Injection – 6 • Cross-site scripting – 6 • Cross-site request forgery – 6 • Ethical considerations - 3 • System usability – 3 • Extra mitigations – 7	40	
Demo: Authentication • Encryption - 3 • Hashing/Salting - 2 • Authentication method – 7 • Authentication usability - 3	15	
Demo: Quality • Content – 2 • Delivery and structure – 2 • Trello - 2 • Question Answering - 4	10	
Client Report about system Clear explanation of security: • Ethical considerations - 3 • Vulnerability mitigations, incl pre-built libraries, hashing/Salting, encryption, usability – 6 • Evidence of user testing – 4 • Evidence of system testing – 4 • Grammar, spelling and references – 3	20	
(MSC Only) Testing Report and Critical analysis • Testing strategy chosen - 4 • Different testing approaches available - 5 • Ideal/most common method and why – 3 • Grammar, spelling and references – 3	15	

Extra Comments: **Total Score**