Cardiff University

School of Computer Science and Informatics

CM3203 - Individual Project Report

Initial Plan

Creating a Cyber Security Educational Platform for Developers

Author: Alexander Hayman

supervised by

Dr. Yulia Cherdantseva

February 7, 2022

1 Project Description

1.1 Problem

Web applications are often a point of interest for cyber criminals as they are publicly available online and personal data, user credentials and financial information. Once the application is compromised, hackers can pivot around the network where the website is being hosted and target more machines. Almost 50% of the data breaches begin in the Web Application and that there is a 34% increase in the number of data breaches each year [1]. It is more critical for developers to secure their web applications.

There are many free resources available online such as PortSwiggier which provide detailed knowledge about the various attack methods and how to mitigate them. There is also a growing market in gamification of cyber security learning. Websites such as HackTheBox and TryHackMe provide real life scenarios for ethical hackers to test their enumeration and exploitation of systems including web applications and provide a leaderboard system. These resources are useful for people in the cyber security field however it does not benefit developers who want to improve their security awareness as they provide brief information on how to patch vulnerabilities.

1.2 Project

The main aim for the project will be creating an educational platform to teach developers how to test web applications for security risks and how to fix them. The platform will consist of a vulnerable web application where developers will use cyber security attack methods structured around the OWASP Top 10 to find and patch vulnerabilities. A tutorial of how to perform and mitigate each attack method will be provided. For the technology stacks, PHP and MYSQL will be used for the backend of vulnerable web application as it is simple to create misconfigurations . And for the tutorials, the Google Code Labs tool will be used. As the web application will contain vulnerabilities on purpose, users will need to fetch the source code and host the application locally. Documentation will be provided to help with the installation process.

1.3 Background Information

The OWASP Top 10 is the standard document for developers and web application security. It consists of the top 10 web application security risks It has recently been updated in 2021 due to an increase in new exploits and frameworks. The risks are ranked by the difficulty of exploitation and the impact of the exploit. Risks that are easy to exploit and have a huge impact are ranked higher.

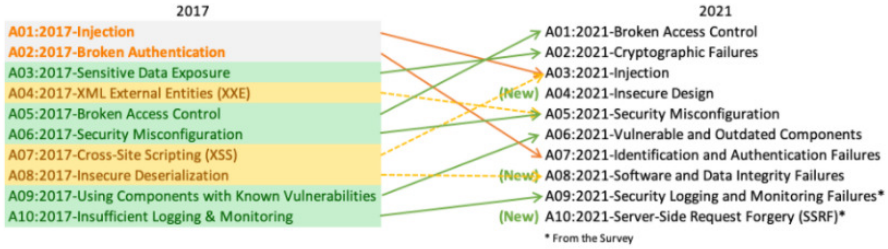

Figure 1: OWASP Top 10

2 Project Aims and Objectives

2.1 Primary Aims and Objectives

- Research modern attack methods surrounding web applications
  - Look at resources surrounding web exploitation such as portswigger and OWASP
  - Read up on the impacts of the attack methods and how to mitigate them
  - Practice exploiting web applications by using the OWASP Juice Shop training platform
  - Order vulnerabilities to implement on the application by their impact and ease of implementation
- Develop a Vulnerable web application
  - Implement at least five vulnerabilities for the system
  - Design a theme surrounding the web application (consistent colour scheme, base application on a fake company)
  - Create features on the website that are susceptible to attacks (login system, submit forms, admin account, upload feature)
  - Vulnerabilities implemented should be possible to patch
  - Provide documentation for the installation process
- Create a set of CodeLabs for the application
  - Research how to use and configure Google CodeLabs Tool
  - Provide a set of Codelabs for each attack method
  - Each Codelabs will provide an overview of the attack method, how to use the method and a solution to mitigate it

2.2 Secondary Aims and Objectives

These aims and objectives will be considered if all the primarily aims have been completed during the implementation phase.

- Create a scoring system
  - Provide flags for successful exploitation and patching of the system
  - Implement centralised submission portal where User's can submit flags for points
  - Create a progress system for Users to see how many vulnerabilities are left on the machine
- Implement a deployable machine system
  - Users would not need to get the source code but instead deploy an external machine hosting the web application
  - Implement a VPN network where users will connect to the machine with a VPN configuration
  - Isolate the hosted machines to prevent pivoting if the machine gets compromised

3 Work Plan

The gantt chart will give the time frame of the primarily objectives to complete excluding the report. Some features are given more time than required to plan around worse case scenarios such as unexpected bug fixes.

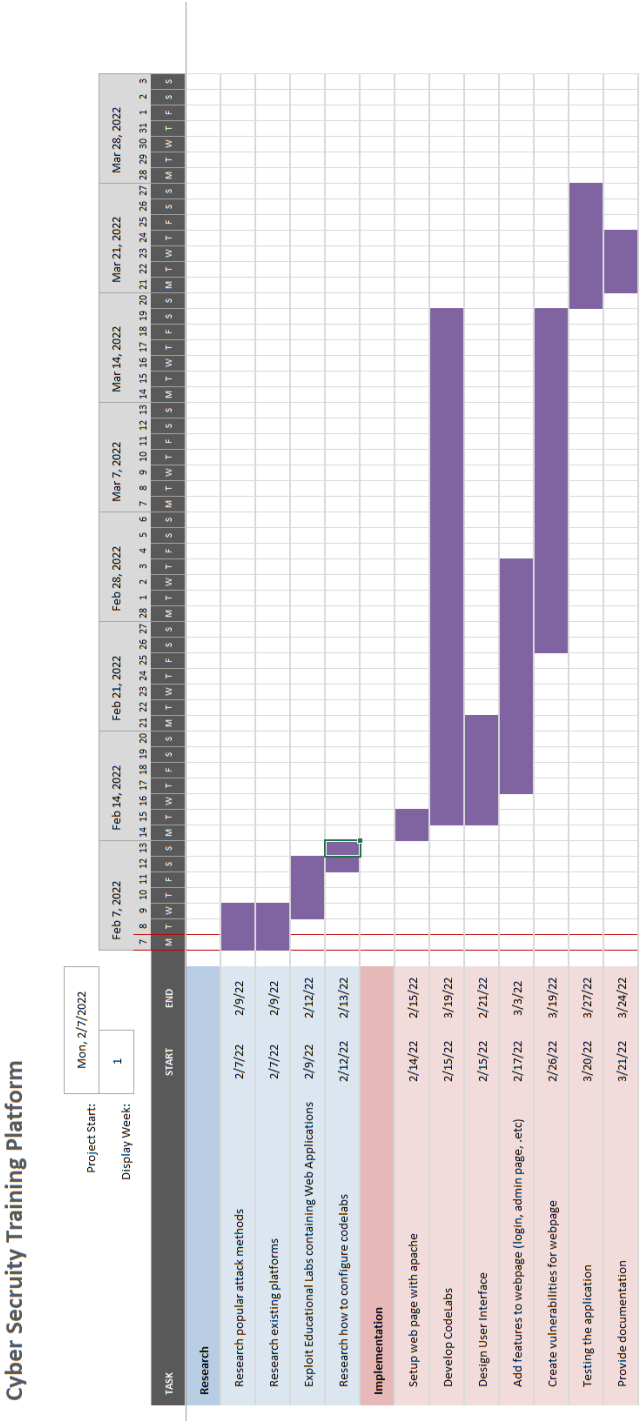

Figure 2: Research and Implementation Gantt Chart

Week	Tasks	Milestone
Week 1	Complete Initial Plan, Research relevant framework and programming languages, **Supervisor meeting**	
Week 2	Start research for the application, Review existing academic literature, **Supervisor meeting**	Initial Plan Submitted
Week 3	Start implementation of the Vulnerable Web Application and Create Codelabs, Make notes about design	Designed User Interface
Week 4	Work on Application and CodeLabs, **Supervisor meeting**	
Week 5	Work on Application and CodeLabs	First vulnerability and CodeLabs implemented
Week 6	Work on Application and CodeLabs, **supervisor meeting**	
Week 7	Work on Application and CodeLabs, **supervisor meeting**	
Week 8	Testing Application and CodeLabs,	Application and CodeLabs fully functional
Week 9	Start work on report, **supervisor meeting**	Begin Final Report, Introduction draft completed
Easter Break	Work on report	Literature review and Research Methodology draft completed
Week 10	Work on report,	Design and Implementation draft complete
Week 11	Work on report, **supervisor meeting**	Evaluation and Conclusion draft complete, full draft report completed
Week 12	Make edits to draft report, **supervisor meeting**	Final Report Submitted

References

[1] Ofir Shaty. *Lessons learned from analyzing 100 data breaches*. URL: https://www.imperva.com/resources/whitepapers/Lessons-learned-from-analyzing-100-data-breaches-Report-210603-v3.pdf. (accessed: 04.02.2022).