**Software report (5%)**

**Feature Description:**

* **How the code functions**
* **Any backend technologies used**
* **How data is transferred**
* **Other important feature – conditional checks and security implementation**

**Secure Development Life Cycle – Security Modelling & Testing (10%)**

The following section of the report pertains to the security policies and procedures adopted by the development team. We’ve chosen to emulate a similar secure development life cycle used by Microsoft. Our SDL has been tailored to the size of our project and encapsulates the most important aspects of the complete SDL used by Microsoft.

**Development Policies:**

1.Use of comments.

2.Research common vulnerabilities associated with libraries before implementation.

3.Test new features before updating repository code.

4.Attempt to fix, mitigate or remove comprised features in a timely manner.

5.Ensure any redundant code is removed from the final software product.

**Threat Modelling:**

Research similar web application as a background procedure to prevent potential bugs in your own design.

Highlight libraries that are used/will be used in the development and their overall security risk.

Design High-level application model to provide visual representation – (lecture 8, page 26)

List high value assets that require protection – example, server, database, and source code.

List threats ranked by risk compared to what features our website has and what libraries are used.

List how these risks will be mitigated or attempted mitigation. Actions – do nothing, remove feature, turn off feature, warn user or counter technology (ie. Conditions written into code).

**Continuous Development Testing:**

This section requires a table containing all website features, tests used, results of the tests and changes made.

**Bug Bar:**

Defines what bugs should be fixed, left or mitigated.

Sets a standard at which the software is secure enough

Catorgories – Critical, Medium and low risk

**ASA (Attack Surface Analysis):**

ASR (Attack surface reduction):

Aims to reduce the possible mechanisms used by attackers- example, adding conditional statements for what character can be entering into input fields on website.

ASR Flowchart (Lecture 8, page 20)

**FSR – Final Security Review:**

Review the website once completed – talk about persistent security flaws

**Security Response Planning:**

\*Locate security vulnerability

\*Create DFD to solve the entire issue rather than fixing the individual bug caused by a common action performed by user such as enter their name in the search bar.

\*Fix the bug and update the source code

**Evidence Of Repository Usage – GitHub (5%)**

**GitHub Commands**

**SSD Work Log**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Time | TODO | Priority (1-5) |
| 23/02/2022 | 9:40 am | Build website home page | 5 |
| 23/02/2022 | 12:00 pm | Build Login/ registration page | 5 |
| 23/02/2022 | 1:00 pm | Build Portfolio & contact page | 5 |
| 23/02/2022 | 3:00 pm | Start adding CSS and dynamic content sizing | 3 |