NGEE ANN
SCHOOL OF INFOCOMM TECHNOLOGY

Secure Software Development

Year 2 (2022/23), Semester 3

SCHOOL OF INFOCOMM TECHNOLOGY
Diploma in Cybersecurity & Digital Forensics

Assignment
Secure Web Application

Duration:	Weeks 11 - 15
Weightage:	40%
Individual/Team/Both:	Both
Deadline:	**31 July 2022, 11:59pm (Sunday)**

Group Name	**Team SAMPLE**
Student IDs	**Student 1xxxxxxx, Student 2xxxxxxx, Student 3xxxxxxx,**
Student Names	**StudentA, StudentB, StudentC**
Module Group	**P0x**

Table of Contents

1 Project Title

The team was engaged in developing a secure web application for XXXX, an organization that aims to help people learn new skills with online courses. Hence, the theme of the application for this project is XXXX XXXXX.

2 Project Vision

The project vision is to properly secure the web application using the knowledge we had gain during the Secure Software Development Course and apply these skills throughout the Secure Software Development Life Cycle (SSDLC). The target of the development team is to design, plan, implement, and test the web application to ensure that malicious attempts to ……………………………

3 Project Description

The objective of this web application is to offer …………………

4 Project Features

Based on the business requirements, the web application consists of mainly four features to implement which are authentication, authorization, auditing, and a CRUD Database.

Firstly, for authentication, this feature is the ……………………

5 Members & Features Worked On

Member	Features
StudentA	Signing up, Logging in, Authenticating users
StudentB	Authorization
StudentC	CRUD Database Auditing

6 3-Tier Application Architecture

In this phase, an overview of the application architecture is carried out by diagramming the application and identifying the attributes of the application.

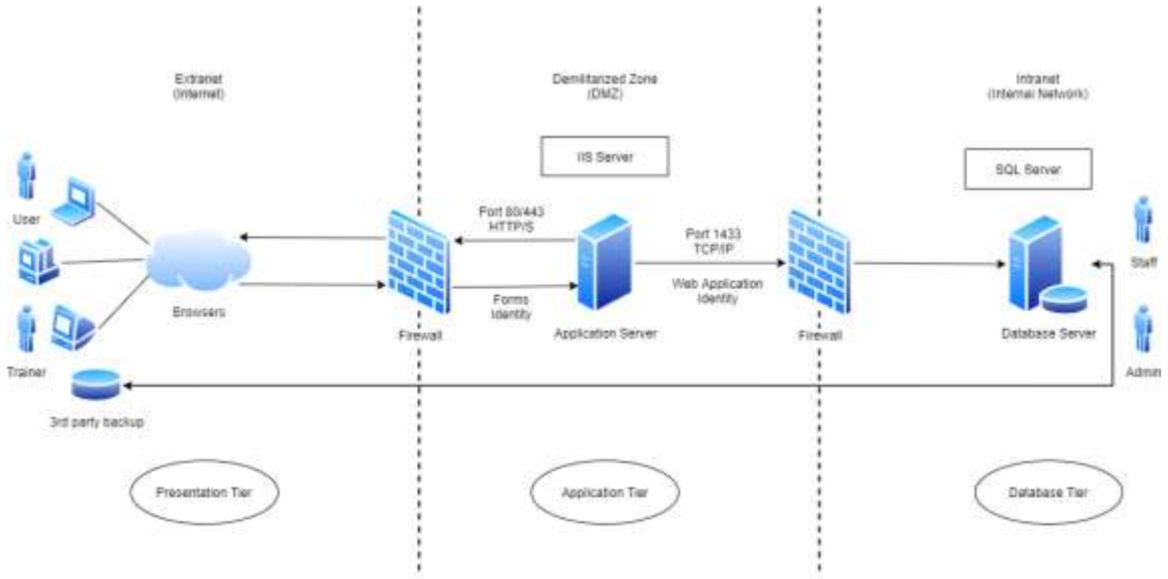


Figure 1 – XXXXX 3-tier application architecture

The software will be logically designed as……………….

.

7 Feature 1 (Login/Registration/Authentication):

Authentication is crucial for security as it enables organizations to keep their networks secure by allowing only authenticated ………………..

7.1 Secure Software Requirements

In authentication, Forms authentication is most commonly seen in online applications, …………………….

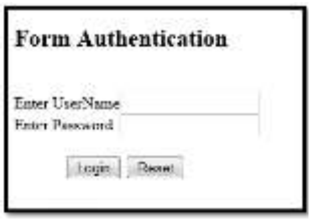


Figure 7.1 – Forms Authentication (Yadav, 2013)

7.1.1 Use Case and Misuse Case Modelling

The method of use case modelling helps in determining software functionality and security requirements. A use case describes …………………..

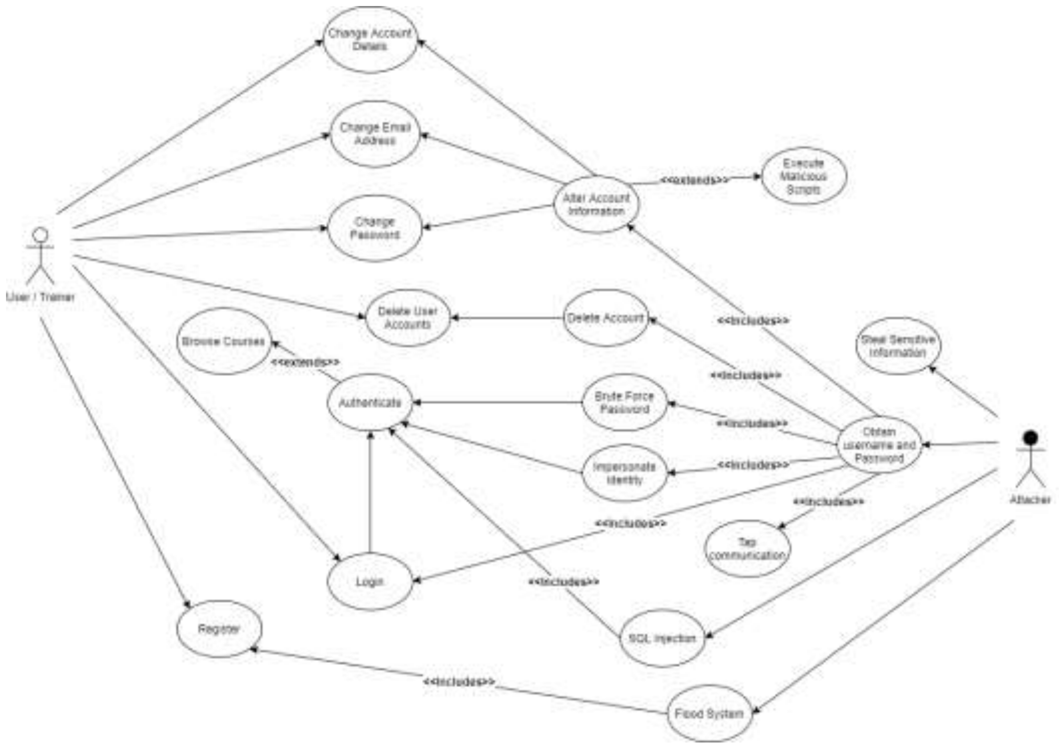


Figure 7.1.1 – Authentication: Use and Misuse Case for XXXX

7.1.2 Data Labelling & Classification

Data and information might be regarded as the most …………………

Name	Information
User Information	Full name, Date of Birth, Email Address
User Credentials	Username and Password
Privileged Account Information	Full name, Date of Birth, Email Address
Privileged Credentials	Username and Password with Admin Access to Application

Figure 7.1.2 Data involved in Authentication

Data	Impact of loss of		
	Confidentiality	Integrity	Availability
User Information	Public	Low	Support
User Credentials	Restricted	High	Critical
Privileged Account Information	Confidential	Low	Support
Privileged Account Credentials	Restricted	High	Critical

Figure 7.1.3 – Authentication: Data Classification and Labelling

7.2 Secure Software Design

Prevention of unauthorized disclosure of protected information can be achieved through cryptographic and masking techniques which was discussed in secure ……………………………………………………….

7.2.1 Threat Modeling

Software faces several threats. Authentication bypass, impersonation, and session hijacking are just a few of the dangers. This section of threat modeling will address the threats identified from the previous sections and implement control measures based on its severity.

7.2.1.1 STRIDE List

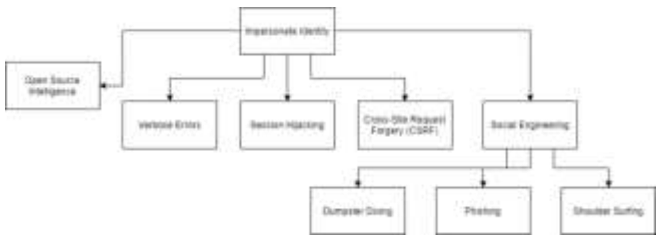
Based on the Use Case and Misuse Case, along with the identified attack surface areas addressed in the previous section, a STRIDE category threat list is used for categorizing the potential threats identified as depicted in Figure 2.3.1.

STRIDE List	Identified Threats
Spoofing	• Impersonate Identity
Tampering	• Alter Account Information • Alter Users Access • Execute Malicious Scripts
Repudiation	
Information Disclosure	• Tap Communication • Steal Sensitive Information
Denial of Service	• Flood System
Elevation of Privilege	• Brute Force Password • SQL Injection

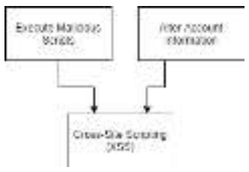
Figure 2.3.1 – Authentication: STRIDE List

7.2.1.2 Attack Tree

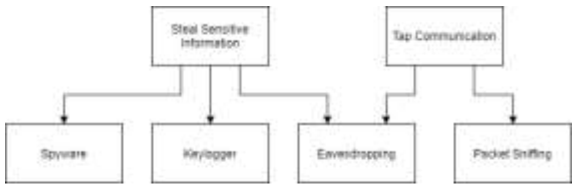
Expanding on the STRIDE List earlier, an attack tree model allows for brainstorming of potential threats from the attacker's perspective. This further strengthens the security of the software by identifying the attack path an attacker my take and implement countermeasures to mitigate or prevent these attacks.



Authentication Attack Tree: Impersonate Identity


Execute Malicious Scripts
Alter Account Information
Cross-Site Scripting (XSS)

Authentication Attack Tree: Execute Malicious Scripts & Alter Account Information



Authentication Attack Tree: Steal Sensitive Information & Tap Communication

7.2.1.3 Average Ranking

Following the identified threats, calculating the average of numeric values assigned to risk ranking categories is another approach for rating the threat's danger shown in Figure 2.3.2. DREAD is one such risk ranking classification methodology as depicted in Appendix A: DREAD.

Threat	D	R	E	A	DI	Average Rank (D+R+E+A+DI)/5
Open-Source Intelligence	2	3	3	2	3	2.6 (High)
Verbose Errors	2	1	2	3	1	1.8 (Medium)
Session Hijacking	2	2	2	1	3	2 (Medium)
CSRF	3	1	1	1	1	1.4 (Medium)
Dumpster Diving	2	3	3	2	2	2.4 (High)
Phishing	2	3	3	2	3	2.6 (High)
Shoulder Surfing	2	3	3	2	3	2.6 (High)
XSS	3	3	3	3	3	3 (High)
Spyware	2	2	2	3	2	2.2 (Medium)
Keylogger	2	2	2	2	2	2 (Medium)
Eavesdropping	2	1	2	3	2	2 (Medium)
Packet Sniffing	2	2	2	2	2	2 (Medium)
Automated Scripts	3	2	2	1	3	2.2 (Medium)
Brute Forcing	2	1	1	3	2	1.8 (Medium)
SQL Injection	3	3	2	3	2	2.6 (High)
Logic Flaws	1	1	1	2	1	1.2 (Low)
High: 2.1 – 3.0; Medium: 1.1 – 2.0; Low: 0.0 – 1.0						

Figure 2.3.2 – Authentication: Average Ranking

7.2.1.4 Control Measures

Following the threat modeling process, ……………………..

7.3 Secure Software Coding

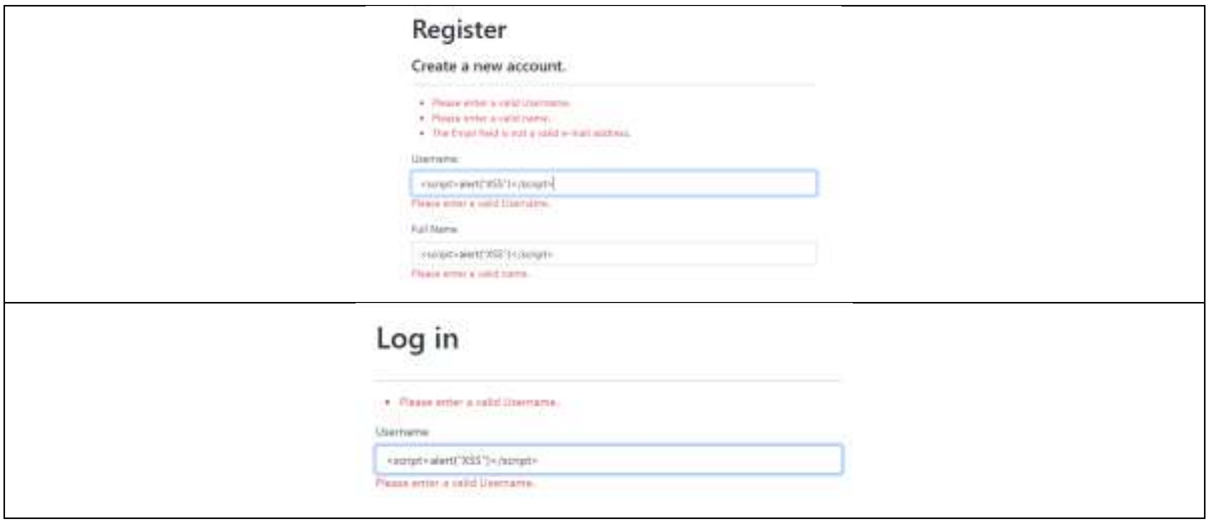
In the previous section, secure software requirements and secure software designs for the authentication feature was addressed, …………………………

7.3.1 Feature related Defensive Coding Practices

Since the entry points and exit points are identified, safeguarding the attack surface area of authentication must be of priority. According to Microsoft ASP.NET Core documentation on preventing cross-site scripting (XSS), the best practice is to always validate untrusted input while also encoding output displayed at the front-end HTML before it is presented to the end-user (Anderson, 2018). More about defensive coding practices in the software will be discussed in this section.

7.3.1.1 Input validation

Validation is a powerful technique for preventing XSS attacks. As stated previously, all input must be validated. This means that input that are not whitelisted as allowable values are immediately rejected. Input validation can be performed in the software by using a…………….



Register

Create a new account.

- Please enter a valid Username.
- Please enter a valid name.
- The Email field is not a valid e-mail address.

Username

<script>alert("XSS")</script>

Please enter a valid Username.

Full Name

<script>alert("XSS")</script>

Please enter a valid name.

Log in

- Please enter a valid Username.

Username

<script>alert("XSS")</script>

Please enter a valid Username.

Figure 2.4.1 – Input Validation on Register and Login Page

7.4 Security Conclusion of the Feature

Throughout the Secure Software Development Cycle (SSDLC), the login, registration, and authentication feature had been thoroughly analysed and all identified attack surfaces (entry points and exit points) ………………..

8 Feature 2 (Authorization)

This Authorization feature is a basic & necessary implementation in any application including the web application in this assignment. …………….

8.1 Secure Software Requirements

There are multiple Access Control Models that can achieve the Authorization requirement that includes Discretionary Access Control (DAC), Non-Discretionary Access Control (NDAC), Mandatory Access Control (MAC), Resource-Based Access Control and Role-Based Access Control (RBAC).

………………………………………………………………………………………………..
………………………………………………………………………………………………..

8.1.1 Brainstorming

Brainstorming is a quick but vague method to determine possible security requirements to extend of the individual’s knowledge. Hence, it is a foundational step to …………………………………...

9 Conclusion

Overall, ……………………………

10 References

Anderson, R. (2018, February 10). *Prevent Cross-Site Scripting (XSS) in ASP.NET Core*. Retrieved from Microsoft: https://docs.microsoft.com/en-us/aspnet/core/security/cross-site-scripting?view=aspnetcore-3.1

Digicert. (n.d.). *Behind the Scenes of SSL Cryptography*. Retrieved from Digicert: https://www.digicert.com/faq/ssl-cryptography.htm

Hasan, F., Anderson, R., & Smith, S. (2019, May 12). *Prevent Cross-Site Request Forgery (XSRF/CSRF) attacks in ASP.NET Core*. Retrieved from Mircosoft: https://docs.microsoft.com/en-us/aspnet/core/security/anti-request-forgery?view=aspnetcore-3.1

Yadav, A. (2013, May 8). *Form Authentication: ASP.NET Security Part 3*. Retrieved from Infosec: https://resources.infosecinstitute.com/topic/form-authentication-asp-net-security-part-3/

11 Appendix A

11.1 DREAD

Damage Potential (Ranks the damage that will be caused when a threat is materialized, or vulnerability exploited)

1. Nothing
2. Individual user data is compromised or affected
3. Complete system of data destruction

Reproducibility (Ranks the ease of being able to recreate the threat and the frequency of the threat exploiting the underlying vulnerability successfully)

1. Very hard or impossible, even for administrators of the application
2. One or two steps required; may need to be an authorized user
3. Just the address bar in a web browser is sufficient, without authentication

Exploitability (Ranks the effort that is necessary for the threat to be manifested and the precautions, if any, that are needed to materialize the threat)

1. Advance programming and networking knowledge, with custom or advance attack tools
2. Malware exists on the Internet, or an exploit is easily performed using available attack tools
3. Just a web browser

Affected Users (Ranks the number of users or installed instances of the software that will be impacted if the threat materializes)

1. None
2. Some users or systems, but not all
3. All users

Discoverability (Ranks how easy it is for external research and attackers to discover the threat, if left unaddressed)

1. Very hard-to-impossible; requires source code or administrative access
2. Can figure it out by guessing or by monitoring network traces