# **Message Board Application(Threat Model**)

Zaid Fadel

Grand Canyon University

SWE-550-O500

08-Dec-2022   
**Document Version 1.0**



.

This Paper prepared for SWE-550-O500

Table of Contents

[Message Board Application(Threat Model) 1](#_Toc121412724)

[Git Sample code: 2](#_Toc121412725)

[Application Name and Description 3](#_Toc121412726)

[Security Objectives 3](#_Toc121412727)

[Application Overview 3](#_Toc121412728)

[End-to-End Deployment Scenario 3](#_Toc121412729)

[Roles 3](#_Toc121412730)

[Application roles are: 3](#_Toc121412731)

[Key Scenarios 3](#_Toc121412732)

[Technologies 4](#_Toc121412733)

[Application Security Mechanisms 4](#_Toc121412734)

[Application Decomposition 4](#_Toc121412735)

[Trust Boundaries 4](#_Toc121412736)

[Identified trust boundaries are: 4](#_Toc121412737)

[Data Flows 4](#_Toc121412738)

[Entry Points 5](#_Toc121412739)

[Exit Points 5](#_Toc121412740)

[Threats 5](#_Toc121412741)

[Vulnerabilities 5](#_Toc121412742)

# Git Sample code:

https://github.com/Zaidusa/Threat\_Modeling\_Assignment/blob/master/hello.py

# Application Name and Description

The Message Board application (Threat Model) is an Internet-facing Web application with a SQL DB as back end.

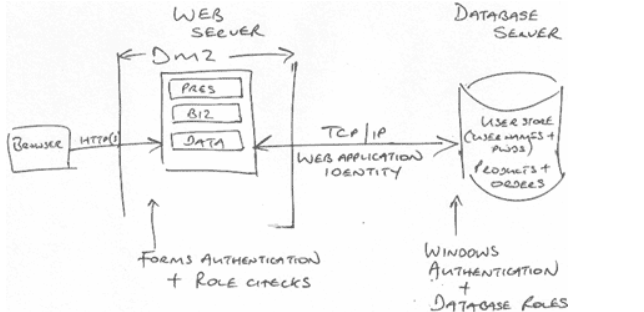
Security Objectives

* Protect customer account details and customer credit history.
* Ensure that the application is available 99.99 percent of the time.
* Prevent unauthorized users from Login

Application Overview

The application is an Internet-facing Web application with a SQL DB back end. The Web server is located in a perimeter network. Business and data access logic resides on the Web server. The application enables internet users to share messages online browser

# End-to-End Deployment Scenario



# Roles

## Application roles are:

* Internet users
* Application administrators

# Key Scenarios

* Check user can able to write a message on the message field till a limit.
* Check whether users can able to delete the existing message or not.
* Check the users can able to forward the existing message to others in a group as well as individuals.
* Check whether the user can able to send a message without a mobile number or not. And as well as the correct error message is displayed.
* Check by trying to send the message to an invalid user
* Check by trying to send a message to our mobile and see the result.
* Check whether you are able to send to any mobile which is temporarily disconnected.
* Authenticated user sends the messages

# Technologies

The application uses the following technologies:

* Web Server: Microsoft Internet Information Server (IIS)
* Presentation logic: Python Stream Lite
* Database Server: Microsoft SQL Server

# Application Security Mechanisms

* The most important application security mechanisms known at this time are:
* Users are authenticated with Forms authentication.
* Application is authenticated at the database by using Windows authentication.
* Roles are used to authorize access to business logic.
* Administration can be performed only by physically logging on to the server computer. No remote administration access is provided.

# Application Decomposition

This section describes the trust boundaries, entry points, exit points, and data flows.

# Trust Boundaries

## Identified trust boundaries are:

* The perimeter firewall.
* The database server trusts calls from the Web application's identity.
* The data access components trust the business components to pass fully validated data.
* An entry point to Message board application administration business component.

# Data Flows

The user logs on. The user submits a name and password through the logon form. The user name and password are handled by the logon page and passed to the membership business logic component. This component passes the data to the data access component, which verifies the credentials with the database to determine their validity.

A application administrator logs on and accesses the restricted Message board application administration page. The application administration component checks the user role at the business layer. If the user is authorized, the business component interacts with the Message board application data access component to view and amend messages.

# Entry Points

* Port XX for Web requests.
* Port XXX for SSL.
* All other ports are restricted by the firewall.
* The logon page, which is accessible to all Internet users. Logon is validated by using client-side and server-side validation controls, together with a common validation library.
* The amend customer details page, which is accessible to authenticated users only. Users are validated by using client-side and server-side validation controls, together with a common validation library. This page invokes functionality that can update customer details.

# Exit Points

* Logout Button to log off from the page/Application

# Threats

* The following threats/attacks could affect the application:
* Brute force attacks occur against the dictionary store.
* Network eavesdropping occurs between the browser and Web server to capture client credentials.
* An attacker captures an authentication cookie to spoof identity.
* SQL injection occurs, enabling an attacker to exploit an input validation vulnerability to execute commands in the database and thereby access and/or modify data.
* Cross-site scripting occurs when an attacker succeeds in injecting script code.
* Cookie replay or capture occurs, allowing an attacker to spoof identity and access the application as another user.
* Information is disclosed and sensitive exception details are revealed to the client.
* An attacker manages to take control of the Web server, gain unauthorized access to the database, and run commands against the database.
* An attacker obtains the encryption keys used to encrypt sensitive data (including client credit card numbers) in the database.
* An attacker or client obtains unauthorized access to Web server resources and static files.

# Vulnerabilities

**The application vulnerabilities are:**

User password storage.

* Lack of password complexity enforcement.
* Lack of password retry logic.
* Missing or weak input validation at the server.
* Failure to validate cookie input.
* Failure to sanitize data read from a shared database.
* Failure to encode output leading to potential cross-site scripting issues.
* Exposing an administration function through the customer-facing Web application.
* Exposing exception details to the client.