Cyber Sentinel (SRS Document)

1.0 PROBLEM DEFINITION

Cyber Sentinel is a vulnerable web application that provides insights and information on the most common yet catastrophic software vulnerabilities and how to mitigate them.

1.1 The user can simulate cyber-attacks on the vulnerable web application by following the detailed steps.

1.2 They can learn how to patch exploitable vulnerabilities to avoid attacks in the future.

2.0 SOFTWARE REQUIREMENTS SPECIFICATION

2.1 INTRODUCTION

2.1.1 Purpose

2.1.1.1 The purpose of this SRS document is to describe the requirements involved in developing a vulnerable web application for testing and learning purposes.

2.1.1.2 The intended audience is anyone who develops software and web applications with basic knowledge of interacting with web applications.

2.1.2 Scope

2.1.2.1 The product is titled Cyber Sentinel.

2.1.2.2 The product will perform the following tasks

2.1.2.2.1 Asks to select one cyber-attack from various options.
2.1.2.2.2 Demonstrates the exploitation of weaknesses in the selected attack.
2.1.2.2.3 Assists on how to defend the software by patching known bugs.

2.1.2.3 The product is adaptable to change, as developers can add new attack modules even after the product’s delivery phase.

2.1.3 Definitions, Acronyms, and Abbreviations

2.1.3.1 VM - Virtual Machine
2.1.3.2 XSS - Cross-site Scripting
2.1.3.3 MITM - Man in The Middle

2.1.3.4 DDoS - Distributed Denial of Service

2.1.4 References

This project is based on an existing vulnerable application, “DVWA” or “Damn Vulnerable Web Application”.

2.1.5 Overview

2.1.5.1 The requirements needed for the smooth development of this web application.

2.1.5.2 The overall description provides the requirements, such as interfaces or tools used, constraints, product functions, and user characteristics for the Cyber Sentinel software.

2.2 THE OVERALL DESCRIPTION

2.2.1 Product Perspective

2.2.1.1 Hardware Interfaces

2.2.1.1.1 It is advised to use this web application in a virtual machine to increase the security of the host operating system.

2.2.1.1.2 This application uses basic hardware interfaces like a keyboard and mousepad.

2.2.1.1.3 The host operating system will be secure by containing the vulnerable application in Docker in the final stages of product development.

2.2.1.2 Software Interfaces

2.2.1.2.1 Front End: HTML, CSS, and PHP.

2.2.1.2.2 Back End: MySQL database.

2.2.1.3 Memory Constraints

2.2.1.3.1 No specific constraints on memory.

2.2.1.3.2 Memory constraints apply only when the application uses virtual machines, but these constraints are due to the VM rather than the application.

2.2.1.4 Operations

2.2.1.4.1 The application takes the user credentials to sign up / log in.

2.2.1.4.2 It provides different options to select a cyber-attack the user wants.

2.2.1.4.3 The application simulates the selected attack along with detailed steps on its execution.

2.2.1.4.4 After the simulation, the patches for the vulnerability and tips on how the user can protect themselves will be displayed.

2.2.2 Product Functions

2.2.2.1 The software validates the user's authentication by extracting their username and password.

2.2.2.2 The application will allow the user to choose different cyber-attack options.

2.2.2.3 The application will simulate the attack with detailed steps on what is happening to the application during the attack process.

2.2.2.4 Once the application demonstrates the attack, it will make the user aware of how to protect themselves from these vulnerabilities by providing fixes and patches.

2.2.3 User Characteristics

2.2.3.1 Few attacks demonstrated on this application need a basic understanding of websites and networks.

2.2.3.2 Most of the attacks simulated by the software do not require the users to have specific knowledge of the application's internal workings.

2.2.3.3 The product does not need the user to possess extensive technical knowledge. Anyone who knows how to use the internet can use this product successfully.

2.2.4 Constraints

2.2.4.1 The user must create a unique username, password, and a secret question during sign-up. If the user forgets their password, they can reset it using the security question.

2.3 SPECIFIC REQUIREMENTS

2.3.1 Logical Database Requirements

2.3.1.1 The system must contain databases for the necessary information for the functioning of the application. This information includes user details and specific data entries for attacks like SQL injection.

2.3.1.2 User details refer to the username, password, and security question and answer.

2.3.1.3 Some attacks will require using the database entries for demonstrations, such as how threat actors can access values inside databases without proper authorization.

2.4 FRONT - END DESCRIPTION

Cyber Sentinel is a web application that is a concoction of web pages. Each group of web pages will host the demonstration of a cyber-attack - a vulnerability, its exploitation, and finally, its risk mitigation.

The main sign-up page takes the username and password. The user must answer a security question, which the software will use if a password reset is requested. Each attack will need a unique set of web pages. There will be steps or prompts displayed to guide the user throughout the demonstration of each attack.

2.5 BACK - END DESCRIPTION

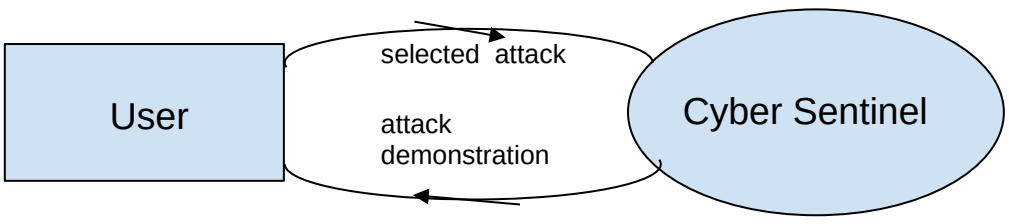
The sign-up or login data of the web application will be stored as a username, password, and answer to the security question. If time permits, additional security features, such as salting the hashed passwords, will be added, which will need more data structure fields.

Few attacks require the usage of additional fields according to their needs. This part of the development process is volatile, as developers can incorporate more attack modules after completing software delivery.

2.6 DATA STRUCTURES

FIELD NAME	TYPE
Username	text with special char
Password	text with numbers and special char
Security answer	text

2.7 DATA FLOW DIAGRAM



Context Diagram