??? Database Security Plan

This document describes the security plan for the ??? Database

|  |  |  |
| --- | --- | --- |
| Version & Date | Change Description | Authors |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Security Overview: Threats and Controls

This section describes an overview of the risk to the ???? Database. It includes:

* Misuse Case Diagrams, Security Use Case Diagrams, and Misuse Deployment Diagram;
* Accountability Table of Evil User Stories and Security Stories with corresponding Test Cases.

## Misuse Case Diagrams

This section includes Misuse Case Diagrams and Descriptions, Security Use Diagrams and Descriptions, and a Misuse Deployment Diagram. See text or lecture notes for examples.

## 1.2 Evil User Stories

This section includes Evil User Stories, corresponding Security Stories, and one or more Test Case numbers and names. See text or lecture notes for examples.

Accountability Table of Evil User Stories, Security Stories, Test Cases

|  |  |  |
| --- | --- | --- |
| **Evil User Story** | **Corresponding Security Story** | **Test Case # and Name** |
| *Section: Patient Management* |  |  |
| As a busy nurse, I sometimes do not provide all required information or enter incorrect information | As a database system, I ensure that all required fields are included and that some fields are checked for type (zip, phone) and accurate options (sex, insurance) | Test case 1: Create Patient Information |
|  |  |  |
| *Section: Handling Disclosures* |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Test Plan

This test plan includes prototypes and test cases for the Evil User Stories described in the previous section. It may include prototype testing forms showing sample data.

## 2.1 Test Case 1: Create Patient Information (Example Only)

|  |
| --- |
| **Test Case: Create Patient Information** |
| **Test Case ID: 1** |
| **Test Purpose:**   1. Ensure a valid new client can be entered into the system, and the appropriate tabs are created as expected. 2. Ensure a duplicate entry is not created for an existing patient. 3. Ensure invalid data is detected, including wrong data types, overflow text, overflow math numbers, blank required fields, and inappropriate data or common attacks. |
| **Primary Actors:**  Medical Administrator (Nurse, Doctor) |
| **Preconditions:** The user is at the main menu. |
| **Flow of Events:**   1. The user case begins when a Medical Admin selects “Manage Patient” or as an extension to Make Appointment 2. The Tester: enters last name and first name for an existing patient and press ‘Create’. 3. While the system finds a matching record    1. The system displays an error message: “Match Exists”, and requests the tester revise the information.    2. The tester changes the name to a new patient name.   4. The system should display multiple tabs, including Patient Information (Form 6.2, Patient Medical History (Form 6.3), and Patient Medical Information (Form 6.4).  5. The system should rename the ‘Create’ button into the ‘Save’ button.  6. The tester enters inappropriate data types for each field of the new Patient and presses ‘Save’. (Example form shown below.)  7. The system should recognize the invalid information and gives error messages.  6. The tester enters too much information for text strings or overflow data for arithmetic fields for each field of the new Patient and presses ‘Save’.  8. The system should recognize the overflow and gives error messages.  9. The tester leaves some required fields empty for the new Patient and presses ‘Save’.  10. The system should recognize the lacking information and gives error messages.  11. The tester enters inappropriate information for many fields of the new Patient and presses ‘Save’ (e.g, illegal state, sex, number of children, etc.)  12. The system should recognizes the errors and gives error messages.  13. The tester enters valid information for the new Patient and presses ‘Save’.  14. The system should display: ‘Record Updated’  15. The system should create a Patient Plan Management (Form 6.5) tab for Patients with health plans, or a Patient Bill Management tab for Patients without. |
| **Business Rules:**   1. Required fields include: First and last name, address, city, state, zip code, one form of contact information (email, phone). 2. Year of birth must be within the last 125 years. |
| **Postconditions:**   1. The new record has been saved into the test database. 2. A new encrypted transaction log has been recorded. 3. For Patients with health plans, a Patient Plan Management tab is available with information about the Patient’s plan. For Patients without, a Patient Bill Management tab is provided. |

Patient Information

Last Name: A First Name: Middle Initial: BC

Address: 1234 Phone: Abc-def-ghij

City: 1234 Email: garbage\*abc.com

State: WW

Family Members: 1234 Allowed contact: 1234

Employer: Address:

Insurance Plan: Red Cross Group Number: ABCD

Last Visit: Doctor: Incompetent

Next Visit: Doctor:

Retrieve

Create

# Security Events, Alarms, Metrics and Assumptions

This section includes the documented events and alarms that are generated by the organization’s main database software. Also included are metrics collected by the database software, and assumptions as to the security provided by the operating system and network environment.

## 3.1 Security Events and Alarms

These events/alarms are generated by the organization’s main database to notify administration about unusual events or potential attacks.

Table: Events and Alarms for Incident Response

|  |  |  |
| --- | --- | --- |
| **Incident** | **Description** | **System Response** |
| Illegal Access | Invalid username and password entered at login screen | Error message displayed to user  At 5 sequential invalid login attempts, lock out account.  Log written when lockout occurs or is cleared.  30-minute or admin command clears lockout |
|  |  |  |

## 3.2 Metrics provided by the Software

Statistics are collected and reported as part of the company’s main database system. These relate to availability, confidentiality and integrity.

## 3.3 List of Assumptions

Here allocation of security functions to network, operating system, administrative procedures, and database are described. The main database assumes that these security features are implemented in support of the database.