VIGILBOARD

VigilNet

Version 1.0

9-14-2023

William Mahoney, David Abbot, Isaac Bamidele, Chance Currie
williammahoney@my.unt.edu
davidabbot@my.unt.edu
oluwafeyisayobamidele@my.unt.edu
chancecurrie@my.unt.edu

Requirements Document

Version 1.0

Contents

A. Project Overview

The purpose of this project is to address any potential security vulnerabilities existing in a healthcare database. The VigilBoard will provide insight to any potential issues regarding security to a healthcare management system. This will be accomplished by providing a vulnerability scan to the database and will give a report about an existing security issue. For the purpose of our project we will be creating the healthcare database as a test subject of the VigilBoard. The VigilBoard will integrate with [insert tool] to collect vulnerability information. User authentication will be used to ensure that only authorized members are able to use the dashboard. The dashboard will provide a visualization for the user to see the potential vulnerabilities, the severity of vulnerabilities and the potential assets that are affected. This will lead to a prioritization of vulnerabilities that need to be reconciled. The VigilBoard will provide recommendations to the security issues that are found and give detailed information about the existing vulnerabilities. This information will be an asset to the database owner as it will give a detailed explanation about any current vulnerabilities and how to address them

B. Current Problems and Proposed Solutions

Healthcare databases and their accompanying management systems are usually outdated due to rules and regulations that prevent the upgrade to better systems. With outdated management systems, there are always security issues, and an attacker could be looking to exploit this, leading to a breach in confidential information, possible data loss, and much more depending on the intentions of the attacker. VigilBoard will run vulnerability scans on these outdated management systems, and it will generate a report of recommended solutions to best counteract any potential vulnerabilities that it finds. This will help the database owner better understand how they can protect their systems despite being outdated, and ensures that the proper steps of protecting an outdated system are followed.

C. Requirements

1. Functional Requirements

Phase Color Code

Blue - Plan Phase | Yellow - Develop Phase | Red - Build Phase

Purple - Test Phase | Pink - Release and Deliver | Green - Deploy Phase

Brown - Operate Phase | Grey - Monitor Phase

ID	Functional Requirements	Team Member Responsible	Effort (in %)
WM1	Project planning	William Mahoney	5%
IB1	System Design	Isaac Bamidele	5%

DA1	Software Requirement Analysis	David Abbot	5%
CC1	Change Management Planning	Chance Currie	5%
CC2	Application code development	Chance Currie	25%
WM2	Application code development	William Mahoney	15%
WM3	Code Review	William Mahoney	5%
IB2	Database Development	Isaac Bamidele	25%
DA2	Documentation	David Abbot	25%
CC3	Build	Chance Currie	5%
WM4	SAST	William Mahoney	10%
DA3	Dependency Vulnerability Checking	David Abbot	5%
IB3	Store Artifacts	Isaac Bamidele	10%
DA4	Release Packaging	David Abbot	5%
CC4	Build Configuration and Audit	Chance Currie	5%
DA5	System Test	David Abbot	10%
CC5	Unit Test	Chance Currie	10%
WM5	Manual Security Test	William Mahoney	20%
DA6	Integration test	David Abbot	10%
IB4	Database security test	Isaac Bamidele	10%
IB5	Test Audit	Isaac Bamidele	10%
CC6	Database functional test	Chance Currie	10%
WM6	Release Go/ No-go decision	William Mahoney	3%
WM7	Deliver Released Artifacts	William Mahoney	2%
DA7	Delivery Results Review	David Abbot	5%
IB6	Ops Team Acceptance	Isaac Bamidele	5%
CC7	Configuration Integration Testing	Chance Currie	5%
CC8	Post-deployment Security Scan	Chance Currie	15%
WM8	Post-deployment Checkout	William Mahoney	15%

DA8	Create linked clone of VM master image	David Abbot	7.5%
IB7	Database Installation	Isaac Bamidele	15%
DA9	Database artifact deployment	David Abbot	7.5%
CC9	Backup	Chance Currie	4%
IB8	Scale	Isaac Bamidele	5%
WM9	Load Balancing	William Mahoney	5%
DA10	Feedback	David Abbot	5%
WM10	Logging	William Mahoney	15%
DA10	Logging Analysis	David Abbot	7.5%
IB9	Database Monitoring and Security Auditing	Isaac Bamidele	7%
CC10	Log Auditing	Chance Currie	15%
IB10	System Performance Monitoring	Isaac Bamidele	7%
DA11	System Security Monitoring	David Abbot	7.5%

2. Non-Functional Requirements

[List your non-functional requirements with the team member responsible to meet the non-functional requirement and his effort. Only one team member per non-functional requirement and group by team member.]

ID	Non-Functional Requirements	Team Member Responsible	Effort (in %)
WM11	GUI Design	William	5%
CC11	Documentation (how to install/run)	Chance Currie	1%
IB11	Demo Video	Isaac Bamidele	1%

3. Constraints

- *Legal - Privacy concerns with regards to healthcare data.*

    - *HIPAA - Mandatory HIPAA compliance is required to safeguard all patient private information, medical and non medical.*
    - *PHI - Emphasize that the scans will not directly access the information in the databases but only provide information regarding the security of the database structure.*

- *Financial - NO BUDGET*
- *Availability - Must provide a consistent uptime to monitor vulnerabilities in the system. Must be available to admins when they need it for compliance and scheduled scans.*
- *Usability - System should be easy to use. Navigating the dashboard must remain clear.*
- *Expertise - This is the first time anyone in our team has participated in creating a technical project of this size.*