Analyse des risques

-

Projet Voltron

Auteur : T. de Lachaux
Dernière version : 02.07.2021

I Introduction

L’objet de ce rapport est de mettre en lumière les enjeux et les risques du projet Voltron en matière de sûreté de fonctionnement et de sécurité. L’objet de ce rapport est également, pour chaque risque identifié, de proposer un ensemble de solutions.

Le projet Voltron est une technologie visant à récolter, centraliser et analyser des données concernant la qualité et la disponibilité du matériel médical et des locaux au sein d’un complexe hospitalier. La solution développée comprend une intelligence artificielle, divers capteurs (humidité, température, etc…), une infrastructure cloud et une interface graphique.

La présente analyse se décompose en deux parties. La première est dédiée aux risques relatifs à des dysfonctionnements internes à la solution tandis que la seconde s’attache à lister les menaces externes.

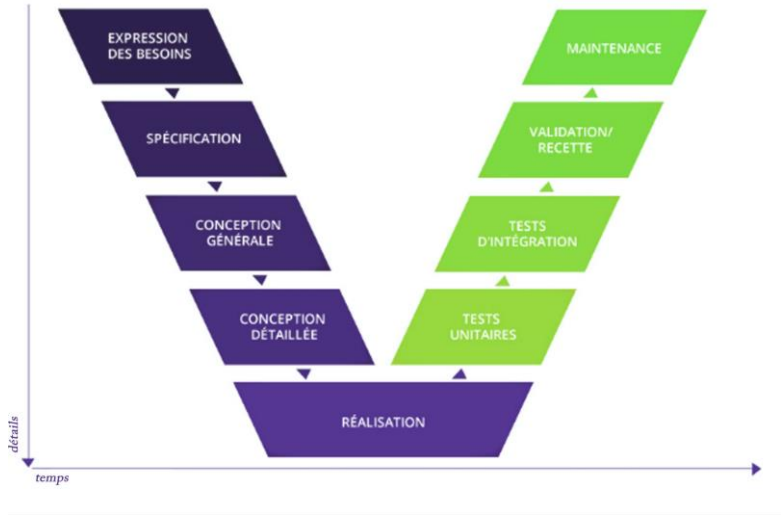
I Enjeux de sûreté : menaces internes

Le projet Voltron, une fois achevé, sera amené à traiter des données sensibles susceptibles, si elles sont erronées, de mettre en danger la vie des patients hospitalisés. Il est donc crucial d'assurer la sûreté de fonctionnement de la solution. S'assurer de la sûreté de la solution, c'est s'assurer de la capacité de la solution à remplir les tâches pour lesquelles elle a été conçue.

1. Méthodologie

Dans le cadre de développements critiques, il est recommandé de suivre un cycle de développement en V. Le cycle en V est constitué de trois phases :

- La définition du besoin (spécification, conception détaillée)
- Le développement,
- Les tests (tests unitaires, tests fonctionnels, tests de validation).



A chaque étape de la phase descendante correspond une étape de la phase ascendante.

2. Monitoring et contrôles

La mise en place de systèmes de monitoring et de contrôle pour s'assurer de la pertinence des données, de l'état des machines et des capteurs est également hautement recommandée.

Par ailleurs, en cas d'utilisation d'algorithmes non-déterministes (notamment pour l'intelligence artificielle), il est crucial de mettre en place des gardes-fous et des entités de contrôle.

I Enjeux de sécurité : menaces externes

Outre les enjeux de sûreté, les équipes de développement et de maintenance vont devoir être confrontées à des menaces externes.

1. Interception des données

Les données issues des capteurs vont nécessairement transiter par un réseau, qu'elle soit local ou qu'il s'agisse d'Internet.

-> Afin de s'assurer que ces données ne puissent pas être lues en cas d'interception, les données doivent être cryptées avant tout transfert.

2. Corruption des données

De la même manière qu'elles peuvent être lues, des données transitant en clair sur un réseau sont susceptibles de subir des modifications.

-> Comme pour le point 1, cette menace est levée en cryptant les données.

3. Pénétration du système

Un attaquant peut tenter de pénétrer à l'intérieur du système. Pour parer cette menace, il convient d'identifier les points d'entrée et de les verrouiller.

-> Si le système est exposé sur Internet, il est primordial de réduire sa surface d'exposition en mettant en place un pare-feux et éventuellement un reverse-proxy.
-> Si le port ssh (permettant de prendre le contrôle d'une machine à distance) est ouvert sur l'une des machines du projet Voltron, il est crucial de le protéger avec une clé RSA.
-> En cas de pénétration réussie, l'attaque peut être amoindrie si la politique d'attribution des droits a été correctement appliquée.

4. Corruption de la base de données

Si les requêtes et données destinées à la base de données ne sont pas vérifiées, il est possible, pour un attaquant, d'endommager ou de détruire la ou les bases de données.

-> Ce type d’attaque peut être évité en vérifiant toutes entrées saisies par l’utilisateur ou issues de capteurs. Lors du développement d’un serveur, il faut partir du principe que toute donnée entrante peut être malveillante.
-> Il est également fortement conseillé d’effectuer des sauvegardes régulières des bases de données afin de réduire l’impact d’une corruption réussie.

5. Déni de service

L’attaque par déni de service consiste à saturer un système de requêtes pour que ce dernier ne parvienne plus à traiter les requêtes légitimes dans un temps raisonnable.

-> Cette attaque peut être évitée en isolant les émetteurs de requêtes malveillantes et en les bannissant temporairement. Il est également possible de paramétrer des alertes en cas de temps de réponse trop élevé des serveurs permettant aux administrateurs de rapidement réagir.