**LES ÉLÉMENTS DE SÉCURITÉ À PRENDRE EN COMPTE LORS DU PROJET**

1. La sécurité BY DESIGN dès l'entame du projet

La sécurité BY DESIGN est un préalable pour nous. Et nous allons l’instaurer au cours de ce projet. C’est un principe de conception visant à intégrer la sécurité dès les premières phases du processus de développement du projet. L'idée fondamentale est de considérer la sécurité comme une composante inhérente à chaque aspect du système, plutôt que comme une couche ajoutée après coup. Cela contribue à minimiser les vulnérabilités et les risques de sécurité tout au long du cycle de vie du produit.

Afin de rendre cela possible, nous allons réaliser :

* Une évaluation des risques avec la méthode EBIOS pour identifier en amont les risques potentiels liés à la sécurité dès le début du projet pour identifier les menaces et les vulnérabilités associées.

Cela nous permettra de prendre des mesures concrètes dès le départ pour éviter de se retrouver dans des situations qui seraient de nature à remettre en question la qualité de notre travail.

Exemple de risque 1 : L’intégrité des données d’entrées avec lesquelles KNIME travail n’est pas garantie.

Exemple de mesure 1 : Un système de vérification des hashs des bonne données d’entrées peut être fait pour se rassurer de leur intégrité.

* Le principe de moindre privilège sera le maître mot lors de notre approche de réalisation de ce projet. Accordez aux utilisateurs et aux composants système uniquement les droits d'accès nécessaires.
* Prise en compte du chiffrement, Intégrez le chiffrement des données dès le début, que ce soit pour les données en transit ou les données stockées. Cela contribue à protéger les informations sensibles contre les accès non autorisés.
* Prendre en compte l’authentification forte : Favorisez l'utilisation de méthodes d'authentification forte pour garantir que seules les personnes légitimes ont accès au système.
* Prévoir dès maintenant comment les mises à Jour régulières se feront avec les correctifs de sécurité. Intégrez un processus de gestion des vulnérabilités.
* Intégrez des tests de sécurité tout au long du processus de développement. Cela inclut des tests d'intrusion, des analyses statiques et dynamiques de code, et d'autres tests de sécurité.
* Penser dès le début à la sécurité des API : notre système utilisera des interfaces de programmation (API), nous allons nous assurer qu'elles sont sécurisées contre les attaques telles que les injections SQL, les attaques par débordement de tampon, etc.

1. La sécurité de KNIME

La sécurisation de KNIME est cruciale pour garantir la confidentialité, l'intégrité et la disponibilité des données dans nos workflows et projets d'analyse. Voici quelques bonnes pratiques que nous allons mettre en pratique pour renforcer la sécurité de KNIME :

* KNIME embarquera l'authentification forte ;
* Nous allons faire l’intégration avec LDAP/Active Directory d’OCI ;
* Audits et Journalisation seront configurés pour suivre les activités des utilisateurs. Ces journaux peuvent être utiles pour la détection des problèmes de sécurité, l'analyse des performances et la conformité aux réglementations.
* Contrôle d'Accès aux Données : Utilisez les fonctionnalités de contrôle d'accès de KNIME pour restreindre l'accès aux données sensibles. Configurez les permissions au niveau du projet et des workflows pour garantir que seuls les utilisateurs autorisés peuvent accéder et modifier les éléments appropriés.
* Surveillez les performances du système KNIME pour détecter toute anomalie ou attaque potentielle. Configurez des alertes pour être informé rapidement des problèmes de sécurité ou des erreurs.
* Sécurisation du serveur Web : Si KNIME est déployé sur un serveur web, appliquez les bonnes pratiques de sécurisation du serveur web pour renforcer la sécurité globale.
* Mettre KNIME dans un VLAN spécifique avec filtrage d’accès serré au VLAN.
* Se connecter à KNIME devra se faire via un serveur de rebond avec enrégistrement de la session sur le BASTION.
* Une procédure de sauvegarde de KNIME sera mise en place pour garantir la continuité d’activité en cas de problème majeur.
* Les comptes capables de se connecter à KNIME devront être très limités et faire l’objet d’une surveillance accrue.
* Le dimensionnement de KNIME devra être fait pour éviter les lenteurs et garantir une certaine disponibilité.
* La seule entité capable d’extraire les données sensibles de KNIME, de les déchiffrer et de les utiliser pour mieux les présenter sur les dashboards de supervision devra être notre solution applicative.

1. Sécurité de l'application

Au niveau de la solution applicative web qui sera développée et couplée à KNIME afin de mettre en oeuvre notre projet, nous allons aussi y embarquer un ensemble de mesures de sécurité à observer comme :

* L’authentification double facteur pour s’y connecter ;
* La prise en compte dans le développement du top 10 OWASP ;
* Protéger cette solution applicative par un WAF ;
* Utiliser une méthode de programmation sécurisée comme la programmation orientée objet ;
* Écrire une procédure de sauvegarde complète pour la solution ;
* Écrire une procédure de mise à jour complète pour la solution ;
* Écrire une procédure de correction des vulnérabilités ;
* Protéger la solution contre les brutes forces par captcha ;
* Nous allons mettre en place un système d’audit et de journalisation poussé pour enregistrer les activités de l’application. Cela peut être utile pour détecter les comportements suspects, comprendre les attaques potentielles, et faciliter les enquêtes en cas d'incident ;
* Nous allons terminer par la mise en place du principe de moindre privilège, que ce soit pour les comptes applicatifs ou les comptes de services et comptes système ou les comptes utilisateurs pour éviter qu’ils soient intrusifs.

1. Optimisation des requêtes et de temps de réponse de l'application.

L'optimisation des requêtes et du temps de réponse d'une application est cruciale pour garantir des performances efficaces, une expérience utilisateur fluide et l'évolutivité du système. Voici quelques astuces que nous allons utiliser pour optimiser les requêtes et le temps de réponse de notre application :

* Indexation des Bases de Données : Nous allons nous assurer que les tables de notre base de données sont correctement indexées. Les index accélèrent la recherche de données, réduisant ainsi le temps nécessaire pour exécuter des requêtes.
* Requêtes SQL Optimisées : Nous allons écrire des requêtes SQL optimisées. Limitez le nombre de colonnes récupérées, utilisez des indices, évitez les opérations coûteuses comme les jointures complexes, et nous allons nous assurer que les requêtes sont bien structurées.
* Nous allons utiliser le caching comme mécanismes de mise en cache pour stocker temporairement les résultats fréquemment demandés. Cela va réduire la charge sur la base de données et accélérer les temps de réponse.
* Nous allons optimiser les images et les ressources en comprimant et optimisant les images et autres ressources statiques pour réduire le temps de chargement des pages web.
* Nous allons procéder à la compression des données pour réduire la taille des données transférées entre le serveur et le client.
* Nous allons optimiser le code de l'application en évitant les boucles inutiles, en minimisant le nombre d'appels réseau, et en utilisant des structures de données efficaces.
* Nous allons optimiser les requêtes HTTP en réduisant le nombre de requêtes HTTP en combinant les fichiers CSS et JavaScript, en utilisant des sprites pour les images, et en limitant le chargement de ressources non essentielles.

1. Les anomalies standard des comptes qui seront embarquées

Cette partie va parler du cœur de notre réponse à la problématique d’Orange Côte d’Ivoire. Lors d'une activité de revue de compte par la sécurité informatique, plusieurs anomalies peuvent être identifiées sur des comptes systèmes, applicatifs ou de bases de données. Ces anomalies peuvent indiquer des risques potentiels pour la sécurité du réseau. Voici quelques exemples d'anomalies courantes que nous allons directement embarquer dans notre solution complète à fournir :

Comptes Inactifs : Des comptes qui n'ont pas été utilisés depuis longtemps peuvent représenter un risque de sécurité s'ils ne sont pas nécessaires. Notre revue automatique va identifier les comptes inutilisés.

Compte avec privilèges excessifs : Les comptes ayant des privilèges excessifs peuvent créer des vulnérabilités. Il est important de s'assurer que les comptes disposent uniquement des droits nécessaires pour accomplir leurs tâches. Nous allons avec l’aide des inputs d’OCI traquer ces comptes.

Utilisateurs avec Accès Administratif : La présence d'utilisateurs avec des privilèges administratifs doit être justifiée et limitée. Des accès administratifs excessifs peuvent être exploités par des attaquants. C’est pourquoi notre solution traquera ces comptes et les remontera.

Mots de Passe Faibles : Des mots de passe faibles ou partagés représentent un risque de sécurité. Notre outil de revue automatisé doit s'assurer que les politiques de mot de passe d’Orange Côte d’Ivoire sont appliquées et que les utilisateurs suivent des bonnes pratiques en matière de sécurité des mots de passe. Les utilisations simultanées à des endroits différents seront traquées.

Les comptes génériques : Les comptes génériques seront identifiés et remontés à la sécurité d’Orange Côte d’Ivoire. Si des anomalies sont détectées sur ces comptes comme la non utilisation, des droits excessifs cela sera remonté pour prise de décision de la part de la sécurité d’OCI.

Comptes Orphelins : les comptes d'application qui ne sont plus nécessaires ou qui sont orphelins (non liés à un propriétaire) peuvent présenter un risque. Il est important de les identifier et de les désactiver. Ils seront tous remontés sur notre Dashboard.

Comptes d'application avec des Privilèges Élevés : Certains comptes d'application peuvent avoir des privilèges élevés qui ne sont plus nécessaires. Réduire les privilèges à un niveau minimal est une bonne pratique. Avec l’aide des équipes métiers et applicatif ces comptes seront remontés dans le dashboard de la solution.

Compte utilisés depuis des sources non cohérentes : Si nous prenons le cas d’un compte d’une franchise ou d’une agence bien précise et que ce compte est utilisé depuis une source non conforme à l’emplacement de cette franchise ou agence, en venant d’une sources inhabituelles, APN, VPN, VDI, etc notre solution remontera cette anomalies.

Compte avec des verrouillages excessifs : Notre solution sera aussi en mesure de détecter les verrouillages excessifs des comptes systèmes, applicatif, et base de données qui pourrait bien être une anomalie.

Compte de service utilisé comme utilisateur : Notre solution sera en mesure de détecter les comptes de services qui sont utilisés comme compte utilisateur. Ce qui peut être une anomalie de sécurité.

Compte de consultant utilisé hors contexte : Notre solution pourra relever les alertes sur les comptes des consultants qui sont utilisés en dehors des rebonds qui leur sont fournis. Ce qui est une anomalie.

Compte de franchise ou d’agence utilisé sur des rebonds internes : Notre solution pourra mettre en évidence les comptes de franchises et d’agence qui se connectent sur les rebonds internes d’OCI qui ne leurs sont pas attribués. Ce qui peut être une anomalie majeure de sécurité.

Compte à désactiver toujours en activité : Notre solution donnera la latitude aux équipes de sécurité d’Orange Côte d’Ivoire de mettre des tags sur les comptes (exemple : à désactiver) et notre solution pourra remonter toutes activités du compte si celui-ci continue à être utilisé sur le réseau. Ce qui peut constituer une anomalie de sécurité.

Compte AD administrateur de domaine : Avec l’aide des inputs système, notre solution sera en mesure de remonter en permanence les comptes qui ont le droits administrateur de domaine sur le domaine d’OCI.

Compte AD administrateur local : Avec l’aide des inputs système, notre solution sera en mesure de remonter en permanence les comptes qui ont les droits administrateur local sur le domaine d’OCI.

Pour réaliser ces contrôles standards, nous nous baserons sur les éléments suivants :

* Horodatage de l'utilisation des comptes dans les logs ;
* Source de la connexion des comptes dans les logs ;
* Les droits des comptes (applicatif, système, Base de données);
* Traquer les événements du genre verrouillages, expirations, réactivations de comptes, etc)
* Les logs applicatifs, système, base de données, fournis par les ressources d’Orange Côte d’Ivoire ;

Bien évidemment cette liste de contrôle n’est pas exhaustive, nous allons la compléter avec les besoins spécifiques de la sécurité d’Orange Côte d’Ivoire après les interviews qui seront menées en début de projet.

1. Sécurité du serveur embarquant KNIME

Nous devons renforcer la sécurité sur le ou les serveurs où sera installé l’outil KNIME et la solution web applicative. Les mesures standard qui seront prises sont les suivantes :

* S’assurer que ces serveurs ont la dernière version de l’antivirus ;
* S’assurer que ces serveurs ont la dernière version de l’EDR d’OCI ;
* S’assurer que ces serveurs sont accessibles seulement via rebond;
* S’assurer que le strict minimum en terme de service y tourne;
* S’assurer que ces serveurs sont bien dimensionnés ;
* S’assurer que ces serveurs ont une politique de sauvegarde ;
* S’assurer que ces serveurs sont dans le VLAN dédié à KNIME;
* S’assurer que seule les comptes habilités s’y connectent ;
* S’assurer qu’ils sont régulièrement mise à jour ;
* S’assurer que ces serveurs sont bien monitorer à l’aide du SIEM ;