Planification SDLC - The Count of Money

Partie 1 : Identifier les Ressources et Estimer le Temps

1. Outils de Sécurité Nécessaires

1.1 Analyse des Dépendances

- **npm audit** : Analyse des vulnérabilités des packages Node.js
- **Snyk** : Scan continu des dépendances
- **GitHub Security Alerts** : Alertes sur les dépendances vulnérables

1.2 Tests de Sécurité

- **OWASP ZAP** : Tests de sécurité dynamiques
- **Postman** : Tests des API
- **Trivy** : Analyse des conteneurs Docker

1.3 Authentification et Autorisation

- **JWT** : Gestion des tokens d'authentification
- **OAuth2** : Authentification Google
- **bcrypt** : Hachage des mots de passe

1.4 Sécurité Frontend

- **Validator.js** : Validation des entrées
- **DOMPurify** : Protection XSS
- **Vue Router Guards** : Protection des routes

1.5 Sécurité API

- **Helmet.js** : En-têtes de sécurité
- **express-rate-limit** : Protection contre les attaques par force brute
- **CORS** : Configuration sécurisée

2. Structure Git

Voir GitHub

3. Planning Détaillé

Voir Jira

4. Répartition des Ressources par Composant

Authentification

- JWT Decoder
- Validateur de mot de passe
- OAuth2 Client
- Session Manager

Routes et Navigation

- Route Guards
- Permission Manager
- Error Handler
- Middleware de sécurité

API et Données

- Input Validator
- Rate Limiter

- CORS Manager

- Data Sanitizer

UI/UX Sécurisé

- XSS Protection

- CSP Configuration

- Error Boundary

- Session Monitor

5. Temps Total Estimé : 25 jours ouvrés

- Configuration et Setup : 5 jours
- Développement Sécurité : 17 jours
- Tests et Audit : 3 jours

Documenter les Mesures de Sécurité

1. Identification des Fonctionnalités Sensibles

Notre application comporte plusieurs fonctionnalités qui nécessitent une attention particulière en matière de sécurité.

Module d'Authentification

Le système d'authentification est critique car il constitue la porte d'entrée de l'application. Il gère :

- La connexion par email/mot de passe
- L'authentification via Google OAuth
- L'enregistrement des nouveaux utilisateurs
- La gestion des sessions utilisateur

Cette fonctionnalité est particulièrement sensible car une faille pourrait compromettre l'ensemble du système et les données des utilisateurs.

Interface Anonyme

L'interface anonyme, bien que publique, nécessite une protection car elle :

- Affiche les N cryptomonnaies les plus populaires
- Présente les K derniers articles
- Gère l'affichage des tendances et pourcentages

La sécurisation est nécessaire pour éviter la manipulation des données publiques et protéger l'intégrité des informations affichées.

Interface Utilisateur

Cette interface gère des données personnalisées :

- Liste personnelle de cryptomonnaies
- Mots-clés pour la revue de presse
- Préférences de profil
- Historique des modifications

La protection de ces fonctionnalités est essentielle pour garantir la confidentialité des données utilisateur.

Interface Administrateur

L'interface administrateur est la plus sensible car elle permet :

- La gestion des préférences globales
- La configuration des cryptomonnaies disponibles
- La gestion des sources d'articles
- Les paramètres système généraux

2. Processus Critiques

Les processus critiques identifiés comprennent :

Gestion des Identités

- Création et validation des comptes
- Gestion des sessions
- Modification des profils
- Réinitialisation des mots de passe

Gestion des Données Financières

- Récupération des cours en temps réel
- Stockage des préférences utilisateur
- Historisation des données
- Synchronisation des informations

Gestion du Contenu

- Agrégation des articles
- Filtrage par mots-clés
- Validation des sources
- Distribution du contenu

3. Attribution des Responsabilités (RACI)

Processus d'Authentification

Role	**Acteurs**
Responsible (Responsable)	- Service d'authentification - Développeurs backend - Service de gestion des tokens
Accountable (Approbateur)	- Administrateur sécurité - Chef de projet
Consulted (Consulté)	- Service utilisateur - Expert sécurité - Service OAuth
Informed (Informé)	- Utilisateurs finaux - Support technique - Équipe monitoring

Processus de Gestion des Cryptomonnaies

Role	**Acteurs**
Responsible (Responsable)	- Service de données crypto - Développeurs API - Service de mise à jour
Accountable (Approbateur)	- Administrateur système - Responsable données
Consulted (Consulté)	- API externes - Expert financier - Service de validation

Informed (Informé)	- Utilisateurs connectés - Service de monitoring - Équipe support

Processus de Gestion des Articles

Role	**Acteurs**
Responsible (Responsable)	- Service d'agrégation d'articles - Développeurs backend - Service de filtrage
Accountable (Approbateur)	- Administrateur contenu - Chef de projet
Consulted (Consulté)	- Parseur RSS - Expert contenu - Service de validation
Informed (Informé)	- Tous les utilisateurs - Équipe support - Service monitoring

Processus de Gestion des Utilisateurs

Role	**Acteurs**
Responsible (Responsable)	- Service de gestion utilisateurs - Développeurs frontend/backend - Service de profils
Accountable (Approbateur)	- Administrateur système - Responsable RGPD
Consulted (Consulté)	- Service juridique - Expert sécurité - Service validation
Informed (Informé)	- Utilisateurs concernés - Support technique

	- Service monitoring

Processus d'Administration

Role	**Acteurs**
Responsible (Responsable)	- Service d'administration - Développeurs système - Service configuration
Accountable (Approbateur)	- Administrateur principal - Direction technique
Consulted (Consulté)	- Expert sécurité - Service technique - Auditeurs système
Informed (Informé)	- Tous les administrateurs - Équipe développement - Service monitoring

4. Conversion des Rôles RACI en RBAC (Role-Based Access Control)

Tableaux RBAC

Pour faciliter l'implémentation et la visualisation du système de contrôle d'accès, les tableaux suivants synthétisent la structure RBAC :

Table 1: Vue d'ensemble des Rôles

Rôle	Niveau d'accès	Portée	Zone d'action
Administrateur Système	Complet	Global	Toutes fonctionnalités
Modérateur de Contenu	Partiel	Contenu	Articles et sources
Utilisateur Authentifié	Restreint	Personnel	Profil et préférences
Utilisateur Anonyme	Minimal	Public	Consultation uniquement

Table 2: Matrice Détaillée des Permissions

Fonctionnalité	Admin	Modérateur	Utilisateur	Anonyme	Justification
Gestion utilisateurs	CRUD	-	Read/Update (self)	-	Sécurité et maintenance
Configuration système	CRUD	-	-	-	Intégrité système
Gestion articles	CRUD	CRUD	Read	Read	Qualité du contenu
Gestion cryptos	CRUD	Read	Read/Save	Read	Service principal
Paramètres profil	CRUD	Read	CRUD (self)	-	Contrôle utilisateur
Sources RSS	CRUD	CRUD	-	-	Alimentation contenu
Statistiques	CRUD	Read	Read	Read	Transparence
Logs système	CRUD	-	-	-	Audit et sécurité

Table 3: Actions Sensibles et Validation

Action	Validation requise	Niveau d'autorisation	Log
Suppression compte	Double auth	Admin	Oui
Modification rôles	Double auth	Admin	Oui
Ajout source RSS	Simple auth	Modérateur	Oui
Modification crypto	Double auth	Admin	Oui
Reset password	Simple auth	Admin/Self	Oui

Cette structuration en tableaux complète la description narrative du système RBAC et fournit une référence claire pour l'implémentation des contrôles d'accès dans l'application.

5. Intégration du Flux de Sécurité dans le Workflow

Pour garantir une sécurité optimale tout au long du cycle de vie de l'application, nous avons intégré des mécanismes de sécurité à chaque étape des workflows utilisateur. Cette approche permet d'assurer une protection continue sans compromettre l'expérience utilisateur.

Workflow d'Accès à l'Application

1. Authentification

- Point d'entrée : Interface de connexion
- Mesures de sécurité intégrées :

    - Validation des entrées en temps réel
    - Vérification de la complexité des mots de passe

- Protection contre les attaques par force brute (rate limiting)
- Double authentification pour les actions sensibles

- Flux sécurisé :

  - Tentative de connexion → Validation des identifiants → Vérification 2FA si activée → Génération du token JWT → Accès accordé

2. Navigation et Consultation

- Point d'entrée : Interface principale
- Mesures de sécurité intégrées :

  - Vérification de l'authenticité du token à chaque requête
  - Protection contre le XSS dans l'affichage du contenu
  - CSRF tokens pour les actions utilisateur
  - Chargement sécurisé des ressources externes

- Flux sécurisé :

  - Requête de page → Vérification du token → Validation des permissions → Affichage sécurisé

Workflow de Gestion des Données

1. Manipulation des Cryptomonnaies

- Points d'interaction : Interface de cours et listes personnalisées
- Mesures de sécurité intégrées :

  - Vérification de l'intégrité des données
  - Protection contre la manipulation des API
  - Rate limiting sur les requêtes de données
  - Validation des données avant affichage

- Flux sécurisé :

  - Demande de cours → Authentification API → Validation des données → Affichage sécurisé → Actions utilisateur validées

2. Gestion des Articles

- Points d'interaction : Interface articles et filtres
- Mesures de sécurité intégrées :

  - Validation des sources RSS

- Sanitization du contenu HTML
- Contrôle d'accès sur les articles restreints
- Protection contre l'injection dans les filtres de recherche

- Flux sécurisé :

  - Récupération articles → Validation source → Sanitization → Filtrage → Affichage sécurisé

Workflow Administratif

1. Gestion Système

- Points d'interaction : Interface d'administration
- Mesures de sécurité intégrées :

  - Double authentification obligatoire
  - Journalisation détaillée des actions
  - Vérification des privilèges pour chaque action
  - Confirmation explicite pour les actions critiques

- Flux sécurisé :

  - Accès admin → 2FA → Vérification privilèges → Action → Confirmation → Journalisation

2. Gestion des Utilisateurs

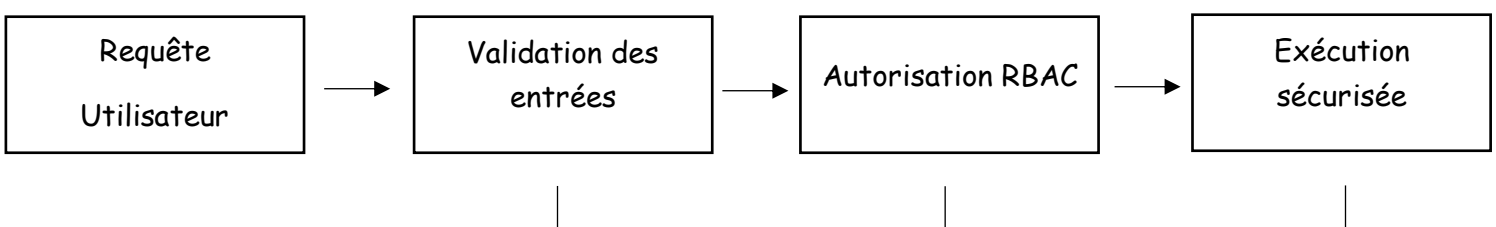
- Points d'interaction : Panel administrateur
- Mesures de sécurité intégrées :

  - Séparation des privilèges
  - Validation en plusieurs étapes pour les actions sensibles
  - Notification aux utilisateurs concernés
  - Audit trail complet

- Flux sécurisé :

  - **Demande action → Validation privilèges → Confirmation → Exécution → Notification → Journalisation**

Diagramme de Workflow Sécurisé



6. Intégration du Flux de Sécurité dans le Workflow

L'analyse de sécurité nous a permis d'identifier les fonctionnalités les plus critiques nécessitant une attention prioritaire lors de l'implémentation des mesures de sécurité. Cette priorisation est essentielle pour allouer efficacement les ressources et garantir que les éléments les plus sensibles sont protégés en premier.

Critères de Priorisation

Nous avons évalué chaque fonctionnalité selon les critères suivants :

- Impact : Conséquences potentielles d'une compromission (1-5)
- Probabilité : Likelihood d'une attaque réussie (1-5)
- Exposition : Niveau d'accessibilité de la fonctionnalité (1-5)
- Dépendances : Nombre de systèmes affectés en cas de compromission (1-5)

Le score de criticité est calculé par la formule : (Impact × 2) + Probabilité + Exposition + Dépendances

Table de Priorisation des Fonctionnalités

Rang	Fonctionnalité	Impact	Probabilité	Exposition	Dépendances	Score	Priorité
1	Authentification	5	5	5	5	25	Critique
2	Panel Administration	5	4	3	5	22	Critique
3	Gestion API Crypto	4	4	4	4	20	Élevée
4	Gestion Utilisateurs	4	3	4	4	19	Élevée

Rang	Fonctionnalité	Impact	Probabilité	Exposition	Dépendances	Score	Priorité
5	Stockage Données Sensibles	5	3	2	3	18	Élevée
6	Validation RSS	3	4	3	3	16	Moyenne
7	Formulaires Utilisateurs	3	4	4	2	16	Moyenne
8	Système de Recherche	2	3	5	2	14	Moyenne
9	Affichage Articles	2	3	5	1	13	Moyenne
10	Préférences Profil	2	2	3	2	11	Faible
11	Statistiques Publiques	1	2	5	1	10	Faible
12	Interface Anonyme	1	2	5	1	10	Faible

Justification des Priorités Critiques

1. **Authentification (Score: 25 - Critique)** L'authentification est la porte d'entrée principale de l'application. Une compromission à ce niveau permettrait un accès non autorisé à l'ensemble du système et des données utilisateur. Sa priorité critique est justifiée par : • Impact maximum en cas de violation • Exposition permanente aux tentatives d'attaque • Dépendances critiques avec tous les autres systèmes
2. **Panel Administration (Score: 22 - Critique)** Le panel d'administration donne accès aux fonctionnalités les plus sensibles du système. Sa priorité critique est justifiée par : • Contrôle complet sur l'application • Accès aux données de tous les utilisateurs • Capacité de modification des paramètres systèmes
3. **Gestion API Crypto (Score: 20 - Élevée)** Les API de cryptomonnaies représentent le cœur fonctionnel de l'application. Leur priorité élevée est justifiée par : • Manipulation possible des données financières • Exposition permanente aux requêtes externes • Impact direct sur la fiabilité du service

Plan d'Implémentation Sécurisé

1. **Phase 1 - Priorité Critique (Semaines 1-2)** • Implémentation des mesures de sécurité pour l'authentification • Sécurisation du panel d'administration • Protection des systèmes de gestion des rôles
2. **Phase 2 - Priorité Élevée (Semaines 3-4)** • Sécurisation des API de cryptomonnaies • Protection des données utilisateurs • Implémentation du stockage sécurisé

3. **Phase 3 - Priorité Moyenne (Semaines 5-6)** • Sécurisation des formulaires et validations • Protection des systèmes de recherche • Sécurisation des flux RSS
4. **Phase 4 - Priorité Faible (Semaines 7-8)** • Sécurisation des interfaces publiques • Protection des statistiques • Finalisation des contrôles d'accès

Cette priorisation permet une approche structurée de l'implémentation des mesures de sécurité, en commençant par les éléments les plus critiques pour garantir une protection efficace des fonctionnalités essentielles du système dès le début du développement.

7. Document Final des Fonctionnalités Sensibles

Ce document synthétise l'ensemble des fonctionnalités sensibles de l'application "The Count of Money" et détaille les mesures de sécurité requises pour chacune d'entre elles.

1. Système d'Authentification

Description : Point d'entrée principal de l'application, gérant l'identification des utilisateurs par email/mot de passe et OAuth2.

Mesures de Sécurité Requises :

- Hachage sécurisé des mots de passe avec bcrypt (facteur de coût ≥ 12)
- Implémentation d'une politique stricte de mots de passe (12 caractères minimum, complexité)
- Rate limiting (max 5 tentatives/minute) pour prévenir les attaques par force brute
- Tokens JWT avec expiration courte (60 minutes) et rotation des refresh tokens
- Double authentification pour les actions sensibles
- Validation complète des tokens OAuth2 avec vérification des domaines
- Déconnexion automatique après 30 minutes d'inactivité
- Journalisation des connexions suspectes (IP, timestamps, User-Agent)

2. Panel d'Administration

Description : Interface réservée aux administrateurs permettant la gestion globale de l'application.

Mesures de Sécurité Requises :

- Authentification renforcée (2FA obligatoire)
- Accès limité aux adresses IP autorisées
- Auditing complet de toutes les actions administratives
- Séparation des privilèges pour les actions critiques
- Délai d'expiration de session réduit (15 minutes)
- Confirmation explicite pour toute modification critique
- Notifications en temps réel des modifications système
- Routes d'accès non prédictibles et non exposées publiquement

3. API de Cryptomonnaies

Description : Services gérant la récupération, le stockage et l'affichage des données de cryptomonnaies.

Mesures de Sécurité Requises :

- Authentification API par tokens avec permissions spécifiques
- Validation des données à la source et à la réception
- Rate limiting (100 requêtes/minute par utilisateur)
- Protection contre les attaques par déni de service
- Chiffrement des données sensibles
- Vérification d'intégrité des données
- Monitoring continu des patterns d'accès suspects
- Caching sécurisé pour limiter les requêtes externes

4. Gestion des Utilisateurs

Description : Fonctionnalités permettant la création, modification et suppression des comptes utilisateurs.

Mesures de Sécurité Requises :

- Validation stricte des entrées utilisateur
- Sanitization des données avant stockage
- Permissions granulaires basées sur les rôles
- Protection contre l'énumération des comptes
- Processus sécurisé de récupération de mot de passe
- Notification des modifications de compte
- Journalisation des changements sensibles
- Vérification des adresses email

5. Stockage des Données Personnelles

Description : Mécanismes de stockage et de gestion des informations personnelles des utilisateurs.

Mesures de Sécurité Requises :

- Chiffrement des données sensibles au repos
- Isolation des données par utilisateur
- Politiques de rétention et suppression automatique
- Conformité RGPD complète
- Accès limité au principe du moindre privilège
- Backup chiffré avec rotation
- Pseudonymisation des données non essentielles
- Audit trail des accès aux données personnelles

6. Système de Flux RSS et Articles

Description : Fonctionnalités d'agrégation, filtrage et affichage d'articles de presse.

Mesures de Sécurité Requises :

- Validation des sources RSS autorisées
- Sanitization du contenu HTML avant affichage
- Protection contre les attaques XSS
- Filtrage du contenu malveillant
- Vérification de l'intégrité des flux
- Rate limiting sur les requêtes d'agrégation
- Isolation des processus de parsing
- Filtrage basé sur les mots-clés sécurisé

7. Formulaires Utilisateurs

Description : Tous les formulaires permettant la saisie et la soumission de données par les utilisateurs.

Mesures de Sécurité Requises :

- Validation côté serveur de toutes les entrées
- Protection CSRF sur chaque formulaire
- Sanitization des données entrantes
- Filtrage des caractères spéciaux
- Limitation de la taille des entrées
- Recaptcha pour les formulaires publics
- Feedback limité des erreurs (sans divulgation d'informations)
- Vérification des types de données

8. Système de Recherche

Description : Fonctionnalités de recherche d'articles et de cryptomonnaies.

Mesures de Sécurité Requises :

- Protection contre les injections SQL/NoSQL
- Validation et sanitization des termes de recherche
- Rate limiting (20 recherches/minute)
- Limitation des résultats par page
- Protection contre les attaques par déni de service
- Indexation sécurisée
- Restriction des champs de recherche sensibles
- Journalisation des recherches inhabituelles

9. Préférences de Profil

Description : Fonctionnalités permettant aux utilisateurs de personnaliser leur expérience.

Mesures de Sécurité Requises :

- Validation des modifications de préférences
- Protection contre la manipulation des préférences
- Limitation des options personnalisables
- Vérification de cohérence des données
- Notification des changements importants
- Restauration des paramètres par défaut sécurisée
- Audit des modifications
- Protection contre les préférences malveillantes

10. Interface Publique

Description : Fonctionnalités accessibles sans authentification.

Mesures de Sécurité Requises :

- Limitation des données exposées
- Protection contre le scraping
- Rate limiting par IP
- Protection DDoS
- Sanitization des données affichées
- Cache sécurisé
- En-têtes de sécurité HTTP
- Content Security Policy stricte

Ce document servira de référence pour l'implémentation et la vérification des mesures de sécurité tout au long du cycle de développement, assurant que chaque fonctionnalité sensible est protégée de manière appropriée.

Partie 2 : Conception et Modélisation

Analyse de la Base de Données MongoDB

L'examen de la base de données MongoDB a été réalisé en utilisant MongoDB Shell, permettant d'accéder directement aux collections et à leurs structures. Cette analyse a révélé que l'application "The Count of Money" repose sur deux collections principales : "users" et "anonyms".

La collection "users" stocke les informations relatives aux utilisateurs enregistrés, incluant leurs données personnelles, identifiants de connexion et préférences. L'analyse d'un document type a permis d'identifier la structure suivante :

```
{
_id: ObjectId('67b8e31624a8db97bda17e73'),
avatar: null,
lastname: "debede",
firstname: 'sole',
username: 'debede',
email: 'debedenouwo@gmail.com',
password: '$2b$10$Tm64j4LOxePHLlXDwpTBteVJuvO0hScIeYdhb9QazH5V9QLRXVvi6',
role: 'user',
articles: [],
cryptos: [],
__v: 0
}
```

La collection "anonyms" contient les paramètres par défaut pour les utilisateurs non authentifiés, avec la structure suivante :

```
  {
    _id: ObjectId('67b3cbab61fd6ac0cf2175d1'),
    name: 'Default',
```

Identification des Risques de Sécurité

L'analyse approfondie de la base de données a permis d'identifier plusieurs vulnérabilités potentielles. Pour cela, diverses commandes de test ont été exécutées afin d'évaluer la robustesse du système :

```
// Test des vulnérabilités d'injection
db.users.find({username: {$regex: /<script>/}})

// Vérification des contrôles d'accès
db.getUsers()
// Résultat: { users: [], ok: 1 }

// Tentative de vérification du schéma
db.users.getSchema()
// Erreur: TypeError: db.users.getSchema is not a function
```

Ces tests ont révélé plusieurs problèmes de sécurité importants :

1. **Absence de Contrôles d'Accès** : L'absence d'utilisateurs MongoDB configurés (comme démontré par la commande db.getUsers()) indique que la base de données ne dispose pas de mécanismes d'authentification dédiés, créant un risque d'accès non autorisé en cas de compromission du serveur.
2. **Manque de Validation de Schéma** : L'absence de validateurs de schéma expose la base de données à l'insertion de données malformées ou malveillantes, compromettant l'intégrité des données.
3. **Stockage Non Sécurisé des Données Sensibles** : Les informations personnelles comme les adresses email et les noms sont stockées en clair, sans mécanismes de chiffrement ou de pseudonymisation.
4. **Vulnérabilité aux Injections NoSQL** : L'application pourrait être vulnérable aux injections NoSQL si elle utilise des requêtes dynamiques construites à partir d'entrées utilisateur non validées.
5. **Indexation Insuffisante** : L'absence d'indexation adéquate peut non seulement affecter les performances, mais également exposer l'application à des attaques par déni de service via des requêtes consommatrices de ressources

Contre-mesures pour les Menaces Identifiées

Notre analyse de modélisation des menaces a mis en évidence plusieurs vulnérabilités potentielles dans l'architecture de l'application "The Count of Money". Pour chacune de ces menaces, nous proposons des contre-mesures spécifiques visant à renforcer la sécurité du système.

Élévation de Privilèges

Les menaces d'élévation de privilèges permettent à un adversaire d'accéder à des fonctionnalités nécessitant des droits supérieurs à ceux dont il dispose légitimement. Pour contrer ces risques, nous recommandons:

- Implémentation d'un système RBAC (Role-Based Access Control) dans MongoDB pour limiter strictement les accès selon les rôles attribués.
- Mise en place d'un middleware d'authentification dans Express.js pour vérifier systématiquement les autorisations avant l'accès aux routes protégées.
- Configuration d'utilisateurs MongoDB avec privilèges limités pour les opérations courantes, évitant ainsi l'utilisation de comptes administrateurs pour les fonctions quotidiennes de l'application.

Divulgation d'Informations

Les vulnérabilités de divulgation d'informations exposent des données sensibles à des utilisateurs non autorisés. Pour protéger ces données:

- Mise en œuvre du chiffrement des données sensibles au repos dans MongoDB, particulièrement pour les informations personnelles des utilisateurs.
- Activation du chiffrement TLS/SSL pour toutes les communications avec MongoDB, empêchant l'interception des données en transit.
- Utilisation systématique de projections sécurisées dans les requêtes MongoDB pour exclure les champs sensibles des résultats.
- Configuration des en-têtes de sécurité HTTP via Helmet.js pour réduire les risques d'exposition d'informations sensibles.

Injections NoSQL

Les attaques par injection NoSQL permettent la manipulation des requêtes de base de données. Pour se protéger:

- Implémentation de Mongoose avec validation stricte des schémas pour chaque collection.
- Utilisation systématique d'express-validator pour valider et assainir toutes les entrées utilisateur avant traitement.
- Évitement des constructeurs de requêtes dynamiques ou des opérateurs dangereux comme $where qui peuvent être exploités dans des attaques par injection.

Répudiation

Les problèmes de répudiation surviennent lorsqu'un utilisateur peut nier avoir effectué une action en raison d'un manque de traçabilité. Pour y remédier:

- Implémentation d'un système complet de journalisation pour enregistrer toutes les actions critiques des utilisateurs.
- Configuration d'un audit trail dans MongoDB pour tracer les modifications importantes des données.
- Mise en place de signatures cryptographiques pour les actions critiques, assurant leur non-répudiation.

Falsification

Les menaces de falsification concernent la modification non autorisée des données. Pour contrer ces risques:

- Implémentation de signatures numériques pour vérifier l'intégrité des données sensibles.

- Mise en place de validateurs de schéma MongoDB rigoureux pour empêcher l'insertion de données incorrectes.
- Utilisation de hooks Mongoose pour valider les modifications avant leur sauvegarde dans la base de données.

Déni de Service

Les attaques par déni de service visent à rendre l'application indisponible. Pour s'en protéger :

- Implémentation d'un rate limiting robuste pour limiter le nombre de requêtes par IP.
- Création systématique d'index MongoDB pour optimiser les performances des requêtes fréquentes.
- Limitation de la taille des requêtes pour prévenir les attaques par saturation de la mémoire ou du réseau.
- Configuration de timeouts appropriés pour les opérations MongoDB afin d'éviter les requêtes infinies.

L'application systématique de ces contre-mesures, combinée à une surveillance continue et des audits de sécurité réguliers, permettra de réduire significativement les risques identifiés lors de la modélisation des menaces et d'assurer un niveau de sécurité approprié pour l'application "The Count of Money".

La suite des reponses aux questions se trouve dans le repository github.