## 1. Contrôle d’accès défaillant

- Définition : Les utilisateurs peuvent accéder à des ressources ou des actions sans autorisation appropriée.
- Recommandations : Implémenter une politique stricte de contrôle d'accès, utiliser des tests automatisés pour vérifier les permissions.
- Développeurs concernés ? : Oui, ils doivent intégrer des vérifications de permissions dans le code.

## 2. Cryptographie insuffisante

- Définition : Une mauvaise gestion des clés cryptographiques ou utilisation d'algorithmes obsolètes.
- Recommandations : Utiliser des algorithmes et des bibliothèques cryptographiques modernes, stocker les clés de manière sécurisée.
- Développeurs concernés ? : Oui, pour choisir et implémenter correctement les solutions cryptographiques.

## 3. Injection

- Définition : Les données non fiables sont envoyées à un interpréteur (SQL, NoSQL, OS).
- Recommandations : Utiliser des requêtes préparées, valider les entrées.
- Développeurs concernés ? : Oui, surtout ceux travaillant avec des bases de données.

## 4. Conception non sécurisée

- Définition : Une mauvaise conception de la sécurité dès le début du projet.
- Recommandations : Adopter des principes de conception sécurisée, réaliser des revues de conception.
- Développeurs concernés ? : Oui, dès la phase de conception.

## 5. Mauvaise configuration de sécurité

- Définition : Une mauvaise configuration des serveurs, des bases de données, ou des applications.
- Recommandations : Automatiser les configurations sécurisées, maintenir les systèmes à jour.
- Développeurs concernés ? : Oui, souvent en collaboration avec les administrateurs systèmes.

## 6. Composants vulnérables et obsolètes

- Définition : Utilisation de bibliothèques, frameworks ou composants avec des failles connues.
- Recommandations : Utiliser des outils de gestion des vulnérabilités, mettre à jour régulièrement les composants.
- Développeurs concernés ? : Oui, pour gérer et maintenir les dépendances.

## 7. Identification et authentification défaillantes

- Définition : Problèmes avec l'authentification des utilisateurs et la gestion des sessions.
- Recommandations : Implémenter des mécanismes d'authentification robustes, sécuriser les jetons de session.
- Développeurs concernés ? : Oui, pour sécuriser les processus d'authentification.

## 8. Défauts d'intégrité des logiciels et des données

- Définition : Les mises à jour logicielles et les données ne sont pas vérifiées pour leur intégrité.
- Recommandations : Utiliser des signatures numériques, vérifier l'intégrité des mises à jour et des données.
- Développeurs concernés ? : Oui, pour garantir l'intégrité des applications et des données.

## 9. Journalisation et surveillance insuffisantes

- Définition : Une absence de journalisation ou de surveillance des événements de sécurité.
- Recommandations : Mettre en place une journalisation détaillée, utiliser des outils de surveillance et d'alerte.
- Développeurs concernés ? : Oui, pour intégrer des journaux de sécurité dans les applications.

## 10. Falsification de requêtes côté serveur

- Définition : Les attaquants peuvent envoyer des requêtes malveillantes du serveur à des systèmes internes.
- Recommandations : Valider et filtrer les requêtes sortantes.
- Développeurs concernés ? : Oui, pour contrôler les requêtes côté serveur.