EXERCICE 5 — Plan de tests de sécurité & validation des exigences

Contexte professionnel

Après avoir dérivé les exigences de sécurité (Exercice 3) et conçu une architecture Zero Trust (Exercice 4), ShopNow veut s'assurer que ces exigences sont réellement implémentées, testées et vérifiables. Le CTO te demande de construire un plan de tests de sécurité complet, aligné sur :

- les menaces STRIDE,
- les exigences de sécurité,
- l'architecture Zero Trust.

Objectifs pédagogiques

À l'issue de cet exercice, l'étudiant doit être capable de :

- **Identifier** les types de tests de sécurité pertinents (tests unitaires, tests d'intégration, tests d'intrusion, fuzzing, etc.).
- **Aligner** chaque test avec une exigence de sécurité donnée.
- **Définir** des critères d'acceptation concrets et mesurables.
- **Prioriser** les tests en fonction de la criticité métier et des risques.
- **Intégrer** les tests dans un pipeline CI/CD (DevSecOps).

Consigne générale

Répondez sous la forme d'un rapport en anglais (page de garde, sommaire, numérotation, conclusion).

Travail demandé

1. **Cartographier les exigences → tests**

Pour un sous-ensemble d'exigences de l'Exercice 3 (au moins 10), complétez un tableau :

2. **Définir une stratégie de tests par zone**
    - Zone Front (client, C1, F1, F3, F5)
    - Zone Backend sécurisé (C2, C5, C4, F4, F6)
    - Zone Données (C3, D1, D2, D3, D6)
    - Zone Externe (C6, A3)

Pour chaque zone, décrivez :

- **Types de tests** (ex : tests d'API, tests de charge, tests de fuzzing, tests de permissions).
- **Objectifs principaux** (ex : détection d'injections, validation RBAC, résilience DoS).

3. **Plan de tests pour les flux sensibles**

Concentrez-vous sur F1 (auth), F2 (paiement), F4 (commandes) :

- Identifiez les **cas de test critiques** (ex : tentative de paiement sans authentification, token expiré, modification de montant).
- Associez chaque cas de test à :
  - une menace STRIDE,
  - une exigence de sécurité,
  - un résultat attendu.

4. **Intégration dans un pipeline CI/CD**
   - Proposez comment intégrer ces tests dans un pipeline DevSecOps :
     - tests automatisés à chaque commit,
     - scans réguliers (SAST, DAST),
     - tests manuels périodiques (pentest).
   - Discutez des **compromis** entre couverture de tests, temps d'exécution et vélocité produit.

- Quels tests sont **non négociables** pour ShopNow ?
- Quels tests peuvent être **progressivement** ajoutés ?
- Comment mesurer l'**efficacité** de la stratégie de tests (KPIs) ?

ANNEXE A — Exigences de sécurité (extrait utile)

ID	Exigence	Menace
S1	Tokens HttpOnly/Secure	Spoofing
S2	MFA obligatoire admin	Spoofing
T1	HMAC sur requêtes sensibles	Tampering
T2	JWT RS256	Tampering
I1	TLS 1.3 obligatoire	Info Disclosure
I2	Chiffrement DB AES256	Info Disclosure
D1	Rate limiting	DoS
E1	RBAC strict	Elevation

ANNEXE B — Types de tests disponibles

- **SAST** : analyse statique du code
- **DAST** : tests dynamiques sur API
- **IAST** : instrumentation
- **Fuzzing** : tests aléatoires
- **Pentest** : tests manuels
- **Tests de charge** : DoS
- **Tests d'intégrité** : signature, HMAC
- **Tests d'autorisation** : RBAC, privilèges
- **Tests de configuration** : secrets, headers, TLS

ANNEXE C — Schéma ASCII des zones à tester