

Exigences Sécurité Applicative

Toolbox Pentest Automatisée

Version : 2.1 — Opérationnelle (consignes pilotage SecOps)
Date : Janvier 2026

1. Exigences d'Authentification — CHECKLIST AUDIT

À Implémenter (Obligatoire)

- **Pas d'accès sans authentification** : tout utilisateur doit s'identifier avant accès
- **Authentification multi-facteurs (MFA)** : mot de passe + code TOTP (authenticator) OU clé USB FIDO2
- **JWT signé RS256** : token 4h pour admin, 8h pour utilisateurs
- **Blocage après 5 tentatives échouées** : lockout 15 minutes (enregistré en logs)
- **Mots de passe** : min 12 caractères, complexité, hachage Argon2 (pas SHA-1, MD5)

À Valider (Checklist SecOps)

Contrôle	Preuve de Conformité	Responsable
Authentification obligatoire fonctionnelle	Test login manuel + captures écran	Backend dev
Tentatives échouées bloquées après 5	Query logs + test automatisé	QA
MFA activé sur tous les comptes	Admin user setup, TOTP QR généré	Backend dev
Tokens JWT non lisibles côté client	Inspection token (base64 décodage rejeté)	SecOps
Pas de credentials en clair en logs	Grep "password", "secret" dans logs.txt	SecOps

Rôles Obligatoires (RBAC)

Rôle	Permissions	Exemple
Admin	Tout (users, logs, configs, scan deletion)	Gestion infrastructure
Pentesteur	Créer/lancer scans, voir rapports, modifier tests	Exécution pentests
Analyste	Lancer scans prédéfinis, consulter rapports	Analyse résultats

À valider : Testez escalade de privilèges (utilisateur "Analyste" essaie d'accéder rôle "Admin" → REJETTÉ)

2. Exigences Chiffrement — CHECKLIST AUDIT

En Transit (HTTPS)

1. Certificat SSL/TLS valide, **TLS 1.3 minimum**
2. Pas de HTTP en production (redirect automatique HTTPS)
3. Cipher suites sécurisées uniquement (TLS_AES_256_GCM, CHACHA20_POLY1305)
4. Headers sécurité requis (HSTS, CSP, X-Frame-Options)

Au Repos (Base Données)

1. Données sensibles chiffrées **AES-256-GCM** (identifiants découverts, hashes, credentials)
2. Clés de chiffrement **non en code source**, stockées dans Vault ou env sécurisé
3. PostgreSQL pgcrypto OR bibliothèque Fernet (Python)

Checklist Validation Chiffrement

Contrôle	Preuve	Test
HTTPS actif, TLS 1.3	curl -I https://app.local → affiche TLS 1.3	SecOps
Certificat valide	openssl s_client -connect app:443 → pas d'erreur	SecOps
HSTS header présent	curl -I → "Strict-Transport-Security" visible	SecOps
BD chiffrée (at-rest)	Query raw file /var/lib/postgresql/data/ → données en binaire, pas lisible en clair	DevOps
Clés pas en git	git log -p grep -i "key secret" → aucun résultat	SecOps
Chiffrement teste en cycle	Insertion donnée sensible → déchiffrement autorisé (Admin) vs rejeté (Analyste)	Backend dev

Action concrète :

- Demander au backend : comment sont stockez les credentials découverts pendant les pentests en BD
 - Vérifier : `SELECT pgp_sym_decrypt(credentials::bytea, 'key') FROM findings` fonctionne (Admin seul)
-

3. Exigences Audit & Logs — CHECKLIST AUDIT

Événements à Logger Impérativement

- Connexions réussies / échouées
- Changements mot de passe, 2FA activé
- Modifications rôles/permissions utilisateurs
- Création/modification/suppression scans
- Lancement de scans (qui, quand, cible)
- Accès données sensibles (report téléchargé, credentials consultées)
- Erreurs système (exceptions, timeouts)
- Actions administrateur (import/export, config critiques modifiées)

Format des Logs

Obligation : JSON structuré, horodaté ISO 8601 UTC

```
{
  "timestamp": "2026-01-23T21:30:45Z",
  "event_type": "SCAN_LAUNCHED",
  "user_id": "alice",
  "action": "Scan lancé",
  "resource": "/scan/12345",
  "result": "SUCCESS",
  "severity": "MEDIUM"
}
```

Stockage Logs

Aspect	Exigence
Destination	Système centralisé (ELK, Graylog, fichier WORM) — pas sur même serveur que app
Immuabilité	Écriture seule, pas de modification retroactive
Rétention	90j en ligne, 2 ans archive froide minimum
Accessibilité	Logs consultables pour enquête forensique

Checklist Validation Logs

Contrôle	Preuve	Test
Logs centralisés	Démonstration ELK/Graylog avec 10+ entrées	Backend dev
Format JSON	Logs parsables en JSON (aucune erreur)	SecOps
Horodatage ISO 8601	Grep timestamp → "2026-01-23T21:30:45Z"	SecOps
Événements critiques présents	Recherche "USER_LOGIN", "SCAN_LAUNCHED" → trouvés	SecOps
Immuabilité testée	Tentative suppression log ancien → échouée	SecOps
Pas de données sensibles en clair	Grep "password", "api_key" dans logs → zéro résultat	SecOps

Test concret :

1. Créez utilisateur de test
2. Authentifiez-vous
3. Lancez un scan
4. Consultez logs centralisés

5. Vérifiez timeline reconstituable (qui a fait quoi, quand)

4. Conformité RGPD — CHECKLIST

Données Sensibles Collectées Pendant Pentests

Pendant un pentest, votre tool découvre :

- Identifiants (emails, usernames, noms)
- Hashes passwords
- Données personnelles (adresses, téléphones, numéros clients)
- Credentials trouvés

Obligation légale : Ces données doivent être traitées conformément RGPD (Article 5).

Minimisation & Suppression

Donnée	Durée Rétention	Action
Identifiants trouvés	1 an	Suppression auto après 365 jours
Hashes passwords	6 mois	Suppression auto après 180 jours
Données personnelles	Durée contrat pentest (ex: 6 mois)	Masquage/chiffrement renfor
Rapports pentests	2 ans (légal)	Archive chiffrée

Checklist RGPD

Contrôle	Preuve	Responsable
Inventaire données collectées	Document listant types données + source	SecOps
Durées rétention codées	Script suppression auto données expirées	Backend dev
Droit d'accès implémenté	API /data-export retourne JSON toutes données user	Backend dev
Droit oubli testé	Suppression compte → toutes données utilisateur party	Backend dev
Consent/Base légale	Contrat pentest mentionne "traitement données découvertes"	Client management

Action :

- Exiger du backend : Montrer script suppression données RGPD
- Tester : Créer finding ancien (>1 an) → vérifier suppression auto

5. Sécurité Docker/Infrastructure (si besoin)

Obligations

- Conteneurs **JAMAIS en mode --privileged** ou root
- Pas de secrets en variables d'env (utiliser Vault)
- Volumes montés en lecture-seule quand possible
- Scan images vulnérabilités (trivy, gype) avant production
- Logs Docker centralisés

Checklist

Contrôle	Commande de Test	Pass/Fail
Pas de mode privileged	<code>docker inspect toolbox \\\ grep Privileged → false</code>	—
Pas de root	<code>docker ps --format "table {{.Names}} {{.User}}" → user ≠ root</code>	—
Pas de secrets en env	<code>docker inspect \\\ grep -i password → zéro résultat</code>	—
Image signée	<code>docker image inspect --format='{{.RepoDigests}}' → digest présent</code>	—
Logs centralisés	Logs Docker → ELK/Graylog	—

6. Risques Non-Acceptables (Red Lines)

Si l'un de ces points échoue → **BLOCAGE** production :

1. **✗** Accès sans authentification → Pas de déploiement
2. **✗** Credentials en clair en code/logs → Pas de déploiement
3. **✗** HTTP sans HTTPS en production → Pas de déploiement
4. **✗** Logs modifiables (pas immuables) → Pas de déploiement
5. **✗** Données sensibles (RGPD) sans chiffrement → Pas de déploiement
6. **✗** Containers en mode root/privileged → Pas de déploiement

7. Outils SecOps Recommandés

Outil	Fonction	Installation
OpenSSL	Vérifier TLS config	apt install openssl
Burp Suite Community	Pen-test app (OWASP)	Gratuit, téléchargeable
OWASP ZAP	Scan sécurité automatisé	Gratuit, Docker
Trivy	Scan vulnérabilités Docker images	Docker : trivy image toolbox:latest
Hashicorp Vault	Gestion secrets/clés	Docker/Cloud
Graylog / ELK	Logs centralisés	Docker Compose