ATELIER 1 : CADRAGE ET SOCLE DE SÉCURITÉ
Application CareerBooster
Version Adaptée - Projet Étudiant

Équipe de 3 développeurs

12 novembre 2025

Table des matières

Liste des tableaux

Chapitre 1

ATELIER 1 : CADRAGE ET SOCLE DE SÉCURITÉ - VERSION RÉALISTE

1.1 Cadre de l'étude

1.1.1 Contexte du projet

- **Type de projet** : Projet étudiant / Proof of Concept
- **Équipe** : 3 développeurs étudiants
- **Budget sécurité** : Aucun budget dédié
- **Phase** : Développement/test (MVP)
- **Utilisateurs cibles** : 300 actuels, objectif 1000 en 6-12 mois
- **Infrastructure** : Self-hosted, environnement de développement

1.1.2 Objectifs adaptés

- **Objectif principal** : Identifier et traiter les risques critiques avec des moyens limités
- **Objectifs secondaires** :
  - Sensibiliser l'équipe aux bonnes pratiques de sécurité
  - Mettre en place des mesures de sécurité de base gratuites
  - Préparer une roadmap sécurité pour la croissance future
  - Documenter les choix techniques pour faciliter les audits futurs

1.1.3 Organisation simplifiée

1.1.4 Cycles temporels réalistes

- **Court terme (1-3 mois)** :
  - Correction des vulnérabilités critiques identifiées
  - Mise en place de l'authentification sécurisée
  - Configuration HTTPS et chiffrement de base
- **Moyen terme (3-6 mois)** :

TABLE 1.1 – Répartition des responsabilités (équipe de 3 personnes)

Activité	**Développeur Lead**	**Équipe (2 autres)**
Définition objectifs sécurité	R	C
Architecture sécurité	R	C
Implémentation sécurité	A	R
Tests sécurité	C	R
Documentation	R	C
Veille sécurité	R	I

  - Tests de sécurité automatisés dans le pipeline
  - Sauvegarde et plan de récupération
  - Documentation sécurité complète
- **Long terme (6-12 mois)** :
  - Premier audit externe (si financement obtenu)
  - Conformité RGPD basique
  - Préparation à la mise en production

1.2 Périmètre

1.2.1 Description du système

CareerBooster est une application mobile en développement permettant :

1. Analyse de CV par IA (Google Gemini)
2. Génération de CV optimisés (templates PDF)
3. Recommandations de formations
4. Suivi de progression

1.2.2 Valeurs métier prioritaires (contexte étudiant)

1. **Données utilisateurs** - Critique (même en phase test)
2. **Fonctionnement de l'application** - Important (pour démonstration)
3. **Code source** - Important (propriété intellectuelle)
4. **Réputation** - Moyenne (impact limité en phase test)

1.2.3 Biens supports simplifiés

1.3 Événements redoutés (ER) - Focus sur l'essentiel

1.4 Socle de sécurité - Approche pragmatique

1.4.1 Référentiels applicables (version allégée)

1. **OWASP Top 10** : Guide de base pour sécuriser l'application web

TABLE 1.2 – Biens supports identifiés

Catégorie	Biens supports	Criticité
Données	Base de données utilisateurs (PostgreSQL)	Critique
	CV stockés	Critique
	Logs d'application	Moyenne
Applications	Application mobile (React Native)	Importante
	API Backend (Spring Boot)	Importante
	Interface d'administration	Moyenne
Infrastructure	Serveur de développement	Importante
	Clés API externes (Gemini)	Critique
	Environnement de développement	Moyenne
Code	Repository Git	Importante
	Secrets et configurations	Critique

TABLE 1.3 – Événements redoutés prioritaires

ID	Événement redouté	Gravité	Justification
ER-01	Fuite des données utilisateurs (CV, informations personnelles)	G4	Impact RGPD même en phase test
ER-02	Compromission des clés API (Gemini, autres services)	G3	Coût financier direct, arrêt du service
ER-03	Injection de code malveillant dans l'application	G3	Compromission des utilisateurs testeurs
ER-04	Perte des données (pas de sauvegarde)	G2	Perte du travail de développement
ER-05	Indisponibilité prolongée du service	G2	Impact sur les démonstrations/tests

2. **RGPD - Principes de base** : Minimisation des données, consentement, droit à l'effacement
3. **Bonnes pratiques ANSSI** : Guide d'hygiène informatique
4. **Sécurité des APIs REST** : Authentification, autorisation, validation

1.4.2 État actuel et écarts

1.5 Plan d'action réaliste (ressources limitées)

1.5.1 Actions immédiates (0-1 mois) - Coût : 0€

1. **Sécurisation des secrets** :
   — Utiliser des variables d'environnement pour les clés API

TABLE 1.4 – Évaluation du socle de sécurité actuel

Domaine	Note	Acquis	À améliorer
Authentification	3/5	— Hachage des mots de passe — Critères de complexité — JWT basique	— Pas de 2FA — Gestion des sessions à améliorer — Rate limiting manquant
Chiffrement	2/5	— HTTPS en développement — Hachage mots de passe	— Chiffrement des données sensibles — Gestion sécurisée des clés — Certificats SSL valides
Validation	2/5	— Validation côté client — Swagger documentation	— Validation côté serveur stricte — Protection contre injections — Sanitisation des entrées
Monitoring	1/5	— Logs de développement	— Logs de sécurité — Monitoring des erreurs — Alertes automatiques
Sauvegarde	1/5	— Git pour le code	— Sauvegarde base de données — Plan de récupération — Tests de restauration

   - Ajouter .env au .gitignore
   - Rotation des clés API exposées
2. **Validation des entrées** :
   - Validation stricte côté serveur
   - Protection contre les injections SQL
   - Sanitisation des uploads de fichiers
3. **Configuration HTTPS** :
   - Certificat Let's Encrypt gratuit
   - Redirection HTTP vers HTTPS
   - Headers de sécurité de base

1.5.2 Actions à court terme (1-3 mois) - Coût : <100€

1. **Monitoring basique** :

   - Intégration d'un service gratuit (ex : Sentry free tier)
   - Logs structurés avec rotation
   - Alertes par email sur erreurs critiques
2. **Tests de sécurité automatisés** :
   - Intégration OWASP ZAP dans CI/CD
   - Scan des dépendances avec npm audit / Snyk
   - Tests unitaires pour les fonctions de sécurité
3. **Sauvegarde** :
   - Script de sauvegarde automatique de la DB
   - Stockage sur service cloud gratuit (Google Drive API)
   - Test de restauration mensuel

1.5.3 Actions à moyen terme (3-6 mois) - Coût : <500€

1. **Audit externe léger** :
   - Audit par un étudiant en cybersécurité
   - Utilisation d'outils gratuits (Nmap, Burp Suite Community)
   - Documentation des vulnérabilités trouvées
2. **Conformité RGPD basique** :
   - Politique de confidentialité
   - Mécanisme de suppression des données
   - Consentement explicite pour l'analyse IA
3. **Préparation production** :
   - Environnement de staging sécurisé
   - Procédures de déploiement sécurisé
   - Plan de réponse aux incidents

1.6 Métriques réalistes

1.6.1 Indicateurs de sécurité adaptés

- **Disponibilité** : 95% (objectif réaliste pour un projet étudiant)
- **Temps de correction des vulnérabilités critiques** : 7 jours
- **Couverture des tests de sécurité** : 60% des endpoints critiques
- **Fréquence des sauvegardes** : Quotidienne (automatisée)
- **Temps de récupération** : 4 heures maximum

1.6.2 Suivi mensuel

- Revue des logs de sécurité
- Mise à jour des dépendances
- Test de restauration des sauvegardes
- Formation continue de l'équipe (1h/mois)

1.7 Conclusion

Cette approche réaliste d'EBIOS Risk Manager permet de :

Points forts :

— Adapter la méthodologie aux contraintes réelles du projet

— Prioriser les risques critiques avec des moyens limités

— Établir une base solide pour la croissance future

— Sensibiliser l'équipe aux enjeux de sécurité

Limitations acceptées :

— Pas d'audit professionnel immédiat

— Monitoring basique

— Conformité RGPD simplifiée

— Tests de pénétration reportés

Évolution prévue : À mesure que le projet grandit et obtient des financements, les mesures de sécurité pourront être renforcées progressivement, en s'appuyant sur cette base solide.

L'atelier 2 se concentrera sur l'analyse détaillée des menaces identifiées et la définition de contrôles techniques spécifiques adaptés aux ressources disponibles.