DOCUMENTATION CELLULE-PRO

Modèle de Sécurité

Frontières de confiance · Ed25519 · JWT EdDSA

cellule.ai · pro.cellule.ai
2026-05-18

Table des matières

0.1 Modèle de menace — Cellule PRO

Ce document inventorie les menaces spécifiques au déploiement Cellule PRO et leur mapping vers les chantiers qui les adressent.

Hypothèses de base

- L'adversaire peut être **externe** (cyberattaque, vol) ou **interne** (employé malveillant, prestataire, admin IT malveillant)
- Toutes les communications sont TLS dans l'intranet (couche transport non-discussée ici)
- Les workers tournent sur des machines que CELLULE ne contrôle pas (postes employés)
- Le pool orchestrator et la DB sont hébergés par le client (admin IT client les gère)

1. Menaces externes

1.1 Exfiltration de données vers internet

Vecteur : un adversaire compromet un poste employé et tente d'exfiltrer les prompts/réponses via un worker malicieux.

Mitigation native : - Workers n'ont pas d'accès sortant internet (pare-feu intranet) - Communication worker pool via WebSocket sur réseau privé uniquement - Pas de dépendance cellule.ai public (aucun push sortant vers le domaine public)

Chantier associé : aucun chantier spécifique, c'est une contrainte de deployment (topologie réseau).

1.2 Vol de laptop employé

Vecteur : un laptop d'employé est volé, le worker continue de répondre aux requêtes de ses collègues.

Mitigation : - Admin IT révoque le token worker → worker disconnect du pool à la prochaine reconnexion - Fenêtre d'attaque : ~5-30 min (heartbeat pool)

Limitation : pendant la fenêtre, l'attaquant peut lire les prompts routés à ce worker. Mitigation : **chantier #4 proof of inference** détecte les anomalies statistiques + admin IT peut forcer un "worker rotation" après incident.

1.3 Man-in-the-middle intranet

Vecteur : attaquant sur le réseau intranet intercepte traffic worker pool.

Mitigation : - TLS intranet (obligatoire dans le deployment guide) - Ed25519 signature sur federation messages (chantiers #1, #3) — impossible de forger un message au nom d'un peer sans la clé privée

2. Menaces internes

2.1 Employé malveillant

Vecteur : employé configure son worker pour : - Logger les prompts de ses collègues - Retourner des réponses cachées au lieu d'inférer vraiment - Saboter les réponses (injection malveillante)

Mitigation : - **Chantier #4 proof of inference** : spot-check 3% + reputation tracking → détection statistique d'un worker qui triche >30% du temps - Mode déterministe max : si temp=0 seed fixé, le mismatch entre deux workers = preuve immédiate - Admin IT peut blacklister un worker suspect via dashboard enterprise

Limitation : un employé très discipliné qui ne triche qu'occasionnellement (<3% des requêtes) passe entre les mailles. Mitigation future : **chantier #4 mode** `required` = 100% redondance pour les tâches sensibles.

2.2 Admin IT malveillant

Vecteur : admin IT a les droits sur le pool orchestrator et la DB. Il peut tout voir en clair ?

Mitigation : - **Chiffrement Fernet zero-knowledge** : même avec accès DB postgres, admin IT voit des ciphertext. La clé est dérivée du token utilisateur (PBKDF2), qui n'est stocké nulle part en clair. - Admin IT voit : qui (token_hash SHA256 anonyme), quand, volume (taille cipher) — **pas** le contenu - Politique RGPD : admin IT signe un NDA strict à son embauche

Limitation : admin IT peut modifier le code pool pour injecter un keylogger. Détection : intégrité du container (`celluleai/pool:<signed-digest>` vérifié au boot via chantier release signing public).

2.3 Prestataire externe

Vecteur : consultant IT a un accès temporaire au réseau + laptop ajouté comme worker.

Mitigation : - Workers provisionnés avec un `expires_at` (ex : 30 jours) - Après expiration, worker rejeté par le pool même si tente de se reconnecter - Log audit : qui a ajouté ce worker, quand, par qui approuvé

3. Menaces cryptographiques / protocolaires

3.1 Replay attack sur federation

Vecteur : un ancien message signé est rejoué pour insérer une fact périmée.

Mitigation : - Timestamp + nonce dans la signature (chantier #1) - Anti-replay window 60s (chantier #3 anti-entropy `AUTH_TS_WINDOW_SEC`)

3.2 Collusion inter-peers

Vecteur : plusieurs peers bondés (trust 3) colluent pour injecter des facts poisons.

Mitigation : - Trust level 3 exige un handshake admin bilatéral (revue manuelle) - Supersede cross-pool sous LWW timestamp : un peer malveillant ne peut pas "revenir en arrière" - Réputation peer tracked — trop de mismatches = downgrade trust

3.3 Extraction d'information via embeddings

Vecteur : un peer cumule les query embeddings d'un user et tente d'inférer les sujets.

Mitigation : - **Chantier #3 privacy modes** : `balanced` ajoute du bruit DP, `paranoid` ne fanout jamais - Rate limiting 60 req/min/signer empêche les attaques par accumulation massive

3.4 Timing attack sur le TEE-less deployment

Vecteur : mesurer les latencies d'inférence pour inférer le contenu.

Mitigation : mode `required` batch les requêtes sur 2+ workers en parallèle, masquant les timings individuels.

Limitation : mitigation partielle. Mitigation complète nécessite un vrai TEE (différé, exclut workers CPU modestes).

4. Menaces sur la disponibilité

4.1 Worker unique critique en panne

Vecteur : un employé malade pendant 2 semaines → son worker n'est plus là.

Mitigation : - N workers redondants (chaque modèle tier a plusieurs workers) - Fallback serveur NPU pour les tâches qui nécessitent un modèle rare - Chantier #8 (futur) : rebalancing automatique par idle-governance

4.2 Serveur NPU tombe

Vecteur : le serveur d'inférence lourde est en panne.

Mitigation : - Workers CPU continuent de servir les tâches légères-moyennes (majorité du volume) - SLA déclare une dégradation acceptable ("réponses 7B au lieu de 30B pendant incident") - Hot-spare NPU recommandée pour clients SLA-stricts

4.3 Intranet isolé du multi-site

Vecteur : coupure VPN entre Paris et Lyon dans un déploiement multi-site.

Mitigation : - Chaque site a son propre pool + workers → service continue sur chaque site - Chantier #1 anti-entropy re-synchronise les facts au retour de la connectivité - Chantier #3 mode `paranoid` fonctionne naturellement sans fanout cross-site

5. Mapping menaces → chantiers

Menace	Chantier principal	Chantier complémentaire
Vol laptop employé	#4 (detection)	Token revocation manual (existe déjà dans public)
Employé malveillant — worker cheating	**#4**	Reputation tracking

Menace	Chantier principal	Chantier complémentaire
Admin IT malveillant	Chiffrement Fernet (public) + signed container	Audit logs
Extraction via embeddings cumulés	**#3**	Rate limit (déjà livré)
Collusion inter-peers	#1 LWW supersede + trust bilatéral	M11 public
Worker unique critique	Redundance workers	**#8** (futur idle governance)
Serveur NPU en panne	Routing fallback CPU	SLA contractuel
Multi-site WAN down	**#1** anti-entropy	Chaque site autonome

6. Threats que nous n'adressons PAS (encore)

- **Side-channels physiques** (keyloggers hardware, écoute radio) — hors scope logiciel
- **Attaques supply-chain sur les dépendances Python** — couverte partiellement par release signing public, pas plus
- **Attaques par corruption du modèle** (prompt injection pour fuir les system prompts) — sujet LLM général, pas spécifique Cellule PRO
- **Censorship résistance** — Cellule PRO tourne chez le client, pas un goal
- **Anonymat des employés entre eux** — les employés sont authentifiés par SSO, leurs actions sont attribuées (c'est volontaire pour l'audit)

7. Processus de réponse incident

1. Détection via monitoring (dashboard admin IT)
2. Isolation : admin IT révoque les workers/tokens suspects
3. Investigation : lecture des logs `inference_verifications` + `worker_reputation`
4. Remediation : reconfiguration workers, rotation secrets, rebuild container si compromission
5. Post-mortem documenté, communication client selon SLA

À retenir

Le modèle de menace Cellule PRO suppose que **l'intranet client est le premier périmètre de sécurité**, et que **chaque couche de défense est un diamètre de confiance réduit** :

1. Périmètre réseau (client + pare-feu client)
2. Trusted peers (trust 3 federation)
3. Workers (machines employés, variable confiance)
4. Pool orchestrator (controlled by IT admin)
5. Database (chiffré par user — zero-knowledge même contre admin)

Les chantiers #1-#9 sont les couches logicielles qui renforcent ce modèle en profondeur.