FR

Guide Pratique

OpenClaw : Securiser en Production
Auth, SSL, Reverse Proxy et Vault

Checklist securite complete pour deploiements agent IA

Mars 2026

Sommaire

Un agent IA avec acces a vos serveurs, vos emails et vos credentials est une surface d'attaque reelle. Ce guide documente comment securiser OpenClaw en production.

1. Pourquoi la securite est critique ici

Avec le self-hosted, la securite est votre responsabilite. La surface d'attaque est reelle :

• L'agent a acces au systeme de fichiers (cles SSH, tokens, configs)

• Il peut executer des commandes shell si le skill est active

• Il recoit des instructions via la messagerie (risque d'injection)

• Son workspace Git peut contenir des secrets dans l'historique

■ *Selon IDC 2025, 43% des incidents IA impliquent une mauvaise configuration des acces, pas une faille dans le modele.*

2. Authentification

Authorized senders

OpenClaw filtre les messages par identifiant d'expediteur. Seuls les expediteurs autorises declenchent un traitement. Tout message inconnu est ignore silencieusement.

Tokens API

Un token par integration, rotation tous les 90 jours, revocation immediate en cas de doute. Les tokens ne transitent jamais dans le workspace Git.

Whitelist de canaux

L'agent n'ecoute que le canal prive configure. Toute autre source est bloquee au niveau du gateway avant d'atteindre le runtime.

> *“Traiter l'agent comme un compte de service systeme. Acces etendus = permissions d'entree maximalement restrictives.”*

3. HTTPS et SSL

L'interface gateway ne doit jamais etre accessible en HTTP clair. Architecture recommandee : reverse proxy Nginx + certificat TLS Let's Encrypt. Le gateway ecoute en local, le reverse proxy fait la terminaison TLS.

```
server {
listen 443 ssl;
server_name agent.votre-domaine.com;
ssl_certificate /etc/letsencrypt/.../fullchain.pem;
location / { proxy_pass http://127.0.0.1:PORT; }
}
```

■ *Le renouvellement automatique certbot s'integre en cron systeme.*

4. Isolation reseau

Regle absolue : l'agent ne doit pas etre accessible depuis internet public. Le gateway peut rester entierement prive. Pour l'acces admin a distance, un reseau prive virtuel permet d'atteindre le gateway sans l'exposer.

Regles UFW recommandees :

• Port 22 (SSH) : IPs de gestion uniquement

• Port 443 : ferme si pas d'endpoint public necessaire

• Port gateway : loopback uniquement (127.0.0.1)

• Tout le reste : DROP par defaut

5. Gestion des secrets — vault obligatoire

Les credentials (tokens API, mots de passe, cles SSH) ne doivent jamais se retrouver dans des fichiers plats, l'historique Git, ou dans des prompts systeme.

Regles absolues :

• Jamais de credentials dans le workspace Git (meme .gitignore — l'historique garde tout)

• Jamais de secrets dans SOUL.md, MEMORY.md, AGENTS.md

• Jamais de tokens dans les prompts systeme

• Variables d'environnement chiffrees — injectees au demarrage, non persistees

> *“Les scripts recuperent les credentials dynamiquement depuis le vault a l'execution. Session expiree = credential inaccessible = blast radius limite.”*

6. Sandboxing — moindre privilege

Utilisateur systeme dedie

OpenClaw tourne sous un compte sans sudo, avec acces uniquement au workspace, logs et donnees des skills actives. Aucun acces aux repertoires systeme ou fichiers d'autres utilisateurs.

Skills limites

N'activer que les skills reellement utilises. Le skill exec est le plus puissant et le plus dangereux — documenter explicitement pourquoi il est present.

Isolation Docker

Pour les deployements avec exec : faire tourner OpenClaw dans un conteneur avec montages de volumes explicites. En cas de compromission, le blast radius reste confine au conteneur.

7. Audit et logs

Le workspace est un depot Git : chaque action significative est commitee. Ce que surveiller :

• Commits inhabituels : modifications de configs, ajout de cles, creation de scripts

• Activite hors heures habituelles : une action a 3h du matin merite investigation

• Tentatives d'acces refusees dans les logs gateway

• Pics inhabituels de tokens consommes

■ *BOTUM fait tourner un cron qui analyse les commits Git des 24h et genere un digest. Modification d'un fichier sensible = alerte automatique.*

8. Checklist securite production

- ■ Authorized senders configures et restreints aux identifiants connus
- ■ Reverse proxy HTTPS devant le gateway (Nginx + Let's Encrypt)
- ■ Gateway non expose sur internet public
- ■ Firewall UFW configure — seuls les ports necessaires ouverts
- ■ Vault de secrets en place — aucune credential en clair dans Git
- ■ Utilisateur systeme dedie sans sudo
- ■ Skills limites au strict necessaire
- ■ Logs Git — monitoring, alertes sur fichiers sensibles
- ■ Rotation des tokens planifiee (90 jours max)
- ■ Tests basiques : verifier que le gateway est inaccessible depuis l'exterieur

Conclusion

Securiser OpenClaw n'est pas une tache de 'phase 2'. Une heure de configuration initiale, quelques regles a maintenir. Le self-hosted donne un controle total — ce controle s'accompagne d'une responsabilite totale.

Billet 4 : Ne jamais exposer ses secrets dans le contexte IA — comment structurer SOUL.md et les fichiers d'agent.

Article complet : blog.botum.ca/openclaw-securiser-auth-ssl-reverse-proxy-vault

Site web : www.botum.ca • contact@botum.ca