EN

Practical Guide

OpenClaw: Securing in Production
Auth, SSL, Reverse Proxy and Vault

Complete security checklist for AI agent deployments

Mars 2026

Table of Contents

1. Why Security Is Critical Here
2. Authentication — Who Can Talk to the Agent
3. HTTPS and SSL — Never Expose in Plain HTTP
4. Network Isolation — Don't Expose to the Public Internet
5. Secrets Management — The Vault Is Mandatory
6. Sandboxing — Principle of Least Privilege
7. Audit and Logs — Native Git Traceability
8. Production Security Checklist

Conclusion

An AI agent with access to your servers, emails, and credentials is a real attack surface. This guide documents how to secure OpenClaw in production.

1. Why Security Is Critical Here

With self-hosting, security is your responsibility. The attack surface is real:

• The agent has filesystem access (SSH keys, tokens, configs)

• It can execute shell commands if the skill is enabled

• It receives instructions via messaging (prompt injection risk)

• Its Git workspace may contain secrets in commit history

■ *Per IDC 2025, 43% of enterprise AI incidents involve misconfigured access controls, not model vulnerabilities.*

2. Authentication

Authorized Senders

OpenClaw filters messages by sender identifier. Only explicitly authorized senders trigger processing. Unknown messages are silently ignored.

API Tokens

One token per integration, rotation every 90 days, immediate revocation when in doubt. Tokens never transit through the Git workspace.

Channel Whitelist

The agent only listens on the configured private channel. Any other source is blocked at the gateway level before reaching the runtime.

> *“Treat the agent like a system service account. Broad access = maximally restrictive input permissions.”*

3. HTTPS and SSL

The gateway interface must never be accessible over plain HTTP. Recommended architecture: Nginx reverse proxy + Let's Encrypt TLS certificate. The gateway listens locally, the reverse proxy handles TLS termination.

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

■ *Automatic certbot renewal integrates as a system cron.*

4. Network Isolation

Absolute rule: the agent must not be accessible from the public internet. The gateway can remain entirely private. For remote admin access, a private virtual network allows reaching the gateway without exposing it.

Recommended UFW rules:

• Port 22 (SSH): management IPs only

• Port 443: closed if no public endpoint needed

• Gateway port: loopback only (127.0.0.1)

• Everything else: DROP by default

5. Secrets Management — Vault Is Mandatory

Credentials (API tokens, passwords, SSH keys) must never end up in flat files, Git history, or system prompts.

Absolute rules:

• No credentials in the Git workspace (even .gitignore'd — history keeps everything)

• No secrets in SOUL.md, MEMORY.md, AGENTS.md

• No tokens in system prompts

• Encrypted environment variables — injected at startup, not persisted

> *"Scripts retrieve credentials dynamically from the vault at execution time. Expired session = inaccessible credential = limited blast radius."*

6. Sandboxing — Least Privilege

Dedicated System User

OpenClaw runs under a no-sudo account, with access only to the workspace, logs, and activated skill data directories. No access to system directories or other users' files.

Limited Skills

Only activate skills actually in use. The exec skill is the most powerful and dangerous — explicitly document why it's present.

Docker Isolation

For deployments requiring exec: run OpenClaw in a container with explicit volume mounts. If compromised, the blast radius stays confined to the container.

7. Audit and Logs

The workspace is a Git repository: every significant action is committed. What to monitor:

• Unusual commits: config file modifications, key additions, script creation

• Activity outside normal hours: a 3am action deserves investigation

• Denied access attempts in gateway logs

• Unusual token consumption spikes

■ *BOTUM runs a cron that analyzes Git commits from the past 24h and generates a digest. Modification of a sensitive file = automatic alert.*

8. Production Security Checklist

- ■ Authorized senders configured and restricted to known identifiers
- ■ HTTPS reverse proxy in front of the gateway (Nginx + Let's Encrypt)
- ■ Gateway not exposed to the public internet
- ■ UFW firewall configured — only necessary ports open
- ■ Secrets vault in place — no plaintext credentials in Git
- ■ Dedicated system user without sudo
- ■ Skills limited to strict necessity
- ■ Git logs — monitoring, alerts on sensitive files
- ■ Token rotation scheduled (90 days max)
- ■ Basic tests: verify gateway is unreachable from the outside

Conclusion

Securing OpenClaw is not a 'phase 2' task. One hour of initial configuration, a few rules to maintain. Self-hosting gives total control — that control comes with total responsibility.

Post 4: Never expose your secrets in the AI context — how to structure SOUL.md and agent files.

Full article: blog.botum.ca/openclaw-securing-auth-ssl-reverse-proxy-vault

Website: www.botum.ca • contact@botum.ca