M78Armor 妙手甲堡

OpenClaw 安全加固完全指南

The Complete Guide to securing OpenClaw

ℹ 18 个步骤覆盖 CVE 修复、网关锁定、凭证保护、工具权限、插件审查与浏览器隔离。
专为中国境内独立部署 OpenClaw 的开发者和小型团队设计。
18 security configuration steps covering CVE patching, gateway lockdown, credential protection, tool policy, plugin auditing, and browser isolation.
Designed for solo developers and small teams running self-managed OpenClaw deployments in China.

Version 1.0· Last Verified: 2026-04-09

Baseline: Official OpenClaw Docs · GitHub Releases · NVD · CNCERT

Recommended Floor: v2026.3.11+ · Current Stable: v2026.4.9

重要免责声明 Important Disclaimer

本指南仅用于对你合法拥有或已获授权的 OpenClaw 环境进行防御性加固。
This guide is for defensive hardening & security configuration of OpenClaw environments you own or are authorized to administer.

本指南不构成对《网络安全法》《数据安全法》《个人信息保护法》或任何行业要求的合规保证。
This guide does not guarantee compliance with the CSL, DSL, PIPL, or sector-specific requirements.

禁止将本指南、相关命令或配套工具用于扫描、探测、攻击未授权系统。
Do not use this guide, its commands, or related tools against unauthorized systems.

本指南不包含任何 PoC 利用代码，仅针对已公开且已修复漏洞提供防御性缓解建议。符合《网络安全法》第二十八条关于防御性安全信息发布的要求。
This guide contains no PoC exploit code. Provides defensive mitigations only for publicly disclosed, patched vulnerabilities. Complies with Article 28 of the Cybersecurity Law.

修改 Docker 网络、文件权限、认证和浏览器控制策略前，请先备份。
Back up before changing Docker networking, file permissions, auth, or browser-control settings.

Move78 International 与 M78Armor 不对因误配置、停机、数据丢失或密钥泄露产生的直接或间接损失负责。
Move78 International and M78Armor disclaim liability for direct or indirect loss from misconfiguration, downtime, data loss, or credential exposure.

文档目的 Purpose of This Document

本指南为在中国境内部署 OpenClaw 的独立开发者和小型团队提供一份可直接执行的安全加固检查清单。涵盖 18 个加固步骤，覆盖已公开 CVE 漏洞修复、网关锁定、凭证保护、工具权限收紧、插件审查和浏览器隔离，目标是将 OpenClaw 从默认危险状态提升到基本防御基线。
This guide provides an actionable security configuration checklist for solo developers and small teams deploying OpenClaw in China. It covers 18 security configuration steps spanning CVE patching, gateway lockdown, credential protection, tool policy tightening, skill auditing, and browser isolation.

为什么现在要做 Why This Matters Now

中国国家互联网应急中心（CNCERT）已公开提示 OpenClaw 存在默认配置脆弱、插件投毒、提示词注入、权限过高和漏洞利用等安全风险。如果你正在公网、VPS、云主机、办公室网络或共享网络中运行 OpenClaw，而没有做下面这些最基本的加固，你的实例很可能已经处于高风险状态。
CNCERT has publicly warned that OpenClaw deployments can be exposed by weak defaults, malicious skills, prompt injection, excessive privileges, and published vulnerabilities. If you run OpenClaw on a public server, VPS, cloud VM, office network, or shared network without these controls, your deployment is likely high risk.

Risk Severity Dashboard 风险等级总览

Severity	Count	Steps
CRITICAL	7	0, 1, 2, 3, 4, 7, 10, 11
HIGH	6	5, 8, 12, 14, 16, 17
MEDIUM	3	6, 9, 13, 15

Key Threat Vectors 主要威胁向量

- CVE-2026-25253: 一键远程代码执行 (1-Click RCE)
  WebSocket hijack — affects versions before v2026.1.29
- CVE-2026-32922: 权限提升漏洞 (CVSS 9.9)
  Privilege escalation — affects versions before v2026.3.11
- CVE-2026-34511: Gemini OAuth PKCE 泄露
  Token theft via redirect URI — affects versions before v2026.4.2
- ClawHavoc: 供应链投毒，1,200+ 恶意技能
  Malicious skills distributed via supply chain attack
- YOLO 模式 (v2026.4.2+): 主机命令无审批默认执行
  Host exec defaults to no-approval mode in latest versions

使用方式 How to Use This Guide

建议执行路径 Recommended Execution Paths

根据你的时间和风险优先级，先选择一种执行路径。不要把它当成一次性通读的白皮书。
Choose one execution path first based on your time and risk priority. Do not treat this like a white paper to read front to back.

- ⚡ 紧急 15 分钟：先做 Step 0、1、2、8、11、17，先堵住最容易直接变成 RCE 或公网暴露的面。
  Emergency 15-minute pass: do Steps 0, 1, 2, 8, 11, and 17 first. These close the fastest paths to remote execution or public exposure.
- ⏱ 1 小时基线：再做 Step 3、4、7、12、14、16，把权限、插件、远程访问和代理信任收紧。
  One-hour baseline: then do Steps 3, 4, 7, 12, 14, and 16. Tighten permissions, plugins, remote access, and proxy trust.
- ☑ 完整加固：最后完成 Step 5、6、9、10、13、15 和 Appendix A。
  Full security configuration pass: finish Steps 5, 6, 9, 10, 13, 15, and Appendix A.
- 📋 通用规则：每一步先看警告框，再执行命令，再做验证；每次升级、改配置、加插件或开新通道后都重跑 openclaw security audit --deep。
  General rule: read the warning first, run the command, then verify; after any upgrade, config change, plugin, or new channel, rerun openclaw security audit --deep.

Step 0 — Pre-Flight: 开始前请先确认
Pre-Flight Checklist

CRITICAL
0 — Pre-Flight of 18

🎯 **在执行任何加固步骤之前，先完成下面 6 项确认。**
Complete these 6 checks before touching any security configuration steps.

1. 先做备份：至少备份 ~/.openclaw/openclaw.json、~/.openclaw/workspace/、~/.openclaw/credentials/。
Back up at minimum: openclaw.json, workspace/, and credentials/.

2. 确认版本：先确认版本不低于 v2026.3.11，最好直接升到当前稳定版。
Make sure you are at least on v2026.3.11, preferably current stable.

3. 确认安装方式：先弄清楚你是原生安装、WSL2、还是 Docker 部署。
Know whether you are running native, WSL2, or Docker.

4. 确认权限：你需要有修改配置、重启 Gateway、调整文件权限的权限。
You need permission to edit config, restart the Gateway, and change file permissions.

5. 先跑一次体检：记录当前状态，后面出错时更容易回滚。
Run a baseline check first so rollback is easier later.

6. 记录关键配置：至少记下当前的 gateway.bind、gateway.auth.mode、插件白名单和是否启用了 browser evaluate。
Note gateway.bind, gateway.auth.mode, plugin allowlist, and browser evaluate status.

```
Terminal
$ openclaw --version
$ openclaw config validate
$ openclaw doctor
$ openclaw security audit --deep
```

快速健康检查 Quick Health Check

```
Terminal
$ openclaw security audit
$ openclaw security audit --deep
```

⚠ **谨慎使用 --fix：运行 openclaw security audit --fix 之前，请确认你已完成上面的备份。--fix 会自动修改配置。**
Use --fix with caution: run openclaw security audit --fix only AFTER completing backup. --fix will modify your configuration automatically.

openclaw security audit --deep 高层检查项 What --deep checks:

- 入站访问策略（陌生人能否触发机器人）
  Inbound access policy
- 工具爆炸半径（提示词注入能否变成文件/命令/网络动作）
  Tool blast radius
- 执行审批漂移（你的 exec 防护是否仍然按预期工作）
  Exec approval drift
- 网络暴露（bind/auth、Tailscale Serve/Funnel 等）
  Network exposure
- 浏览器控制暴露（浏览器控制服务、远程 CDP 风险）
  Browser control exposure
- 本地磁盘卫生（权限、软链接、配置包含、同步目录）
  Local disk hygiene
- 插件与策略漂移（插件、allowlist、sandbox/工具策略不一致）
  Plugin and policy drift

ℹ **注意：本指南覆盖了大部分高信号风险，但不覆盖所有策略漂移与高级架构问题。**
This guide covers most high-signal risks, but not every policy-drift or advanced architecture edge case.

Step 0: 升级到安全版本
Upgrade to a Safe Version

CRITICAL
0 of 18

🎯 **先把版本抬到安全基线，再继续其它步骤。**
Reach a safe version floor before doing anything else.

如果你的版本低于 v2026.3.11，你暴露于 CVE-2026-32922。
If below v2026.3.11, exposed to CVE-2026-32922.
如果你的版本低于 v2026.4.2，你还暴露于 CVE-2026-34511 — Gemini OAuth PKCE 泄露。
If below v2026.4.2, you are also exposed to CVE-2026-34511 — Gemini OAuth PKCE verifier leak.

⚠ **推荐升级目标：v2026.3.11 是最低安全基线，实际建议始终升级到当前稳定版（截至 2026-04-06 为 v2026.4.5）。**
v2026.3.11 is minimum safe floor; recommended target is always current stable (v2026.4.5 as of 2026-04-06).

```
Terminal
$ openclaw --version
$ openclaw update
$ openclaw --version
```

Docker 用户：不要自己发明镜像更新流程。优先使用官方 Docker 安装/设置流程。
Docker users: Do not invent your own image update flow. Use the official Docker install/setup flow.

Step 1: 仅绑定 loopback
Keep the Gateway on Loopback

CRITICAL
1 of 18

🎯 **让 Gateway 默认只对本机开放。**
Keep the Gateway reachable only from localhost unless remote access is explicitly required.

```
Terminal
{ "gateway": { "bind": "loopback" } }
```

检查监听状态 Check the actual listener:

```
Terminal
$ ss -tlnp | grep 18789
```

- 看到 127.0.0.1:18789 = 仅本机可达（安全）
  127.0.0.1:18789 = localhost only (safe)
- 看到 0.0.0.0:18789 或公网 IP = 已暴露（危险）
  0.0.0.0:18789 or public IP = exposed (dangerous)

重要：ss/netstat 显示的是监听 IP；配置文件里应写 bind mode，不是 127.0.0.1/0.0.0.0 这样的别名。
ss/netstat show listener IPs; the config should use bind modes, not host aliases.

```
Terminal
$ openclaw gateway restart
```

⚠ **Docker 注意：即使应用内 bind 设为 loopback，docker-compose.yml 里 "18789:18789" 仍可能对外监听。必须写为：ports: - "127.0.0.1:18789:18789"**
Even if app bind is loopback, Docker may still publish broadly. Use: ports: - "127.0.0.1:18789:18789"

Step 2: 启用网关认证
Turn On Gateway Authentication

CRITICAL
2 of 18

🎯 **任何非 loopback 暴露都必须有认证。**
Require authentication whenever the Gateway is reachable beyond localhost.

lan/tailnet/custom/auto（非 loopback）场景下认证是必须的。
On lan/tailnet/custom/auto, auth is mandatory.

Token Mode (Recommended) 推荐：Token 模式

```
Terminal
$ openclaw doctor --generate-gateway-token
# Or manually:
$ node -e "console.log(require('crypto').randomBytes(32).toString('base64url'))"
$ export OPENCLAW_GATEWAY_TOKEN="YOUR_LONG_RANDOM_TOKEN"
# In openclaw.json:
{ "gateway": { "auth": { "mode": "token" } } }
```

Step 3: 收紧状态目录与凭证权限
Lock Down State and Credential Permissions

CRITICAL
3 of 18

🎯 **先把状态目录、工作区和凭证文件收紧到最小权限。**
Lock down state, workspace, and credential files to minimum necessary permissions.

```
Terminal
$ chmod 700 ~/.openclaw
$ chmod 600 ~/.openclaw/openclaw.json
$ chmod 700 ~/.openclaw/workspace
$ chmod -R 600 ~/.openclaw/credentials/ 2>/dev/null
$ ls -la ~/.openclaw/
```

- ~/.openclaw/openclaw.json
  Main configuration
- ~/.openclaw/workspace/
  Agent workspace
- ~/.openclaw/credentials/
  Stored credentials
- ~/.openclaw/agents/<agentId>/agent/auth-profiles.json
  Auth profiles

ℹ **Windows 用户：OpenClaw 官方仍建议 WSL2 作为完整体验的更稳定路径。**
Windows users: OpenClaw officially recommends WSL2 for the full experience.

Step 4: 清除明文密钥并改用 SecretRef
Replace Plaintext Secrets with SecretRef

CRITICAL
4 of 18

🎯 **把明文密钥从配置里清出去，并在每次升级后复查。**
Remove plaintext secrets from config and re-check after every upgrade or config write.

```
Terminal
$ grep -i -E "(token|secret|key|password|api_key)" ~/.openclaw/openclaw.json
$ openclaw secrets audit
```

- "${VAR_NAME}" 字符串替换
  "${VAR_NAME}" string substitution
- SecretRef 对象
  SecretRef objects: { source: "env", provider: "default", id: "VAR_NAME" }

```
Terminal
$ openclaw secrets reload
$ openclaw secrets audit
```

⚠ **重要现实风险：官方 issue tracker 仍有报告指出升级、迁移或配置写回路径可能把明文重新写入磁盘。每次跑完 openclaw doctor、configure、升级或迁移之后，重新检查配置文件。**
Issue tracker reports that upgrades and migrations may write resolved secrets back to disk. Re-check config after openclaw doctor, configure, upgrades, or migrations.

Step 5: 隔离私信会话
Isolate DM Sessions

HIGH
5 of 18

🎯 **避免不同发送方共享同一条私信上下文。**
Stop DM users from inheriting each other's context.

```
Terminal
{ "session": { "dmScope": "per-channel-peer" } }
```

多账号渠道可考虑：
For multi-account channels, consider:

```
Terminal
{ "session": { "dmScope": "per-account-channel-peer" }, "dmPolicy": "pairing",
"contextVisibility": "allowlist" }
```

```
Terminal
$ openclaw pairing list <channel>
$ openclaw pairing approve <channel> <CODE>
```

- dmPolicy 决定陌生人能否触发机器人。
  dmPolicy controls whether unknown senders can trigger the bot.
- session.dmScope 决定不同发送方是否共用上下文。
  session.dmScope controls whether different senders share context.

Step 6: 加上防火墙与网络边界
Add a Firewall and Network Boundary

MEDIUM
6 of 18

🎯 **在应用层之外再加一层网络边界。**
Add a second network boundary outside the application itself.

```
Terminal
$ sudo ufw default deny incoming
$ sudo ufw default allow outgoing
$ sudo ufw allow 22/tcp
$ sudo ufw enable
```

⚠ **Docker 注意：UFW 不能可靠覆盖 Docker 暴露端口；优先 127.0.0.1 绑定，必要时在 DOCKER-USER 链中加规则。**
UFW does not reliably protect Docker-published ports; prefer localhost binds and enforce in DOCKER-USER chain.

云主机注意：在阿里云、腾讯云上同时在安全组中封锁网关端口。
Also block the gateway port in cloud security groups (Aliyun, Tencent Cloud).

IPv6 注意：如果双栈主机，确认 IPv6 没有暴露同一服务。
If dual-stack, confirm IPv6 is not exposing the same service.

Step 7: 审查技能并删除可疑项
Audit Skills and Remove Suspicious Ones

CRITICAL
7 of 18

🎯 **先确认你运行的每个 skill 都可信且必要。**
Verify every installed skill is trusted and truly needed.

```
Terminal
$ openclaw skills list
$ openclaw skills info <skill-name>
$ openclaw skills check
```

High-risk permission combinations 高风险权限组合

- filesystem:write + child_process → RCE 风险
  RCE risk
- filesystem:write + network:outbound → 数据外传
  Data exfiltration
- credentials:read + network:outbound → 密钥泄露
  Key leakage

ℹ **删除注意：官方 CLI 没有 openclaw skills uninstall。安全做法是 list/info/check 后删除可疑 skills/ 目录。**
Official CLI has no uninstall command. Run list/info/check, then remove suspicious skill directories manually.

Step 8: 收紧工具执行权限
Restrict Host Execution

HIGH
8 of 18

🎯 **把主机命令执行权限收到最小必要范围，并强制人工审批。**
Restrict host exec to a minimum allowlist and force human approval on every execution.

🚨 **v2026.4.2+ 用户特别注意：gateway/node host exec 默认已切到 YOLO mode（security="full", ask="off"）。不主动覆盖则主机命令可能无审批直接执行。**
For v2026.4.2+: host exec defaults to YOLO mode. If you do not override it, host commands may run without any approval prompt.

Terminal
```
{
  "tools": {
    "profile": "messaging",
    "deny":
["group:automation","group:runtime","group:fs","sessions_spawn","sessions_send"],
    "fs": { "workspaceOnly": true },
    "exec": { "security": "allowlist", "ask": "always" },
    "elevated": { "enabled": false }
  }
}
```

Terminal
```
$ openclaw approvals get --gateway
```

allowlist + ask=always 覆盖 v2026.4.2+ 默认 YOLO 行为，保留经过人工确认的合法 exec 工作流。
allowlist + ask=always overrides YOLO default while preserving legitimate exec workflows behind human approval.

如果完全不需要 host exec，最严格做法是直接 deny exec。
If you never need host exec, deny it entirely.

Step 9: 最小化日志与留存
Minimize Logging and Retention

MEDIUM
9 of 18

🎯 **保留足够排障信息，但不要把敏感内容长期写进日志。**
Keep enough logs for troubleshooting without turning logs into a second secrets store.

Terminal
```
{ "logging": { "level": "info", "redactSensitive": "tools" } }
```

Step 10: 保护工作区记忆文件

CRITICAL
10 of 18

Protect Workspace Memory Files

🎯 **封住最容易被忽略的持久化入口。**
Protect the memory files attackers often use for quiet persistence.

```
Terminal
$ cat ~/.openclaw/SOUL.md
$ cat ~/.openclaw/MEMORY.md
$ cat ~/.openclaw/AGENTS.md
```

Look for 检查是否存在

- 不是你自己写的系统规则
  Instructions you did not add
- 不认识的外链、下载地址、shell 片段
  Unknown links, download URLs, or shell fragments
- "忽略安全规则""自动批准执行" 等提示
  "ignore security policy" or "auto-approve execution" rules

```
Terminal
$ chmod 700 ~/.openclaw/workspace
$ chmod 444 ~/.openclaw/SOUL.md
$ chmod 444 ~/.openclaw/MEMORY.md
```

更强选项（谨慎使用）：
Stronger option (use carefully):

```
Terminal
$ sudo chattr +i ~/.openclaw/SOUL.md
$ sudo chattr +i ~/.openclaw/MEMORY.md
```

⚠ **chattr +i 会阻止合法更新；只在文件应长期稳定时使用。**
chattr +i blocks legitimate updates; use only when you want those files frozen.

Step 11: 关闭高风险浏览器执行面

Disable High-Risk Browser Execution Surfaces

CRITICAL
11 of 18

🎯 **关闭高风险浏览器执行面，并只在必要时开放 CDP。**
Disable the riskiest browser execution surfaces and restrict CDP to known-safe paths.

```
Terminal
{ "browser": { "evaluateEnabled": false } }
```

sandbox browser 用户还应限制 CDP 来源：
Sandboxed browser users should also restrict CDP ingress:

```
Terminal
```

```
{ "agents": { "defaults": { "sandbox": { "browser": { "cdpSourceRange":
"172.21.0.1/32" } } } } }
```

ℹ 只在清楚 **Docker/sandbox 网络拓扑时再设置 cdpSourceRange**。
Only set cdpSourceRange if you understand your Docker/sandbox network topology.

Step 12: 显式控制插件白名单
Use an Explicit Plugin Allowlist

HIGH
12 of 18

🎯 **明确只允许你真正需要的插件。**
Allow only the plugins you intentionally need.

```
Terminal
{ "plugins": { "allow": ["browser"] } }
$ openclaw plugins list
$ openclaw plugins inspect <id>
```

⚠ **关键陷阱：仅 browser.enabled=true 还不够；browser 必须出现在 allowlist 里。用 openclaw plugins list/inspect 确认真实插件 ID。**
browser.enabled=true is not enough; browser must be in the allowlist. Verify real plugin IDs with list/inspect.

🚨 **v2026.4.2 之前 Channel Shadow 风险：恶意 workspace plugin 可能在 channel setup 期间执行代码。不要在不信任的克隆目录中启动 OpenClaw，优先升级到 v2026.4.2+。**
Before v2026.4.2, malicious workspace plugins could execute during channel setup. Never launch in untrusted cloned workspaces.

Step 13: 关闭 mDNS/Bonjour 发现
Disable mDNS and Bonjour Discovery

MEDIUM
13 of 18

🎯 **如果不需要自动发现，就把它关掉。**
Disable auto-discovery unless you have a real need for it.

```
Terminal
{ "discovery": { "mdns": { "mode": "off" } } }
# Or via environment variable:
$ export OPENCLAW_DISABLE_BONJOUR=1
```

需要最小发现能力可用 minimal。
If you need limited discovery, use minimal.

Step 14: 安全开放远程 UI
Expose Remote UI Safely

HIGH
14 of 18

🎯 **远程访问只在显式来源白名单和强认证下开放。**
Expose remote UI only with explicit origins and strong authentication.

Official audit flags as HIGH risk: 以下配置会被审计标记为高风险

- 非 loopback Control UI 没有显式 gateway.controlUi.allowedOrigins
  Without explicit allowedOrigins
- allowedOrigins: ["*"] 全放开
  Wildcard origin allowlists
- Loopback + SSH/Tailscale Serve 是最安全默认值
  Loopback + SSH/Tailscale Serve is the safest default
- Funnel 是公网暴露，不要当成方便的远程访问
  Funnel is public exposure, not just convenient remote access
- 非 loopback 暴露必须配合 token/password/identity-aware proxy
  Non-loopback must use token, password, or identity-aware proxy

Step 15: 限制自动化 cron 任务
Restrict Automated Cron Jobs

MEDIUM
15 of 18

🎯 **不要让定时任务成为静默持久化通道。**
Keep scheduled jobs from becoming a quiet persistence path.

```
Terminal
$ openclaw cron add --name "Daily check" --every "24h" \
  --message "Review logs and summarize only" \
  --tools read,web_fetch --no-deliver
```

除非任务确实需要 exec 或 browser，否则不要给。
Do not grant exec or browser unless truly needed.

Step 16: 仅信任显式代理来源
Trust Only Explicit Proxies

HIGH
16 of 18

🎯 **只信任明确配置的代理，不信任可伪造的来源头。**
Trust only explicitly configured proxies, not spoofable client-IP headers.

```
Terminal
{ "gateway": { "allowRealIpFallback": false } }
```

ℹ **命名陷阱：审计标签 gateway.real_ip_fallback_enabled，配置键 gateway.allowRealIpFallback。写入时用配置键。**
Audit label is gateway.real_ip_fallback_enabled, config key is gateway.allowRealIpFallback. Use the config key when writing.

已知相关漏洞：CVE-2026-32029 影响 v2026.2.21 之前版本。
Known related issue: CVE-2026-32029 affects versions before v2026.2.21.
正确做法：使用 trustedProxies 明确信任代理 IP，不要把 X-Real-IP 回退当默认安全配置。
Trust only explicit proxy IPs with trustedProxies.

Step 17: 阻止浏览器访问内网

HIGH

Block Browser Access to Private Networks

🎯 **阻止浏览器工具探测你的内网。**
Stop the browser tool from reaching private-network targets unless you deliberately allow it.

```
Terminal
{
  "browser": {
    "ssrfPolicy": {
      "dangerouslyAllowPrivateNetwork": false
    }
  }
}
```

ℹ **v2026.4.5 中旧别名 allowPrivateNetwork 已废弃；使用 dangerouslyAllowPrivateNetwork。**
Legacy alias allowPrivateNetwork deprecated in v2026.4.5; use dangerouslyAllowPrivateNetwork.

Appendix A: 理解 YOLO 模式 Understanding YOLO Mode (v2026.4.2+)

YOLO 模式不是单独漏洞，是官方为受信任单操作员场景的高便利默认值。问题：很多用户升级后没意识到 host exec 默认从 ask before run 变成了 run by default。
YOLO mode is not a vulnerability — it is an intentional convenience default. The problem is users upgraded without realizing host exec changed from ask-before-run to run-by-default.

- exec.security = "full" → 不做命令过滤
  No command filtering
- exec.ask = "off" → 不弹审批提示
  No approval prompts

如果实例同时存在提示词注入、恶意技能、过宽权限或公网暴露，YOLO 模式直接放大成主机命令执行风险。
Combined with prompt injection, malicious skills, or public exposure, YOLO mode amplifies into host-command execution risk.

How to check: 检查当前状态

```
Terminal
$ openclaw config get tools.exec
$ openclaw approvals get --gateway
```

Recommended exit path: 推荐配置

```
Terminal
{ "tools": { "exec": { "security": "allowlist", "ask": "always" } } }
```

覆盖 v2026.4.2+ 默认 YOLO 行为，拉回到 allowlist + 人工确认的保守基线。
Overrides the v2026.4.2+ default and pulls host exec back to allowlist plus human confirmation.

> ℹ **M78Armor 自动化：m78armor.js --harden 会检测 YOLO 模式，并提示切换到更安全的 allowlist 基线。**
> *m78armor.js --harden detects YOLO mode and prompts you to move to a safer allowlist baseline.*

最终验证 Final Verification

```
Terminal
$ openclaw security audit --deep
```

Target: 0 critical findings

Audit Finding Category	Relevant Steps
Inbound DM allowlist	Step 1, 2, 5
Tool blast radius / exec drift	Step 8
Network exposure / Tailscale bind	Step 1, 6, 14
Browser exposure / CDP / SSRF	Step 11, 17
Local disk / permissions / memory	Step 3, 4, 10
Plugins allowlist	Step 7, 12
Discovery leakage	Step 13
Header spoofing / proxy IP	Step 16
Scheduled persistence / cron	Step 15

建议每次修改配置、升级版本、增加插件或启用新通道后重新跑一次。
Re-run after config changes, upgrades, plugin changes, or enabling new channels.

本指南未完全覆盖的内容 Scope Limitations

- Hook ingress token 设计与复用问题
  Hook ingress token design and reuse problems
- 全量 trusted-proxy 部署细节
  Full trusted-proxy deployment details
- 多租户/敌对用户共享 Gateway 的隔离架构
  Isolation for hostile multi-user shared gateways
- 全部 plugin migration / policy-drift 边界条件
  Every plugin migration / policy-drift edge case
- 企业级日志、保留、取证与回滚流程
  Enterprise logging, retention, forensics, and rollback workflows

下一步 Next Step

如果你已经完成手工基线，并且后续还要重复执行、交给同事执行或降低误配置概率，可以再用 M78Armor 自动化这些重复检查。
If you have completed the manual baseline and need to repeat these checks, hand them to teammates, or reduce configuration mistakes, use M78Armor to standardize the recurring work.

```
Terminal
$ node m78armor.js --scan
$ node m78armor.js --harden
```

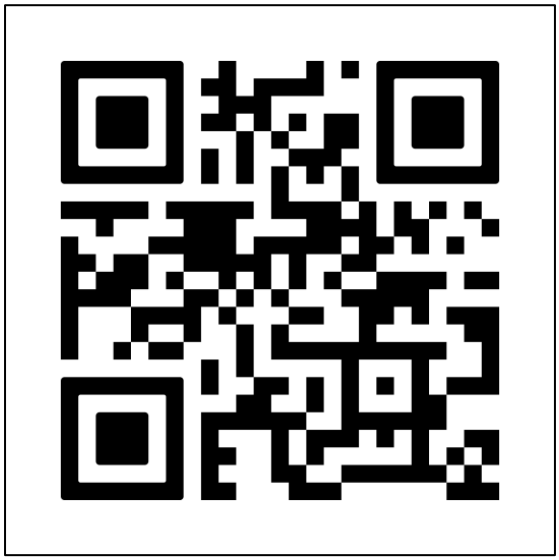
你刚刚花了 60–90 分钟手工完成这些步骤。M78Armor ¥88 版本可以把下次的重复工作压缩到 10 分钟内，并自动识别 ClawHavoc 恶意技能签名。
You just spent 60–90 minutes doing this manually. The ¥88 tier automates the recurring work down to 10 minutes and auto-detects ClawHavoc malicious-skill signatures.

功能 Feature	基础版 ¥88	专业版 ¥188
自动扫描 + 加固 *Automated scan + security configuration*	✓	✓
ClawHavoc 恶意技能签名库 *Malicious-skill signature set (ClawHavoc)*	✓	✓
安装脚本 + 入门文档 + FAQ *Install scripts + docs + FAQ*	✓	✓
本地运行，不上传数据 *Local execution, no data upload*	✓	✓
部署清单工作簿 *Deployment inventory workbook*	—	✓
安全基线配置手册 *Secure baseline config manual*	—	✓
应急响应速查卡 *Incident-response cheatsheets*	—	✓
合规触发地图 *Compliance trigger map*	—	✓

联系我们 Contact Us

support@m78armor.com
www.m78armor.com

用微信扫码直达官网 / WeChat QR to visit site:



Want the faster path?

Get the paid M78Armor automated security configuration script for a quicker OpenClaw security review and remediation workflow.
Scan the QR code to open the order request page.

想走更快的路径?

获取付费版 M78Armor 加固脚本，更快完成 OpenClaw 安全检查与加固处理。
扫码打开订购申请页面。