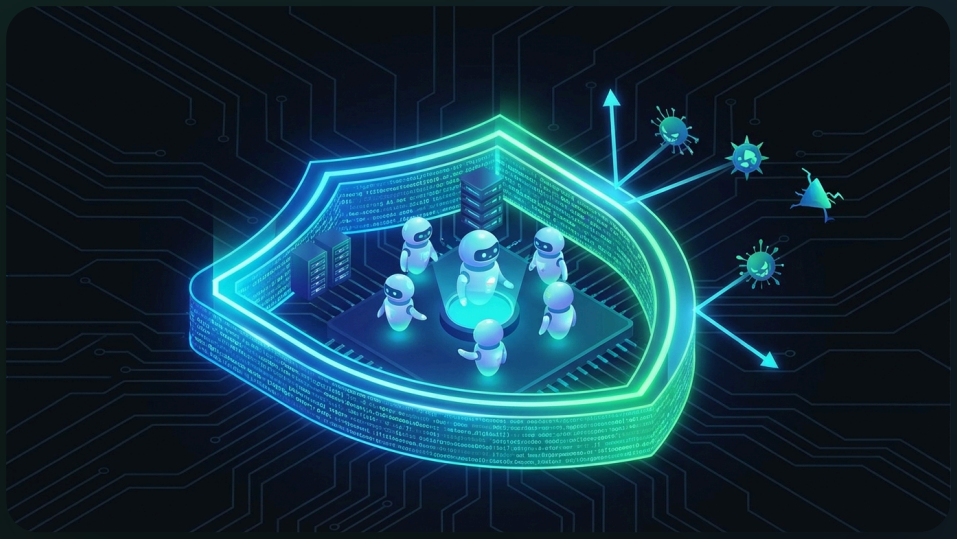
OPENCLAW

Lock It Down

Everything you need to secure your AI assistant — and what to NEVER do.



Version 1.0 · February 2026 · openclaw-guide.com

Table of Contents

Version 1.0 | February 2026

Everything you need to secure your AI assistant — and what to NEVER do.

Why This Matters

Your OpenClaw assistant has access to:

- Your files and code
- Your messages and emails
- Your calendar and contacts
- Your API keys and credentials

A compromised agent = a compromised life.

This checklist covers the essential security practices.

The 10 Commandments of OpenClaw Security

1. NEVER Put API Keys in Chat

❌ **Wrong:**

```
TERMINAL
Use this API key: sk-ant-api03-xxxxx
```

✅ **Right:**

```
TERMINAL
The API key is in ~/.openclaw/secrets/api.env
```

Why: Chat logs are stored. If compromised, keys are exposed.

2. Use Environment Variables

❌ **Wrong:**

```
JSON
{
  "anthropicApiKey": "sk-ant-xxxxx"
}
```

✅ **Right:**

```
JSON
{
  "anthropicApiKey": "${ANTHROPIC_API_KEY}"
}
```

Implementation:

```bash
# In ~/.zshrc or ~/.bashrc
export ANTHROPIC_API_KEY="sk-ant-xxxxx"
export OPENAI_API_KEY="sk-xxxxx"

# Reload shell
source ~/.zshrc
```

3. Separate Information vs Action Channels

Information channels (lower risk):

- Discord public channels
- Telegram groups
- Team Slack channels

Action channels (higher risk):

- Telegram DM (can execute commands)
- Discord DM with permissions
- Slack DM with workspace access

Recommended Setup

```json
{
  "channels": {
    "discord": {
      "publicChannels": ["general", "announcements"],
      "readonly": true
    },
    "telegram": {
      "dmAccess": ["your_chat_id"],
      "requireAuth": true
    }
  }
}
```

> **Rule:** Only authenticated DMs should trigger actions.

4. Audit Your Permissions

BASH

```
# List all configured channels
openclaw config channels

# Check file access
cat ~/.openclaw/config.json | grep -i "path\|dir\|folder"

# Review API integrations
openclaw config apis
```

The Principle of Least Privilege

Only grant access that's actually needed:

- Don't give Gmail access if you only need Calendar
- Don't give full filesystem if you only need one folder
- Don't give admin Slack if you only need to post messages

5. Rotate Keys Regularly

Recommended rotation schedule:

Key Type	Rotation Frequency
Anthropic API key	Every 90 days
Telegram bot token	Every 6 months
Slack bot token	Every 6 months
Discord bot token	Every 6 months
Database passwords	Every 90 days

Quick Rotation Script

BASH

```
#!/bin/bash
# Save as rotate-keys.sh

echo "Rotating Anthropic key..."
# 1. Generate new key at console.anthropic.com
# 2. Update environment
read -p "Enter new Anthropic key: " NEW_KEY
```

```
echo 'export ANTHROPIC_API_KEY="'$NEW_KEY'"' >> ~/.zshrc

echo "Key rotated. Restart OpenClaw to apply."
```

6. Secure Your Memory Files

What's in your memory files?

BASH

```
# Check for sensitive data
grep -r "password\|secret\|key\|token" ~/.openclaw/
```

If found, MOVE IT:

BASH

```
# Create secrets directory
mkdir -p ~/.openclaw/secrets

# Move sensitive data
mv sensitive-file.md ~/.openclaw/secrets/

# Reference it instead
echo "See ~/.openclaw/secrets/ for credentials" >> ~/.openclaw/memory.md
```

7. Use Webhook Authentication

For external integrations:

JSON

```
{
  "webhooks": {
    "enabled": true,
    "secretToken": "${WEBHOOK_SECRET}",
    "verifySignature": true
  }
}
```

Never expose webhooks without authentication.

8. Monitor Agent Activity

JSON

```
{
  "logging": {
    "level": "info",
    "file": "~/.openclaw/logs/activity.log",
    "rotateDaily": true
  }
}
```

Review weekly:

BASH

```
# Check for suspicious activity
tail -100 ~/.openclaw/logs/activity.log | grep -i "delete\|remove\|api\|key"

# Check file modifications
grep -i "write\|edit\|create" ~/.openclaw/logs/activity.log
```

9. Sandbox File Operations

> □ **Dangerous:**

JSON

```
{
  "fileAccess": {
    "allowed": ["/"]
  }
}
```

> □ **Safe:**

JSON

```
{
  "fileAccess": {
    "allowed": ["/home/user/projects"],
    "denied": ["/home/user/.ssh", "/home/user/.config"]
  }
}
```

Never allow access to:

- `~/.gnupg/` (GPG keys)
- `~/.config/` (application configs)
- `~/.aws/` (AWS credentials)
- Environment files (`.env`)

10. Have an Emergency Kill Switch

Setup:

```JSON
{
  "emergencyStop": {
    "enabled": true,
    "command": "pkill -f openclaw",
    "telegramTrigger": "/emergency_stop",
    "allowedUsers": [123456789]
  }
}
```

Telegram command: `/emergency_stop`

This immediately kills all OpenClaw processes.

Security Audit Checklist

Run this monthly:

- ☐ No API keys in chat logs or memory files
- ☐ All secrets in environment variables
- ☐ File access restricted to project directories
- ☐ Action channels require authentication
- ☐ Keys rotated in last 90 days
- ☐ Logging enabled and reviewed
- ☐ Webhooks use authentication
- ☐ Emergency stop configured
- ☐ Sensitive directories denied
- ☐ Unused integrations removed

Common Security Mistakes

Mistake 1: Public GitHub Repos

Never commit:

- `config.json` with real keys
- `.env` files
- Memory files with personal info
- Log files

> **Solution:** Use `.gitignore`

GITIGNORE

```
# .gitignore
.env
config.json
*.log
secrets/
memory/
```

Mistake 2: Sharing Screenshots

Before sharing:

- Blur API keys
- Blur tokens
- Blur file paths with usernames
- Blur email addresses

Mistake 3: Public Discord/Slack Bots

> ⚠ **Risk:** Anyone can message your bot.

> **Solution:**

JSON

```
{
  "discord": {
    "allowedGuilds": [123456789],
    "allowedChannels": [987654321],
    "dmRequiresAuth": true
  }
}
```

Mistake 4: No Rate Limiting

⚠ **Risk:** Someone spams your bot, draining API credits.

□ **Solution:**

JSON

```
{
  "rateLimit": {
    "maxRequestsPerMinute": 10,
    "maxTokensPerDay": 100000
  }
}
```

Incident Response

If you suspect a breach:

1 Kill the agent immediately

```bash
pkill -f openclaw
```

2 Rotate all keys

- Anthropic: console.anthropic.com
- Telegram: @BotFather → revoke token
- Slack: api.slack.com → rotate tokens
- Discord: discord.com/developers → regenerate

3 Check logs for damage

```bash
cat ~/.openclaw/logs/activity.log | grep -i "delete\|send\|api"
```

4 Audit file changes

```bash
git diff
```

5 Report if needed

- If financial data accessed: contact bank
- If API keys used: check provider dashboards

Security Resources

- **OpenClaw Security Docs:** docs.openclaw.ai/security
- **Report Vulnerabilities:** security@openclaw.ai
- **Community Support:** discord.gg/clawd

Next Steps

Want the complete system?

 The Autonomous Agent Playbook ($49)

- ▸ Full security configurations
- ▸ Production deployment checklist
- ▸ Monitoring and alerting setup
- ▸ Incident response playbooks

Get all guides at: openclaw-guide.com

Built with ❤ by the OpenClaw community

Stay safe out there.