THE COMPLETE

SECURITY HARDENING GUIDE

Open WebUI on Mac Mini (macOS)

Apple Silicon | Docker & Direct Install | Beginner-Friendly

Your Mac Mini is a powerful, energy-efficient AI server. This guide walks you through every security measure to protect it, whether you run Docker, a direct installation, or local models like Ollama. Each step explains what you are doing and why.

Vertical Systems | verticalsystems.io | March 2026

Always verify commands are current before running on a production server.

Before You Begin

Open Terminal (Cmd+Space, type Terminal). Run these commands to figure out your setup:

```
docker ps                                # Works? Docker setup
which openclaw && openclaw --version     # Works? Direct Install
which ollama && ollama --version         # Works? Local model
```

Cloud-only users (ChatGPT API, Claude API) can skip local model steps. If you run Ollama, Cortex, or LM Studio, do everything.

The Threat Landscape

Open WebUI has 31+ known vulnerabilities including two unpatched zero-day RCE flaws. 17,000+ instances are exposed and under active attack. On a Mac Mini the attack surface is naturally smaller than a VPS, but AI-specific threats apply equally.

Vulnerability	Severity	What It Does	Status
CVE-2026-0765	Critical (8.8)	Command injection via crafted skill	UNPATCHED
CVE-2026-0766	Critical (8.8)	Code injection via tool-loading	UNPATCHED
CVE-2025-9074	Critical (9.3)	Docker container escape to host Mac	Patched v4.44.3
CVE-2026-25253	Critical (8.8)	One-click RCE via crafted link	Patched v2026.1.29

1 Lock Your Local AI Model to Localhost

This step applies ONLY if you run Ollama, Cortex, LM Studio, or any local model. Skip if you only use cloud APIs. Local model servers have zero authentication: no username, no password, no API key.

Check What Your Model Is Listening On

```
lsof -iTCP -sTCP:LISTEN -n -P | grep -E '11434|1234|39281'

# You MUST see 127.0.0.1
# If you see * or 0.0.0.0, your model is exposed to your entire network
```

WHY THIS MATTERS 127.0.0.1 means only your Mac can connect. 0.0.0.0 means anyone on your Wi-Fi can access your models, read your data, inject prompts, and burn your GPU. There is no login to stop them.

Fix Ollama

```
launchctl setenv OLLAMA_HOST "127.0.0.1"
launchctl setenv OLLAMA_ORIGINS "http://localhost:3000"
# Quit and reopen Ollama from the menu bar
```

Verify the Fix

```
# This should succeed:
curl -s http://localhost:11434

# This should FAIL:
curl -s --connect-timeout 2 http://$(ipconfig getifaddr en0):11434
```

TIP If the second command succeeds, the fix did not take effect. Restart Ollama and try again.

2 Install Tailscale for Zero-Trust Remote Access

If you need to access your Mac Mini from outside your home or office, Tailscale creates an encrypted private tunnel using WireGuard, with zero open ports.

Install and Connect

```
brew install --cask tailscale
sudo tailscale up --ssh

# Your Mac gets a private IP like 100.x.y.z
# Access from any device on your tailnet
```

WHY THIS MATTERS Traditional remote access requires opening ports to the public internet, inviting brute-force attacks. Tailscale works in reverse: your Mac makes an outbound connection, so zero inbound ports need to be open. All traffic is WireGuard-encrypted end-to-end.

Critical: serve vs funnel

- tailscale serve: Private, tailnet-only. SAFE for AI applications.
- tailscale funnel: Exposes to the PUBLIC internet. NEVER use for Open WebUI.

WARNING Funnel traffic is anonymous with no identity headers. Serve provides authentication. Always use serve for AI applications.

Once Tailscale is running, close every public port. Access your Mac only via the 100.x.y.z address. Free for personal use.

3 Enable the Three-Layer macOS Firewall

macOS has three distinct firewall mechanisms. Most people only know about one. Using all three creates overlapping protection.

Layer 1: Application Firewall (Built-In)

```
sudo /usr/libexec/ApplicationFirewall/socketfilterfw --setglobalstate on
sudo /usr/libexec/ApplicationFirewall/socketfilterfw --setstealthmode on
sudo /usr/libexec/ApplicationFirewall/socketfilterfw --setblockall on
sudo /usr/libexec/ApplicationFirewall/socketfilterfw --setloggingmode on
```

WHY THIS MATTERS setglobalstate on activates the firewall. setstealthmode on makes your Mac invisible to port scanners. setblockall on blocks all unsolicited incoming connections. This does NOT break Tailscale, which uses outbound connections.

Layer 2: PF Packet Filter (Network-Level Rules)

Create /etc/pf.anchors/com.local.server:

```
ext_if = "en0"
block in all
pass out all flags S/SA keep state
pass quick on lo0 all
pass quick on utun3 all                    # Tailscale
pass in on $ext_if proto tcp to any port 22 flags S/SA keep state \
  (max-src-conn 5, max-src-conn-rate 3/30, overload <bruteforce> flush global)
block in quick from <bruteforce>
sudo pfctl -Ef /etc/pf.conf
```

WHY THIS MATTERS PF provides IP-level filtering the Application Firewall cannot do. The bruteforce table auto-bans IPs that make too many SSH attempts.

Layer 3: LuLu (Outbound Traffic Monitor)

```
brew install --cask lulu
# Or download from objective-see.org/products/lulu.html
```

LuLu alerts you whenever a new process tries to connect to the internet. Allow Docker Desktop and Tailscale. Question everything else.

WARNING Do not run LuLu and Little Snitch at the same time. They conflict. Use one outbound monitor.

4 Set the Critical Security Variables

Three environment variables block the three most common attacks: predictable login tokens, open registration, and the unpatched command injection zero-day.

Generate Your Secret Key

```
openssl rand -hex 32    # Copy the output
```

If You Are on Docker

Add to your .env file:

```
WEBUI_SECRET_KEY=paste_your_key_here
ENABLE_SIGNUP=False
ENABLE_PIP_INSTALL_FRONTMATTER_REQUIREMENTS=False

docker compose down && docker compose up -d
```

If You Are on Direct Install

Add to ~/.zshrc (modern macOS default shell):

```
export WEBUI_SECRET_KEY=paste_your_key_here
export ENABLE_SIGNUP=False
export ENABLE_PIP_INSTALL_FRONTMATTER_REQUIREMENTS=False

source ~/.zshrc
```

WHY THIS MATTERS WEBUI_SECRET_KEY signs every login token cryptographically. ENABLE_SIGNUP blocks strangers from creating accounts. The pip install variable blocks CVE-2026-0765, an unpatched flaw that lets attackers run commands through crafted skill files.

Also lock down the RBAC permission:

- Admin Panel > Workspace > Permissions > Remove workspace.tools from non-admins

WARNING workspace.tools allows users to push arbitrary Python code that executes on your Mac. Only trusted administrators should have this permission.

All Security Variables

Variable	Value	What It Does
WEBUI_SECRET_KEY	(openssl rand -hex 32)	Signs login tokens
JWT_EXPIRES_IN	7d	Token lifetime. Never -1.
ENABLE_SIGNUP	False	Blocks account creation
DEFAULT_USER_ROLE	pending	Require admin approval
ENABLE_PIP_INSTALL_FRONTMATTER_REQUIREMENTS	False	Blocks CVE-2026-0765
CORS_ALLOW_ORIGIN	https://yourdomain.com	Cross-origin restriction

WEBUI_SESSION_COOKIE_SECURE	True	HTTPS-only cookies
AUDIT_LOG_LEVEL	REQUEST	Logs for investigation
ENV	prod	Hides API docs

5 Enable FileVault & Verify macOS Security

FileVault (Full-Disk Encryption)

If someone steals your Mac Mini, FileVault prevents them from reading your data, AI models, API keys, or conversations without your password.

```
fdesetup status          # Check if enabled
sudo fdesetup enable     # Enable if not
# Save the recovery key in a password manager, NOT on the Mac
```

WHY THIS MATTERS Apple Silicon uses hardware Secure Enclave encryption with virtually zero performance impact. Your data is always encrypted at the chip level. FileVault ensures a password is required at boot to decrypt it.

System Integrity Protection (SIP)

```
csrutil status    # Must say: enabled
```

SIP prevents even root-level users from modifying critical macOS system files.

Gatekeeper

```
spctl --status    # Must say: assessments enabled
```

Gatekeeper blocks unverified applications from running.

XProtect

```
system_profiler SPInstallHistoryDataType | grep -A 5 XProtect
```

Apple's built-in malware protection. Verify it has received recent updates.

6 Update Docker Desktop (Container Escape Patch)

Skip if you do not use Docker. CVE-2025-9074 (CVSS 9.3) exposed the Docker Engine API inside containers without authentication. A malicious container could escape and take over your entire Mac.

Update via App or Terminal

```
# Via Terminal:
brew upgrade --cask docker
docker --version    # Must be 4.44.3 or later

# Or: Docker Desktop > Settings > General > Check for Updates
```

Docker Desktop Security Settings

- Use Apple Virtualization Framework (Settings > General)
- Use VirtioFS for file sharing (fastest, most secure)
- Set resource limits for CPU, memory, and disk
- Never mount /var/run/docker.sock into application containers

7 Manage API Keys with macOS Keychain

Your Mac has a built-in encrypted vault called Keychain, far more secure than .env files or config files.

Store Secrets

```
security add-generic-password -a "$USER" -s "openwebui_secret" -w "your-key"
security add-generic-password -a "$USER" -s "anthropic_api_key" -w "sk-ant-xxx"
```

Retrieve Secrets

```
security find-generic-password -a "$USER" -s "openwebui_secret" -w
```

Use in Launch Scripts

```
# In ~/.zshrc or startup script:
export WEBUI_SECRET_KEY="$(security find-generic-password -a $USER -s openwebui_secret
-w)"
export ANTHROPIC_API_KEY="$(security find-generic-password -a $USER -s anthropic_api_key
-w)"
docker compose up -d
```

WHY THIS MATTERS Keychain encrypts secrets using your login password and the Secure Enclave chip. Even if someone copies your drive, they cannot read the keys. Compare this to a plain .env file that anyone with file access can read.

Alternative: envchain

```
brew install envchain
envchain --set openwebui WEBUI_SECRET_KEY ANTHROPIC_API_KEY
envchain openwebui docker compose up -d
```

8 Enable Automatic Updates

```
sudo defaults write /Library/Preferences/com.apple.SoftwareUpdate AutomaticCheckEnabled
-bool true
sudo defaults write /Library/Preferences/com.apple.SoftwareUpdate AutomaticDownload
-bool true
sudo defaults write /Library/Preferences/com.apple.SoftwareUpdate CriticalUpdateInstall
-bool true
sudo defaults write /Library/Preferences/com.apple.SoftwareUpdate
AutomaticallyInstallMacOSUpdates -bool true
```

Keep Open WebUI Updated

Docker:

```
docker pull ghcr.io/open-webui/open-webui:latest
docker compose down && docker compose up -d
```

Direct Install:

```
npm update -g @openclaw/cli
openclaw --version   # Must be 2026.1.29 or later
```

WARNING 9+ CVEs since January 2026. OpenClaw v2026.1.29 patches the CVSS 8.8 one-click RCE. Update weekly minimum.

9 AI-Specific Threats on macOS

Malicious Skills (800+ Found in Marketplace)

20% of ClawHub marketplace skills were malicious, deploying the AMOS stealer malware family that captures API keys, browser credentials, crypto wallets, and SSH keys.

- Never install marketplace skills without reviewing source code
- Use LuLu (Step 3) to catch skills that try to phone home
- Audit every skill for os.environ access, external URLs, or file operations

Prompt Injection (OWASP LLM #1 Risk)

Crafted inputs trick the AI into executing commands. Successful injection chains into tool execution for arbitrary code execution on your Mac.

- Restrict workspace.tools to trusted admins only
- Never let untrusted users upload RAG documents

Model Poisoning

Download safety: SafeTensors (most secure) > GGUF (audit templates) > ONNX > Pickle (avoid entirely). For GGUF files, inspect the Jinja2 chat template for suspicious code.

API Key Exfiltration

Malicious skills can read environment variables and send keys to attackers, even via DNS queries.

- Use Keychain (Step 7) instead of plain environment variables
- Use LuLu to catch unexpected outbound connections
- Rotate API keys every 90 days

Security Checklist

- ✓ Local model bound to 127.0.0.1 (or cloud-only, N/A)
- ✓ Tailscale installed for secure remote access
- ✓ Application Firewall on with stealth mode
- ✓ PF packet filter rules loaded
- ✓ LuLu installed for outbound monitoring
- ✓ WEBUI_SECRET_KEY set to 64-character random string
- ✓ ENABLE_SIGNUP=False
- ✓ ENABLE_PIP_INSTALL_FRONTMATTER_REQUIREMENTS=False
- ✓ workspace.tools restricted to admins only
- ✓ FileVault enabled, recovery key stored safely
- ✓ SIP enabled, Gatekeeper enabled, XProtect updated
- ✓ Docker Desktop updated to 4.44.3+ (if using Docker)
- ✓ API keys stored in Keychain, not in files
- ✓ Automatic macOS updates enabled
- ✓ Open WebUI updated to latest version

Vertical Systems | verticalsystems.io | March 2026

This is NOT 100% security. For pen-testing and compliance, consult a cybersecurity professional.