



# Security & Architecture Guide

How Cachard is built, what it can and can't do on your caching Macs, what data it keeps, and how to run it safely. Written for district and enterprise security reviews.

# Summary

Cachard is a web console that monitors and manages Apple content caching servers (Macs running Content Caching) across an organization's sites. It is self-hosted software: it runs on a server you control and connects to your caching Macs over SSH.

|                           |                                                                                                                                                                                                      |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hosting                   | On your premises: one small Linux VM (Ubuntu 22.04/24.04, Debian 12), a Mac (macOS 12 or newer) or a Docker container. No vendor cloud, no tenant shared with other customers.                       |
| Where data lives          | Files in <code>/var/lib/cachard</code> on that server, readable only by the unprivileged <code>cachard</code> service account (keys and secrets 0600). No database server.                           |
| Data leaving your network | None by default except update checks (version, channel and, if licensed, the license ID), which can be turned off. Alerts, AI insights, single sign-on and the map only connect out if you use them. |
| Access to the Macs        | SSH with Cachard's own Ed25519 key. No agent. Root limited by a published sudoers file to the content-caching and software-update tools; read-only in monitoring-only mode.                          |
| Arbitrary commands        | The console can't run commands of its own choosing while the remote terminal is off (the default), and the terminal can be locked off at the server. The SSH key itself is covered in section 3.     |
| Sign-in                   | Local accounts (PBKDF2), TOTP two-step, Microsoft Entra ID or Google single sign-on (OIDC + PKCE). Four roles.                                                                                       |
| Audit                     | Every change, action, security setting, override and terminal session is logged with who, when, from where, exact commands and results.                                                              |
| Updates                   | Ed25519-signed releases, verified twice, installed by a root-owned updater with automatic backup and rollback. Offline update files for air-gapped servers.                                          |
| Data about people         | Device IP address, type and OS build from the Macs' caching logs (kept 30 days). No user names, Apple Accounts or content.                                                                           |

## Contents

|   |                                              |    |                                                  |
|---|----------------------------------------------|----|--------------------------------------------------|
| 1 | Architecture and requirements                | 6  | Data collected, stored and retained              |
| 2 | Network and ports                            | 7  | Telemetry and outbound connections               |
| 3 | Privileges on the caching Macs               | 8  | Secrets, keys and backups                        |
| 4 | Monitoring-only mode and the remote terminal | 9  | Updates and the software supply chain            |
| 5 | Accounts, roles and sessions                 | 10 | Safe operations: the site safeguard and auditing |

11 Disaster recovery

---

12 Vulnerability disclosure and support

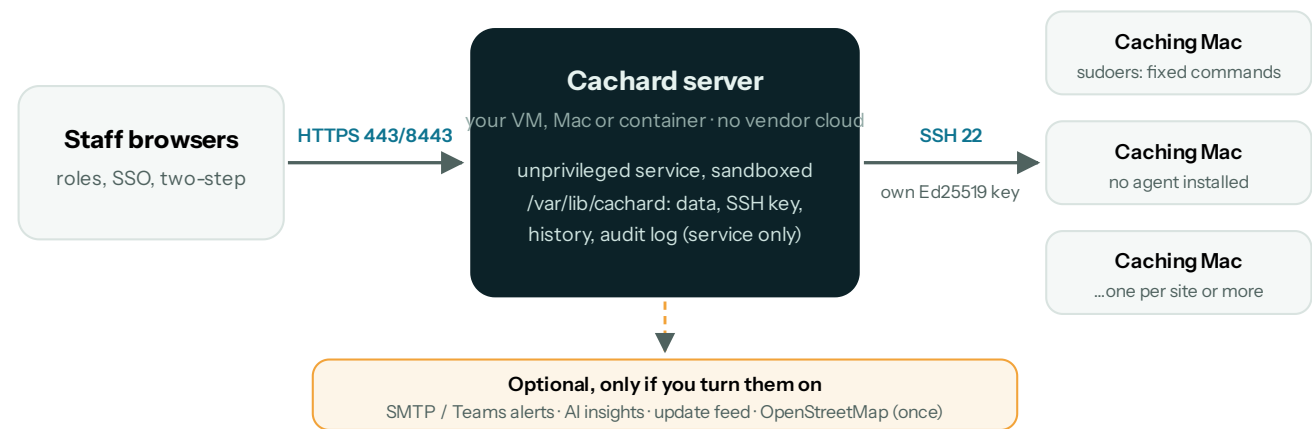
---

13 Deployment checklist

---

# 1 Architecture and requirements

Three parts: staff browsers, the Cachard server, and the caching Macs. Cachard is a single Python service. It polls each Mac about every 15 seconds over a persistent SSH connection and serves the console over HTTPS.



## Components

|                  |                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cachard service  | Python application (FastAPI, uvicorn, asyncssh, cryptography) in <code>/opt/cachard</code> , run by systemd as the unprivileged <code>cachard</code> account with sandboxing: <code>ProtectSystem=strict</code> , <code>ProtectHome</code> , <code>PrivateTmp</code> , <code>NoNewPrivileges</code> , kernel and control-group protection, write access only to its data folder.           |
| Data folder      | <code>/var/lib/cachard</code> : Mac list, SSH key, pinned host keys, history, device records, accounts, audit log, certificates, settings. JSON and text files.                                                                                                                                                                                                                            |
| Service settings | <code>/etc/cachard/cachard.env</code> (root-owned): port, bind address, operating mode, terminal lock.                                                                                                                                                                                                                                                                                     |
| Updater          | A root-owned helper (a systemd path unit, or a launchd job on macOS) that installs verified releases on request; the service itself can't modify its program files.                                                                                                                                                                                                                        |
| On macOS         | The same application runs under launchd as a hidden <code>_cachard</code> account that can't log in and has no administrator rights. Program files and Cachard's Python (downloaded once, SHA-256 pinned) are owned by root; data and settings are readable only by root and <code>_cachard</code> . macOS has no equivalent of the systemd sandbox, so file permissions are the boundary. |
| Admin command    | <code>/usr/local/bin/cachard</code> : status, logs, backup and restore, updates, mode, terminal lock, support bundles.                                                                                                                                                                                                                                                                     |
| Caching Macs     | Unchanged apart from Cachard's public key in one account's <code>authorized_keys</code> and the sudoers file <code>/etc/sudoers.d/zz-cachard</code> . Optionally: power settings to stay awake and restart after a power cut.                                                                                                                                                              |

## System requirements

- Server: Ubuntu 22.04 or 24.04, or Debian 12; 1 vCPU, 1 GB RAM, 2 GB free disk. Or a Mac with macOS 12 or newer (runs as a launchd service under a hidden account). Or Docker (image and compose file included).
- Caching Macs: any Mac running Apple's Content Caching and `AssetCacheManagerUtil`, Apple silicon or Intel, with Remote Login (SSH) on.
- Browsers: current Chrome, Edge, Firefox or Safari.

## Installation

One command, then a browser wizard: `sudo ./install.sh` (or `docker compose up -d`). The wizard is protected by a one-time setup code printed on the server, so nobody else on the network can claim a new installation. Typical time from a fresh VM to monitored caches: about ten minutes.

## 2 Network and ports

Cachard needs only to reach the Macs on SSH and to be reachable by staff on HTTPS. Everything else is optional.

| FROM                    | TO                                                                         | PORT                                               | WHY                                                               | NEEDED       |
|-------------------------|----------------------------------------------------------------------------|----------------------------------------------------|-------------------------------------------------------------------|--------------|
| Staff browsers          | Cachard server                                                             | TCP 8080 (HTTP) or your HTTPS port (e.g. 443/8443) | The console                                                       | Yes          |
| Cachard server          | Each caching Mac                                                           | TCP 22                                             | Status, settings, logs, actions (SSH)                             | Yes          |
| Cachard server          | Your SMTP relay                                                            | 25 / 465 / 587                                     | Email alerts and reports                                          | Optional     |
| Cachard server          | Microsoft Teams webhook                                                    | 443                                                | Teams alerts                                                      | Optional     |
| Cachard server          | <code>api.anthropic.com</code>                                             | 443                                                | AI insights (off by default)                                      | Optional     |
| Cachard server          | <code>updates.cachard.app</code>                                           | 443                                                | Update checks and downloads                                       | Optional     |
| Cachard server          | <code>overpass-api.de</code> ,<br><code>nominatim.openstreetmap.org</code> | 443                                                | Build the street map, once; look up site addresses you search for | Optional     |
| Staff browsers / server | Microsoft or Google sign-in                                                | 443                                                | Single sign-on                                                    | Optional     |
| Cachard server          | PyPI                                                                       | 443                                                | Python packages at install time (or use the offline bundle)       | Install only |

Nothing on the internet needs to reach Cachard. The site checks ask each Mac to run Apple's `networkQuality` test, so the Macs (not Cachard) contact Apple's test servers. An optional read-only network map for signed-out viewers (for example a NOC screen) is **off by default** ( `ACC_PUBLIC_MAP=1` turns it on). When on, it shows Mac and site names, locations, status, traffic rates, cache fill, content categories and anonymous device icons, without IP addresses, settings or errors.

Air-gapped networks: install from the offline bundle (Python wheels included), apply offline update files, and leave update checks off. Cachard works fully without internet access.

### 3 Privileges on the caching Macs

This is the part a reviewer should read most closely. Cachard holds an SSH key that can sign in to each caching Mac. What it can do there as root is limited by a sudoers file that the Mac enforces.

#### How a Mac is prepared

1. An administrator enters a Mac admin username and password once in the setup wizard. The password is held in memory for this step only; it is never written to disk or logs.
2. Cachard connects, records the Mac's SSH host key and shows its fingerprint (trust on first use; it's verified on every later connection).
3. It adds Cachard's public key to that account's `~/.ssh/authorized_keys`.
4. It installs `/etc/sudoers.d/zz-cachard`, validated with `visudo` before it is put in place.
5. It reconnects with the key alone and confirms `sudo -n AssetCacheManagerUtil -j status` works. From then on, the password isn't needed.

Macs can also be prepared by hand with `scripts/prepare-mac.sh`, for example when password sign-in over SSH is disabled.

#### The sudoers file, full mode (published in full)

```
# Installed by Cachard. Lets the console's SSH account run only these commands as root.
Cmdnd_Alias CACHARD_UTIL    = /usr/bin/AssetCacheManagerUtil
Cmdnd_Alias CACHARD_PREFS   = /usr/bin/defaults write /Library/Preferences/com.apple.AssetCache.plist *,
/usr/bin/defaults delete /Library/Preferences/com.apple.AssetCache.plist *
Cmdnd_Alias CACHARD_REBOOT  = /sbin/shutdown -r now
Cmdnd_Alias CACHARD_SWU     = /usr/sbin/softwareupdate
Cmdnd_Alias CACHARD_SWPREF  = /usr/bin/defaults write /Library/Preferences/com.apple.SoftwareUpdate *
cacheadmin ALL=(root) NOPASSWD: CACHARD_UTIL, CACHARD_PREFS, CACHARD_REBOOT, CACHARD_SWU, CACHARD_SWPREF
```

## The sudoers file, monitoring-only mode

```
# Installed by Cachard (monitoring-only). Lets the console's SSH account run only these read-only
commands as root.
Cmdnd_Alias CACHARD_READ = /usr/bin/AssetCacheManagerUtil -j status, /usr/bin/AssetCacheManagerUtil -j
settings, /usr/bin/AssetCacheManagerUtil status, /usr/bin/AssetCacheManagerUtil settings,
/usr/bin/AssetCacheManagerUtil canActivate
cachearmin ALL=(root) NOPASSWD: CACHARD_READ
```

`cachearmin` is replaced by the account you choose. Settings → Security in the console shows the exact file for the server's mode.

**Be clear about two things.** Without root, the SSH key can do whatever that macOS account can do. And if that account is a Mac administrator (the setup wizard installs the key into the account you prepare with), it can still use `sudo` with its own password for anything; the sudoers file only decides what runs *without* a password. Cachard never keeps that password, unless you choose to save an administrator password for scheduled macOS updates on Apple silicon (removed automatically in monitoring-only mode). We recommend a dedicated account for Cachard, and restricting its key to the Cachard server's address by adding `from="10.0.0.5"` (your server's address) in front of the key in `authorized_keys`.

## What the console runs

With the remote terminal off, Cachard runs only its own fixed commands: status polls ( `AssetCacheManagerUtil -j status/settings`, `sw_vers`, `sysctl`, `df`, `plutil` on the caching preferences), log reads, the listed actions (activate, deactivate, reload settings, flush caches, restart), setting changes and `softwareupdate`. Setting values from the browser are validated (types, ranges, IP addresses) and shell-quoted on the server; the browser never sends a command.

## 4 Monitoring-only mode and the remote terminal

Two switches let a cautious organization start with the smallest possible footprint and widen it deliberately. The operating mode and the terminal lock are set on the server by root. No account in the web console can change them, so a stolen console password can't either.

### Monitoring-only mode

```
sudo ./install.sh --monitor-only or sudo
cachard mode monitor (Docker:
CACHARD_MODE=monitor). Cachard reads cache
status, statistics and logs but can't change any Mac.
```

### Remote terminal: off by default

A platform administrator can turn it on in Settings → Security. `--no-terminal` or `sudo cachard terminal forbid (CACHARD_TERMINAL=off)` makes that impossible.

## In monitoring-only mode

- ✓ Caching configuration, actions, Mac updates (manual, scheduled and automatic) and the terminal are switched off for every account, including platform administrators. The check runs on every request, and

again whenever a job, a scheduled update or an automatic update starts, so updates scheduled before the switch don't run.

- ✓ Macs prepared in this mode get the read-only sudoers file in section 3, so without a Mac administrator password even a compromised Cachard server couldn't change caching, restart a Mac or install software as root. Switching to monitoring-only also deletes any Mac administrator password saved for scheduled updates.
- ✓ Monitoring, the map, history, alerts, reports, site checks and diagnostics keep working.
- ✓ Switching to full mode later is a deliberate step on the server, followed by preparing the Macs again to install the full sudoers file.

## The remote terminal, when enabled

- Gives platform administrators a live SSH shell to one Mac, and fleet-wide commands, for troubleshooting.
- Runs as Cachard's SSH account with the same sudoers limits; running anything else as root needs that account's password.
- Every session is written to the audit log when it opens and closes, with a transcript of its output (up to 5 MB per session), and closes after 30 minutes without input.
- Turning it on or off is itself audited.

# 5 Accounts, roles and sessions

## Roles

Each role can do everything the one before it can. The server enforces roles on every request and WebSocket; the interface hides what an account can't do.

| ROLE                          | PERMISSIONS                                                                                                                                                                                                                |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Viewer</b>                 | Cache health and statistics, network map, history, alerts and the audit log. Changes nothing.                                                                                                                              |
| <b>Technician</b>             | Also: site checks and diagnostics, AI questions, update scans, restarting or reloading the caching service.                                                                                                                |
| <b>Administrator</b>          | Also: caching configuration, other actions (flush, restart Mac, deactivate), Mac updates and schedules, standards, sites, map and alerts.                                                                                  |
| <b>Platform administrator</b> | Also: adding and removing Macs, accounts and invitations, sign-in and single sign-on, two-step policy, the remote terminal, security settings, Cachard updates, license, branding, HTTPS, AI settings and support bundles. |

The built-in `admin` account is always a platform administrator, for break-glass access. Its password is chosen in the setup wizard and stored as a PBKDF2 hash (reset it with `sudo cachard reset-admin-password`). An alternative break-glass password can be set in the server's settings file (`ACC_ADMIN_PASSWORD`), where it is stored in plain text; leave it unset unless you need it. Accounts from Cachard 1.0/1.1 called "operator" become technicians.

## Authentication

|                  |                                                                                                                                                                        |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Local passwords  | PBKDF2-SHA256, 310,000 iterations, per-password salt; minimum 10 characters. Sign-in is rate limited to 8 failures per address per 5 minutes.                          |
| Two-step sign-in | TOTP (RFC 6238, any authenticator app) with one-time recovery codes; replayed codes are refused. Can be required for administrators or for everyone.                   |
| Single sign-on   | Microsoft Entra ID or Google via OpenID Connect, authorization-code flow with PKCE. Allowed email domains, and a default role (or none) for people without an account. |
| Invitations      | Single-use links stored as hashes, expiring after 72 hours.                                                                                                            |
| First run        | The setup wizard requires a one-time code generated on the server.                                                                                                     |

## Sessions and browser protections

- Signed, HTTP-only session cookies, `SameSite=Lax`, `Secure` when HTTPS is on, 12-hour lifetime. Role changes and deleted accounts take effect on the next request. Sessions are stateless: signing out ends the session in that browser, but a copied cookie stays valid until it expires, so protect staff browsers as usual.
- Cross-site request forgery: browsers always send an `Origin` header on cross-site requests, and any state-changing request or WebSocket with an `Origin` other than the console is refused (scripts and the command line, which send none, are unaffected). Cookies are `SameSite=Lax` as well.
- Headers: Content-Security-Policy on pages ( `default-src 'self'`, `object-src 'none'`, `base-uri 'none'`, `frame-ancestors 'self'`; scripts and styles also allow `'unsafe-inline'` for the console's own inline code), `X-Frame-Options`, `nosniff`, `Referrer-Policy`, `Permissions-Policy`, HSTS with HTTPS, `no-store` on API responses.
- HTTPS with your own certificate (PFX or PEM, wildcards supported), managed in Settings → HTTPS; HTTP then redirects to HTTPS.

## 6 Data collected, stored and retained

Cachard collects only what it needs to show how caching is working. It reads nothing from the iPads, Macs and other devices that use your caches; it sees what the caching Macs themselves log.

### Collected from each caching Mac

|                             |                                                                                                                           |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------|
| Caching status and settings | From <code>AssetCacheManagerUtil</code> : cache size and usage, bytes served and pulled, peers, parents, configuration.   |
| System facts                | macOS version, model, computer name, boot time, free disk space, power settings.                                          |
| Software updates            | Available macOS updates and update history ( <code>softwareupdate --list</code> , <code>--history</code> ).               |
| Content-caching log         | For each request: the client device's IP address, device type, OS build and what was downloaded (e.g. "iOS 18.1 update"). |

**Not collected:** user names, Apple Accounts, serial numbers of client devices, files, app data, browsing, location or anything else about the people using the devices.

## Stored and retained on your server

| DATA                                               | KEPT                     |
|----------------------------------------------------|--------------------------|
| Device records (IP, type, OS build, downloads)     | 30 days                  |
| Per-minute traffic samples (for replaying the map) | 35 days                  |
| Daily totals per Mac and site                      | 400 days                 |
| Audit log, alert history, terminal transcripts     | Until you remove them    |
| Accounts, settings, Mac list, certificates         | Until changed or removed |

All of it is visible only to signed-in accounts (see the optional public map in section 2). Uninstalling with `--purge` deletes Cachard's data, settings and automatic update backups from the server; backups you saved elsewhere, and the key and sudoers file on the Macs (section 11), are yours to remove.

## Student data

Cachard doesn't receive or store student records. In a school, the device records are network metadata (an IP address and device model) from your own caching servers, kept on your own server for 30 days.

# 7 Telemetry and outbound connections

Your cache telemetry stays on your infrastructure. Cachard sends no usage analytics, and the vendor has no access to your server.

| CONNECTION                    | WHAT IS SENT                                                                                                                                                                                                                                                                                          | DEFAULT               |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| Update check (every 6 hours)  | Installed version and release channel, and the license ID if licensed (nothing for a trial). Like any web request, it reveals your public IP address.                                                                                                                                                 | On; can be turned off |
| Update download               | Also the license key, so the vendor can confirm support is current.                                                                                                                                                                                                                                   | Only when installing  |
| AI insights                   | An aggregated fleet snapshot (organization name, Mac and site names, states, settings, your public IP addresses, traffic totals, findings) sent to Anthropic under your own API key. No client device IP addresses or identities, passwords or keys. Settings → AI insights shows the exact snapshot. | Off                   |
| Email / Teams alerts          | Alert and report text to your own mail relay or Teams channel.                                                                                                                                                                                                                                        | Off                   |
| Street map and address search | A map-area request to OpenStreetMap when you build the map, and the address text when you search for a site's location.                                                                                                                                                                               | Only when used        |
| Single sign-on                | Standard OpenID Connect sign-in with Microsoft or Google.                                                                                                                                                                                                                                             | Off                   |

License keys are verified offline with a public key; there is no license check-in, and Cachard keeps monitoring even when a license lapses (it stops making changes to the Macs).

## 8 Secrets, keys and backups

|                                                         |                                                                                                                                                                                                                         |
|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Cachard's SSH private key</b>                        | Ed25519, generated on the server, permissions 0600 for the service account. Never shown in or downloadable from the console (only the public key is). Not passphrase-protected, because the service uses it unattended. |
| <b>Pinned SSH host keys</b>                             | Stored on first contact and verified on every connection; a changed host key stops the connection.                                                                                                                      |
| <b>Mac admin password (preparation)</b>                 | Memory only, for the duration of the preparation step.                                                                                                                                                                  |
| <b>Mac admin password (scheduled updates, optional)</b> | Apple silicon Macs need one to install macOS updates. Enter it per run (memory only), or save it encrypted for scheduled runs.                                                                                          |
| <b>AI API key</b>                                       | Encrypted with a key derived from the server's secret.                                                                                                                                                                  |
| <b>SMTP password, Teams webhook, SSO client secret</b>  | Files readable only by the service account; never sent back to the browser.                                                                                                                                             |
| <b>Account passwords</b>                                | PBKDF2 hashes (except the optional break-glass <code>ACC_ADMIN_PASSWORD</code> , section 5).                                                                                                                            |

**Encryption at rest:** the encrypted items above are protected by a secret on the same server, which guards against disclosure through copied files or backups of individual settings, not against someone with root on the server. Encrypt the server's disk (most hypervisors and cloud platforms do this per VM), and treat the Cachard server like any other infrastructure management system.

### Backups

`sudo cachard backup` writes the data folder and service settings to one file with permissions 0600. It contains the SSH key and secrets, so store it like a password vault. Every Cachard update also makes its own backup before installing.

### Support bundles

When you ask for help, a support bundle (Settings → Help & support, or `sudo cachard support-bundle`) includes logs, settings and each Mac's last status, with passwords, password hashes, two-step secrets, SSH and TLS private keys, API keys, webhooks and the license key removed. Device records and terminal transcripts are never included. The command-line bundle also includes the last 7 days of the service log, with invitation links, sign-in codes and the setup code removed; IP addresses can be masked ( `--mask-ips` , or the option in the console). Nothing is sent automatically: you send it yourself.

## 9 Updates and the software supply chain

- Releases are signed with an Ed25519 release key that is separate from the license key and kept offline by the vendor. The console and the root updater each verify the feed's signature and the archive's SHA-256 before anything is installed. A tampered feed or mirror is rejected.
- The console runs unprivileged and can't change its own program files. It asks for an install; a root-owned updater re-verifies the release, backs up the data, installs it, checks the new version starts, and rolls back automatically if it doesn't.
- You choose the policy: notify only, install fixes automatically, or install everything automatically, inside a maintenance window. Automatic installs wait while the caching Macs are updating macOS.
- Versions follow major.minor.patch. Patch releases (1.2.x) contain fixes only, including security fixes for Cachard and its dependencies.
- Dependencies: a small set of widely used Python packages (FastAPI, uvicorn, asyncssh, PyYAML, itsdangerous, cryptography) installed from PyPI at install time, or from the wheels bundled with the offline installer. Third-party licenses are listed in THIRD-PARTY-NOTICES.md.
- Every update request, install and rollback is recorded in the audit log.

## 10 Safe operations: the site safeguard and auditing

Cachard is designed so that managing caches never quietly takes a site's caching offline.

### Site safeguard

"Minimum available caches per site" (default 1, Settings → Security):

- A macOS update on a caching Mac waits while its site would be left with fewer available caches than the minimum, then skips that Mac with an explanation if the site doesn't recover within 30 minutes. With two caches at a site, Cachard won't update the second while the first is down or still updating.
- Restarting a Mac or switching caching off asks for confirmation in the same situation; going ahead anyway is recorded in the audit log.
- Rolling updates also go one site at a time per batch, update parent caches last, can start with a pilot Mac, stop on the first failure, wait for each Mac to come back and confirm caching is running again, and pause alerts for Macs being updated.
- Sites with a single cache can't stay covered during a restart; the update plan points them out before you start.

### Audit log

Recorded: every configuration change and action (with the exact commands run on each Mac and the results), Mac updates (with the `softwareupdate` commands), Mac preparation, accounts, roles, invitations, password and two-step changes, single sign-on settings, security settings, safeguard overrides, terminal sessions, Cachard updates, and server-side changes made with `sudo cachard mode`, `terminal`, `reset-admin-password` and `reset-2step`. Each entry has who, when and, for console actions, from which address. Read-only diagnostics and site checks appear in the console's activity list but not in the audit log.

The audit log is visible in the History tab and written to `/var/lib/cachard/audit.jsonl`, one JSON line per event, appended by the service. For tamper evidence, forward it to your log server or SIEM with your usual

tooling.

## 11 Disaster recovery

- **The caches don't depend on Cachard.** Apple's content caching keeps serving devices if the Cachard server is down; only monitoring, alerts and changes pause.
- **Restore:** install Cachard on a new server and run `sudo cachard restore FILE` with your latest backup. Because the SSH key is in the backup, no Mac needs preparing again.
- **Failed update:** rolled back automatically; `sudo cachard update rollback` returns to the previous version by hand.
- **Lost sign-in:** `sudo cachard reset-admin-password` and `sudo cachard reset-2step USER` on the server.
- **Removing Cachard from a Mac:** delete `/etc/sudoers.d/zz-cachard` and Cachard's line in the account's `authorized_keys` (it ends in `cachard-console` for keys Cachard created).

## 12 Vulnerability disclosure and support

- Report a vulnerability to **security@cachard.app** with details and steps to reproduce. We acknowledge within two business days, keep you informed, credit you if you wish, and ask for 90 days to ship a fix before publication. Please test only systems you own.
- Fixes ship as signed patch releases; customers on automatic fix installation receive them without action.
- Support: support@cachard.app. Sales, quotes and purchase orders: sales@cachard.app.

# 13 Deployment checklist

A conservative rollout that most security teams can approve quickly.

| STEP |                                                                                                                                                                              |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Create a small VM on an encrypted datastore; allow inbound HTTPS from staff networks and outbound SSH to the caching Macs only.                                              |
| 2    | Install in monitoring-only mode with the terminal forbidden: <code>sudo ./install.sh --monitor-only --no-terminal</code> .                                                   |
| 3    | Create a dedicated macOS account for Cachard on each caching Mac (or via your MDM) and prepare the Macs in the wizard. Optionally restrict the key with <code>from=</code> . |
| 4    | Install your HTTPS certificate (Settings → HTTPS).                                                                                                                           |
| 5    | Turn on single sign-on and require two-step sign-in for administrators; give most staff the Viewer or Technician role.                                                       |
| 6    | Leave AI insights off unless approved; decide on update checks (on, or offline files).                                                                                       |
| 7    | Schedule <code>sudo cachard backup</code> and store backups securely.                                                                                                        |
| 8    | When you're ready to manage caches, run <code>sudo cachard mode full</code> and prepare the Macs again. Keep the site safeguard at 1 or more.                                |

© 2026 Bartlett Electronics. Cachard is a trademark of Bartlett Electronics. Apple, Mac and macOS are trademarks of Apple Inc.; Cachard is not affiliated with or endorsed by Apple Inc. This guide describes Cachard 1.2; settings and defaults may change in later versions, and the current version is always at [cachard.app/security](https://cachard.app/security).