ARES
Server Guardian

ARES SUITE

ARES Server Guardian

Technical Blueprint

Detects attacks on your servers and blocks them at the firewall.

VERSION	PRODUCTION	DATE
1.6.0	ares-guardian.nacino.net	23 November 2026
AUTHOR	DEMO	CLASSIFICATION
Alfredo Nacino	ares-guardian-demo.nacino.net	Internal

ARES Server Guardian Technical blueprint	DRAWING ARES-SERVER-GUARDIAN-BP	REVISION v1.6.0
	DATE 23 November 2026	SHEETS A-01 to A-11
PRODUCTION https://ares-guardian.nacino.net	DEMO https://ares-guardian-demo.nacino.net	AUTHORED BY Alfredo Nacino

The drawing set for ARES Server Guardian 1.6.0: each sheet is one view of the system with its specification. Environments are named `prod` and `pre`, never by server; server identities live only in the platform inventory and the local host list on the release machine. Defaults are those of `ares/config.py`.

Drawing register

A-01 System context and boundaries

A-02 Production deployment (host install)

A-03 Agent install

A-04 Preproduction (Demo)

A-05 Detection pipeline

A-06 Firewall rendering (nftables)

A-07 ARES Central link

A-08 Data and storage

A-09 Sign-in and access control

A-10 Interfaces

A-11 Operations

A-01 System context and boundaries

ares-server-guardian
rev 1.6.0

> Every interface of the service, its direction and its protection. Only the dashboard accepts inbound connections, and an agent does not have one.

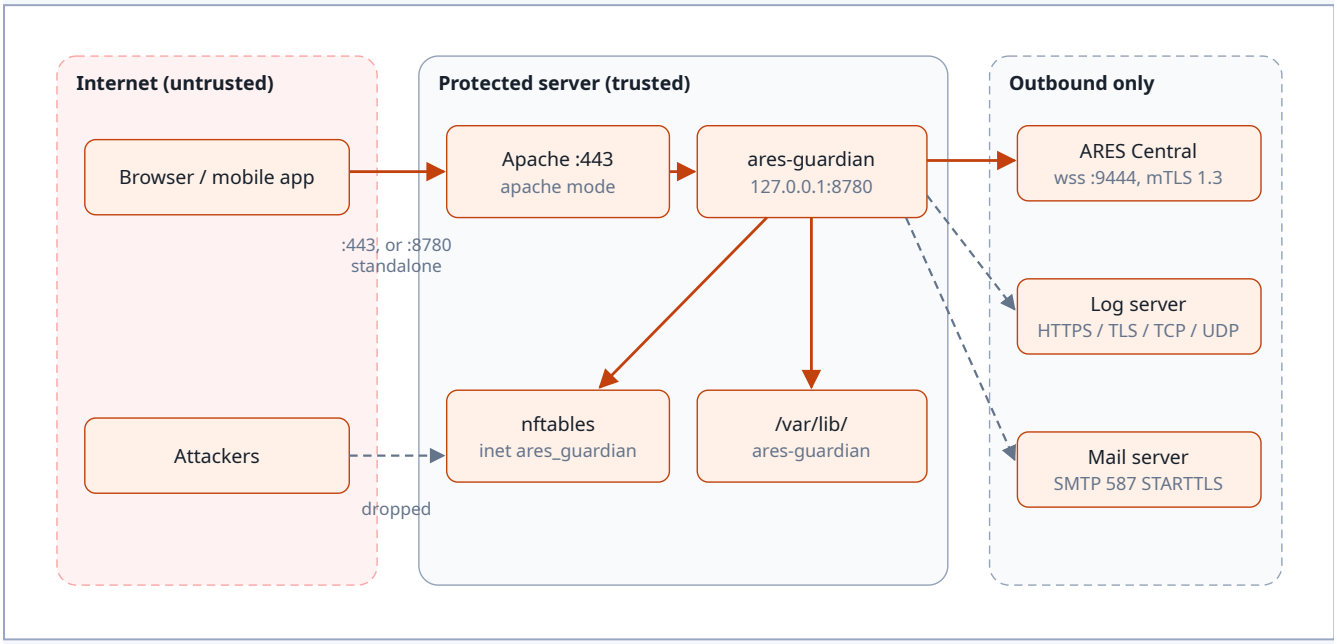


Figure 1. Context and trust boundaries

Interface	Direction	Protocol, port	Authentication
Dashboard and API	in	HTTPS; `:8780` standalone, `127.0.0.1:8780` behind Apache `:443`; `:8443` plain default	password + TOTP, JWT (30 min access, 30 d refresh)
ARES Central	out	WebSocket over TLS 1.3, gateway `:9444`, path `/agent`	mutual TLS: Central's private CA pinned; per-server key and certificate
Central commands	in (over the link)	JSON, Ed25519 signature	signed by Central, `agent_id` must match, valid 5 min, never replayed
Log forwarding	out	HTTPS/HTTP POST JSON array; TLS/TCP JSON lines; UDP one record per datagram	optional `Authorization` header
Webhook	out	HTTPS POST JSON	none (URL is the secret)
Health	in	`GET /healthz` → `200 {"status":"ok"}`	none; reveals nothing

A-02 Production deployment (host install)

ares-server-guardian
rev 1.6.0

> Each protected server runs one systemd service. The install modes differ only in how the dashboard is reached.

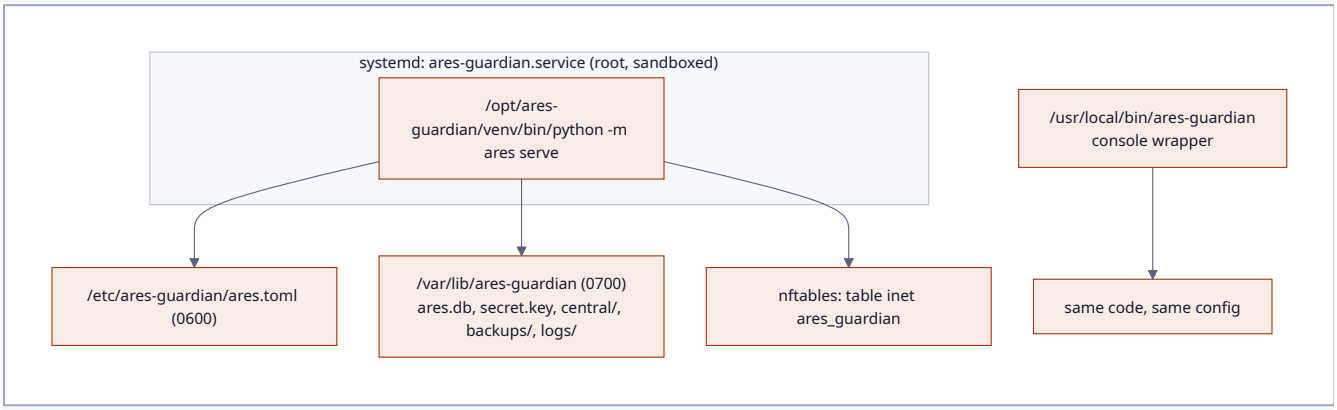


Figure 2. Files and processes on a protected server

Item	Value
Code	`/opt/ares-guardian/ares` (previous release kept as `ares.old`), venv `/opt/ares-guardian/venv`
Configuration	`/etc/ares-guardian/ares.toml`, mode 0600, never overwritten by an upgrade
Data	`/var/lib/ares-guardian`, mode 0700 (`StateDirectory=ares-guardian`)
Service	`ares-guardian.service`, `Restart=on-failure`, `RestartSec=5`, after `network-online.target` and `nftables.service`
Sandbox	`ProtectSystem=strict`, `ReadWritePaths=/var/lib/ares-guardian`, `ProtectHome=read-only`, `PrivateTmp`, `NoNewPrivileges`, `ProtectKernelModules`, `ProtectControlGroups`, `RestrictSUIDSGID`, `LockPersonality`
Capabilities	`CAP_NET_ADMIN`, `CAP_NET_RAW`, `CAP_KILL`, `CAP_SYS_PTRACE`, `CAP_DAC_READ_SEARCH`
Modes	`deploy-apache.sh` (vhost + Let's Encrypt, app on `127.0.0.1:8780`), `deploy-standalone.sh` (HTTPS on `:8780`, self-signed unless a certificate is given), `install-agent.sh` (no web)
Estate (prod)	3 Kubernetes nodes in apache mode, 5 standalone servers; all enforcing with nftables; the nodes on 1.6.0, the standalone servers on 1.5.1

A-03 Agent install

ares-server-guardian
rev 1.6.0

> An agent is the same service with `[server] web = false`: no web app, no users, no port.

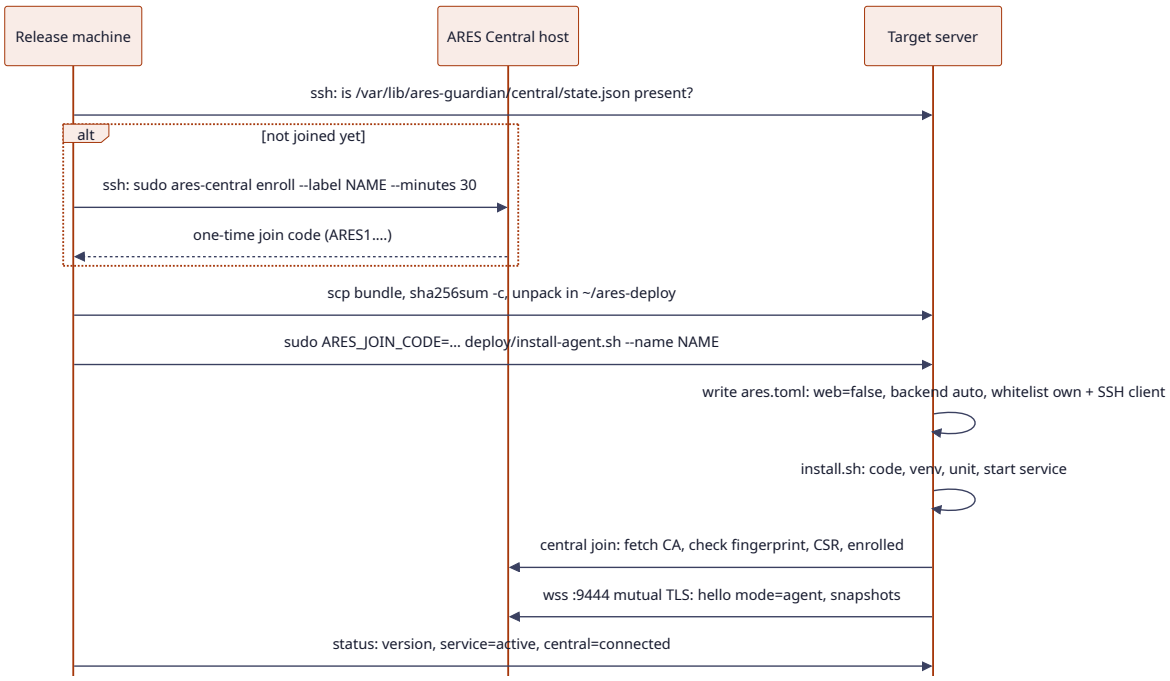


Figure 3. Installing an agent on a chosen server

Setting written by `install-agent.sh`	Value
`[server] web`	`false`
`[firewall] backend`	`auto`: `nftables` if `nft list tables` works, else `iptables`, else `dry-run`; or `--backend`
`[firewall] whitelist`	`127.0.0.0/8`, `::1/128`, every address of `hostname -I`, the SSH client of the installer, each `--whitelist`
`[central] allow_remote_firewall`	`true`, or `false` with `--read-only`
Join code	from `--join` or `ARES_JOIN_CODE`; one use, Central's default lifetime 60 min (30 min from `install-remote.sh`)
Health criterion	`systemctl is-active` and Central link state `connected` within 90 s

A-04 Preproduction (Demo)

ares-server-guardian
rev 1.6.0

> The Demo at https://ares-guardian-demo.nacino.net shows the product web site at `/` and the dashboard at `/app`, with the firewall in dry-run. It runs unprivileged, so it cannot enforce.

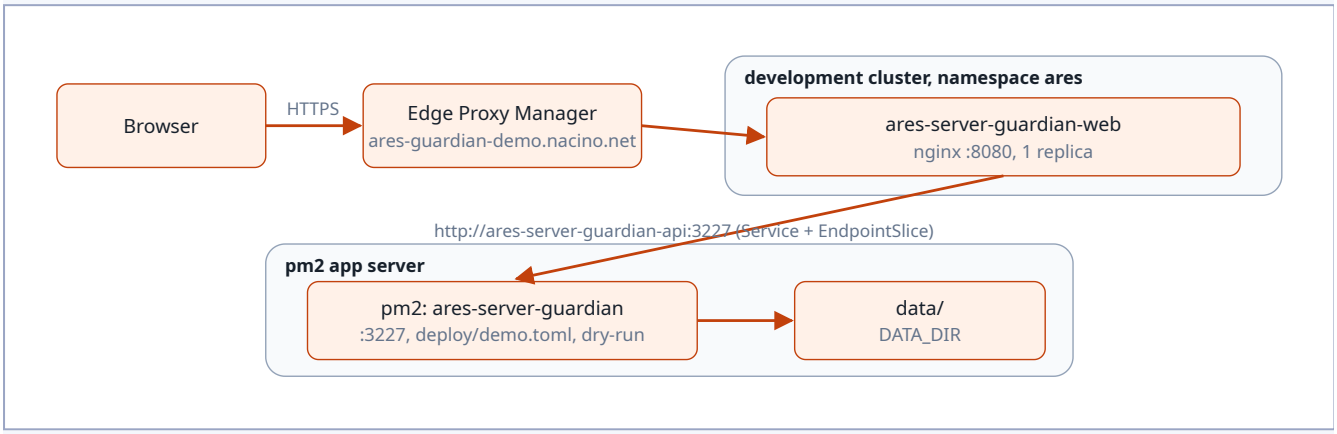


Figure 4. pre environment

Item	Value
Server	pm2 process `ares-server-guardian`, `.venv/bin/python -m ares serve`, port 3227, watch on `ares/`, `requirements.txt`, `deploy/demo.toml`
Configuration	`ARES_CONFIG=deploy/demo.toml`: dry-run, Central off, no system logs, private ranges as trusted proxies
Web tier	image `ares-server-guardian-web:1.6.0` (nginx-unprivileged), 1 replica, WebSocket upgrade forwarded
Product site	`PRODUCT_SITE_DIR=website/public`, built by `deploy.conf` (`website/build.sh`) before every deploy
Deploy	`suite docker push`, then `suite deploy pre` (pm2 via `deploy.sh`, then Kubernetes)
prod web site	https://ares-guardian.nacino.net: image `ares-server-guardian-site:1.6.0` (nginx-unprivileged, port 8080, `/healthz`), 2 replicas, `k8s/site/site.yaml`, Ingress behind each node's Apache

A-05 Detection pipeline

ares-server-guardian
rev 1.6.0

> Every line becomes at most a few events; each event updates sliding-window counters for its detector and address. Reaching the threshold raises one incident, an alert and a ban.

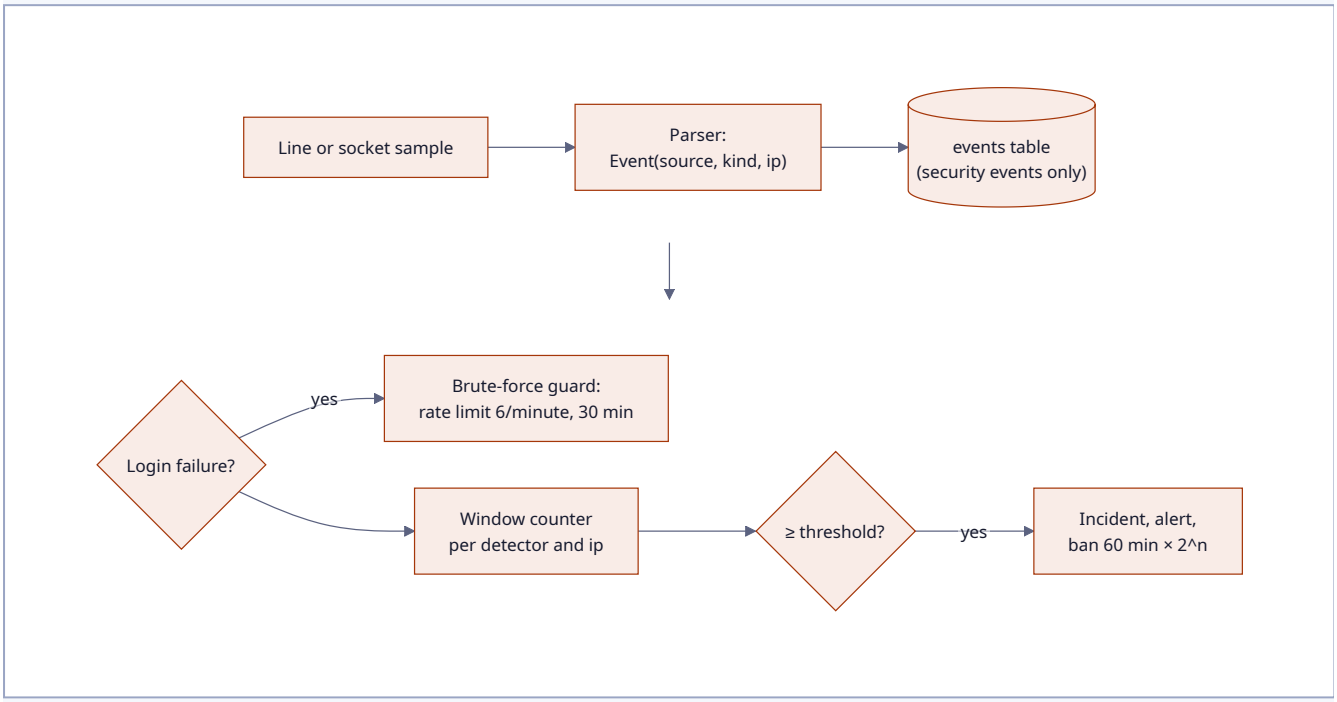


Figure 5. From a line to a rule

Detector	Threshold	Window	Action
`ssh_bruteforce`	5	300 s	ban
`web_login_bruteforce`	10	300 s	ban
`http_flood`	600 requests	60 s	ban + rate limit `25/second` on ports 80, 443 for 60 min
`web_scan`	20	300 s	ban
`port_scan`	15 ports	60 s	ban (needs kernel log lines)
`conn_flood`	150 connections	live	ban + rate limit `50/second` on all TCP for 60 min
`syn_flood`	300 half-open	live	SYN rate limit
`guardian_login`	8	600 s	ban

Repeat offenders: the ban time doubles each time, up to 43 200 min; the 3rd automatic ban is permanent (`permanent_after_bans = 3`).

A-06 Firewall rendering (nftables)

ares-server-guardian
rev 1.6.0

> The desired rules live in the database; the backend is rewritten from them atomically. The nftables backend owns one table and touches nothing else.

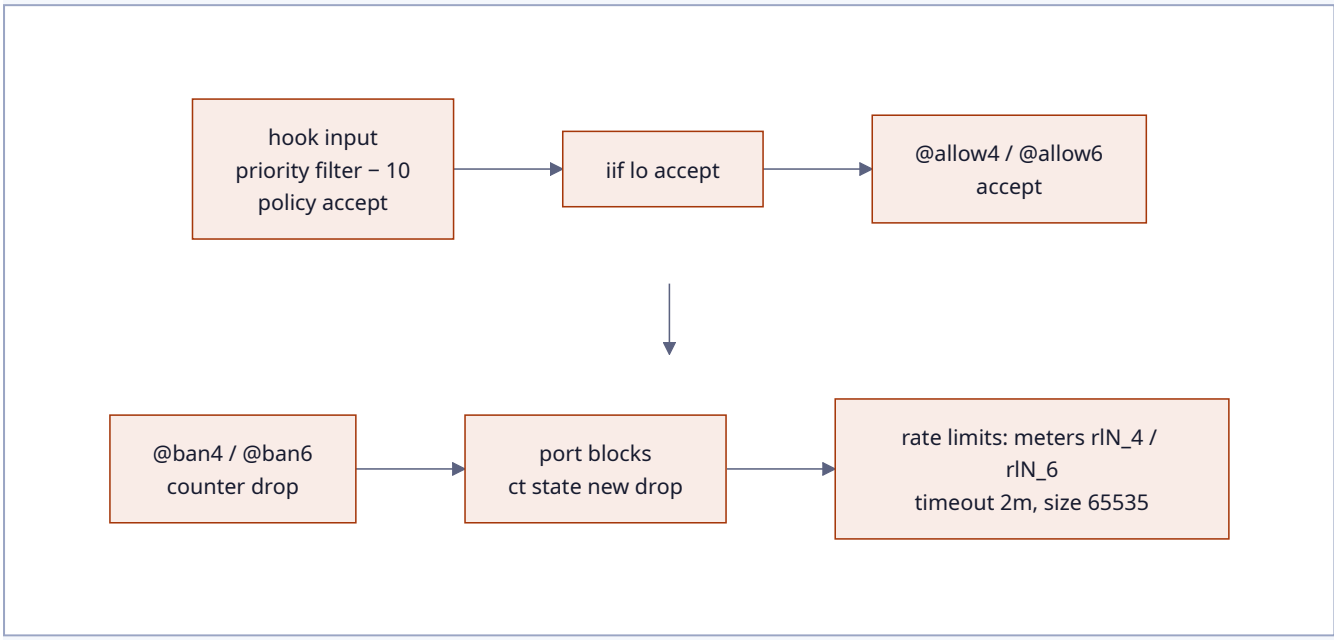


Figure 6. Order of the input chain in table inet ares_guardian

Backend	Objects	Apply
`nftables`	table `inet ares_guardian`; interval sets `ban4`, `ban6`, `allow4`, `allow6`; chain `input`	one `nft -f` transaction (delete + recreate the table)
`iptables`	chains `ARES` (IPv4) and `ARES6` (IPv6), jumped to from `INPUT`	`iptables-restore`
`ufw`	individual rules, diffed against those ARES applied before	`ufw insert 1` / `ufw delete`
`dry-run`	none	commands logged only

Never banned: the whitelist, private, loopback and reserved ranges (`never_ban_private = true`) and ARES Central's address.

A-07 **ARES Central link**

`ares-server-guardian`
`rev 1.6.0`

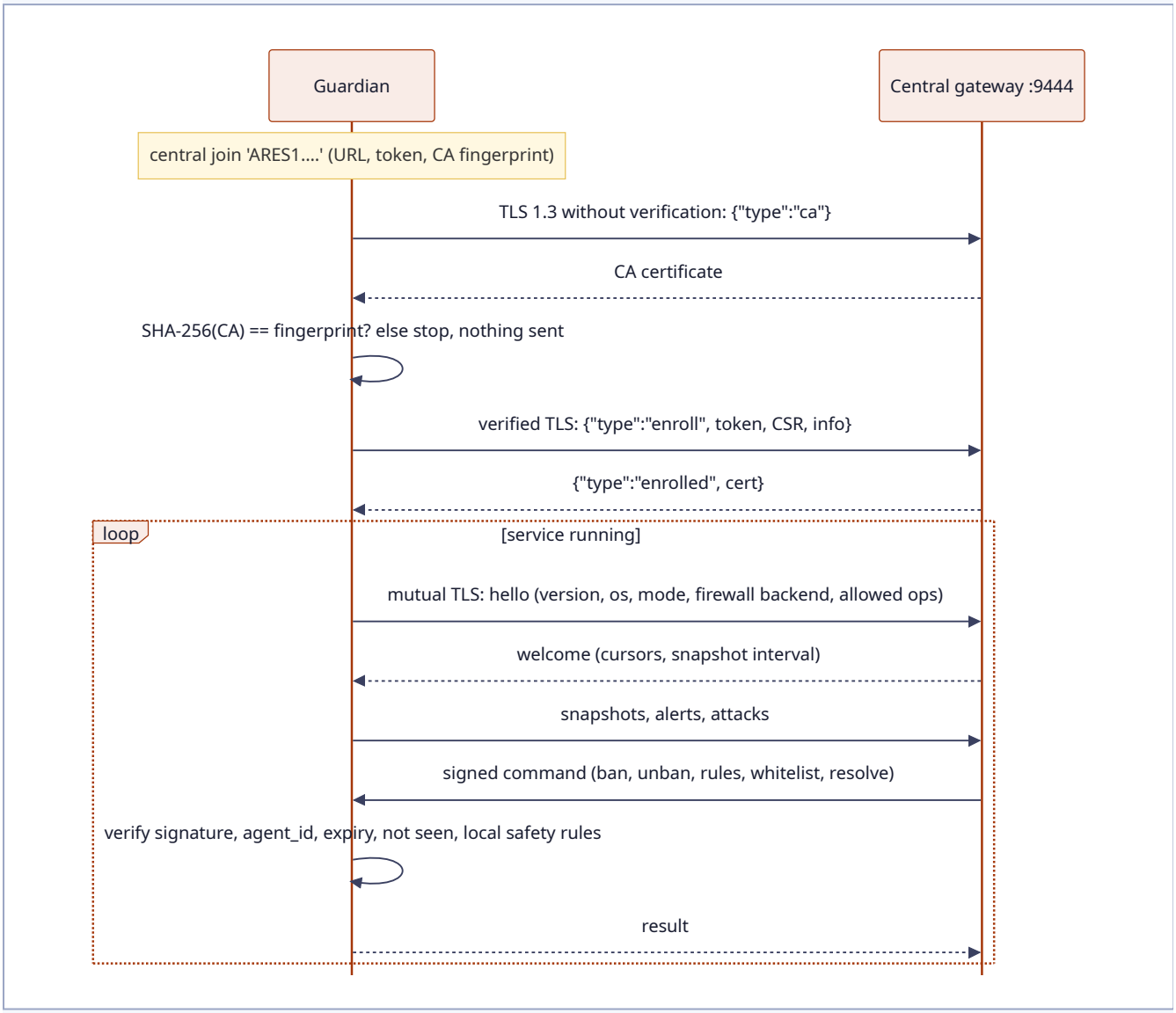


Figure 7. Join once, then a permanent outbound link

Item	Value
State	`/var/lib/ares-guardian/central/` (0700): `agent.key` (0600), `agent.crt`, `ca.crt`, `state.json`, `status.json`
Certificate	issued by Central, renewed automatically before expiry
Reconnect	exponential back-off from 1 s to 120 s with jitter
Limits	`[central] enabled`, `allow_remote_firewall`, `allowed_ops` in `ares.toml`

A-08 Data and storage

ares-server-guardian
rev 1.6.0

Path	Content	Mode
`ares.db` (+ `-wal`, `-shm`)	SQLite, WAL mode, 21 tables	0600
`secret.key`	64-byte URL-safe random key for token signing, made on first start	0600
`central/`	link keys and state (sheet A-07)	0700
`backups/ares-backup-*.tar.gz`	database snapshot, `secret.key`, Central keys, optionally `ares.toml`	0600
`logs/application.log`	the service's own log as JSON lines	0600
`claude/`	Claude Code sign-in for AI analysis (never backed up)	0700

Retention	Default
Events, resolved alerts, web attacks	3 months (choice of 1, 2, 3, 6)
Health samples	14 days
Expired bans	removed by the scheduler every 30 s

A-09 Sign-in and access control

ares-server-guardian
rev 1.6.0

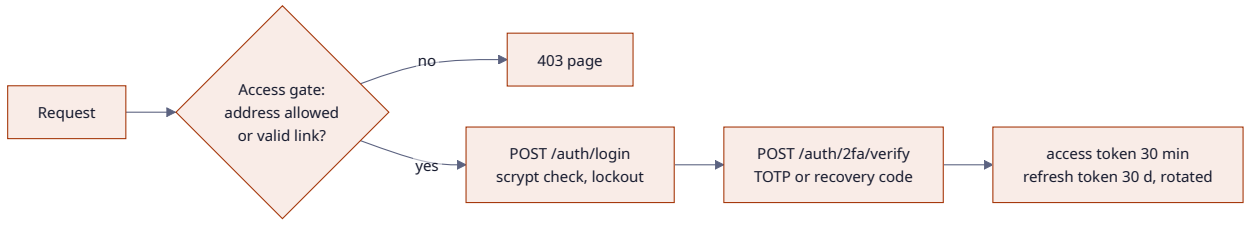


Figure 8. Signing in to the dashboard

Control	Value
Passwords	scrypt N=2^15, r=8, p=1, 10+ characters, mixed
Lockout	5 failures, 15 minutes
2FA	TOTP mandatory, 10 recovery codes
Roles	viewer, operator, admin
Allowed addresses	IPs, networks, DNS names; loopback always allowed unless proxied through an untrusted proxy
Remote access link	emailed, valid 24 h to sign in, session at most 12 h, one account, password + 2FA still required
Exempt from the gate	`/healthz` only

A-10 Interfaces

ares-server-guardian
rev 1.6.0

Interface	Endpoints and commands
REST API	`/api/v1` : `auth` , `users` , `monitoring` (dashboard, alerts, attackers, health), `firewall` , `waf` , `reports` , `ai` , `logs` , `mobile` (info, devices, stream), `settings` ; OpenAPI at `/api/v1/openapi.json` , docs at `/api/docs`
Live stream	WebSocket `/api/v1/stream`
Console	`ares-guardian serve` , `agent` , `status` , `create-admin` , `reset-2fa` , `unlock` , `unban` , `whitelist` , `flush-firewall` , `dns …` , `access …` , `central join/status/leave` , `backup create/list/restore` , `report` , `claude-login` , `claude-status`
Remote install	`scripts/install-remote.sh [--mode agent/standalone/apache] [--central TARGET] HOST…` , `scripts/deploy-remote.sh [--check] [filter…]`

A-11 Operations

ares-server-guardian
rev 1.6.0

Task	How
Health	`GET /healthz` ; `sudo ares-guardian status` ; `systemctl status ares-guardian`
Logs	`journalctl -u ares-guardian -f` ; `/var/lib/ares-guardian/logs/application.log`
Upgrade	`scripts/deploy-remote.sh <hosts>` or `scripts/install-remote.sh <hosts>` ; config, data and keys are kept
Emergency	`sudo ares-guardian flush-firewall` (rules return at the next start); `sudo ares-guardian access disable`
Backup and restore	Settings > Backups or `ares-guardian backup create` ; restore with the service stopped: `ares-guardian backup restore <file>`
Remove	`sudo deploy/uninstall.sh [--keep-data]` ; revoke the server in ARES Central
Release	`packaging/build_release.sh` (tests, wheel, bundle, SHA256SUMS), `scripts/release.sh` (ARES Repository)