ARES
Server Guardian

ARES SUITE

ARES Server Guardian

Executive Summary

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

VERSION	PRODUCTION	DATE
1.6.0	guardian.server1.nacino.net	19 November 2026
AUTHOR	DEMO	CLASSIFICATION
Alfredo Nacino	guardian-demo.nacino.net	Internal

ARES Server Guardian protects Linux servers against attacks. It detects password guessing, floods, scans and web application attacks, and blocks attackers in the server's own firewall. It runs with a web dashboard, or as a headless agent managed from ARES Central.

It is for the people who run internet-facing Linux servers (web sites, mail, VPN, Kubernetes nodes) and want attacks stopped within seconds, without watching logs by hand.

At a glance

8
kinds of attack detected and blocked automatically

10
groups of web attack signatures, including known exploited CVEs

3
firewall backends: nftables, iptables, ufw (plus dry-run)

8
production servers protected today

The problem it solves

Pain	How ARES solves it
Constant password guessing on SSH and web logins, thousands of attempts a day.	One failed login already limits the address; repeated failures ban it, and the third ban is permanent.
Web applications probed for known holes (WordPress, `.env` files, router and VPN exploits).	Every request in the access logs is checked against attack signatures; the attacker is banned at the network firewall, not only in the web server.
Floods that take a site down (HTTP floods, connection and SYN floods).	Per-address bans plus temporary rate limits for every source while a flood lasts.
No overview across servers. Each server keeps its own logs and its own bans.	Every server reports to ARES Central, which shows the whole fleet and changes any firewall with signed commands.
Rules typed by hand go wrong and lock administrators out.	ARES never bans its own whitelist, private addresses or Central, and starts in dry-run until the whitelist is right.

How it works

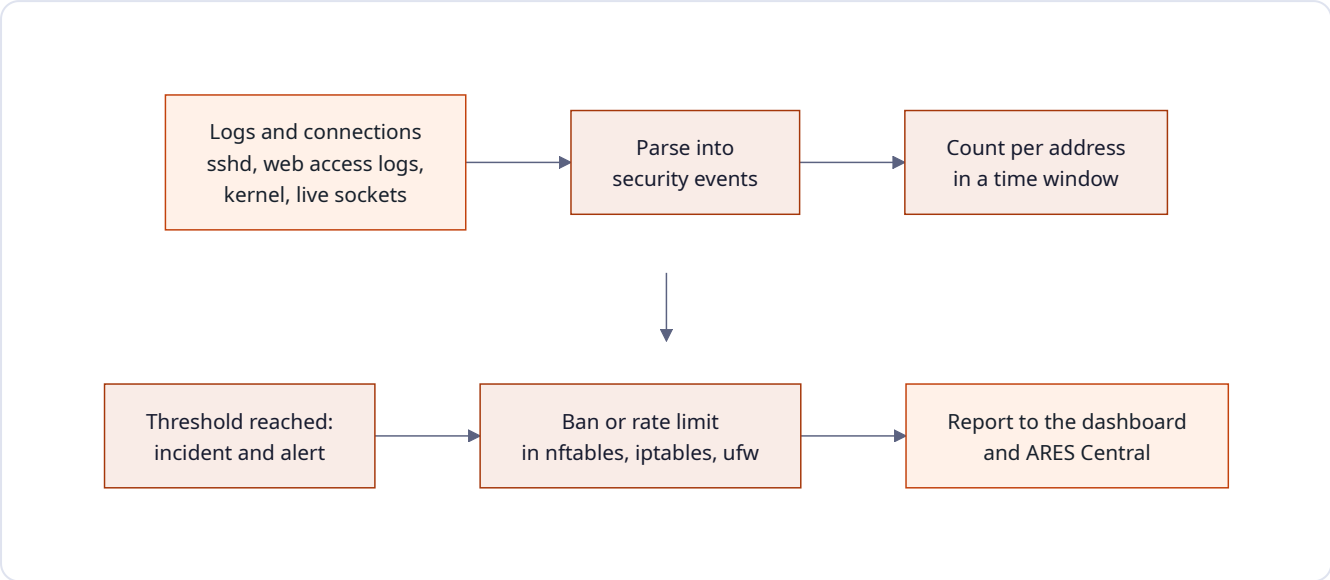


Figure 1. From a log line to a blocked attacker

ARES runs on the server it protects, as the `ares-guardian` system service. It reads the logs the server already writes, so nothing changes in the applications. Bans expire on their own, and repeat offenders are banned for longer each time.

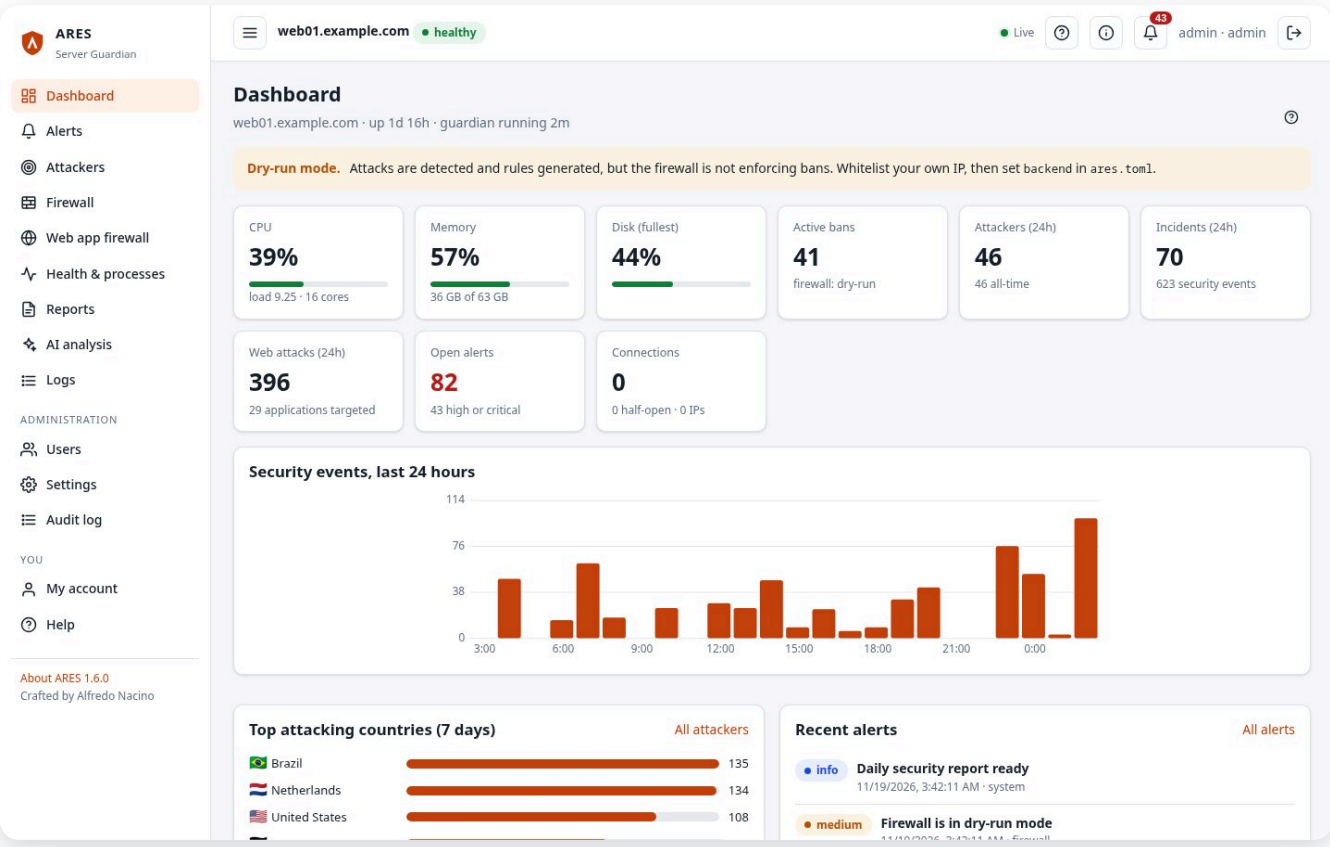


Figure 2. The dashboard of a protected server: health, active bans, attackers, incidents and the security events of the last 24 hours (demo data)

What is new in version 1.6.0

🛰️

Agent mode

A server can run ARES with no dashboard, no users and no open port. It blocks attacks locally and is managed only from ARES Central.

🚀

Install on chosen servers

One command installs or upgrades ARES on the servers you name, over SSH, and joins them to Central with a one-time code.

🧹

Clean removal

An uninstall script removes the service, its firewall rules and, if wanted, its data.

Users and access

- **Viewers** see the dashboard, alerts, attackers, reports and logs.
- **Operators** also ban and unban addresses, add firewall rules and acknowledge alerts.
- **Administrators** also manage users, settings, backups, AI analysis and the audit log.
- Everyone signs in with a password **and an authenticator app** (TOTP), with recovery codes and lockout after repeated failures. Access can be limited to listed addresses, with one-time emailed links for people on the move.

Security and privacy

- ARES dials **out** to ARES Central over mutual TLS 1.3; no management port is opened on a server. Central's commands are signed with Ed25519, name one server, expire after 5 minutes and run once.
- Central can only change the firewall and alerts: it cannot run programs or read files.
- Passwords are hashed with scrypt; sessions use short-lived tokens; every change is in the audit log.
- Logs stay on the server unless forwarding to your own log server is turned on. AI analysis is off by default and never changes the firewall.

How it is delivered

Channel	What
Production	On each protected server as the `ares-guardian` service: 3 Kubernetes nodes with the dashboard behind each node's Apache, 5 standalone servers
Demo (preproduction)	https://guardian-demo.nacino.net: the dashboard with the firewall in dry-run, for trying changes
Agents	Servers managed only from ARES Central, installed with `scripts/install-remote.sh`
ARES Suite	Listed in the ARES Application Suite with this summary, the presentation, the architecture and the description

Risks and limits

> **Warning** A firewall can lock out its own administrators. List the addresses you manage a server from before switching from dry-run to enforcing, and keep console access (or a provider's rescue console).

- **One server, one view**: each server only sees attacks on itself; fleet-wide bans need ARES Central.
- **Logs are the source**: an attack that leaves no log line (for example a TLS-level flood) is only seen through connection counts.
- **Agents are configured in a file**: an agent has no web settings; changes are made in its `ares.toml`.
- **API tokens**: the REST API uses sign-in with 2FA; long-lived API tokens in the Suite format are not implemented yet.

Next steps (proposed)

1. **Route the demo**: add `guardian-demo.nacino.net` to the ARES Edge Proxy Manager.
2. **Move more servers to agents** where nobody needs the dashboard, with `scripts/install-remote.sh`.
3. **Add Suite-format API tokens** and a `/healthz/pingmora` status for APEX Pingmora.
4. **Publish 1.6.0** to the ARES Repository.