

SECURITY

Security in Oktavia and Argos

Vulnerabilities

Hardening

Argos

11 October 2026 · 10 minute read

Network security is not solved with a list of CVEs or a score. It is solved by knowing which device is exposed and why, what to change to fix it, and being sure the tool that found it has not become a new risk itself. This article explains how Oktavia and its assistant, Argos, do it.

Hundreds of CVEs and no decision

Any scanner hands you hundreds of vulnerabilities for every operating system version. Sorted by CVSS score, the list says everything is urgent, and when everything is urgent nothing is. What a network lead needs to know is something else: which ones are being exploited today, on which devices, which version to move to, and whether that version is already part of the standard.

The same goes for configuration. An «SSH exposed» finding without the line that backs it, or the change that fixes it, is a conversation, not a task.

A security finding is useful when it comes with its evidence, its real priority and its fix.

Vulnerabilities by software version

Oktavia takes the platform and version that discovery obtained from each device and matches them against three public catalogs:

- **NVD (CVE API 2.0)**, the NIST database: the CVEs affecting that version, with their severity and the versions that fix them.
- **CISA KEV**: the catalog of vulnerabilities being actively exploited.
- **FIRST EPSS**: the estimated probability that a vulnerability will be exploited in the next thirty days.

The priority order is explicit: KEV first, then severity, exploit probability and score. From that comes a rating per device, per site and per tenant, along with the reason for each: how many actively exploited vulnerabilities, how many critical ones, the highest probability.

For each device, Oktavia suggests the minimum safe version within the same software train—it will not propose jumping from IOS to IOS XE to fix a CVE—and says how many of the severe vulnerabilities it fixes.

That minimum version can be promoted, through an explicit action, to the per-model software standard, and from then on compliance evaluates it in every audit like any other rule.

- **Lab images and virtual nodes.** They are marked as approximate: their version does not allow an exact match and they do not represent real exposure, so they are listed but do not count toward the rating.
- **When it runs.** With every full audit and daily through the task scheduler. A device replacement changes the version, and the analysis is updated on the next audit.
- **What leaves your network.** The catalogs are only queried with product and version, or CVE identifiers. Nothing that identifies your organization or your devices.
- **Printable report.** A CVE report per network or per site, ready to print to PDF, with devices sorted by exposure. Automatic weekly delivery is coming soon.

Security posture with evidence

Security posture is computed from the collected configuration, deterministically, per device and per site, in four domains:

DOMAIN	WHAT IT CHECKS
Perimeter	What goes in and out unfiltered: overly permissive policies, open access lists, services exposed to the internet.
Management plane	SSH and HTTPS reachable from the WAN, telnet or HTTP enabled, virtual lines without an access list, SNMP with default values.
Identities	Admin accounts without a restricted source or second factor, privileged accounts with an unencrypted password.
Segmentation	Unprotected zones, policies without logging, unhardened access ports, orphan objects.

Each finding comes with the line or object that backs it and a concrete recommendation. On FortiGate and Junos, firewall policy analysis is added: permissive rules and what is exposed to the internet. Coverage is multi-vendor: Cisco IOS and IOS XE, FortiGate, Junos, FortiSwitch and Cisco SD-WAN, each read with its own model, not through a common translation.

Per-role hardening, as intent

Detecting a weakness is half the job. The other half is making sure it does not come back. That is why hardening in Oktavia is not a separate document: it is a per-role standard that lives in the same intent as the rest of the configuration, and that a new device inherits when it is onboarded.

- **Baseline.** The minimum per platform: restricted management access, time and logging servers, SNMP without defaults, session timeouts.

- **Weak values.** What the standard declares that contradicts the baseline, such as telnet on the virtual lines, with its fix.
- **Exceptions.** Devices that depart from their role's standard, visible so you can decide whether they are justified or should be cleaned up.
- **Forbidden lines on IOS.** Configuration that must not exist on the device, even if nobody declared it. It is the only extra configuration compliance flags.

Saving the standard does not touch the network. Taking it to the devices goes through the same closed loop as any change: digital twin, approval, apply and verify. Oktavia never proposes secrets: communities, users, passwords and keys are defined by a person.

Argos: network health and diagnosis without a ticket

Argos, the Oktavia assistant, brings together in a single analysis the health of a site, a device or the whole network, across five dimensions: network state, resources, hardware and software (including CVE and KEV), security, and configuration against intent.

What makes it useful is correlation. Every state failure is matched against configuration and classified into one of three causes:

- **The device's own configuration** drifted from intent in the failing service.
- **A neighbor's configuration** explains the failure: the far end of the tunnel or of the BGP session.
- **Operational:** the configuration is fine and the cause is the link, the carrier or the remote device.

With that, Argos builds a diagnosis that does not need a ticket to start: it groups failures into incidents with one cause and several symptoms, and sorts them by impact —isolated site, affected users, degraded redundancy, no user impact—. Each incident proposes what comes next: close the loop where the cause is, test connectivity, or diagnose the device.

All of this is deterministic. Artificial intelligence comes in only when the person asks for it, and only over a bounded summary of the device or site. It never receives full configurations or secrets.

Keeping the tool from becoming the risk

A platform that logs in to every device in the network is, by definition, an attractive target. The decisions that protect it:

- **Credentials in a vault,** with precedence per device, site or tenant. On the test network, day 0 carries only the password hash.
- **Per-tenant isolation.** Each organization sees only its own inventory, results and analyses.
- **SSH identity verification.** Known keys are stored per tenant, and a changed key is rejected until the operator confirms it.

- **Exports without secrets.** CSV exports leave out the lines carrying passwords, communities or keys, and a retired device's archive package is stored without secrets.
- **Nothing sensitive to the language model.** Only bounded summaries, and only on request.

✓ The principles behind it

- **Least change.** What is missing or forbidden is fixed; the full configuration is not rewritten.
- **Human approval.** No change reaches the network without a person reviewing the exact diff.
- **Traceability.** Every change is kept as a digital twin change ticket, with what was proposed, approved, applied and verified.
- **No secrets to AI.** What the model never receives, it cannot leak.

Security with evidence, fixes through the same loop as everything else, and a tool that adds no attack surface.

Do you know today which devices run a version with actively exploited vulnerabilities?

If the answer takes more than a minute, it is worth seeing on your own network. To quote, we only need a rough idea of how many devices you manage.

About Oktanet

Oktanet is a Mexican network automation company focused on controlled, traceable, and scalable operations for enterprise environments across Latin America.

[LinkedIn](#) · [YouTube](#)

© 2026 Oktanet · Oktavia · Public article, with no confidential customer information.