Automate system security and configuration with GUI, API and YAML

Rudder

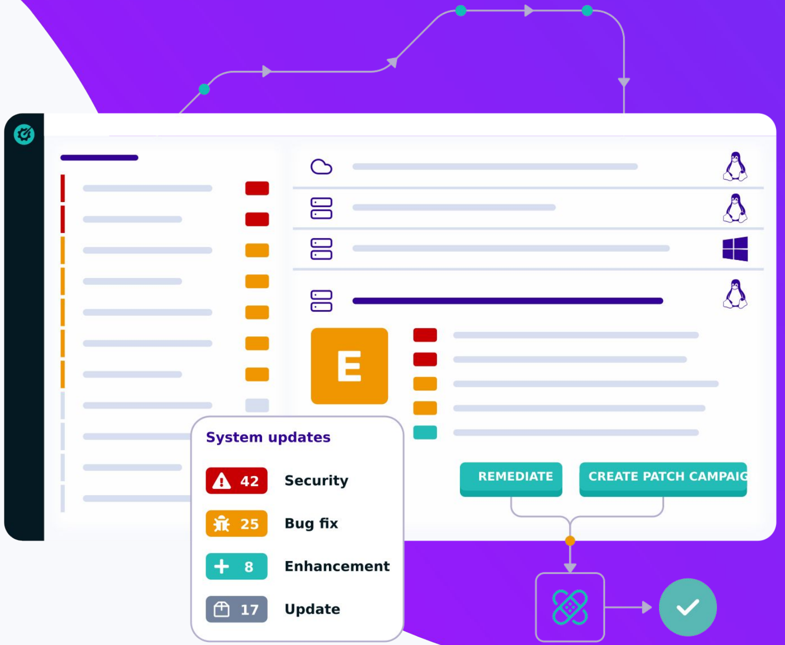
Config Management Camp

Rudder

Security Configuration Management
Audit and remediation of your hardening

Patch & Vulnerability Management
Proactive vulnerability remediation

Policy & Benchmark Compliance
Deployment and proof of compliance



Rudder is a **system infrastructure automation platform** to ensure and improve security posture.

Designed for ops to secure in-depth with proofs,
it allows to implement and maintain **<u>your</u> security model.**

Product license

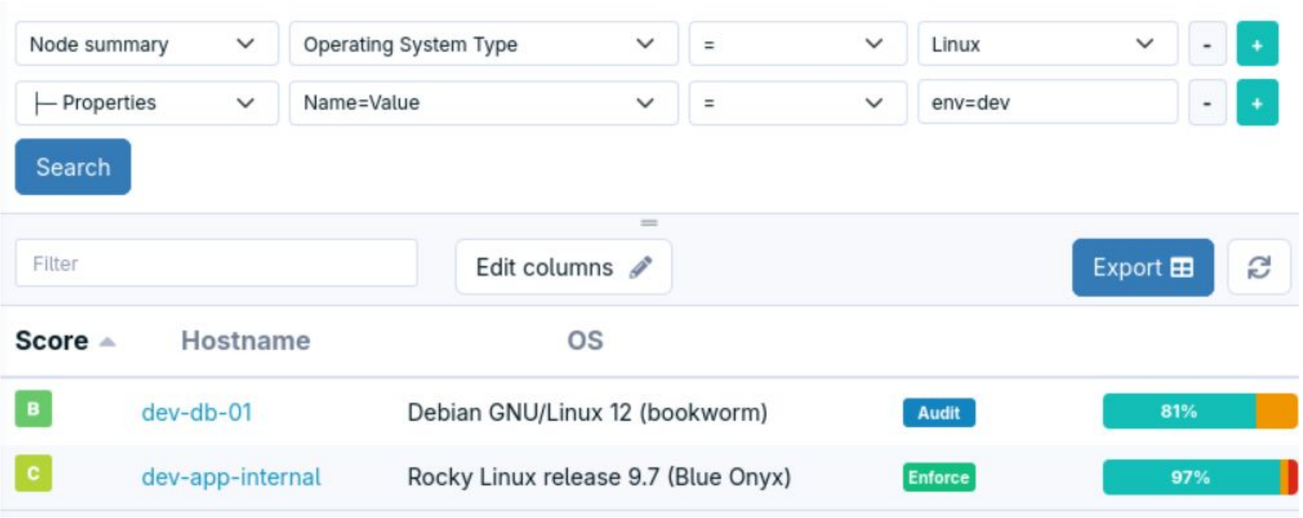
Rudder has an **open-core** model GPLv3 - https://github.com/Normation/rudder

- Most of it is free and open source (web interface, linux agent, binaries, API) with packages available for main distros
- Some plugins, and older OS support are open source, without binaries freely available (won't be in this presentation)
- Some plugins are closed sources (won't be in this presentation)

How it works?

In Rudder, we manage Nodes: **A Node is a system with a rudder agent installed and registered to the Rudder server** (it needs an operating system)

Nodes are classified within **Groups** (dynamic queries on inventories)



Node summary | Operating System Type | = | Linux

├ Properties | Name=Value | = | env=dev

Score	Hostname	OS		
B	dev-db-01	Debian GNU/Linux 12 (bookworm)	Audit	81%
C	dev-app-internal	Rocky Linux release 9.7 (Blue Onyx)	Enforce	97%

How it works?

We define the **desired states**, in Audit or Enforce mode

State definition is a lot like a playbook:

- Don’t include targets
- use agent native methods to inspect/modify the system

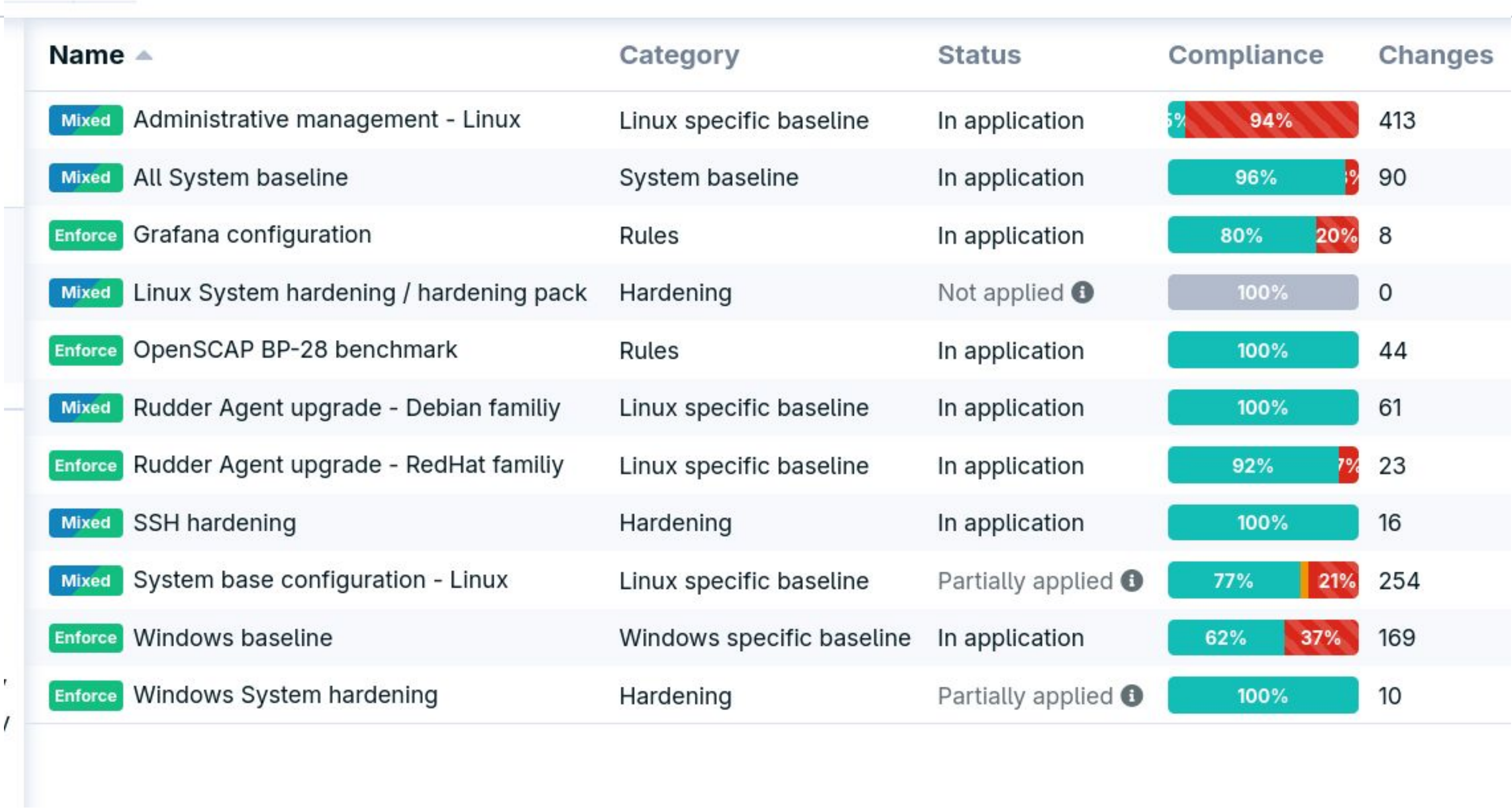
How it works?

We apply these states to the Groups using **Rules** (there can be as many groups and rules as necessary)

The Agent compares these states with the actual status of the Nodes, optionally fixes, and reports the results.

Rudder **computes the compliance** based on these results

How it works?



Name	Category	Status	Compliance	Changes
Mixed Administrative management - Linux	Linux specific baseline	In application	5% 94%	413
Mixed All System baseline	System baseline	In application	96% %	90
Enforce Grafana configuration	Rules	In application	80% 20%	8
Mixed Linux System hardening / hardening pack	Hardening	Not applied ℹ	100%	0
Enforce OpenSCAP BP-28 benchmark	Rules	In application	100%	44
Mixed Rudder Agent upgrade - Debian familiy	Linux specific baseline	In application	100%	61
Enforce Rudder Agent upgrade - RedHat familiy	Linux specific baseline	In application	92% 7%	23
Mixed SSH hardening	Hardening	In application	100%	16
Mixed System base configuration - Linux	Linux specific baseline	Partially applied ℹ	77% 21%	254
Enforce Windows baseline	Windows specific baseline	In application	62% 37%	169
Enforce Windows System hardening	Hardening	Partially applied ℹ	100%	10

Why a GUI?

Goal: onboard everyone and make it “easy” to share knowledge and information

It’s not expected that everyone will become an expert of Rudder, infrastructure management, and compliance but can at least:

- Have a look in the tool and understand what is going on
- Check what’s being done
- Collaborate with others based on their expertise knowledge
- Onboard new members

GUI

Everything can be done within the GUI

- Nodes management/Inventory/Grouping
- Defining Techniques/Directives/Rules
- Checking compliance
- Rudder User management

There are **access rules**

- Access in read/write/nothing to pages

API

Everything can be done with the REST API

- Avoid unnecessary clicks
- Automate Rudder
- Export/Import data from/to Rudder

Acces rules apply also on the API

YAML

We are at the YAMLConf, right ?

- Techniques are written in YAML (and/or the graphical editor)
- In Git
- Compiled by the Rudder server to be understood by the agent

We developed the YAML syntax while writing CIS benchmarks - to ensure that the language is powerful enough to express everything

Demo!

A Rudder server in the cloud, with some hardening configurations.
A new instance is created and is automatically configured.
An API call to define some properties for this instance
A website appears!

Product Roadmap

We release minor versions every 6 months, and major ~ 2 years

Rudder 9.0 was released in October 2025:

- Focus on compliance (augeas module (in the next talk))
- Share data more easily (CSV export)
- HTTPS protocol for client/server communication (used in SecNumCloud environment)
- Better templating engine & command execution

Product Roadmap

Rudder 9.1 is scheduled for April 2026

- Improved traceability on all actions
- Improved dashboard
- Support for SLES 16

Product Roadmap

A **Technique Marketplace** is on its way!

- Expect it for early of Q2 2026

Want to test it?

We have a repo with packages for most distros on https://repository.rudder.io

- Getting started: https://docs.rudder.io/get-started/current/home.html
- Community chat: chat.rudder.io

Thank you!

Any questions?

Rudder

Config Management Camp