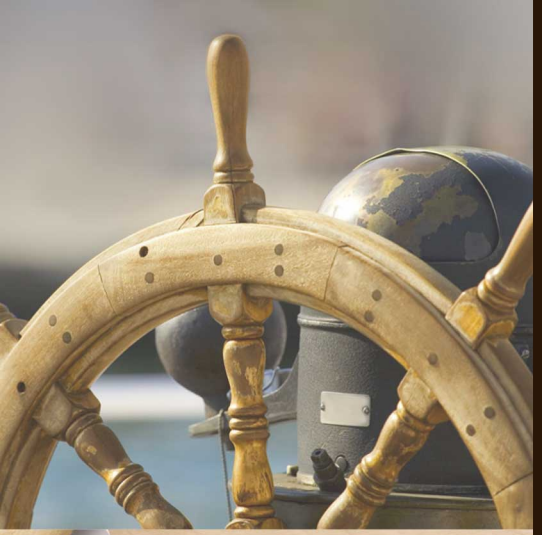
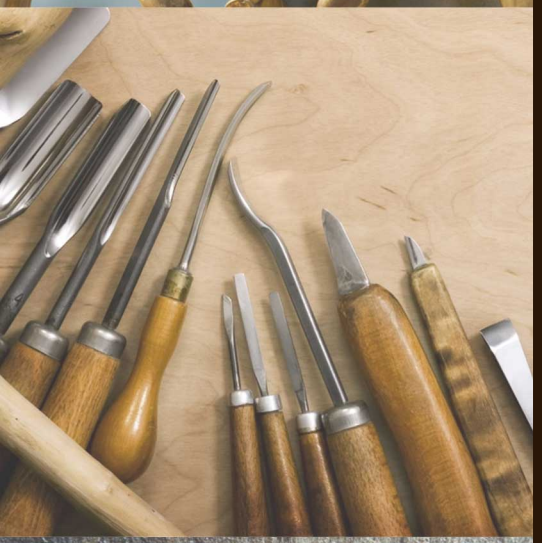
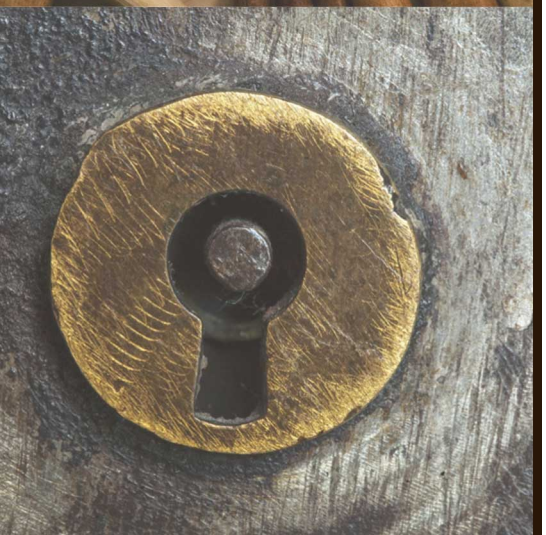






Rudder

flossUK

Automating security policies

From deployment to auditing with Rudder

Jonathan CLARKE – jcl@normation.com

Who am I ?

- Jonathan Clarke
  - **Job:** Co-founder and CTO at Normation
  - **Line of work:**
    - Initially system administration, infrastructure management...
    - Now a whole load of other stuff !
  - **Free software:**
    - Co-creator of Rudder
    - Developer in several LDAP projects: LSC, LTB, OpenLDAP …
    - Contributor to CFEngine

Contact info
Email: jcl@normation.com
Twitter: @jooooooon42 (that's 7 'o's!)

IT infrastructure

Rudder

IT infrastructure

Automation ✔

IT infrastructure

Automation 

Motivations:

Avoid human error	Build new hosts quickly	Rebuild hosts quickly	Scale out quickly

IT infrastructure

Automation ✔

Tools:

CFEngine®

Rudder

SALTSTACK

ANSIBLE

IT infrastructure

Compliance?

IT infrastructure

Compliance?

Motivations:

Know about config **drift**	Get a **complete** overview	Get an **objective** overview	**Prove** compliance

IT infrastructure

Compliance to what?

IT infrastructure

Compliance to what?

Rules come from everywhere:

Laws	Industry regulations	Corporate regulations	Best practices

IT infrastructure

Compliance to what?

Practical examples

MOTD “warning”	Password policy	Enforce *some* parameters in a service	Tripwire (disk contents)

Automation
vs
Compliance

How different is this technically?

How is this different from “just” automation?

Frequency

The more often you check, the more reliable your compliance reporting is.

How can you reach this goal?

Lightweight, efficient agent

CFEngine®

Run “slow” checks in the background (file copying over network...)

Focus on the security checks

Reporting can be done later

How is this different from “just” automation?

All or nothing

Compliance matters on **each and every** system.
Not “most”. All of them.

How can you reach this goal?

Make sure you know what systems exist: rely on an inventory DB

Support all the {old,weird,buggy} {OS,software, versions}

Two systems may be alike on paper, they very rarely are in reality.

How is this different from “just” automation?

Rudder

You cannot get it wrong.
You cannot get it wrong.
You cannot get it wrong.

If you care about compliance,
“prod” is usually pretty real.

How can you reach this goal?

Fake ID +
Prebook flight
to Cayman
islands?

How is this different from “just” automation?

You cannot get it wrong.
You cannot get it wrong.
You cannot get it wrong.

If you care about compliance,
“prod” is usually pretty real.

How can you reach this goal?

Don't touch stuff you don't need to. Be specific.

(One line in a file?)

Start with no changes. Just check. Dry-run?

Cover full cycles (days, weeks, months...)

Classic quality control (reviews...)

So, what have we actually done?

Applied these principles in

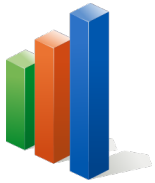
Rudder

http://rudder.cm/

Specifically designed for automation & compliance

Simplified user experience via a Web UI

Based on CFEngine 3



Graphical reporting

Multi-platform (packaged for each OS)

Open Source

Vagrant config to test:

https://github.com/normation/rudder-vagrant/

Introducing Rudder

Rudder

List of configuration rules

Add a new rule

Name	Description	Status	Deployment status	Compliance	Policy instance	Target node group
Danger MOTD		Enabled	In application	Applying	Warning MOTD	Recent OS
Name resolution		Enabled	In application	100%	Name resolution 1, ...	SingleOS
Some Users		Enabled	In application	75%	Basic Users	Without SLES
SSH everywhere		Enabled	In application	16%	SSH prod	AllLinux
SSH srv prod		Enabled	In application	100%	SSH prod	All debian
Users and Fstabs CR	Test CR	Enabled	In application	100%	Fstab on Public Share, ...	SingleOS

First | Previous

Node compliance detail

Rule: Custom Rule
Directive: Time settings

Show 10 entries

Search

Node	Status
orchestrateur-1.labo.normation.com	Success

Component	Status
Time synchronization (NTP)	Success
Hardware clock (RTC)	Success
Time zone	Success

Value	Message	Status
None	Time zone was already correctly configured	Success

Key points for security compliance

Rudder

Continuous checking
Every 5 minutes

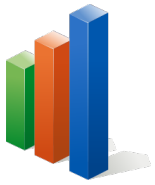
→ **High freqency, trust in compliance reporting**

Separate configuration from implementation

→ **Reuse implementations, less bugs, shared code...**
Clear separation of roles

Multi-platform
Linux, Unix, Windows, Android...

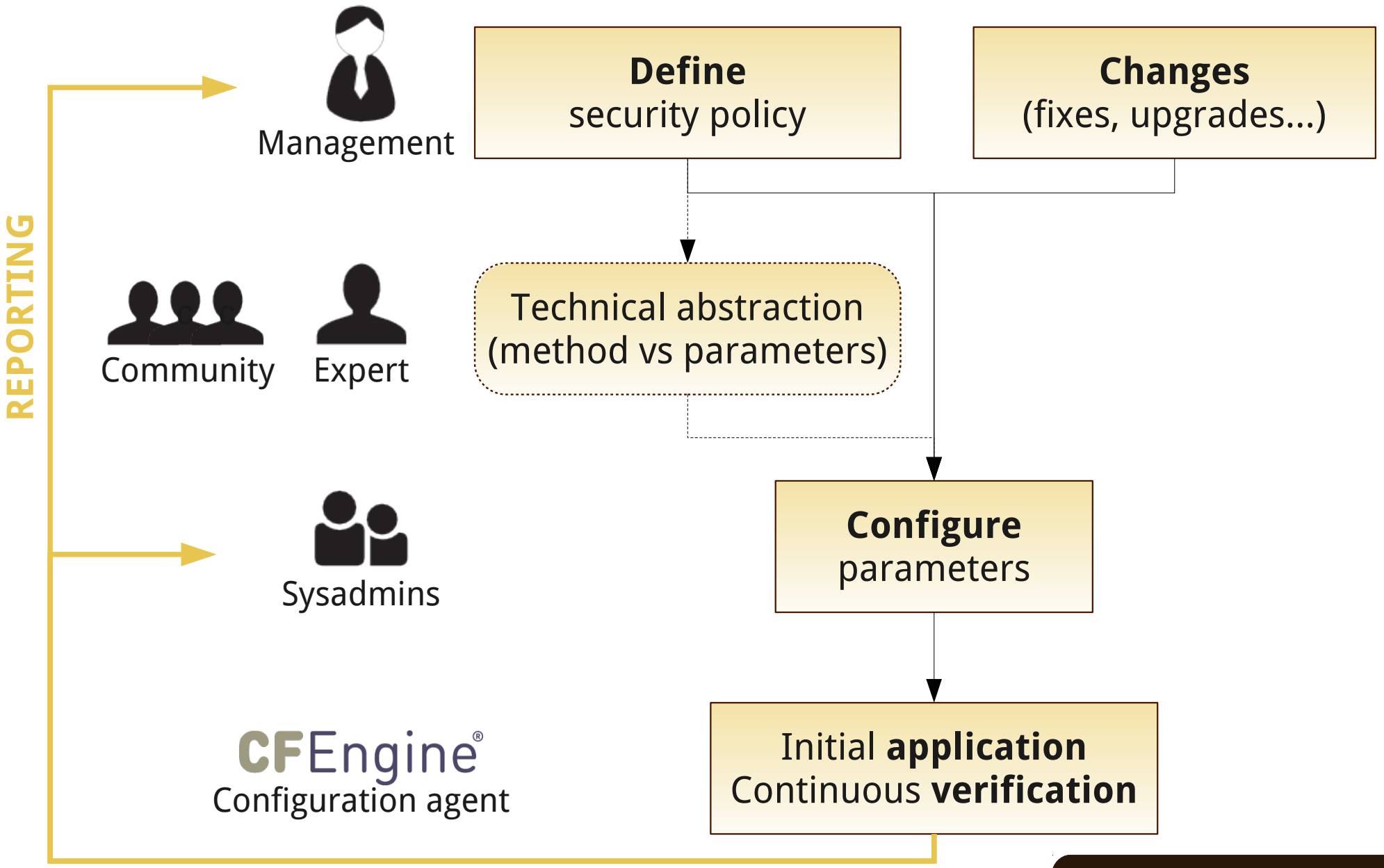
→ **Cover as many systems as possible**



Reporting
Done after the checks, separate process

→ **Avoid bottleneck**
Different report types

Rudder - workflow



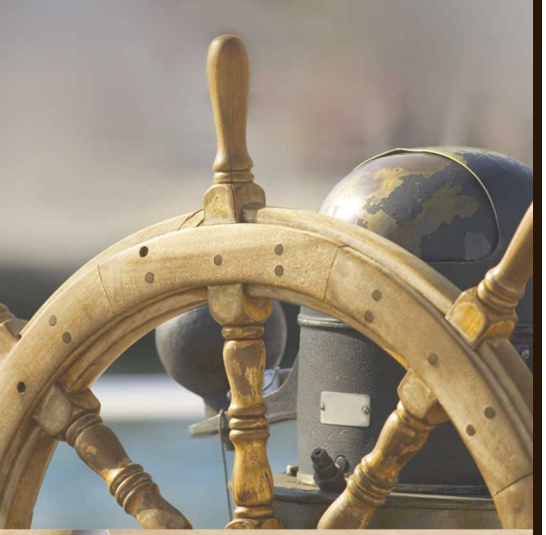
Final thoughts

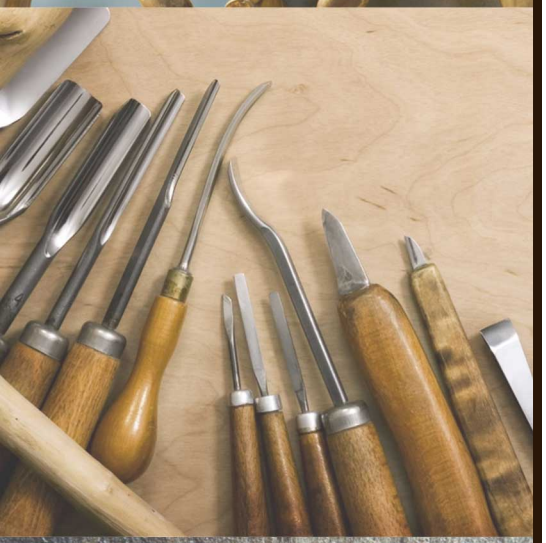
Rudder

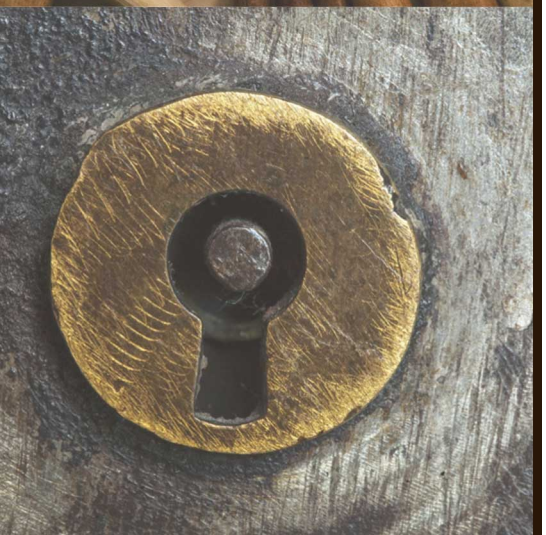
Summary:
- Security compliance is a very demanding type of automation
- Possible today with open source tools
- Main issue is about **how** you use them!

Next steps?
- Authorizations: who can change which parameters?
  (law vs regulations vs policy...)
- Correlate with monitoring data: determine root causes, cross effects...

It works but the tools can be improved:
- detect changes (inotify?) - even 1 minute not always enough
- dry-run iterations automatically?







Rudder

flossUK

Questions?

Follow us on Twitter:
@RudderProject

Jonathan CLARKE – jcl@normation.com