Compliance as Code

April 2019

Anthony Rees
Solutions Architect APJ

What Is InSpec?

Introducing InSpec

InSpec helps express security & compliance requirements as code and incorporate it directly into the delivery process.

Systems shall have a Mandatory Access Control system installed and enabled.

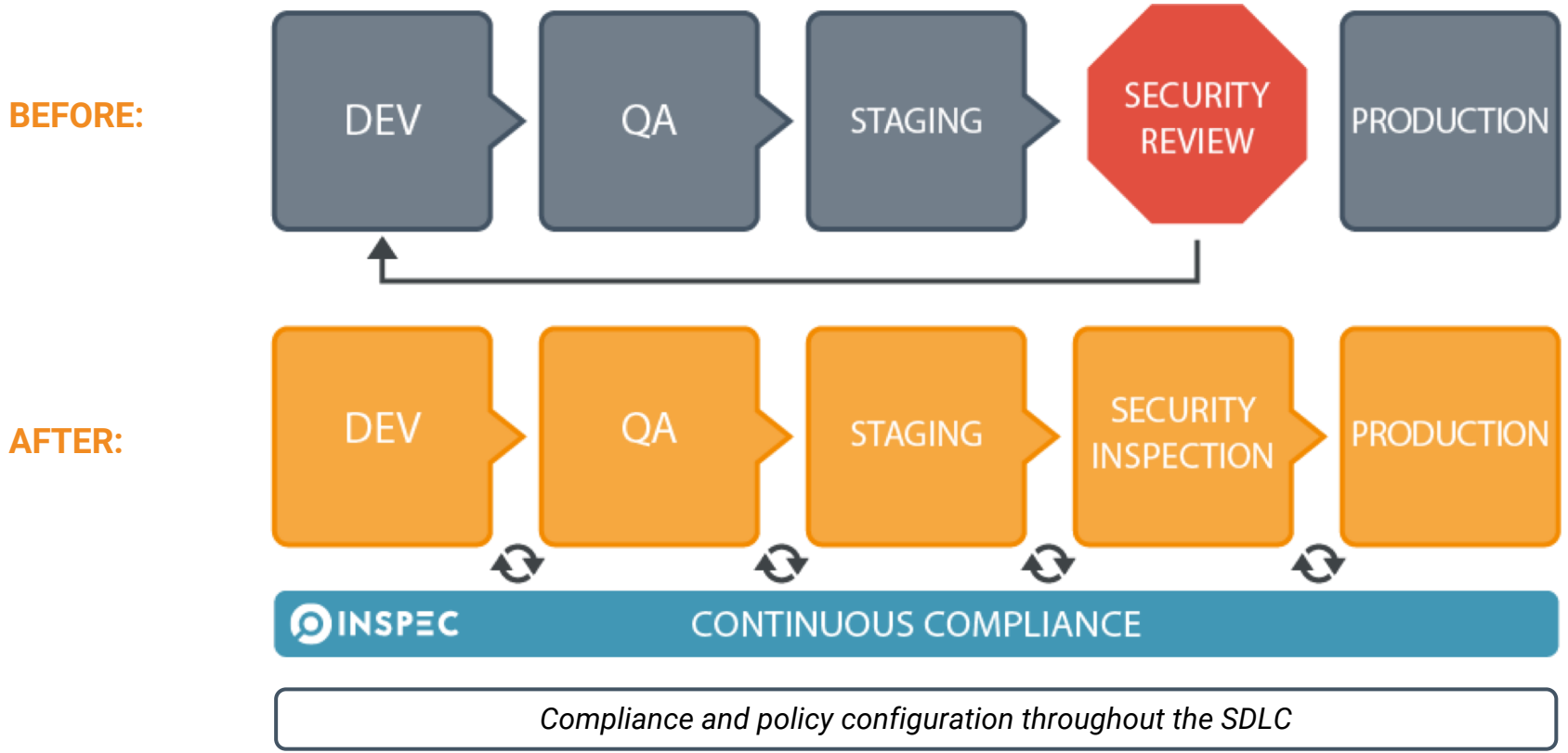
```
control "ensure_selinux_installed" do
  title "Ensure SELinux is installed"
  desc  "SELinux provides Mandatory Access Control"

  impact 1.0
  describe package("libselinux") do
           it { should be_installed }
  end
end
```

Compliance is the Business Requirement

InSpec enables **DevSecOps** by allowing cross-functional application, infrastructure, and security teams to assess & remediate compliance issues through the entire software delivery process.

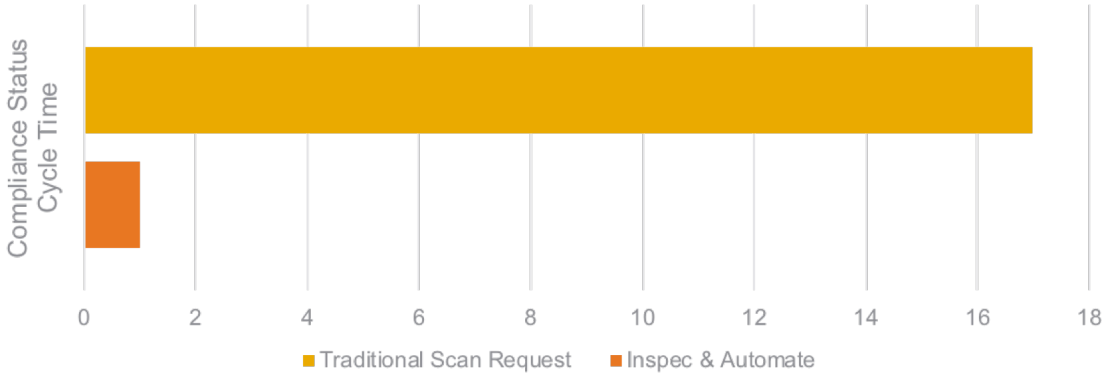
Traditional Compliance vs. InSpec



Customer Benefits

InSpec helps organizations:

- Maintain an up-to-date view of compliance status in production
- Detect security issues long before they reach production
- Reduce risk while delivering applications faster



Example: Major healthcare services provider reduced audit cycle times by 95% by continuously detecting and remediating compliance errors.

InSpec

Turn security and compliance into code

- Translate **compliance** into Code
- **Clearly** express statements of policy
- Move risk to build/test from **runtime**
- Find issues **early**
- Write code **quickly**
- Run code **anywhere**
- **Inspect** machines, data and APIs
  - 100+ built-in resources

INSPEC

PART OF A PROCESS OF CONTINUOUS COMPLIANCE

A SIMPLE EXAMPLE OF AN INSPEC CIS RULE

```
control 'cis-1.4.1' do
   title '1.4.1 Enable SELinux in /etc/grub.conf'
   desc '
Do not disable SELinux and enforcing in your GRUB
configuration. These are important security features that
prevent attackers from escalating their access to your systems.
For reference see …
'
   impact 1.0
   expect(grub_conf.param 'selinux').to_not eq '0'
   expect(grub_conf.param 'enforcing').to_not eq '0'
end
```

One Framework for ALL

One Framework

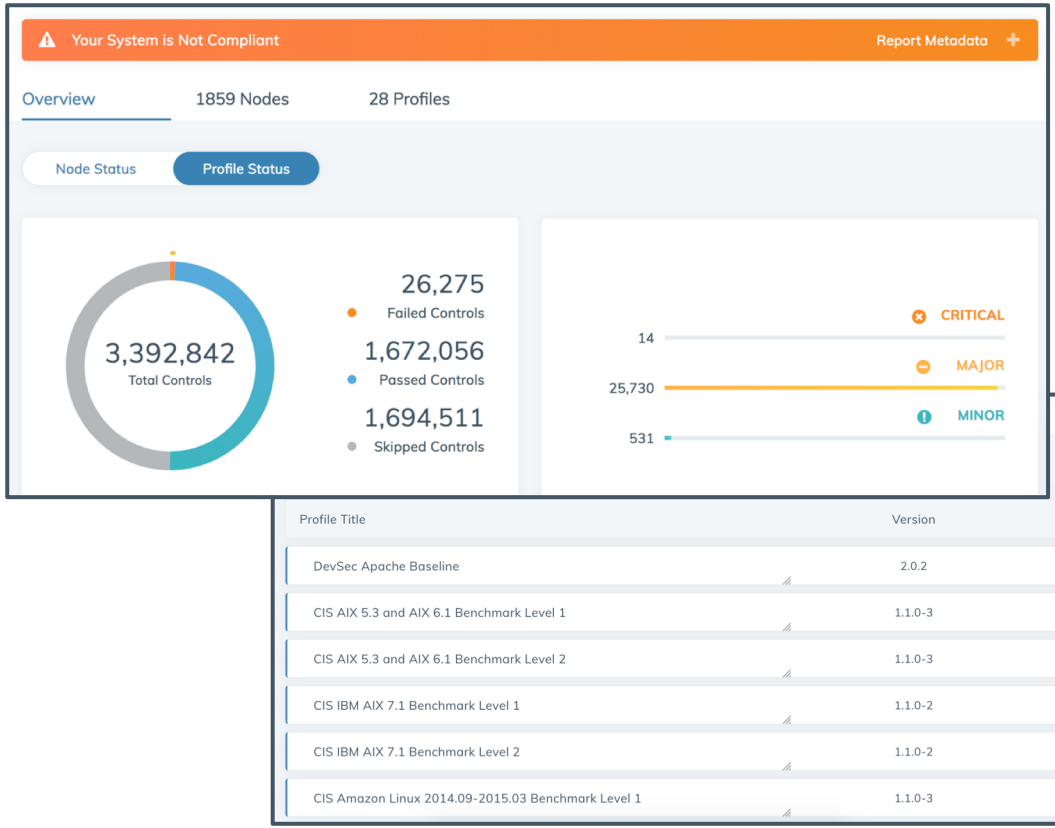
Linux, Windows, MacOS, Solaris, AIX, ...

Bare-metal, VMs, Containers

Databases, APIs, Cloud Platforms, ...

Continuous Compliance with Chef Automate

- Real-time enterprise fleet compliance dashboard
- 140+ built-in baselines for standard compliance frameworks
- Compliance report generation and sharing/exporting



Chef Automate Compliance Profiles

140+ Commercially-supported profiles

- CIS Benchmarks for commercially-supported operating systems
- US government STIG profiles for operating systems
- Database and Application frameworks
- Patching baselines
- Cloud and virtualization platforms
- Network devices
- Frequent updates and new releases

Examples of Available Resources

apache_conf
apt
audit_policy
auditd_conf
auditd_rules
bond
bridge
command
crontab
directory
etc_group
file
gem
group
host
inetd_conf
interface
iptables
kernel_module
kernel_parameter
limits_conf
login_defs
mount
mysql_conf
mysql_session
npm
ntp_conf
oneget
os
os_env
package
parse_config
parse_config_file
passwd
pip
port
postgres_conf
postgres_session
powershell
processes
registry_key
security_policy
service
ssh_config
sshd_config
user
windows_feature
yum

Cloud

Cloud Verification

CIS Center for Internet Security®

Chef is first CIS Partner Certified on AWS, Azure and GCP! Write compliance policies for all aspects of cloud configuration:

- Virtual machines
- Security groups
- Block storage security policies
- Networking
- Identity and access management
- Log management

Example: AWS S3 Bucket Policy

```
describe aws_s3_bucket(bucket_name: 'my_secret_files') do
  it { should exist }
  it { should_not be_public }
  it { should have_access_logging_enabled }
end
```

Example: AWS EBS Volume Policy

```
describe aws_ebs_volume('vol-01a2349e94458a507') do
  it { should exist }
end

describe aws_ebs_volume(name: 'data-vol') do
  it { should be_encrypted }
end
```

Example: Azure Security Group Policy

```
title 'Network Security Group Properties'
control 'azure-generic-network-security-group-1.0' do
  impact 1.0
  title 'Ensure that the webserver security group has been set up as expected'
  describe azure_generic_resource(group_name: 'production', name: 'webservers')
  do

    its('location') { should cmp 'westeurope' }

    it { should_not have_tags }

    its('properties.provisioningState') { should eq 'Succeeded' }
  end
end
```

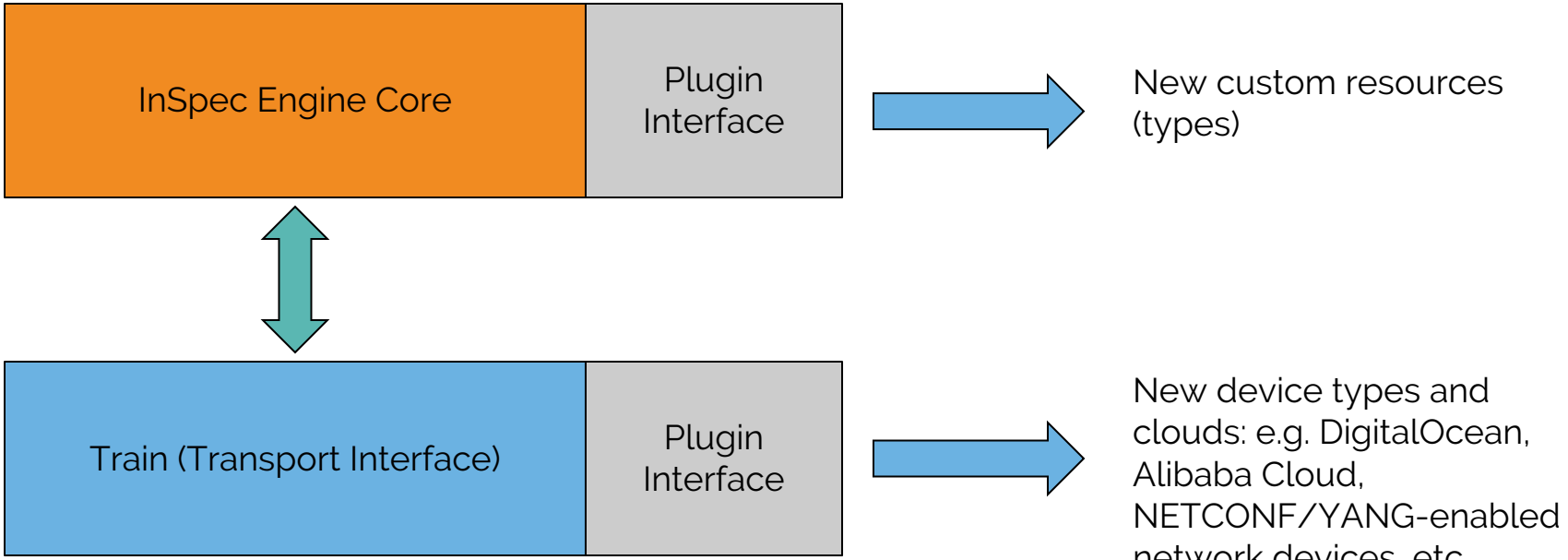
Databases

DB Testing

```
describe mysql_session.query("SELECT user, host FROM mysql.user WHERE host = '%'") do
  its(:stdout) { should be empty  }
end
```

Plugins

Introducing InSpec's Plugin Layer



Example: DigitalOcean Support Using Plugin System

```
describe digitalocean_droplet(id: '112209628') do
  it { should exist }
end

describe digitalocean_ssh_key(id: '112209628') do
  it { should exist }
end
```

A Train plugin under the hood brokers the communication with the DigitalOcean API including authorization, authentication, making the correct API calls, etc.

Terraform

Compliance for Hashicorp Terraform

- Introduces compliance-as-code directly into the infrastructure provisioning process with Hashicorp Terraform

- **InSpec Provisioning Plugin for Terraform** runs InSpec tests after a "terraform apply" operation for servers and clouds

- **InSpec Generator ("Iggy")** generates a starter set of InSpec controls by parsing an existing Terraform state file

Terraform Plugin Example

```
resource "digitalocean_droplet" "web" {
  image = "ubuntu-16-04-x64"
  size = "s-1vcpu-1gb"
  region = "${var.digitalocean_region}"

  # installs inspec and executes the profiles against the newly-created/modified machine
  provisioner "inspec" {
             profiles = [
               "supermarket://dev-sec/linux-baseline",
               "compliance://jsmith/cis-ubuntu16-level1-benchmark",
             ]

             on_failure = "continue" # or error out if desired
  }
}
```

Example: Multiple Descriptions Per Control

```
control 'cis-rule-5.2.9_Ensure_SSHPermitRootLogin_disabled' do
  impact 'critical' # previously only 0.0-1.0 allowed
  title 'Ensure SSH PermitRootLogin is disabled'
  desc 'Do not allow root user to log in directly'
  desc 'cis-rule', '5.2.9'
  desc 'pci-requirement', '8'
  describe sshd_config do
       its('PermitRootLogin') { should eq 'no' }
  end
end
```

What is it not?

- IDS / IPS
- Firewall
- Antivirus
- Pentesting tool

CHEF™