CHEF™

InSpec: Automated Tests for Compliance and Security

Mandi Walls | mandi@chef.io

INSPEC BY CHEF

HI!

- Mandi Walls
- Technical Community Manager for Chef, EMEA
- mandi@chef.io
- @lnxchk
- https://www.chef.io/
- https://www.inspec.io/

INSPEC BY CHEF

EVERY business is a software business

> We're going to be a software company with airplanes.
>
> – CIO, Alaska Airlines

INSPEC BY CHEF

ATTACKS/BREACHES

6/21/2017
05:15 PM

WannaCry Forces Honda to Take Production Plant Offline

Jai Vijayan
News

Connect Directly

1 COMMENT
COMMENT NOW

Login

Work on over 1,000 vehicles affected at automaker's Sayama plant in Japan while systems were restored.

In an example of just how persistent modern cyberthreats can be, automaker Honda Motors had to temporarily stop production at its Sayama plant in Japan this week after being hit by WannaCry, a malware threat the company thought it had mitigated just one month ago.

The nearly 48-hour shutdown impacted production of about 1,000 vehicles at the facility, which does engine production and assembly for a line of vehicles including the Odyssey minivan and the Accord.

A statement from Honda North America said the interruption at the Sayama Auto Plant was caused by the shutdown of several older production-line computers infected with the WannaCry virus.

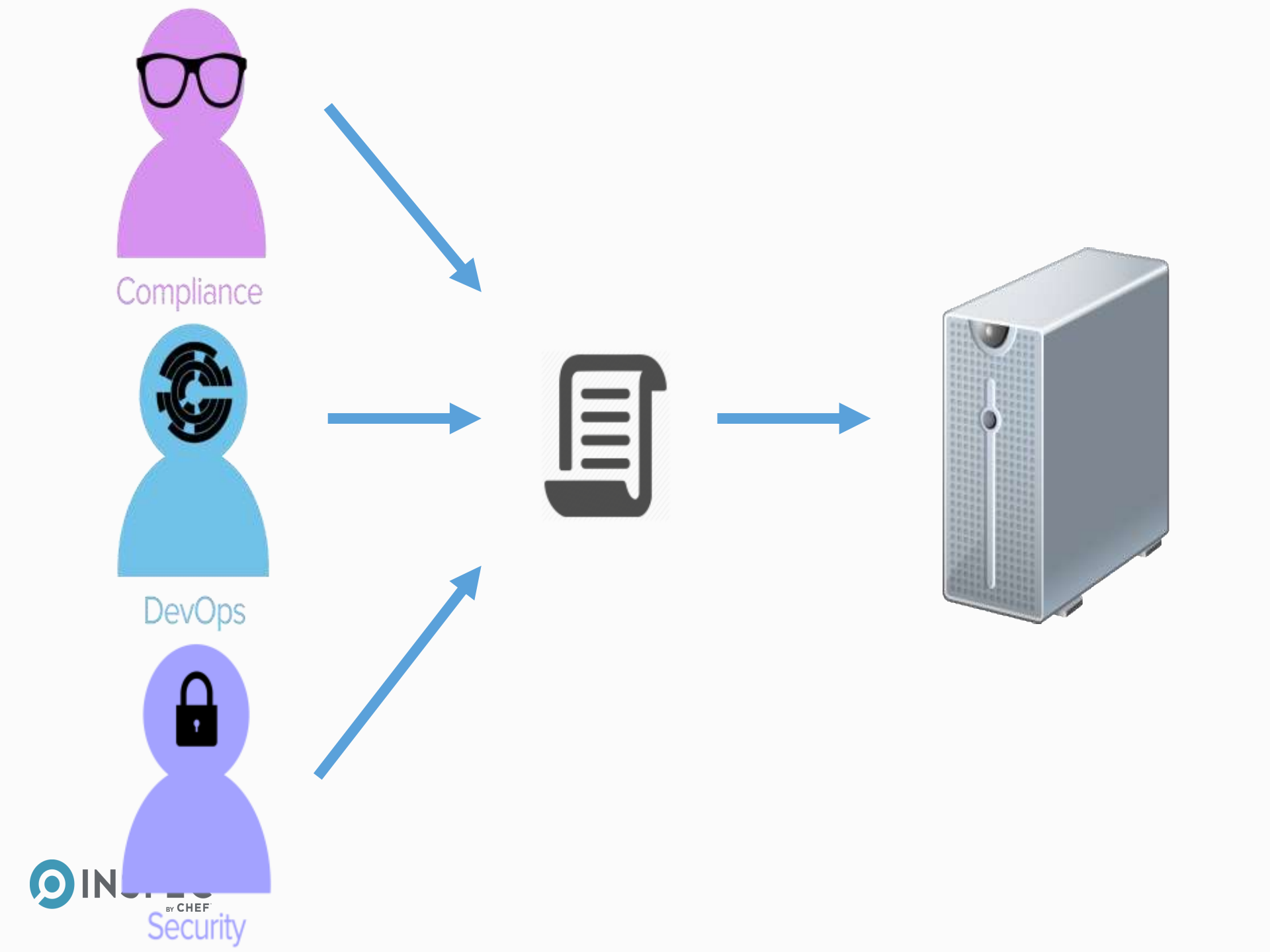
Different Sources for the Same Goals

Compliance

Security

DevOps

INSPEC BY CHEF


Compliance
DevOps
Security
BY CHEF

InSpec

- Human-readable language for tests related to security and compliance
- Create, share, and reuse complex profiles
- Extensible language - build your own rules
- Command-line tools plug into your existing workflow, build, deploy
- Integrates with Test Kitchen for fast feedback
- Test early, test often!

INSPEC BY CHEF

Create and Consume

- Complex compliance requirements can slow you down
- Share information and expertise
- Compliance as code leverages cross-team knowledge
- InSpec is code – check into repos, publish as artifacts
- Include InSpec before code checkin
- Include InSpec in integration and pre-production
- Continue InSpec checks in production to guard against new threats

SSH Requirement

- If your security team sends you a directive:

  SSH supports two different protocol versions. The original version, SSHv1, was subject to a number of security issues. All systems must use SSHv2 instead to avoid these issues.

Checking and Fixing

- Identify the file and file location on your platforms
  - In all environments
- What setting to change
  - Do we check it first or just push a new one everywhere?
- What's the plan for the OS images?
  - Rebuild? Remediate at instantiation?
- Do you test before pushing changes?

Lifecycle – How Often Do You Check Security?

- Single big scan, report mailed out with a “due date”?
  - Considered done, not checked again
- Yearly or twice-yearly massive scans with remediation emergencies?
  - Common audit cycles, large projects around fixing found issues
- Part of the software development lifecycle?
  - “To the left”
  - Regularly part of what is included in builds

INSPEC BY CHEF

Photo: https://www.flickr.com/photos/tarn-aveyron/2124972713/

Check that sshd_config

```
describe sshd_config do
  impact 1.0

  title 'SSH Version 2'

  desc <<-EOF
    SSH supports two different protocol versions. The original version, SSHv1, was subject to a number of security issues. Please use SSHv2 instead to avoid these.
  EOF

  its('Protocol') { should cmp 2 }
end
```

Resources

- InSpec includes built-in resources for common services, system files, and configurations
- Built-in resources work on several platforms of Linux.
  - There are also Windows-specifics
- A resource has characteristics that can be verified for your requirements, and Matchers that work with those characteristics

Sample Resources

- System resources: directory, file, user, group, crontab, service, package
- Specific services: apache, nginx, rabbitmq, postgresql, IIS
- Programming language components: gem, npm, powershell
- Network services: port, http, sshd
- Cloud resources: AWS, Azure
- https://www.inspec.io/docs/reference/resources/

Characteristic Tests

- it { should exist } – files, directories, groups
- it { should be_installed } – packages
- it { should be_enabled } – services
- its('max_log_file') { should cmp 6 } – rotate auditd logs
- its('exit_status') { should eq 0 } – run any command

Run InSpec

- InSpec is command line
  - Installs on your workstation as a ruby gem or as part of the ChefDK
- Can be run locally, test the machine it is executing on
- Or remotely
  - InSpec will log into the target and run the tests for you
- Also a REPL
  - https://www.inspec.io/docs/reference/shell/

Create a Basic Test

- Basic test to make sure /tmp is a directory
- It also should be owned by root
- And its mode should be 01777 – open to all (plus sticky bit!)

test.rb

```
describe file("/tmp") do
  it { should exist }
  it { should be_directory }
  it { should be_owned_by 'root' }
  its('mode') { should cmp '01777' }
end
```

INSPEC BY CHEF

Test Any Target

```
inspec exec test.rb

inspec exec test.rb -i ~/.ssh/key.pem -t
ssh://ec2-user@54.152.7.203

inspec exec test.rb -t
winrm://Admin@192.168.1.2 --password super

inspec exec test.rb -t docker://3dda08e75838
```

INSPEC BY CHEF

Execute InSpec

```
[chef@host ~]$ inspec exec ./test.rb
Profile: tests from ./test.rb
Version: (not specified)
Target:  local://
  File /tmp
     ✓  should exist
     ✓  should be directory
     ✓  should be owned by "root"
     ✓  mode should cmp == "01777"

Test Summary: 4 successful, 0 failures, 0 skipped
```

INSPEC BY CHEF

Execute in Build Pipelines

- InSpec runs with failed tests return a non-zero return code

```
Profile Summary: 0 successful, 1 failures, 0 skipped
$ echo $?
1
```

- Passing tests have 0 return code

```
Profile Summary: 1 successful, 0 failures, 0 skipped
$ echo $?
0
```

INSPEC BY CHEF

Profiles

- InSpec profiles allow you to package and share sets of InSpec tests for your organization or for a specific application set
- Each profile can have multiple test files included
- Flexible!
  - Create your own profiles for specific software you use
  - Use included matcher libraries or write your own – they live in the profile
- https://dev-sec.io/

INSPEC BY CHEF

Sample Profile: *linux-baseline*

```
control 'os-02' do
  impact 1.0
  title 'Check owner and permissions for /etc/shadow'
  desc 'Check periodically the owner and permissions for
/etc/shadow'
  describe file('/etc/shadow') do
    it { should exist }
    it { should be_file }
    it { should be_owned_by 'root' }
    its('group') { should eq shadow_group }
    it { should_not be_executable }
    it { should be_writable.by('owner') }
...
```

INSPEC BY CHEF

Demo Scenario

- My security team has produced a centralized profile – demo_inspec_small
- It is shared on github: https://github.com/lnxchk/demo_inspec_small
- I am working on a dev machine, and I want to make sure what I'm doing isn't in violation
- I need a user account to run my application; the account is *servusr*
- I create the user, do my work, and then test my machine with the centralized profile
- There are things I need to fix!

Profile Dependencies

my-app-profile

```
control 'myapp-1'
control 'myapp-2'
control 'myapp-3'

include_controls 'my-baseline' do
  skip_control 'baseline-2'
end
```

my-baseline

```
control 'baseline-1'
control 'baseline-2'
```

INSPEC BY CHEF

Skipping Individual Controls

```
include_controls 'linux-baseline' do
  skip_control 'os-10’
  skip_control 'os-08’
  skip_control ‘package-08'
  skip_control 'sysctl-14'
end

include_controls ‘demo_inspec_small’ do
  skip_control ‘sshd_config-01’
end
```

INSPEC BY CHEF

Fast Feedback with Test Kitchen

- Test Kitchen is a tool for your team to create fast-feedback loops for development
- Add InSpec tests to TK so that any change can also be certified with the security profile before it is pushed to source code repository
- More info at http://kitchen.ci/

INSPEC BY CHEF

Include InSpec in Your Workflow

- Infrastructure developers rely on InSpec profiles while working on configurations
- App devs consume InSpec profiles: new features don’t violate security requirements
- Security and compliance work with all teams to create profiles to meet requirements and not prevent work
- Build, Integration, Test environments built to meet InSpec requirements
- Production systems checked regularly to manage drift, ensure against new threats

Other Features: Cloud Checks

- InSpec has tests for common objects in public cloud services

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

INSPEC BY CHEF

Resources

- https://inspec.io
- http://www.anniehedgie.com/inspec-basics-1
- https://blog.chef.io/2018/06/19/inspec-gcp-deep-dive/
- https://blog.chef.io/2018/06/14/understanding-singular-and-plural-inspec-resources/
- https://blog.chef.io/2018/05/23/automatically-generating-inspec-controls-from-terraform/
- https://blog.chef.io/2018/05/23/inspec-now-available-in-azure-cloud-shell/

INSPEC
BY CHEF

Gratuitous Hiring Slide

- Work on InSpec in our Belfast office!
- Also integration engineers, support in Belfast
- Sales, customer architects in London
- https://www.chef.io/careers/open-positions/
- Meet our Belfast team on Tuesday

  https://events.chef.io/events/belfast-office-launch-celebration/

CHEF™

INSPEC BY CHEF