INSPEC

Intro and 1.0 release

Alex Pop – Software Engineer @ Chef Software
https://uk.linkedin.com/in/al3xpop

CHEF

Agenda

- **Why infrastructure tests as code**
- **What is InSpec and how it works**
- **Core and custom resources**
- **What's new in InSpec 1.0 (released Sept 26, 2016)**
- **Documentation and installation**
- **Integrations**
- **Demo**
- **Chef Community Summit**

Why infrastructure tests as code

- **Regression and Integration testing**
- **Security and compliance testing**
- **Tests to run during incidents**
- **Checklist tests before upgrade / code release / etc**
- **Frameworks (InSpec, ServerSpec, Testinfra, Pester, Bats, etc)**

What is InSpec?

- Open-source framework and rule language to specify compliance, security and policy for testing any node in your environment.
- Includes a collection of resources to write rules quickly and easily
- Profiles and custom resources provided by the community

CHEF

What is InSpec and how it works

	chef-client	inspec
Open Source	https://github.com/chef/chef	https://github.com/chef/inspec
First Commit	March 2008	April 2015
Language	Ruby DSL	Ruby DSL

Code	`service 'iptables' do`
	`  action [ :enable, :start ]`
	`end`
		`describe service('iptables') do`
		`  it { should be_enabled }`
		`  it { should be_running }`
		`end`

Execution	Local	Local & Remote (SSH, WinRM, Docker)
Artifacts	Cookbooks & Recipes & Resources	Profiles & Controls & Resources
Share Artifacts	Chef Supermarket, Github, Bitbucket, etc	Chef Supermarket, Github, Bitbucket, etc

InSpec's execution overview

train gem

auth & coms

Target:

- **Local (shell out)**
- **SSH (remote commands)**
- **WinRM (remote commands)**
- **Docker (exec, remote commands)**

CHEF

InSpec's execution example

```
$ inspec exec /tmp/passwd.rb
$ inspec exec /tmp/passwd.rb -t docker://3dda08e75838
$ inspec exec /tmp/passwd.rb -t winrm://John@10.0.0.4 --password xyz
$ inspec exec /tmp/passwd.rb -t ssh://john@10.0.0.5 -i ~/john.pem
```

/tmp/passwd.rb content:

```
describe file('/etc/passwd') do
  it { should exist }
  it { should be_file }
  its('mode') { should cmp '0644' }
  its('owner') { should eq 'root' }
  its('group') { should eq 'root' }
  its('size') { should be > 100 }
end
```

train gem

Commands executed on target:

Basic OS detection(like Ohai):

```
uname
test -f /etc/os-release && cat /etc/os-release
test -f /etc/redhat-release && cat /etc/redhat-release
```

Resource specific commands:

```
test -e /etc/passwd
stat -L /etc/passwd
```

CHEF

InSpec's controls examples

```
control 'windows-rdp-101' do
  impact 1.0
  title 'Strong Encryption for Windows Remote Desktop Required'
  describe registry_key('HKLM\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services') do
    it { should exist }
    its('MinEncryptionLevel') { should eq 3 }
  end
end

control 'sshd-11' do
  impact 0.6
  title 'Server: Set protocol version to SSHv2'
  desc "Set the SSH protocol version to 2.
        Don't use legacy insecure SSHv1 connections anymore."
  tag 'ssh', 'sshd', 'openssh-server'
  tag cce: 'CCE-27072-8'
  tag remediation: 'https://supermarket.chef.io/cookbooks/ssh-hardening'
  ref 'NSA-RH6-STIG-3.5.2.1', url: 'https://www.nsa.gov/ia/_files/os/redhat'
  describe sshd_config do
    its('Protocol') { should eq '2' }
  end
end
```

CHEF

60+ resources: github.com / chef / inspec / lib / resources

apache, apache_conf, apt, audit_policy, auditd_conf, auditd_rules, bash, bond, bridge, **command**, **csv**, **directory**, etc_group, **file**, gem, groups, grub_conf, host, iis_site, inetd_conf, ini, interface, iptables, **json**, kernel_module, kernel_parameter, limits_conf, login_def, mount, mssql_session, **mysql**, mysql_conf, mysql_session, npm, ntp_conf, oneget, os, os_env, **package**, parse_config, passwd, pip, **port**, postgres, postgres_conf, postgres_session, **powershell**, **processes**, **registry_key**, security_policy, **service**, shadow, ssh_conf, **ssl**, sys_info, **users**, vbscript, windows_feature, wmi, xinetd, **yaml**, yum

Community profiles and custom resources

https://supermarket.chef.io/tools?type=compliance_profile

```
$ inspec supermarket profiles
== Available profiles:
 * dev-sec/cis-docker-benchmark
 * dev-sec/nginx-hardening
...

$ inspec supermarket info dev-sec/cis-docker-benchmark
name:    cis-docker-benchmark
owner:   dev-sec
url:     https://github.com/dev-sec/cis-docker-benchmark
description: This InSpec compliance profile implements the CIS Docker 1.11.0
Benchmark in an automated way to provide security best-practice tests around
Docker daemon and containers in a production environment.
```

InSpec 1.0

- Integration with Chef Automate
- Dependency management(overlay profiles and custom resource packs)
- Performance and Auth improvements for Windows, including Nano
- Profile Attributes
- Ability to write and test controls in InSpec shell
- Lots of bug fixes and improvements for the core resources
- Supported by docs and demo at http://inspec.io

CHEF

Install and documentation

- Ruby gem (Included in Chef Development Kit)
- msi / dmg / rpm / deb package @ downloads.chef.io/inspec

- http://inspec.io/docs
- https://github.com/chef/inspec / docs
- inspec> help resource

CHEF

Integrations

CHEF AUTOMATE

CHEF COMPLIANCE

Compliance as Code (*if it hurts...*)

CHEF

All Servers

All Orgs

Node State >

Converge Status Compliance Status

chef-client.solo | **Scan ID:** 45b3be22-ae1b-435e-9138-98a1057b678f | view scan results

This node is uncompliant. Too many Critical and Major scored tests failed in the scan for this node to be deemed compliant. You can view the scan results below or view JSON scan results.

SCAN DURATION
2:31 PM - 2:32 PM

SCAN TIME
a few seconds

SCAN INITIATOR
Scheduled

INSPEC VERSION
1.0.0.beta2

PROFILES SCANNED
1

PLATFORM(S)

TOTAL CONTROLS	CRITICAL CONTROLS	MAJOR CONTROLS	MINOR CONTROLS	COMPLIANT CONTROLS	SKIPPED CONTROLS
205	83	0	2	82	38

Status	Score	Control	Profile	Failed	Skipped	Passed	Details
	1	Create Separate Partition for /tmp		1	0	0	details
	1	Set nodev option for /tmp Partition		2	0	0	details

Scan History

9/23/2016

8 5 3 0

a few seconds
09/23/16 | 14:31:47 - 14:32:02

a few seconds
09/23/16 | 14:27:11 - 14:27:12

a few seconds
09/23/16 | 14:10:33 - 14:10:34

a few seconds
09/23/16 | 14:09:55 - 14:09:56

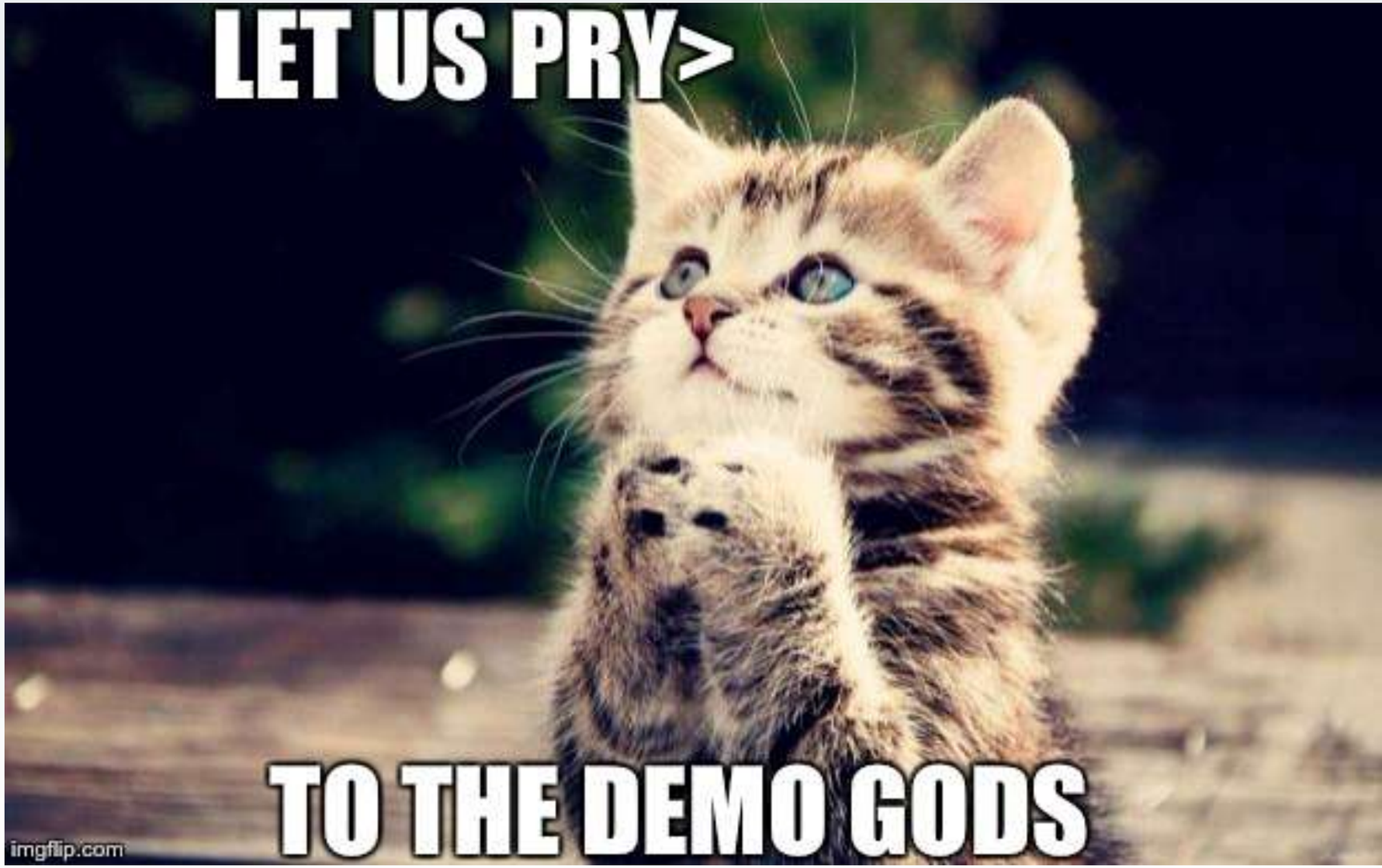
a few seconds
09/23/16 | 14:08:44 - 14:08:45

a few seconds
09/23/16 | 14:07:54 - 14:07:55

a few seconds
09/23/16 | 14:07:26 - 14:07:27

a few seconds
09/23/16 | 13:50:59 - 13:51:00

Demo time!



CHEF

Demo: Use the inspec shell on a remote target

```
$ inspec shell -t ssh://root@10.0.0.5 -i ~/mykey.pem
```

```
inspec> os.params
{:name=>"centos", :family=>"redhat", :release=>"6.6", :arch=>"x86_64"}
inspec> help file
Name: file

Description:
Use the file InSpec audit resource to test all system file types,...SNIP...

Example:

describe file('path') do
  it { should exist }
  it { should be_file }
  its('mode') { should cmp '0644' }
end
```

Demo: Use the file resource and test it in the shell

```
$ inspec shell -t ssh://root@10.0.0.5 -i ~/mykey.pem
```

```
inspec> file('/etc/hosts').exist?
=> true
inspec> file('/etc/hosts').size
=> 183
inspec> describe file('/etc/hosts') do
inspec>   its('size') { should be < 100 }
inspec> end


  File /etc/hosts
     ✗  size should be < 100
     expected: < 100
          got:   183
inspec> ls file('/etc/hosts') # see all available methods for the file resource
  basename  directory? exist?  group  link_path  mount_options  path
  sgid  source_path  type  sha256sum  sticky uid  md5sum  size  mode
```

Demo: Profile structure

```
$ inspec init profile mycerts
$ tree ~/tmp/mycerts
```

```
├── README.md
├── inspec.yml
├── controls
│   └── example.rb
└── libraries
```

Demo: Create and inherit ssl-certificate-profile

```
$ cat inspec.yml
```

```
name: mycerts
title: InSpec Profile
maintainer: The Authors
copyright: The Authors
copyright_email: you@example.com
license: All Rights Reserved
summary: An InSpec Compliance Profile
version: 0.1.0
depends:   # << added in order to use a resource defined in this profile
  - name: ssl-certificate-profile
    git: https://github.com/alexpop/ssl-certificate-profile
    version: '< 1.0.0'
```

Demo: Wrap(overlay) the inherited profile

```
$ cat controls/example.rb
```

```
# allows us to inherit all controls, but skip a few if needed
include_controls 'ssl-certificate-profile' do
  skip_control 'CHECK expired.badssl.com'
  # we can also override, controls, in this example changing the impact
  control 'CHECK sha1-2016.badssl.com' do
    impact 1.0
  end
end

# allows us to cherry pick the controls we want from the dependent profile
require_controls 'ssl-certificate-profile' do
  control 'CHECK github.com'
end
```

Demo: Use the resource from the dependent profile

```
$ cat controls/example.rb
```

```ruby
# example on how to define a profile attribute
val_path = attribute('path', default: '/etc/cert', description: 'Example attr...')


# use the ssl_certificate custom resource from the dependent profile
describe ssl_certificate(host: 'github.io') do
  it { should exist }
  it { should be_trusted }
  its('hash_algorithm') { should eq 'SHA256' }
  its('expiration_days') { should be >= 30 }
  its('key_size') { should be >= 2048 }
end
```

Demo: Execute the profile with cli and json formats

```
$ inspec exec ~/tmp/mycerts
```

```
 ✓  CHECK github.com: Verify github.com`s SSL certificate
     ✓  ssl_certificate github.com:443 should exist
     ✓  ssl_certificate github.com:443 should be trusted
     ✓  ssl_certificate github.com:443 expiration_days should be >= 30
...SNIP...
Profile Summary: 1 successful, 0 failures, 0 skipped
```

```
$ inspec exec ~/tmp/mycerts --format json | jq
```

```
{
  "version": "1.0.0",
  "profiles": [
    {
      "name": "mycerts",
      "title": "InSpec Profile",
      "version": "0.1.0",
```

Demo: Dissect the ssl_certificate custom resource

```
$ cat ssl-certificate-profile/libraries/ssl_certificate.rb
```

```ruby
class SslCertificate < Inspec.resource(1)
  name 'ssl_certificate'
  desc 'The `ssl_certificate` allows to test ...SNIP..'
  example 'describe ssl_certificate do ...SNIP...'
  def initialize(opts = {})
    # ... plain Ruby or other InSpec resources here like: inspec.file(path)
    @cert = http.peer_cert
  end
  # Called by: its('signature_algorithm') { should eq 'something' }
  def signature_algorithm
    return @cert.signature_algorithm
  end
  def expiration_days
    return ((@cert.not_after - Time.now) / 86_400).to_i
  end
end
```

Pry breakpoints (in profiles, resources, recipes etc)

```
$ inspec exec ~/tmp/mycerts
```

```
  def signature_algorithm
    require 'pry'
    binding.pry        # execution will stop here and give me a pry> shell
    return @cert.signature_algorithm
  end
  def expiration_days
    return ((@cert.not_after - Time.now) / 86_400).to_i
  end
end

[1] pry> @cert.signature_algorithm
=> "sha256WithRSAEncryption"
[2] pry> ls @cert
OpenSSL::X509::Certificate#methods:
  add_extension      extensions=  not_before    public_key   serial=   subject
  to_pem    verify   check_private_key     inspect      not_after   not_before
```

https://summit.chef.io/london

Presentations
Collaboration
Food & _ _ _ _

CHEF COMMUNITY SUMMIT

OCTOBER 12-13, 2016 | LONDON, UK

CHEF

Q&A

CHEF