What did you InSpec?

https://github.com/gdha/inspec-cfgmgmtcamp-ghent-2019

Gratien D'haese
IT3 Consultants

Who am I?

- Gratien D'haese
- IT3 Consultants (company)
  - > 30 years Unix experience
  - Unix/Linux Engineer (incl. DevOps)
  - Web: it3.be
- Relax-and-Recover (ReaR)
  - Linux disaster recovery framework
- Open Source pages: https://github.com/gdha

Bit of history

pre-historic times

Bit of history

power to the Ops

- system administrators = Ops
- powerful shell scripts used for:
  - Update
  - Control
  - Security
  - Monitor
- Battle between Ops and Devs
  - No CI/CD
  - Lots of Change Controls

Ops -> DevOps

- it’s everyone’s job now
- Ops tools for devs
- Software engineers (devs) learn ops
- Admins transition to devs

DevOps and the rest

- Developers want tests
- Operations want peace
- Compliance Officers want √
- Security Officers do not want holes

What is InSpec

- InSpec is an open-source testing framework provided by **Chef**
- Human-readable language for specifying compliance, security and policy requirements
- Extensible language
- Re-usable
- Command-line
- Integrates with Test Kitchen

What InSpec is **not**

- Is not a capacity planning tool
- Is not a monitoring tool
- Is not a logging tool
- Is not a configuration management tool
- Is not a firewall tool
- Is not a intrusion detection tool

Why using InSpec?

- Less scripts for verification required
- One Language for many platforms
- Easy to read
- Easy to hand-over
- Easy to share
- Big collection of ready to use profiles
- Excellent documentation
- No need to be a nerd

Features of InSpec

- Supports many Operating Systems
  - Linux
  - Mac/OS
  - BSD, Solaris, AIX, HP-UX
  - Windows
- Supports many Hypervisors, VMs, bare-metal
- Support different Cloud Providers
- Supports docker
- Supports DBs

Why should you care?

- Do you want to be the next?
  New Data Breach exposes 57 million records
  https://blog.hackenproof.com/industry-news/new-data-breach-exposes-57-million-records/
- Protection of your assets – data (security)
- IQ/OQ Compliance
- System validation after major changes
- CI/CD integration checks

Idiot proof

```
# inspec exec https://github.com/lnxchk/inspec-
profile-wannacry-exploit/archive/master.tar.gz -i
./insecure_keys/vagrant.private -t
ssh://root@server

Profile: WannaCry Exploit Mitigation Status
(wannacry-exploit)
Version: 0.2.0
Target:  ssh://root@server:22

  ×  WannaCry Vulnerability Check: Hot-fix mitigation
check for WannaCry Ransomware vulnerability (23
failed)
```

Can you guess what is wrong with above test?

Wannacry on Windows

```
# inspec exec https://github.com/lnxchk/inspec-
profile-wannacry-exploit/archive/master.tar.gz -t
winrm://administrator@10.180.4.12 --password xxxx
```

```
Profile: WannaCry Exploit Mitigation Status (wannacry-exploit)
Version: 0.2.0
Target:
winrm://administrator@http://10.180.4.12:5985/wsman:3389

  ✔  WannaCry Vulnerability Check: Hot-fix mitigation check for
WannaCry Ransomware vulnerability
     ✔  WMI with {:class=>"win32_quickfixengineering",
:filter=>"HotFixID = 'KB4012213'"} InstalledOn should not eq
nil

Profile Summary: 1 successful control, 0 control failures, 0
controls skipped
Test Summary: 1 successful, 0 failures, 0 skipped
```

InSpec Basics

- Start with a demo – mychefdk container
- Launch the container and use inspec to check for my account
  - Check inside the container
  - Check from outside the container
  - Run cookbook myaccount inside the container
  - Re-run the checks again

InSpec Basics (continued)

- Download from https://www.inspec.io/
- Open Source at GitHub: https://github.com/inspec/inspec
- **Resources**
  - InSpec uses built-in resources for common services, system files and configurations https://www.inspec.io/docs/reference/resources/
  - Resources work on many Linux platforms, and also on Windows

InSpec resources

- OS resources
  - apache
  - bond
  - command
  - directory
  - docker
  - etc_fstab
  - group
  - mssql_session
  - and so on

- Cloud resources
  - AWS
  - Azure
  - Google

```
unless os.windows?
  # This is an example test, replace with your own test.
  describe user('root') do
    it { should exist }
  end

  describe user('gdha') do
    it { should exist }
    its('uid') { should eq 501 }
    its('group') { should eq 'users'}
    its('home') { should eq '/home/gdha' }
    its('shell') { should eq '/bin/bash' }
  end
end

describe port(80) do
  it { should_not be_listening }
end
```

Resource

Matcher

Matchers

- should exist
- should be_in
- should_not match /blabla/
- should eq
- should_not eq
- should cmp
- https://www.inspec.io/docs/reference/matchers/

Profiles

- Profiles is about sharing and caring
- Built around “controls” that can be reviewed
- Each profile can have multiple tests
- Include profiles from outside this test
- Profiles can be published to be re-used
- More at https://www.inspec.io/docs/reference/profiles/

Profiles (continued)

```
$ inspec init profile dockerprofile
Create new profile at
/Users/gdha/data/projects/inspec/dockerprofile
 • Create directory libraries
 • Create file README.md
 • Create directory controls
 • Create file controls/example.rb
 • Create file inspec.yml
 • Create file libraries/.gitkeep
```

Add more tests under this directory

Inspec shell

```
root@c26e2f2d7904:/# inspec shell
Welcome to the interactive InSpec Shell
To find out how to use it, type: help

You are currently running on:

    Name:      ubuntu
    Families:  debian, linux, unix, os
    Release:   18.04
    Arch:      x86_64

inspec> help
inspec> command('uname -s').stdout
=> "Linux\n"
```

Inspec shell (continued)

```
inspec> describe file('/etc/gshadow') do
inspec>   it { should be_owned_by 'root' }
inspec> end

Profile: inspec-shell
Version: (not specified)

  File /etc/gshadow
     ✔  should be owned by "root"

Test Summary: 1 successful, 0 failures, 0 skipped
```

Example: source control file

```
$ cat inspec-path-check/controls/path.rb

title 'DOT in PATH variable'

control 'path-1.0' do                 # A unique ID for this control
  impact 1.0                        # The criticality, if this control fails.
  title 'DOT in PATH variable'
  desc 'An optional description...'
  describe os_env('PATH') do       # The actual test
    its('split') { should_not include('') }
    its('split') { should_not include('.') }
  end
end
```

Profiles

- InSpec profiles allow you to share and pack sets of tests
- Built around controls (see previous example)
- Profiles can have multiple tests
- May depend on external profiles
- Publishing of your profiles is possible
- *inspec init profile <profile-name>*
- *inspec check <profile-name>*

Running inspec

- Inspec on command line
- Can run locally on this machine
  - inspec exec profile-name
- Run remotely via target option
  - inspec exec profile-name -i pub.key -t ssh://user@system
  - inspec exec profile-name -t winrm://administrator@system --password secret
- Run via test kitchen

Excute a local path

```
$ inspec exec inspec-path-check
$ inspec exec /full/path/to/inspec-path-check
```

```
Profile: PATH check InSpec Profile (path-check)
Version: 0.1.0
Target:  local://

  ✔  path-1.0: DOT in PATH variable
     ✔  Environment variable PATH split should not include ""
     ✔  Environment variable PATH split should not include "."


Profile Summary: 1 successful control, 0 control failures, 0
controls skipped
Test Summary: 2 successful, 0 failures, 0 skipped
```

Execute a GIT repo

```
$ inspec exec https://github.com/gdha/inspec-path-check
[2019-01-16T18:10:26+01:00] WARN: URL target
https://github.com/gdha/inspec-path-check transformed to
https://github.com/gdha/inspec-path-
check/archive/master.tar.gz. Consider using the git fetcher

Profile: PATH check InSpec Profile (path-check)
Version: 0.1.0
Target:  local://

  ✔  path-1.0: DOT in PATH variable
     ✔  Environment variable PATH split should not include ""
     ✔  Environment variable PATH split should not include "."


Profile Summary: 1 successful control, 0 control failures, 0
controls skipped
Test Summary: 2 successful, 0 failures, 0 skipped
```

Execute in a docker container

```
$ inspec exec -t docker://c26e2f2d7904 inspec-path-check

Profile: PATH check InSpec Profile (path-check)
Version: 0.1.0
Target:
docker://c26e2f2d79041252b2646baea3d64f18f52eea9b45a2443f3325a9
4221e10a4e

  ✔  path-1.0: DOT in PATH variable
     ✔  Environment variable PATH split should not include ""
     ✔  Environment variable PATH split should not include "."


Profile Summary: 1 successful control, 0 control failures, 0
controls skipped
Test Summary: 2 successful, 0 failures, 0 skipped
```

Execute inspec remotely

```
$ inspec exec -t ssh://client --password vagrant
../path-check/

Profile: PATH check InSpec Profile (path-check)
Version: 0.1.0
Target:  ssh://root@client:22

  ✔  path-1.0: DOT in PATH variable
     ✔  Environment variable PATH split should not include ""
     ✔  Environment variable PATH split should not include "."


Profile Summary: 1 successful control, 0 control failures, 0
controls skipped
Test Summary: 2 successful, 0 failures, 0 skipped
```

```
$ inspec exec -t winrm://admin@windows --password
xx ../patch-check
```

Using InSpec with Test Kitchen

```
driver:
  name: vagrant

provisioner:
  name: chef_zero

verifier:
  name: inspec

platforms:
  - name: centos-7.6

suites:
  - name: default
    run_list:
      - recipe[nginx_test::default]
    verifier:
      inspec_tests:
        - test/integration/default
```

Kitchen verify

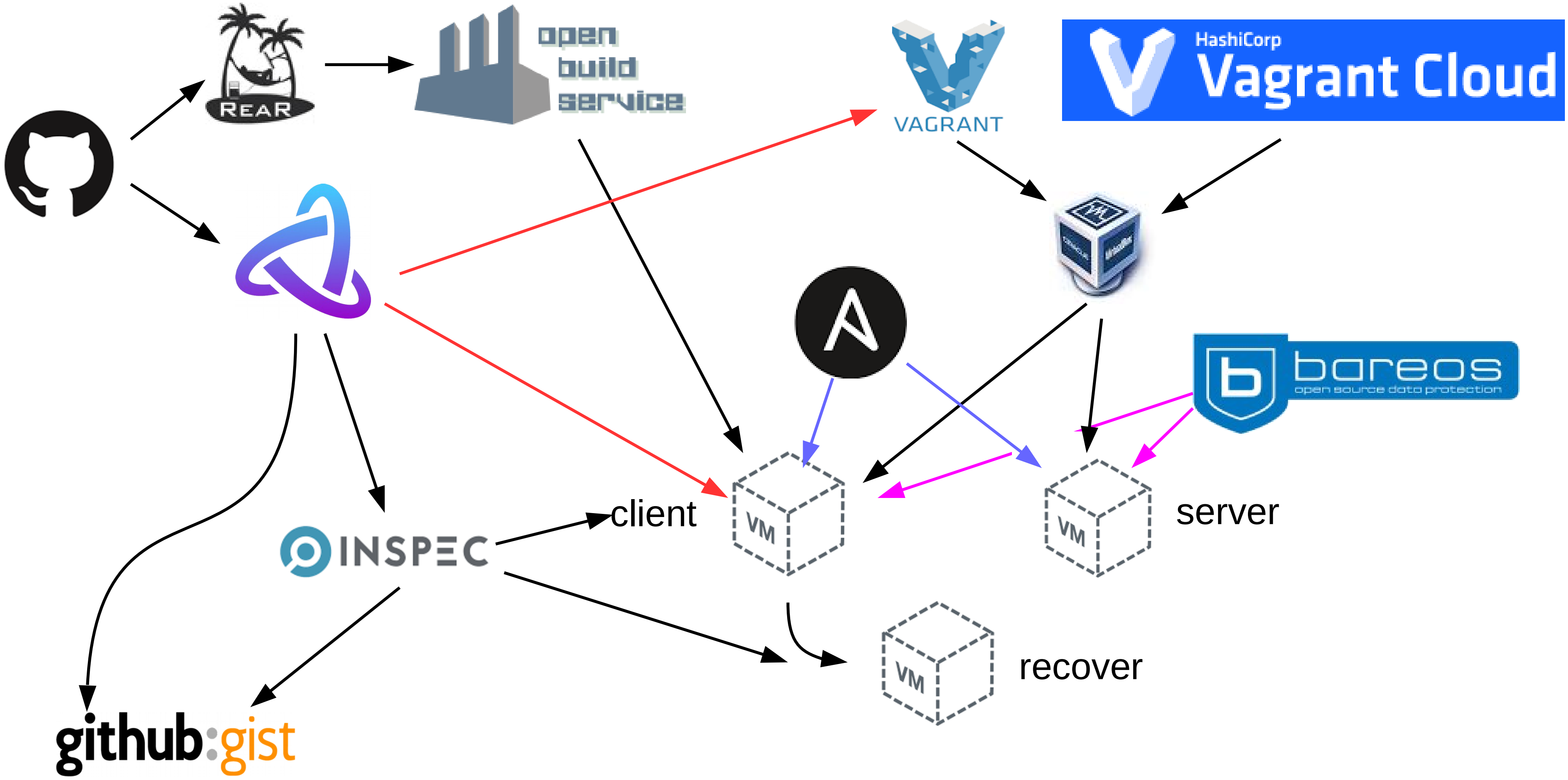
```
$ kitchen verify
-----> Starting Kitchen (v1.24.0)
-----> Verifying <default-centos-76>...
       Loaded tests from
{:path=>".Users.gdha.data.projects.inspec.inspec-cfgmgmtcamp-
ghent-2019.cookbooks.nginx_test.test.integration.default"}

  User root
     ✔  should exist
  Port 80
     ✔  should be listening
  System Package nginx
     ✔  should be installed
  File /etc/nginx/sites-available/default
     ✔  should exist
  Command: `curl localhost`
     ✔  stdout should match "Welcome"

Test Summary: 5 successful, 0 failures, 0 skipped
       Finished verifying <default-centos-76> (0m0.81s).
-----> Kitchen is finished. (0m7.83s)
```

DevSec Linux Security Baseline (linux-baseline)

```
# docker ps
CONTAINER ID        IMAGE                       COMMAND
1e2ef5665f9f        openshift/base-centos7      ...

# inspec exec https://github.com/dev-sec/linux-baseline -t
docker://1e2ef5665f9f

  ✔  os-01: Trusted hosts login
     ✔  File /etc/hosts.equiv should not exist
  ✔  os-02: Check owner and permissions for /etc/shadow
     ✔  File /etc/shadow should exist
     ✔  File /etc/shadow should be file

Profile Summary: 14 successful controls, 3 control
failures, 37 controls skipped
Test Summary: 53 successful, 8 failures, 37 skipped
```

More details at https://dev-sec.io/

ReaR Automated Testing



-Sylvia Tops-2018-

INSPEC

Links

- https://github.com/gdha/inspec-cfgmgmtcamp-ghent-2019
- https://github.com/inspec/inspec
- https://www.inspec.io/
- https://www.inspec.io/docs/reference/resources/
- http://www.it3.be/
- https://gdha.github.io/rear-automated-testing/
- mailto:gratien.dhaese@it3.be