InSpec CLI

Command Line Interface

Objectives

Create and execute a simple compliance check

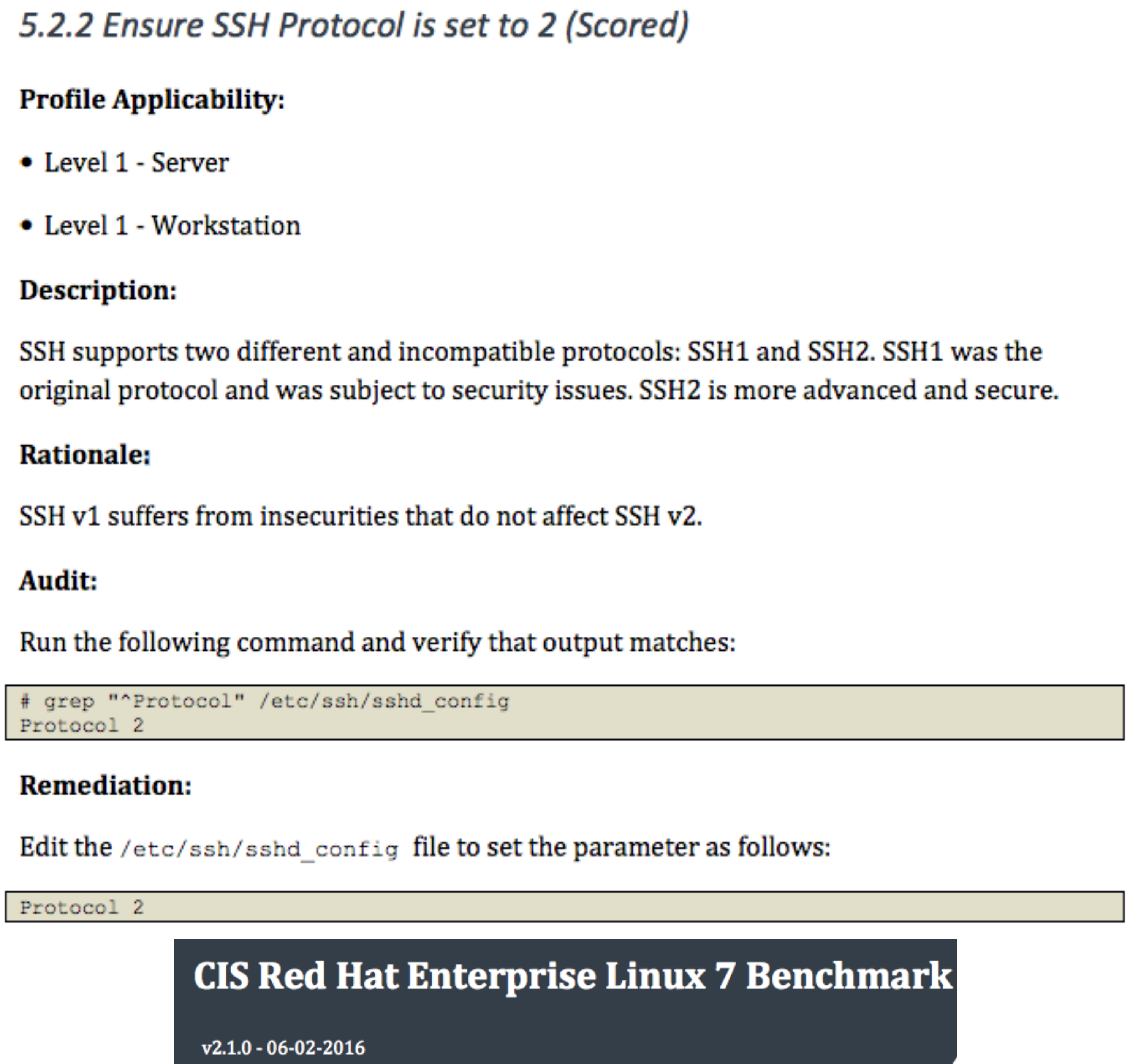
LAB

Ensure SSH Protocol is set to 2

- □ Review the Center for Internet Security control
- □ Create an InSpec profile to verify the control
- □ Execute the InSpec profile to determine current system state

Compliance Mandate

Center for Internet Security



5.2.2 Ensure SSH Protocol is set to 2 (Scored)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

SSH supports two different and incompatible protocols: SSH1 and SSH2. SSH1 was the original protocol and was subject to security issues. SSH2 is more advanced and secure.

Rationale:

SSH v1 suffers from insecurities that do not affect SSH v2.

Audit:

Run the following command and verify that output matches:

```
# grep "^Protocol" /etc/ssh/sshd_config
Protocol 2
```

Remediation:

Edit the `/etc/ssh/sshd_config` file to set the parameter as follows:

```
Protocol 2
```

CIS Red Hat Enterprise Linux 7 Benchmark

v2.1.0 - 06-02-2016

Run InSpec

```
$ inspec
```

```
Commands:
  inspec archive PATH                # archive a profile to tar.gz (default) or zip
  inspec artifact SUBCOMMAND ...     # Sign, verify and install artifacts
  inspec check PATH                  # verify all tests at the specified PATH
  inspec compliance SUBCOMMAND ...   # Chef Compliance commands
  inspec detect                      # detect the target OS
  inspec env                         # Output shell-appropriate completion configuration
  inspec exec PATHS                  # run all test files at the specified PATH.
  inspec help [COMMAND]              # Describe available commands or one specific command
  inspec init TEMPLATE ...           # Scaffolds a new project
  inspec json PATH                   # read all tests in PATH and generate a JSON summary
  inspec shell                       # open an interactive debugging shell
  inspec supermarket SUBCOMMAND ...  # Supermarket commands
  inspec vendor PATH                 # Download all dependencies and generate a lockfile in a `vendor` directory
  inspec version                     # prints the version of this tool

Options:
  l, [--log-level=LOG_LEVEL]         # Set the log level: info (default), debug, warn, error
      [--log-location=LOG_LOCATION]  # Location to send diagnostic log messages to. (default: STDOUT or STDERR)
      [--diagnose], [--no-diagnose]  # Show diagnostics (versions, configurations)
```

DISCUSSION

Questions

What version of InSpec is installed?
What type(s) of projects with inspec init generate?

Build scaffold for an ssh profile

```
$ inspec init profile ssh
```

```
Create new profile at /home/ec2-user/ssh
 * Create file README.md
 * Create directory controls
 * Create file controls/example.rb
 * Create file inspec.yml
 * Create directory libraries
```

Open the example control

/home/ec2-user/ssh/controls/example.rb

```
# encoding: utf-8
# copyright: 2015, The Authors
# license: All rights reserved

title 'sample section'

# you can also use plain tests
describe file('/tmp') do
  it { should be_directory }
end

# you add controls here
control 'tmp-1.0' do                        # A unique ID for this control
  impact 0.7                                # The criticality, if this control fails.
  title 'Create /tmp directory'             # A human-readable title
  desc 'An optional description...'
  describe file('/tmp') do                  # The actual test
    it { should be_directory }
  end
end
```

Execute the example control

```
$ inspec exec ssh
```

```
Profile: InSpec Profile (ssh)
Version: 0.1.0
Target:  local://


  ✔  tmp-1.0: Create /tmp directory
     ✔  File /tmp should be directory

  File /tmp
     ✔  should be directory

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

Rename the example control

```
$ mv ~/ssh/controls/example.rb ~/ssh/controls/server.rb
```

Rewrite the control

```
/home/ec2-user/ssh/controls/example.rb
```

```ruby
# 5.2.2 Ensure SSH Protocol is set to 2
#
# grep "^Protocol" /etc/ssh/sshd_config
# Protocol 2
#
describe file('/etc/ssh/sshd_config') do
  its('content') { should match /^Protocol 2/ }
end
```

Execute the control

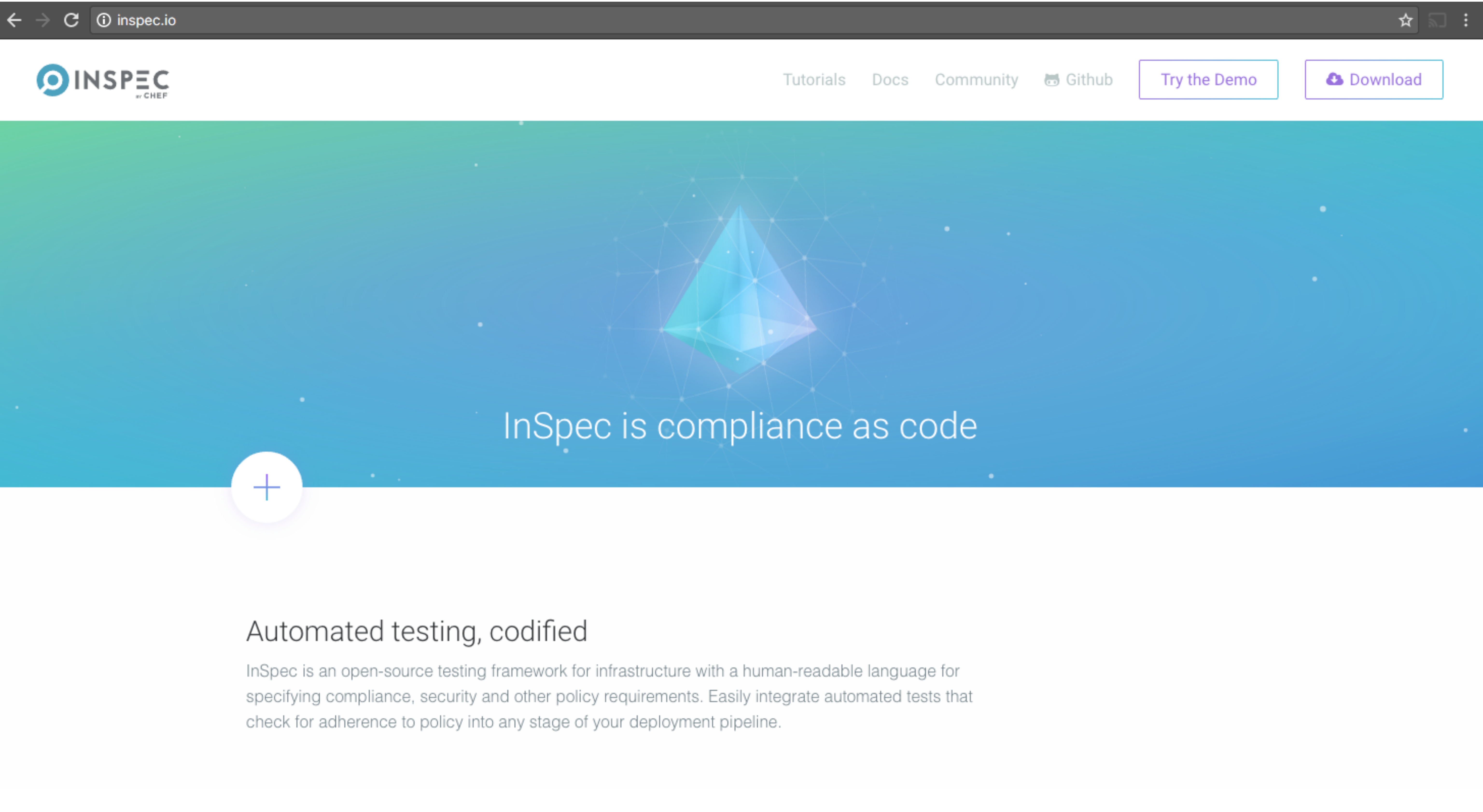
```
$ inspec exec ssh
```

```
Profile: InSpec Profile (ssh)
Version: 0.1.0
Target:  local://



  File /etc/ssh/sshd_config
     ∅  content should match /^Protocol 2/
     expected "# This config file was generated by Chef\n\n#       $OpenBSD: sshd_config,v 1.93 2014/01/10 05:59:19...XMODIFIERS\n\n# override default of no subsystems\nSubsystem sftp  /usr/libexec/openssh/sftp-server" to match /^Protocol 2/
     Diff:
     @@ -1,2 +1,78 @@
     -/^Protocol 2/
     +# This config file was generated by Chef
...
 +Subsystem sftp  /usr/libexec/openssh/sftp-server


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

DISCUSSION

Wait a minute...

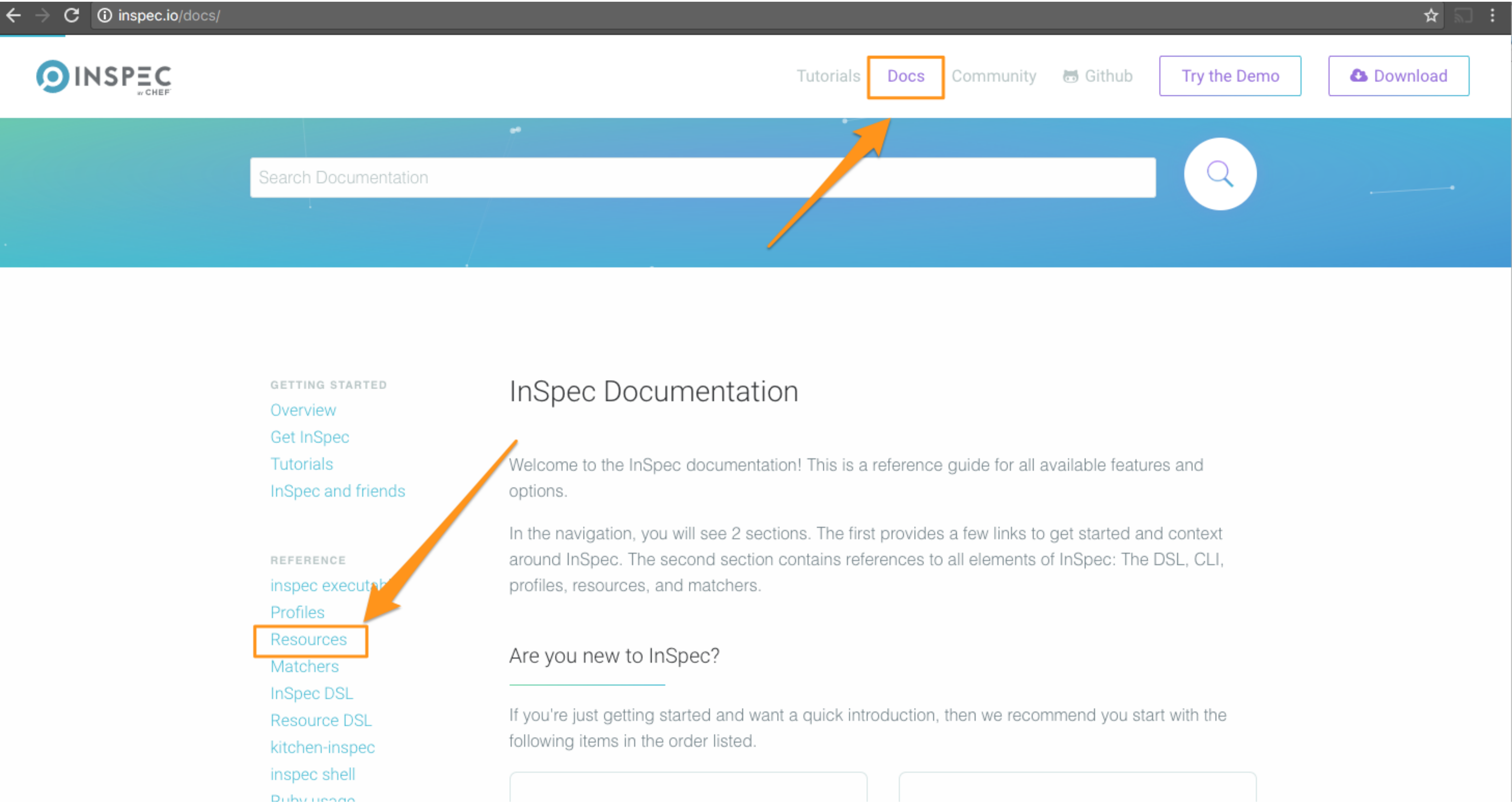
Where did `its('content')` come from?
What other file attributes can we write tests for?
Where does one go to find out more information about these resources?



InSpec is compliance as code

Automated testing, codified

InSpec is an open-source testing framework for infrastructure with a human-readable language for specifying compliance, security and other policy requirements. Easily integrate automated tests that check for adherence to policy into any stage of your deployment pipeline.



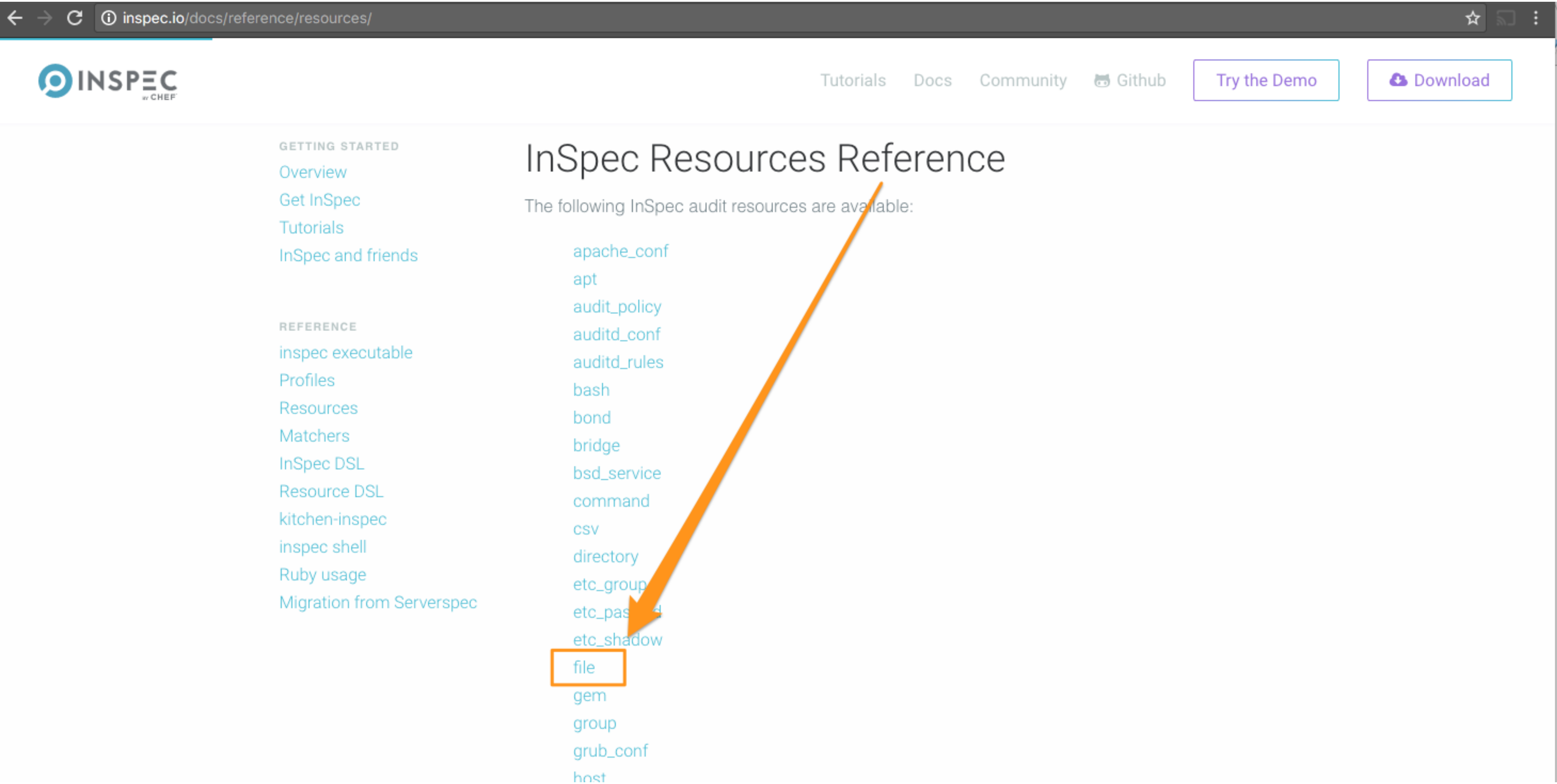
InSpec Documentation

Welcome to the InSpec documentation! This is a reference guide for all available features and options.

In the navigation, you will see 2 sections. The first provides a few links to get started and context around InSpec. The second section contains references to all elements of InSpec: The DSL, CLI, profiles, resources, and matchers.

Are you new to InSpec?

If you're just getting started and want a quick introduction, then we recommend you start with the following items in the order listed.



GETTING STARTED

- Overview
- Get InSpec
- Tutorials
- InSpec and friends

REFERENCE

- inspec executable
- Profiles
- Resources
- Matchers
- InSpec DSL
- Resource DSL
- kitchen-inspec
- inspec shell
- Ruby usage
- Migration from Serverspec

InSpec Resources Reference

The following InSpec audit resources are available:

- apache_conf
- apt
- audit_policy
- auditd_conf
- auditd_rules
- bash
- bond
- bridge
- bsd_service
- command
- csv
- directory
- etc_group
- etc_passwd
- etc_shadow
- file
- gem
- group
- grub_conf
- host

file

Use the `file` InSpec audit resource to test all system file types, including files, directories, symbolic links, named pipes, sockets, character devices, block devices, and doors.

Syntax

A `file` resource block declares the location of the file type to be tested, what type that file should be (if required), and then one (or more) matchers:

```
 describe file('path') do
  it { should MATCHER 'value' }
end
```

where

`('path')` is the name of the file and/or the path to the file
`MATCHER` is a valid matcher for this resource
`'value'` is the value to be tested

Matchers

This InSpec audit resource has the following matchers:

be

Use the `be` matcher to use a comparison operator—`=` (equal to), `>` (greater than), `<` (less than), `>=` (greater than or equal to), and `<=` (less than or equal to)—to compare two values: `its('value') { should be >= value }`, `its('value') { should be < value }`, and so on.

be_block_device

File Resource

```
describe file('path') do
  it { should MATCHER 'value' }
end
```

Use the file resource to test all system file types, including files, directories, symbolic links, named pipes, sockets, character devices, block devices, and doors.

Test if a file exists

```
describe file('/tmp') do
 it { should exist }
end
```

Test if a path is a directory

```
describe file('/tmp') do
 its('type') { should eq :directory }
 it { should be_directory }
end
```

Content Matcher

```
describe file('/etc/ssh/sshd_config') do
  its('content') { should match /^Protocol 2/ }
end
```

The content matcher tests if contents in the file match the value specified in a regular expression. The values of the content matcher are arbitrary and depend on the file type being tested and also the type of information that is expected to be in that file

LAB

Ensure SSH Protocol is set to 2

- ✓ Review the Center for Internet Security control
- ✓ Create an InSpec profile to verify the control
- ✓ Execute the InSpec profile to determine current system state

PROBLEM

There are some problems!

Location of the SSH server configuration is hard-coded
Regular expressions are difficult

LAB

Refactor our control

- ❑ Use a different resource

DISCUSSION

Which resource

Is there a better resource that we could use?
What might a refactored test look like?

Refactored Control

/home/ssh/controls/server.rb

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

Resource: sshd_config

```
describe sshd_config('path') do
  its('name') { should include('foo') }
end
```

where

`name` is a configuration setting in sshd_config

`('path')` is the non-default /path/to/sshd_config

`{ should include('foo') }` tests the value of name as read from sshd_config versus the value declared in the test

Use the sshd_config resource to test configuration data for the OpenSSH daemon located at /etc/ssh/sshd_config on Linux and Unix platforms.

Execute the control

```
$ inspec exec ssh
```

```
 SSH Configuration
     ∅  Protocol should cmp == 2

     expected: 2
          got:

     (compared using `cmp` matcher)



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

Refactor our control

- ✓ Use a different resource

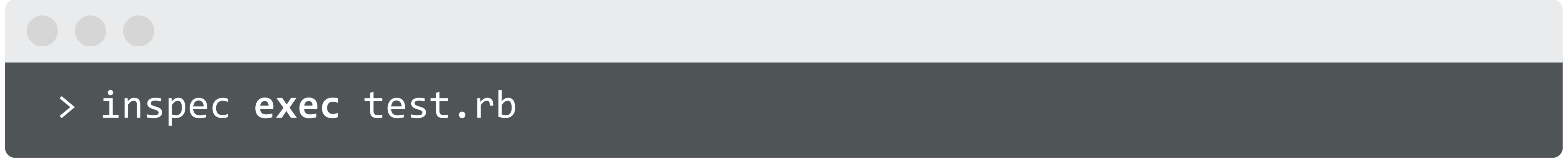
LAB

Execute profile on a remote machine

- ❑ Execute your ssh profile against the instructor’s machine

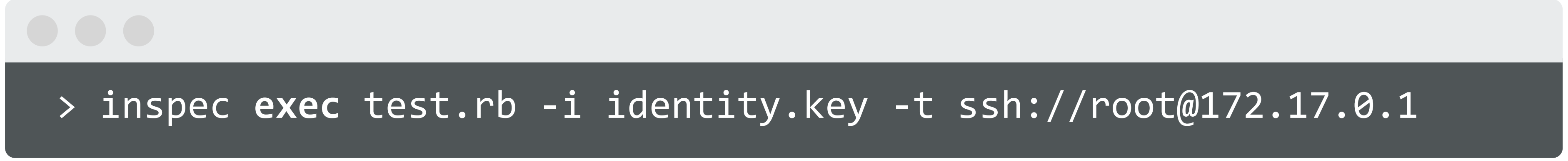
Different ways to run InSpec

Test your machine locally



```
> inspec exec test.rb
```

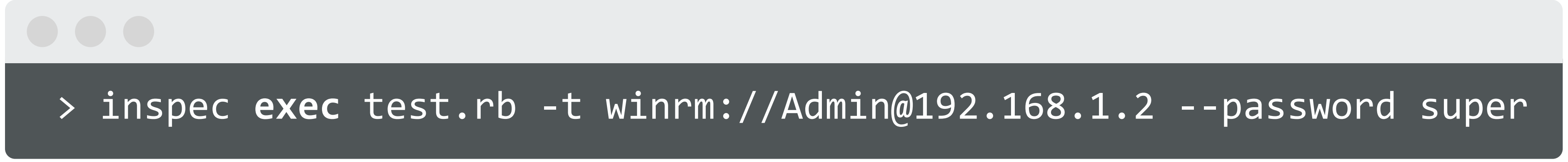
Test a machine remotely via SSH



```
> inspec exec test.rb -i identity.key -t ssh://root@172.17.0.1
```

No ruby/agent on the node

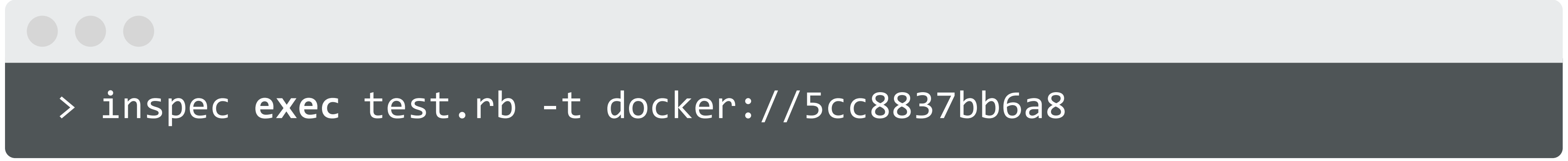
Test a machine remotely via WinRM



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

No ruby/agent on the node

Test Docker Container



```
> inspec exec test.rb -t docker://5cc8837bb6a8
```

no SSH/agent in the container

Execute the control

```
$ inspec exec ssh -t ssh://ec2-35-156-226-39.eu-central-1.compute.amazonaws.com --user=chef --password=chef
```

```
 SSH Configuration
     ∅  Protocol should cmp == 2

     expected: 2
          got:

     (compared using `cmp` matcher)



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

LAB

Execute profile on a remote machine

- ✓ Execute your ssh profile against the instructor’s machine

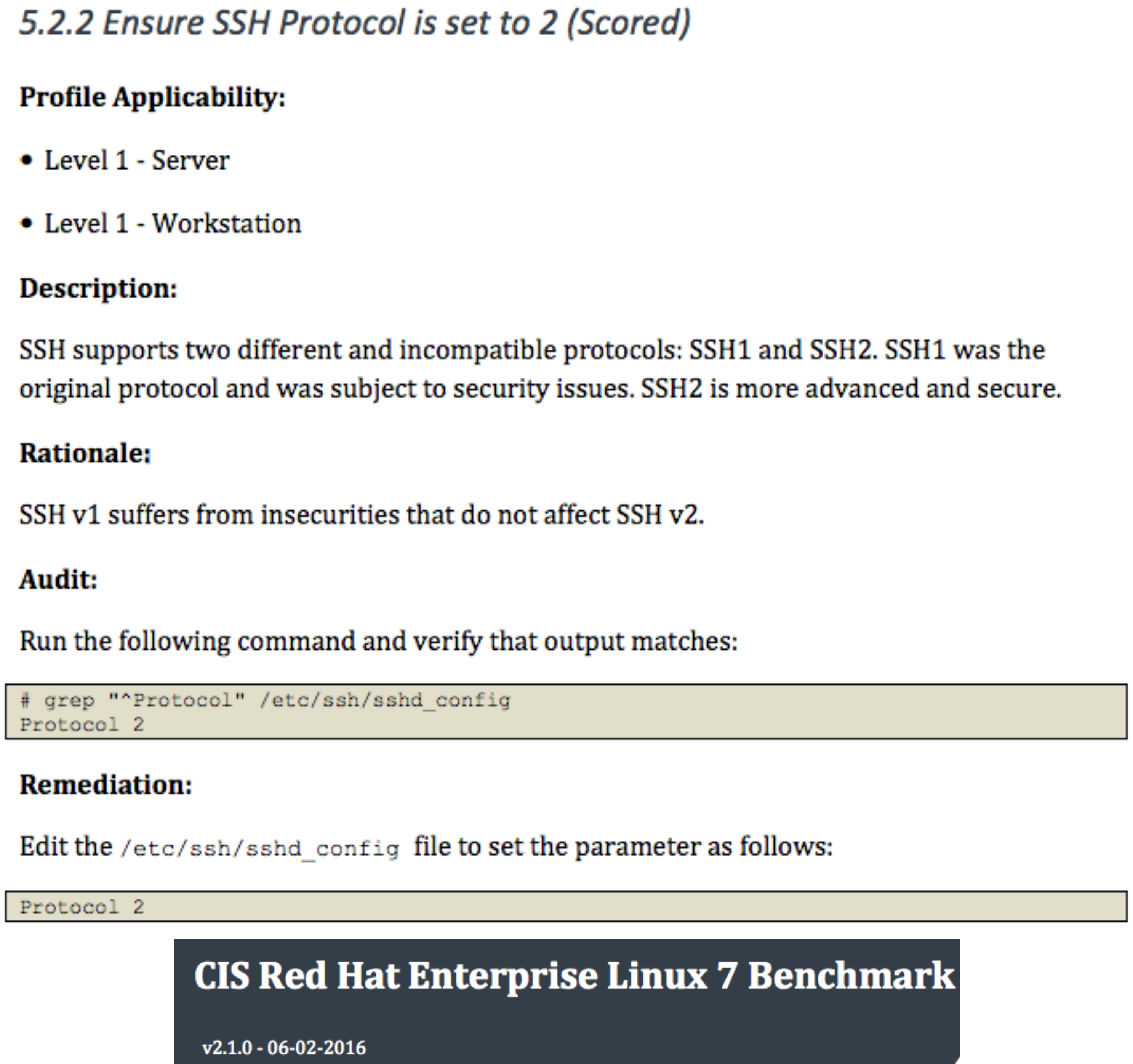
LAB

Enrich our profile

- ❑ Add additional metadata to our control

Compliance Mandate

Center for Internet Security



5.2.2 Ensure SSH Protocol is set to 2 (Scored)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

SSH supports two different and incompatible protocols: SSH1 and SSH2. SSH1 was the original protocol and was subject to security issues. SSH2 is more advanced and secure.

Rationale:

SSH v1 suffers from insecurities that do not affect SSH v2.

Audit:

Run the following command and verify that output matches:

```
# grep "^Protocol" /etc/ssh/sshd_config
Protocol 2
```

Remediation:

Edit the `/etc/ssh/sshd_config` file to set the parameter as follows:

```
Protocol 2
```

CIS Red Hat Enterprise Linux 7 Benchmark

v2.1.0 - 06-02-2016

Server Control

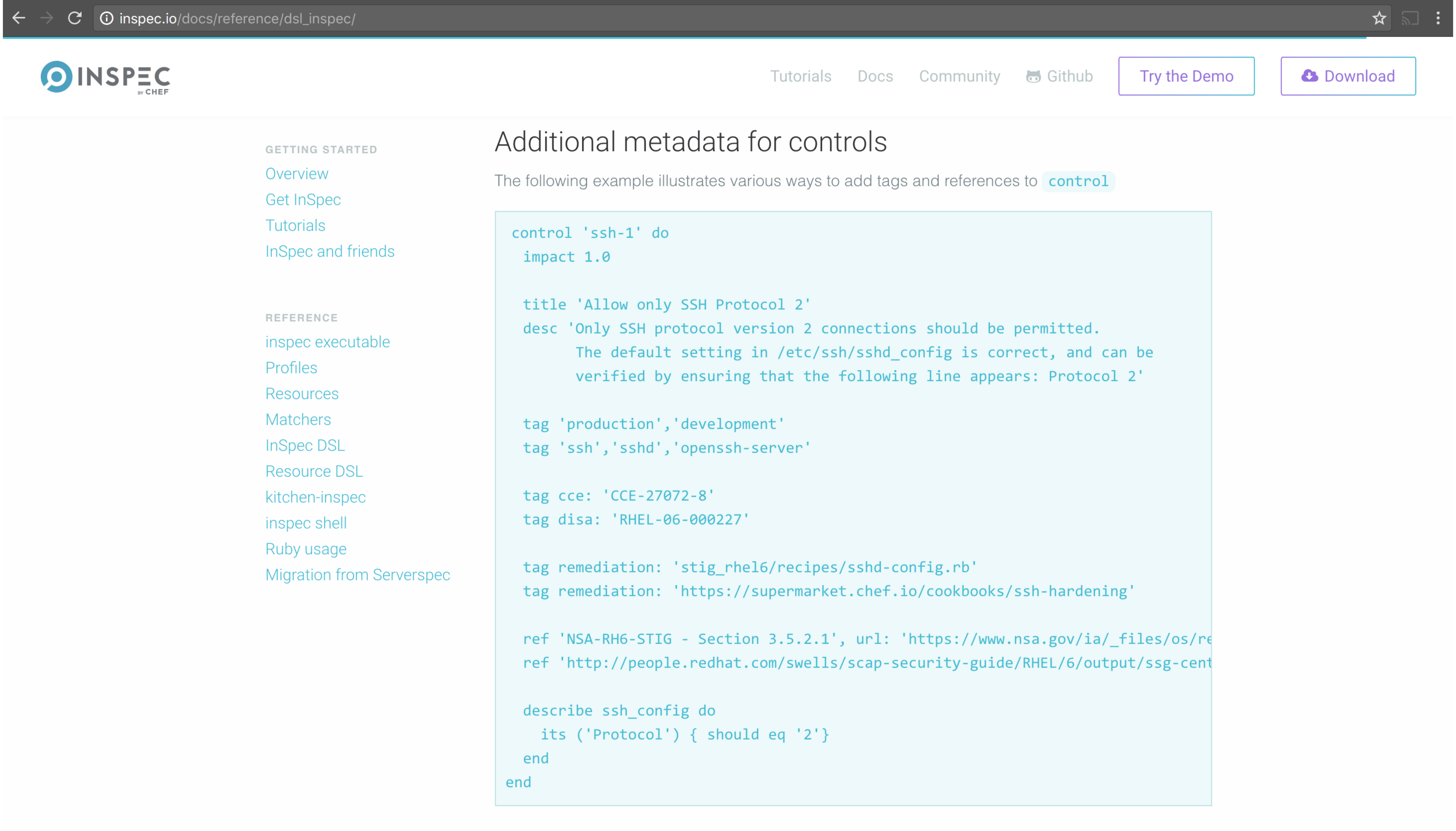
/home/ssh/controls/server.rb

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

Enriched Server Control

/home/ssh/controls/server.rb

```
control '5.2.2' do
    impact 1.0

    title 'Ensure SSH Protocol is set to 2'

    desc <<-EOF
      SSH supports two different and incompatible protocols: SSH1 and SSH2.
      SSH1 was the original protocol and was subject to security issues.
      SSH2 is more advanced and secure.

      SSH v1 suffers from insecurities that do not affect SSH v2.
    EOF

    tag 'ssh', 'sshd', 'server', 'workstation'

    ref 'SSH Protocol', url: 'https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide...'

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



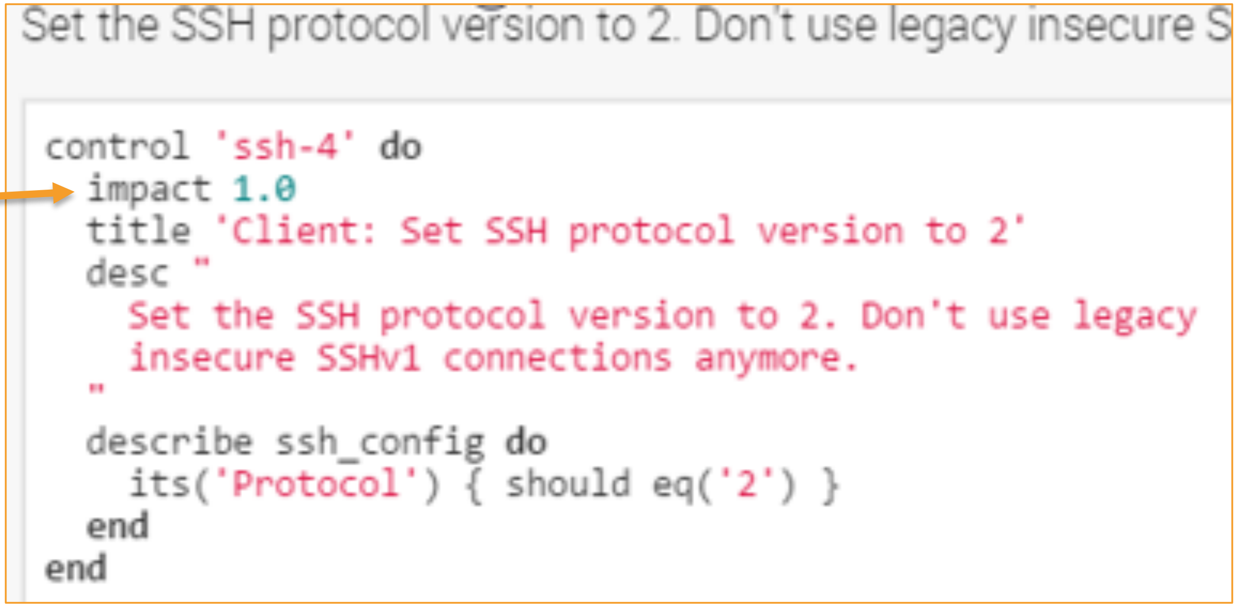
Additional metadata for controls

The following example illustrates various ways to add tags and references to `control`

```
control 'ssh-1' do
  impact 1.0

  title 'Allow only SSH Protocol 2'
  desc 'Only SSH protocol version 2 connections should be permitted.
        The default setting in /etc/ssh/sshd_config is correct, and can be
        verified by ensuring that the following line appears: Protocol 2'

  tag 'production','development'
  tag 'ssh','sshd','openssh-server'

  tag cce: 'CCE-27072-8'
  tag disa: 'RHEL-06-000227'

  tag remediation: 'stig_rhel6/recipes/sshd-config.rb'
  tag remediation: 'https://supermarket.chef.io/cookbooks/ssh-hardening'

  ref 'NSA-RH6-STIG - Section 3.5.2.1', url: 'https://www.nsa.gov/ia/_files/os/re
  ref 'http://people.redhat.com/swells/scap-security-guide/RHEL/6/output/ssg-cent

  describe ssh_config do
    its ('Protocol') { should eq '2'}
  end
end
```

Compliance Profile Severity Mapping

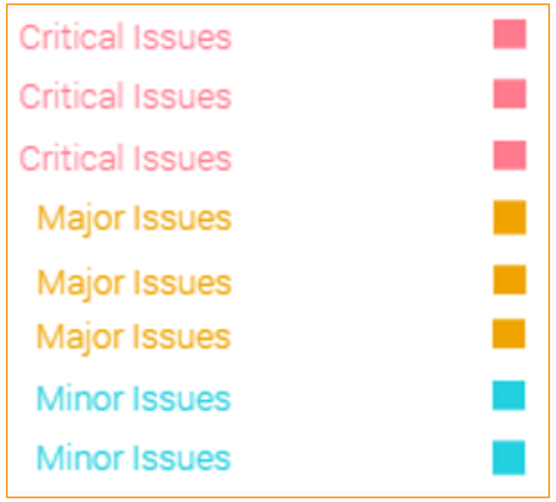
The table below shows the current mapping of Compliance Profile **impact** numbering to severity.



Set the SSH protocol version to 2. Don't use legacy insecure S

```
control 'ssh-4' do
  impact 1.0
  title 'Client: Set SSH protocol version to 2'
  desc "
    Set the SSH protocol version to 2. Don't use legacy
    insecure SSHv1 connections anymore.
  "

  describe ssh_config do
    its('Protocol') { should eq('2') }
  end
end
```

Impact Numbering	Severity Designation
0.7 - 1.0	Critical Issues
0.4 - <0.7	Major Issues
0 - <0.4	Minor Issues



Critical Issues	🟥
Critical Issues	🟥
Critical Issues	🟥
Major Issues	🟧
Major Issues	🟧
Major Issues	🟧
Minor Issues	🟦
Minor Issues	🟦

https://nvd.nist.gov/cvss.cfm

Execute the control

```
$ inspec exec ssh
```

```
×  5.2.2: Ensure SSH Protocol is set to 2 (
   expected: 2
        got:

   (compared using `cmp` matcher)
   )
   ×  SSH Configuration Protocol should cmp == 2

   expected: 2
        got:

   (compared using `cmp` matcher)
```

LAB

Enrich our profile

✓ Add additional metadata to our control

Objectives

- ✓ Execute an InSpec test on a local machine
- ✓ Execute an InSpec test on a remote machine
- ✓ Generate an InSpec profile

Add InSpec-based integration test to a Chef cookbook

Run InSpec-based integrations tests during Chef cookbook development

List additional resources and places to look for support with InSpec