InSpec

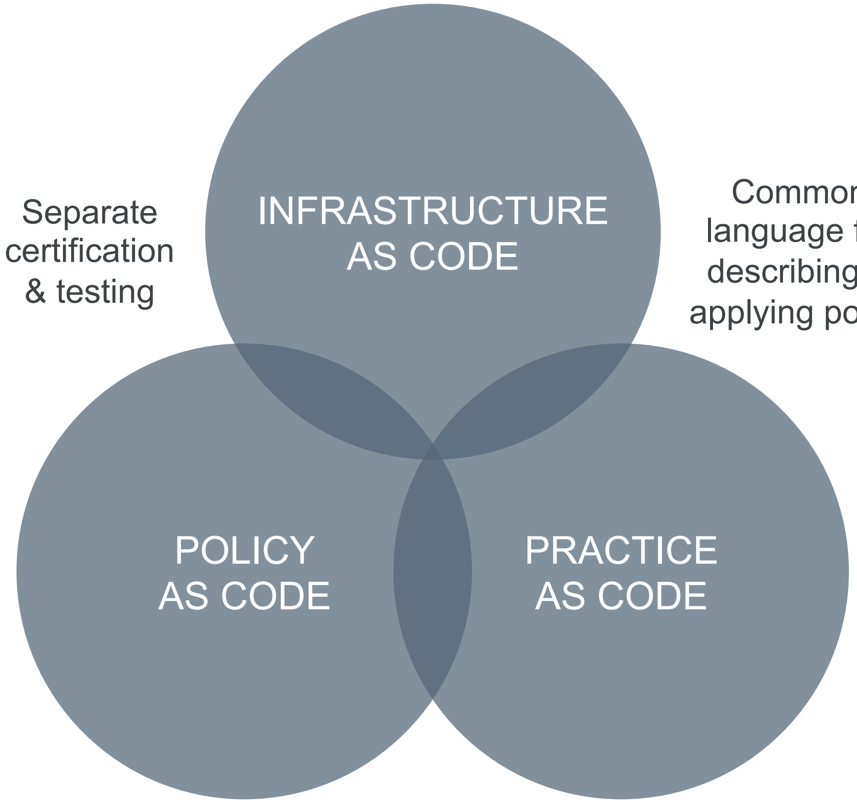
Compliance as Code

INSPEC

InSpec is compliance as code – a human-readable language for automating the continuous testing and compliance auditing of your entire infrastructure.

Compliance as Code

ACCELERATED CYCLE



INFRASTRUCTURE AS CODE

POLICY AS CODE

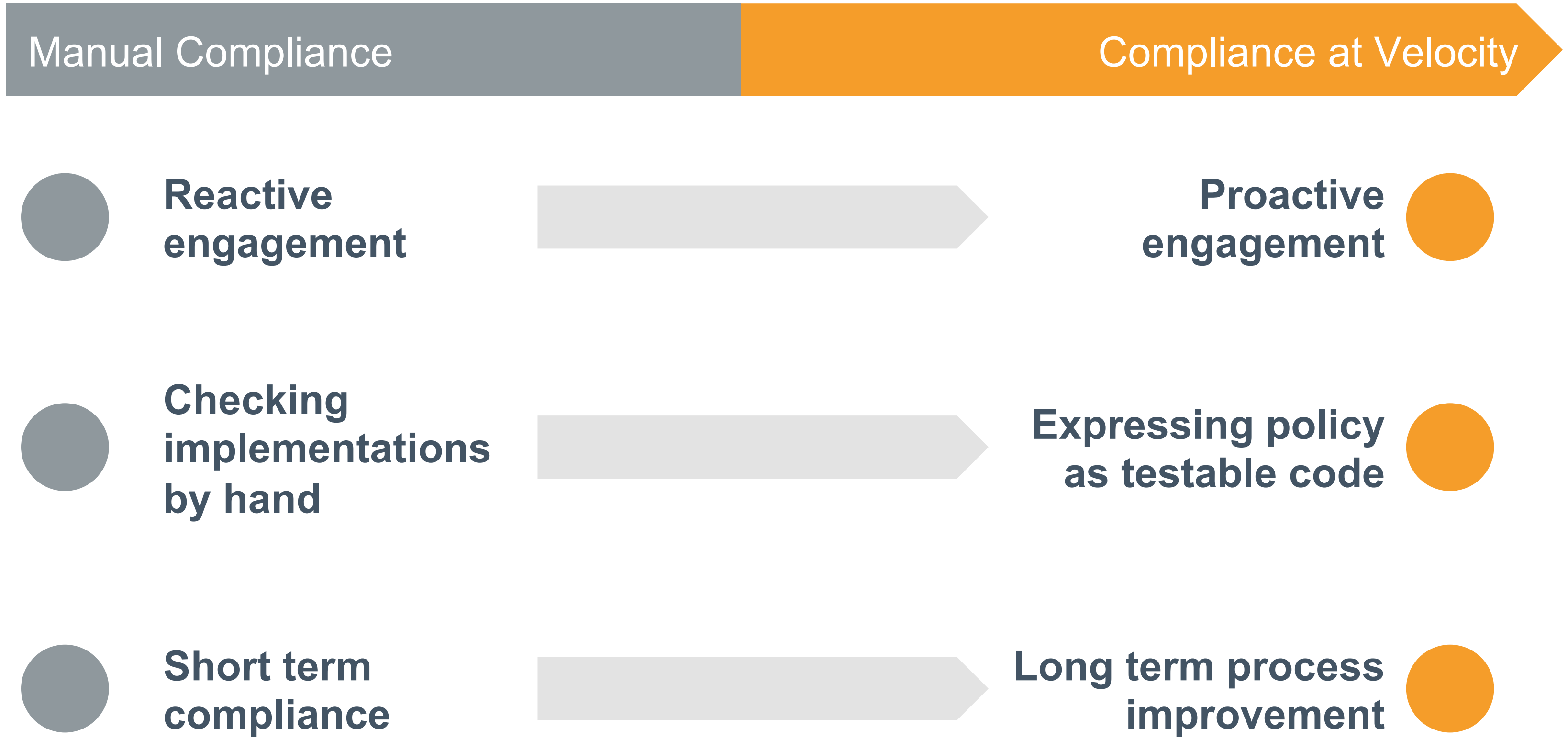
PRACTICE AS CODE

Separate certification & testing

Common language for describing & applying policy

Compliance at velocity

ROLE OF THE COMPLIANCE OFFICER



Manual Compliance → Compliance at Velocity

Reactive engagement → Proactive engagement

Checking implementations by hand → Expressing policy as testable code

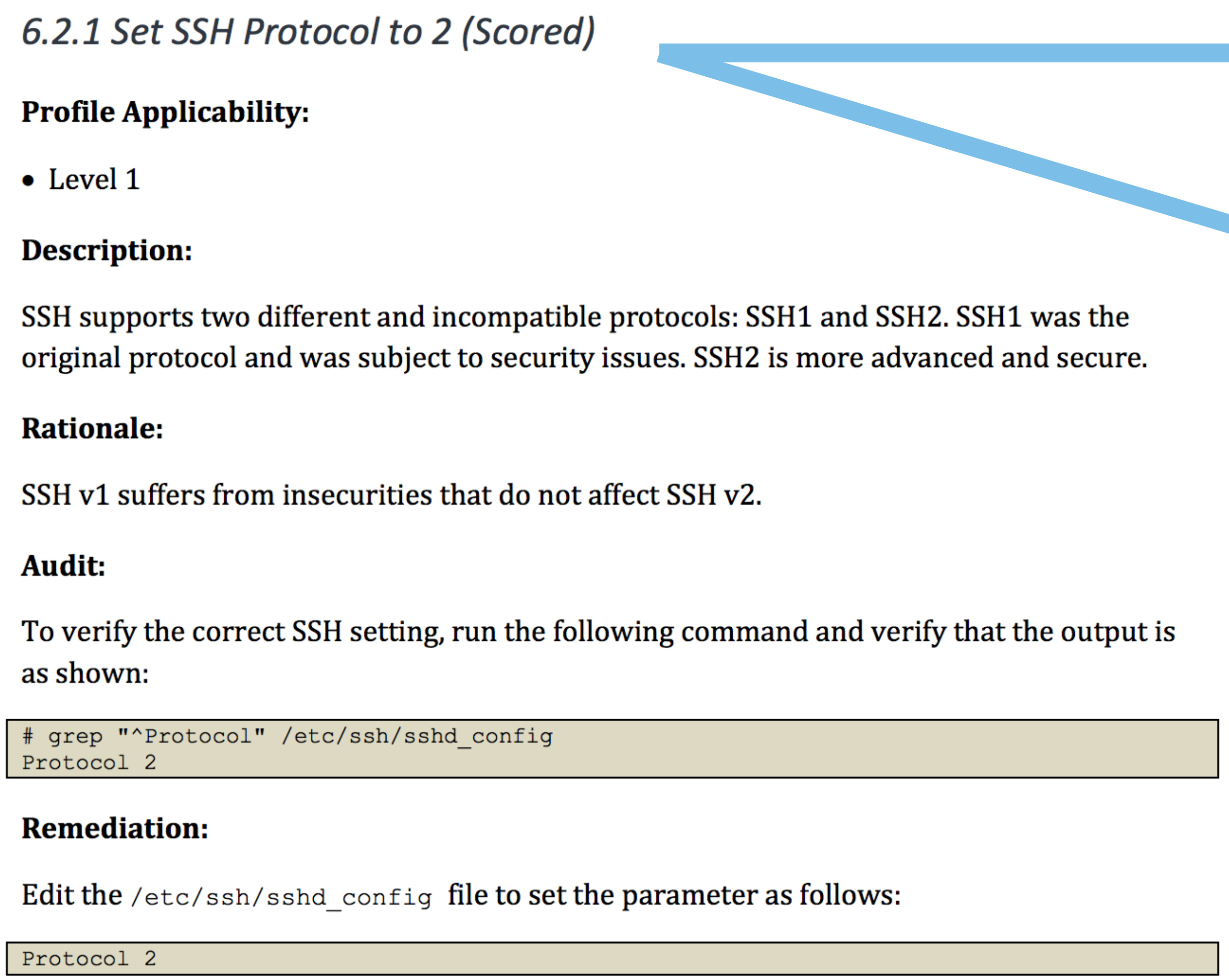
Short term compliance → Long term process improvement

One language, One workflow

SSH Control

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.

Mapping Compliance Document to InSpec



6.2.1 Set SSH Protocol to 2 (Scored)

Profile Applicability:

- Level 1

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:

To verify the correct SSH setting, run the following command and verify that the output is as shown:

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

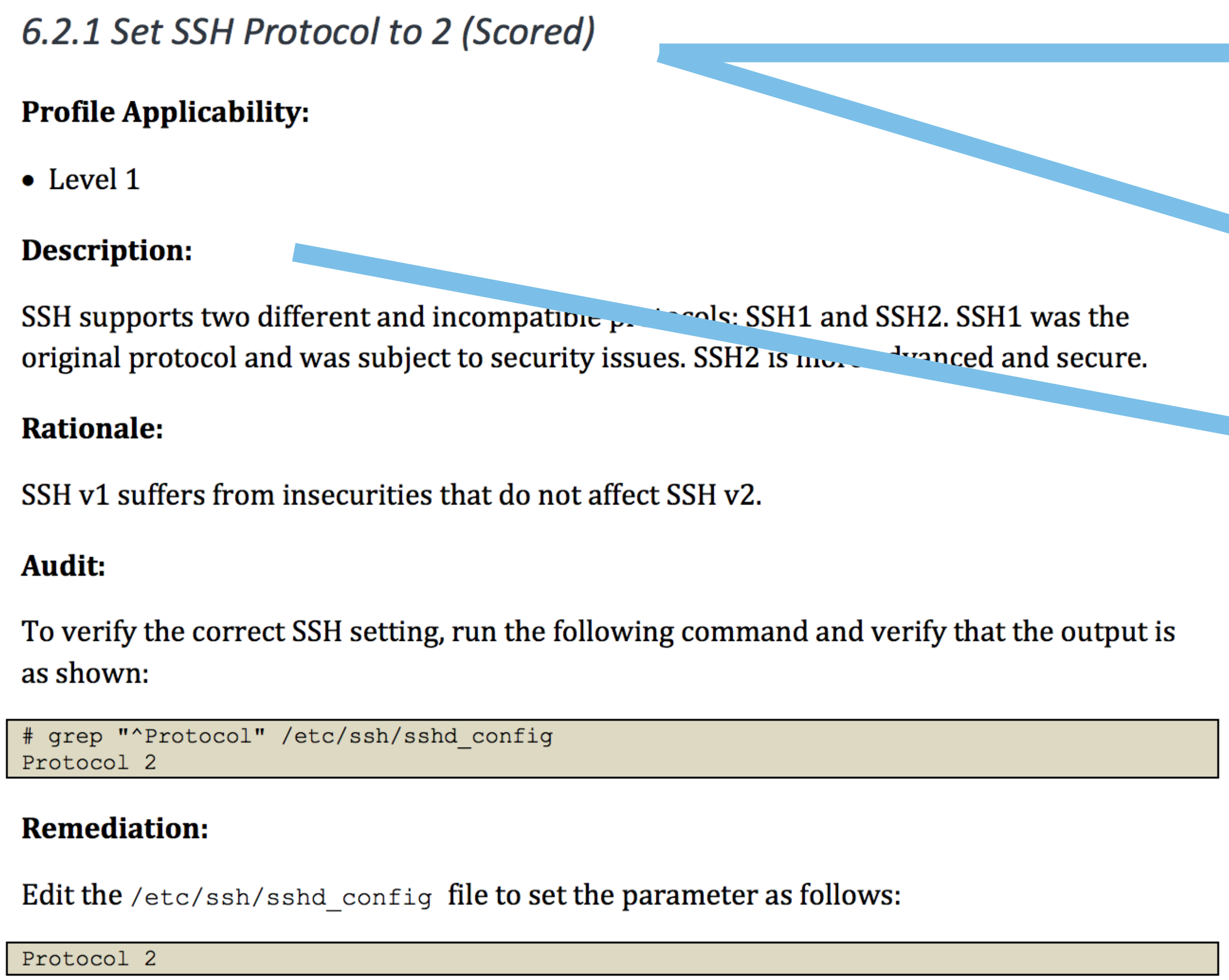
Remediation:

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

```
Protocol 2
```

```
control 'ssh-6.2.1' do

  title 'Set SSH Protocol to 2'









end
```

Mapping Compliance Document to InSpec



6.2.1 Set SSH Protocol to 2 (Scored)

Profile Applicability:

- Level 1

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:

To verify the correct SSH setting, run the following command and verify that the output is as shown:

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

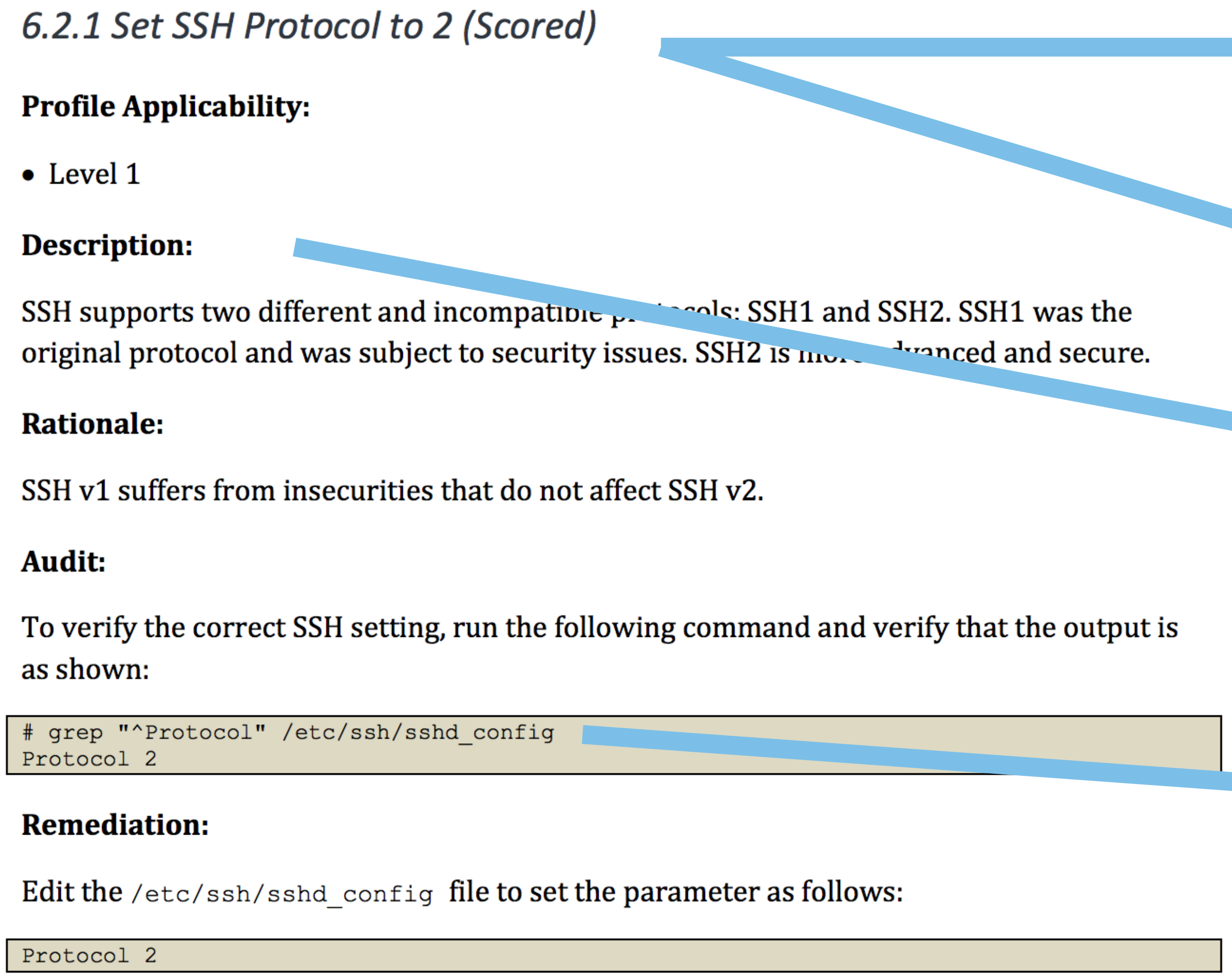
Remediation:

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

```
Protocol 2
```

```
control 'ssh-6.2.1' do

  title 'Set SSH Protocol to 2'
  desc "
    SSH supports two different ...
  "






end
```

Mapping Compliance Document to InSpec



6.2.1 Set SSH Protocol to 2 (Scored)

Profile Applicability:

- Level 1

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:

To verify the correct SSH setting, run the following command and verify that the output is as shown:

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

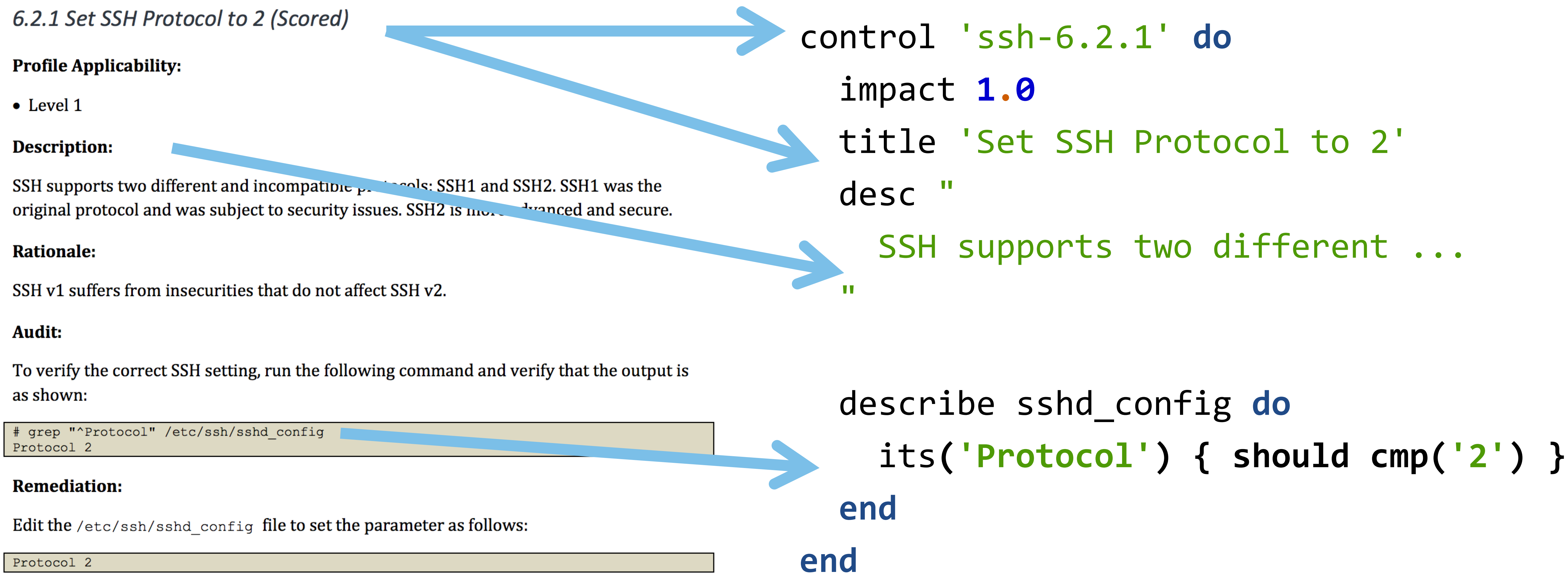
Remediation:

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

```
Protocol 2
```

```
control 'ssh-6.2.1' do

  title 'Set SSH Protocol to 2'
  desc "
    SSH supports two different ...
  "

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

Mapping Compliance Document to InSpec



6.2.1 Set SSH Protocol to 2 (Scored)

Profile Applicability:

- Level 1

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:

To verify the correct SSH setting, run the following command and verify that the output is as shown:

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

Remediation:

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

```
Protocol 2
```

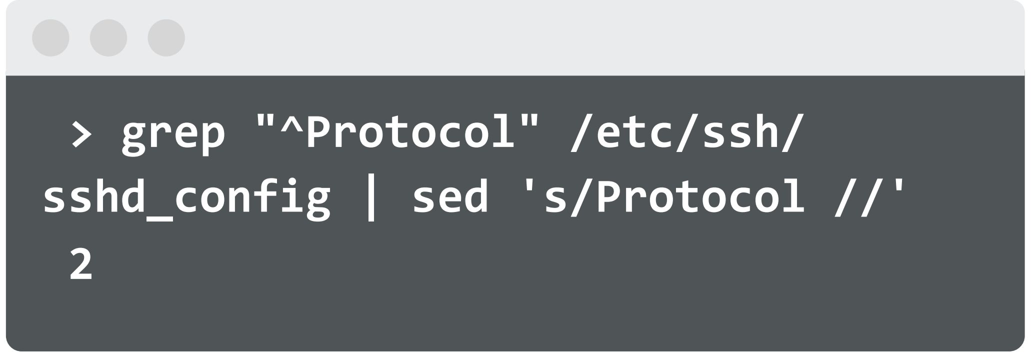
```
control 'ssh-6.2.1' do
  impact 1.0
  title 'Set SSH Protocol to 2'
  desc "
    SSH supports two different ...
  "

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

Differences in verifying compliance policy

DOCUMENTATION

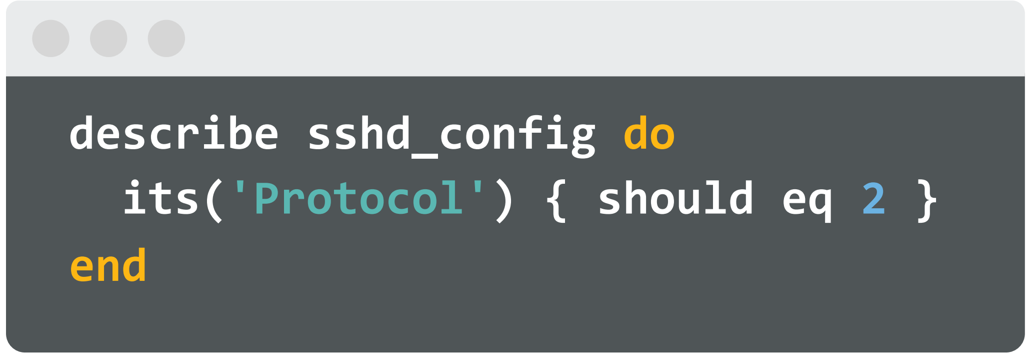
SSH supports two different protocol versions. The original version, SSHv1, was subject to a number of security issues. Please use SSHv2 instead to avoid these.

SCRIPTING TOOLS



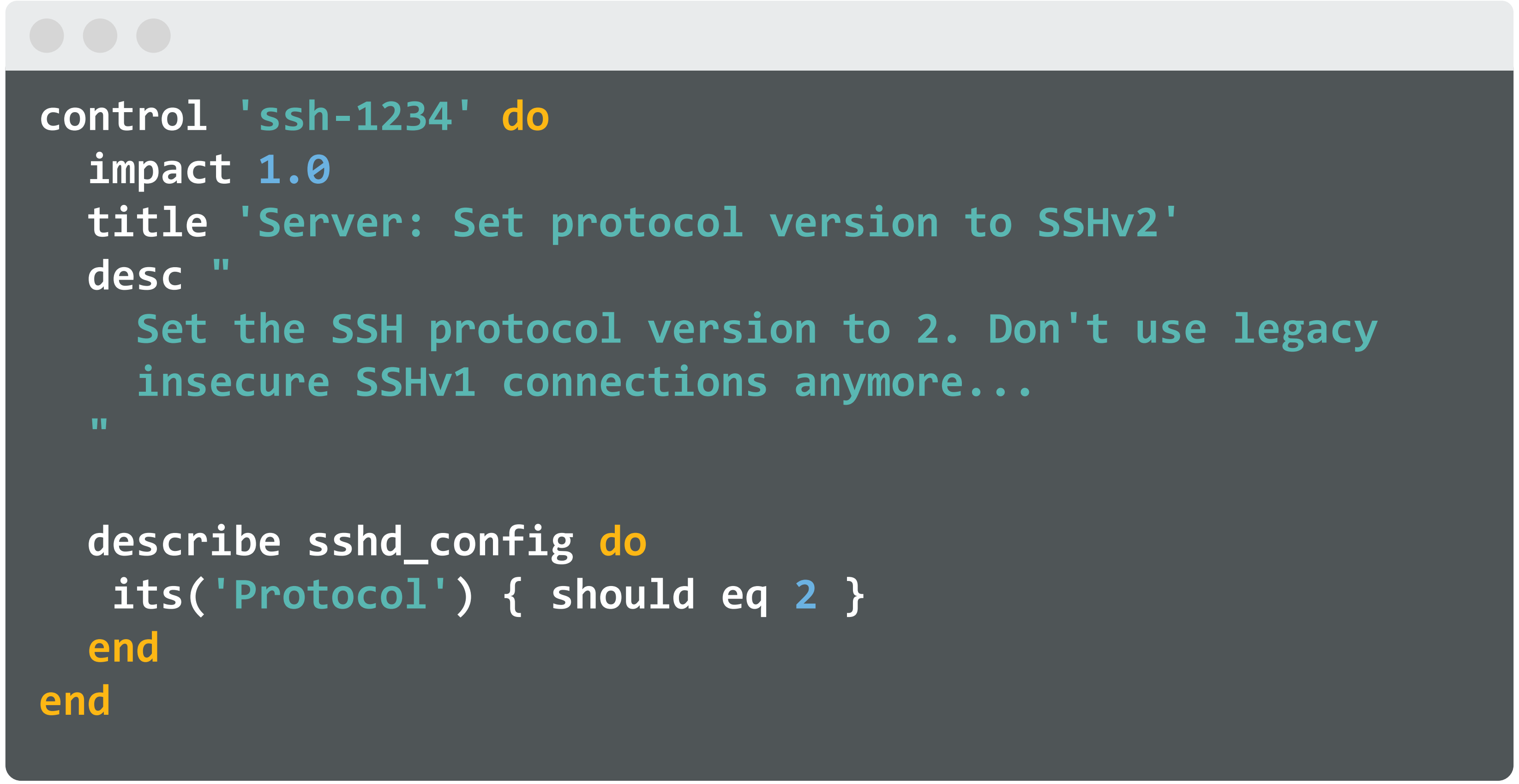
```
 > grep "^Protocol" /etc/ssh/
sshd_config | sed 's/Protocol //'
 2
```

COMPLIANCE LANGUAGE



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

COMPLIANCE LANGUAGE



```
control 'ssh-1234' do
  impact 1.0
  title 'Server: Set protocol version to SSHv2'
  desc "
    Set the SSH protocol version to 2. Don't use legacy
    insecure SSHv1 connections anymore...
  "

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

InSpec

ONE LANGUAGE

Linux

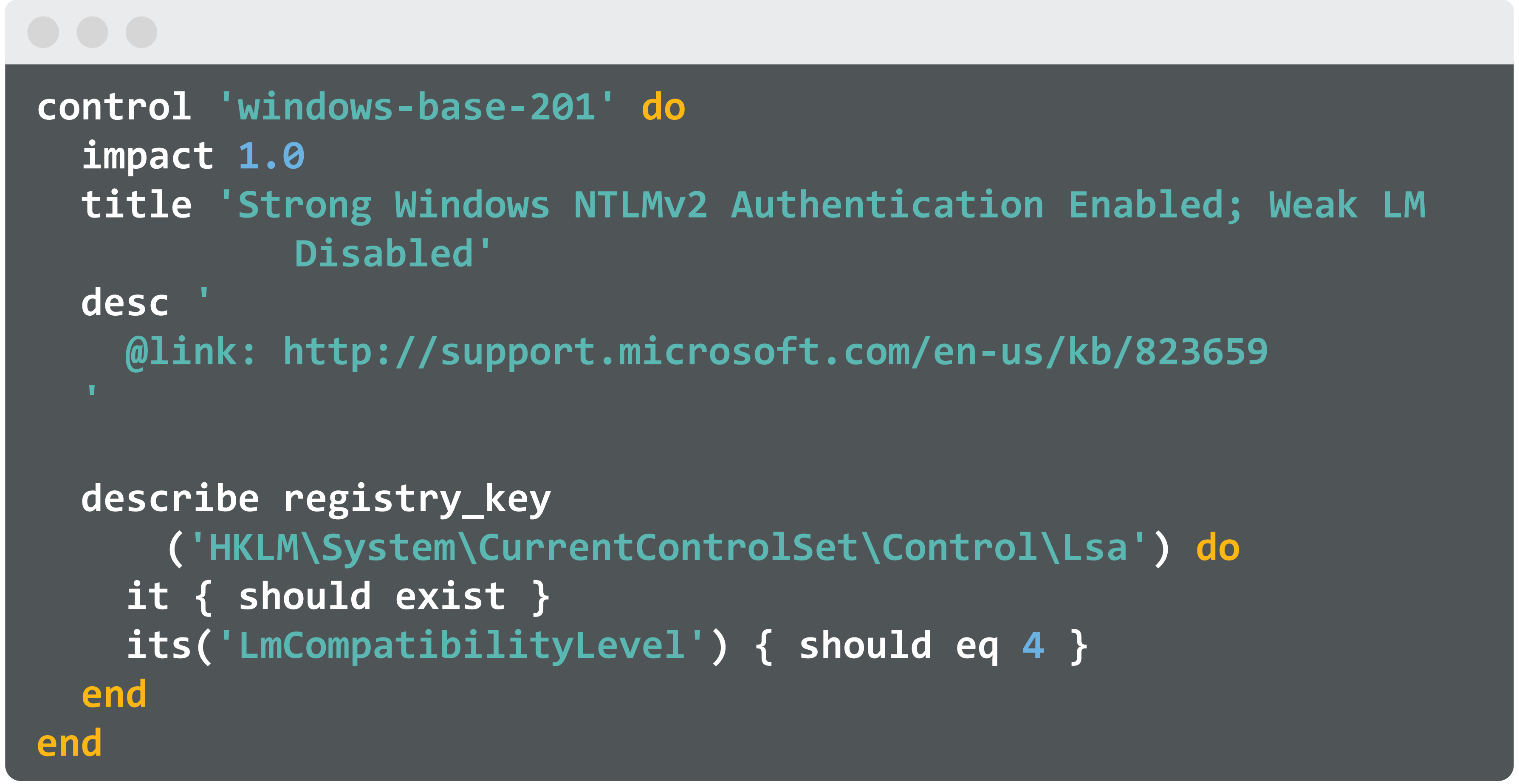
Windows

FreeBSD

Oracle Solaris 11

AIX

InSpec for Windows



```
control 'windows-base-201' do
  impact 1.0
  title 'Strong Windows NTLMv2 Authentication Enabled; Weak LM
            Disabled'
  desc '
    @link: http://support.microsoft.com/en-us/kb/823659
  '

  describe registry_key
      ('HKLM\System\CurrentControlSet\Control\Lsa') do
    it { should exist }
    its('LmCompatibilityLevel') { should eq 4 }
  end
end
```

InSpec

ONE LANGUAGE

Linux

Windows

FreeBSD

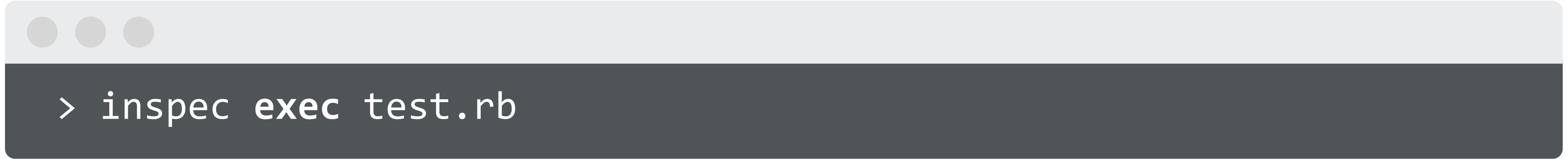
Oracle Solaris 11

AIX

- Baremetal
- VMs
- Containers

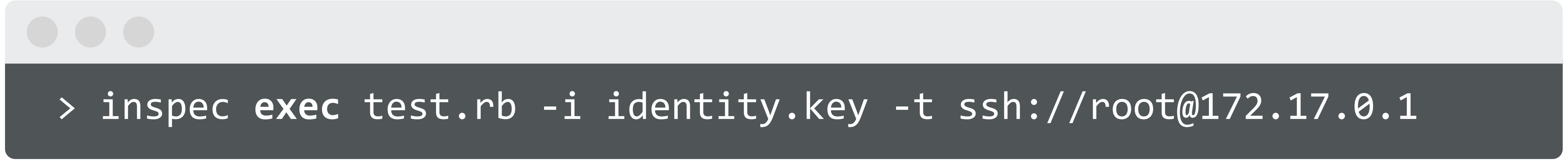
Different ways to run InSpec

Test your machine locally



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

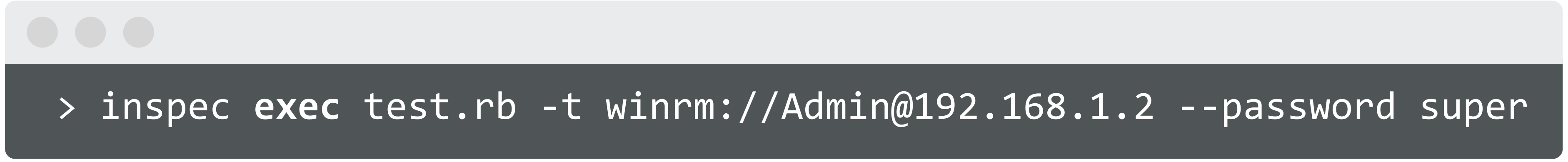
Test a machine remotely via SSH



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

No ruby/agent on the node

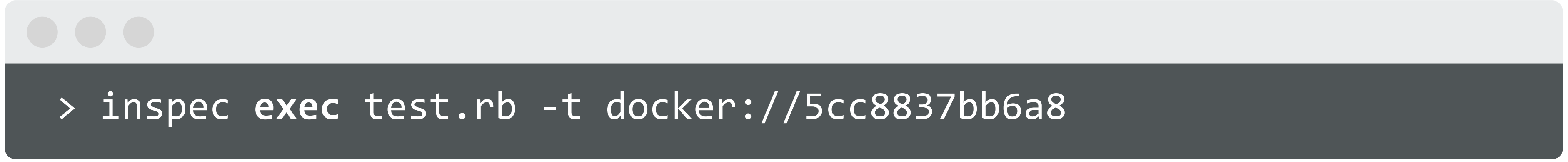
Test a machine remotely via WinRM



```
> 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

InSpec

ONE LANGUAGE

Linux

Windows

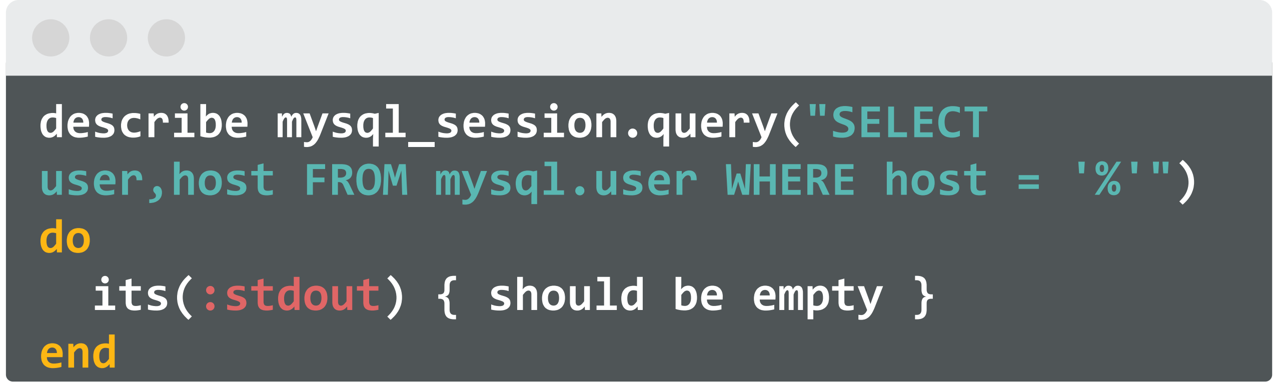
FreeBSD

Oracle Solaris 11

AIX

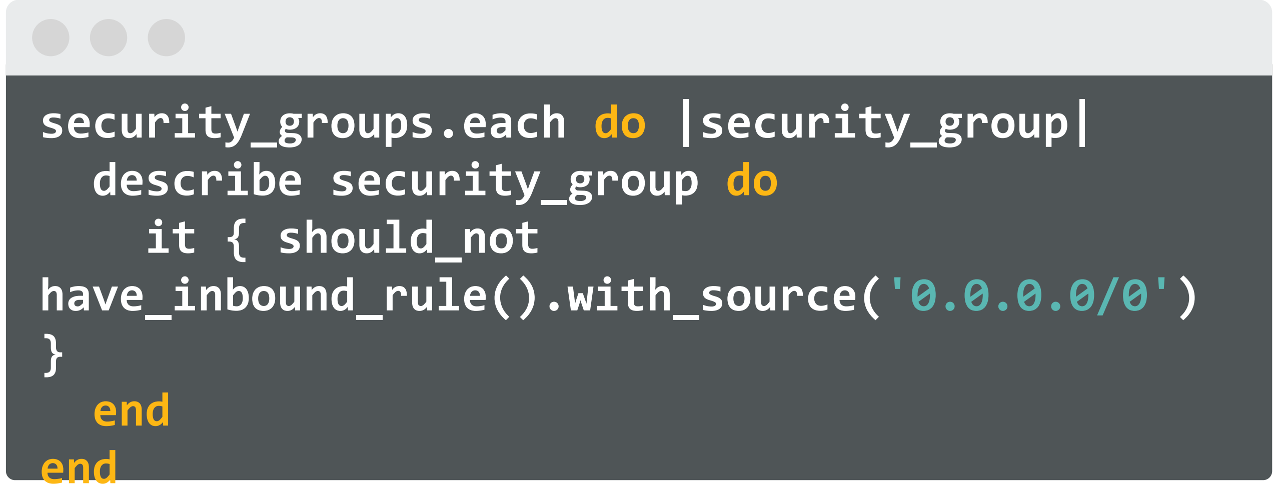
- Baremetal
- VMs
- Containers
- **Databases**
- **API endpoints (e.g. cloud)**

Database Testing



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

Cloud Provider Testing



```
security_groups.each do |security_group|
  describe security_group do
    it { should_not
have_inbound_rule().with_source('0.0.0.0/0')
}
  end
end
```

InSpec

ONE WORKFLOW

Security meets operations

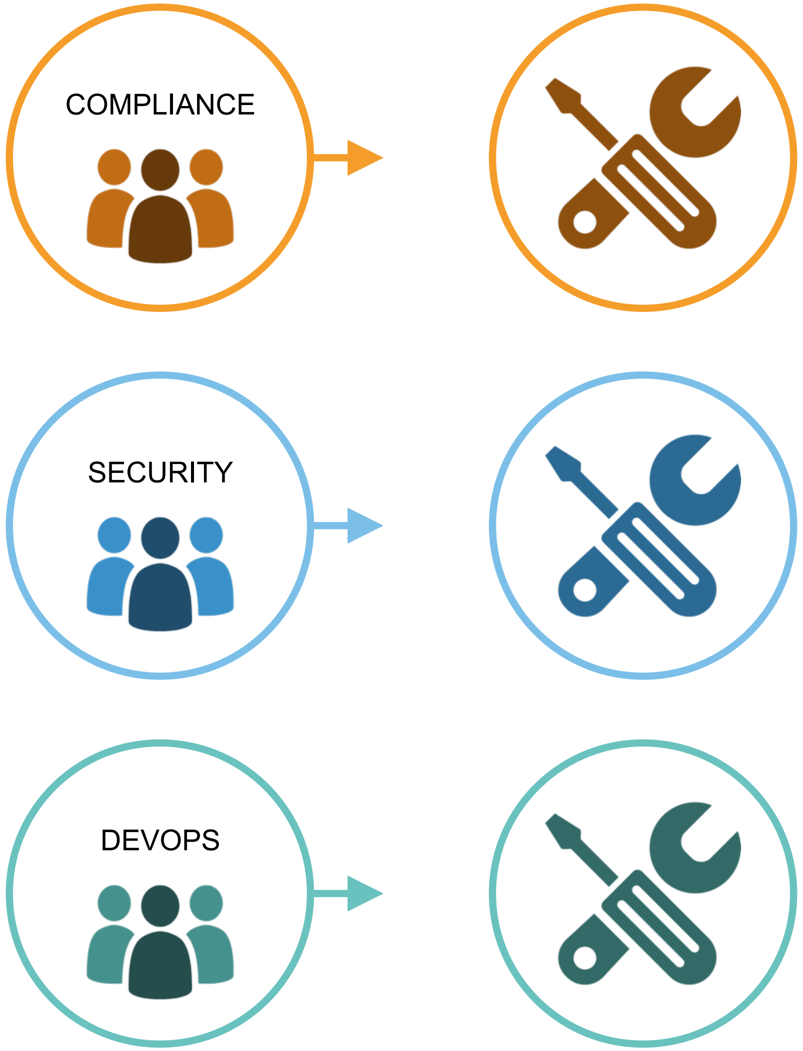
COMPLIANCE

SECURITY

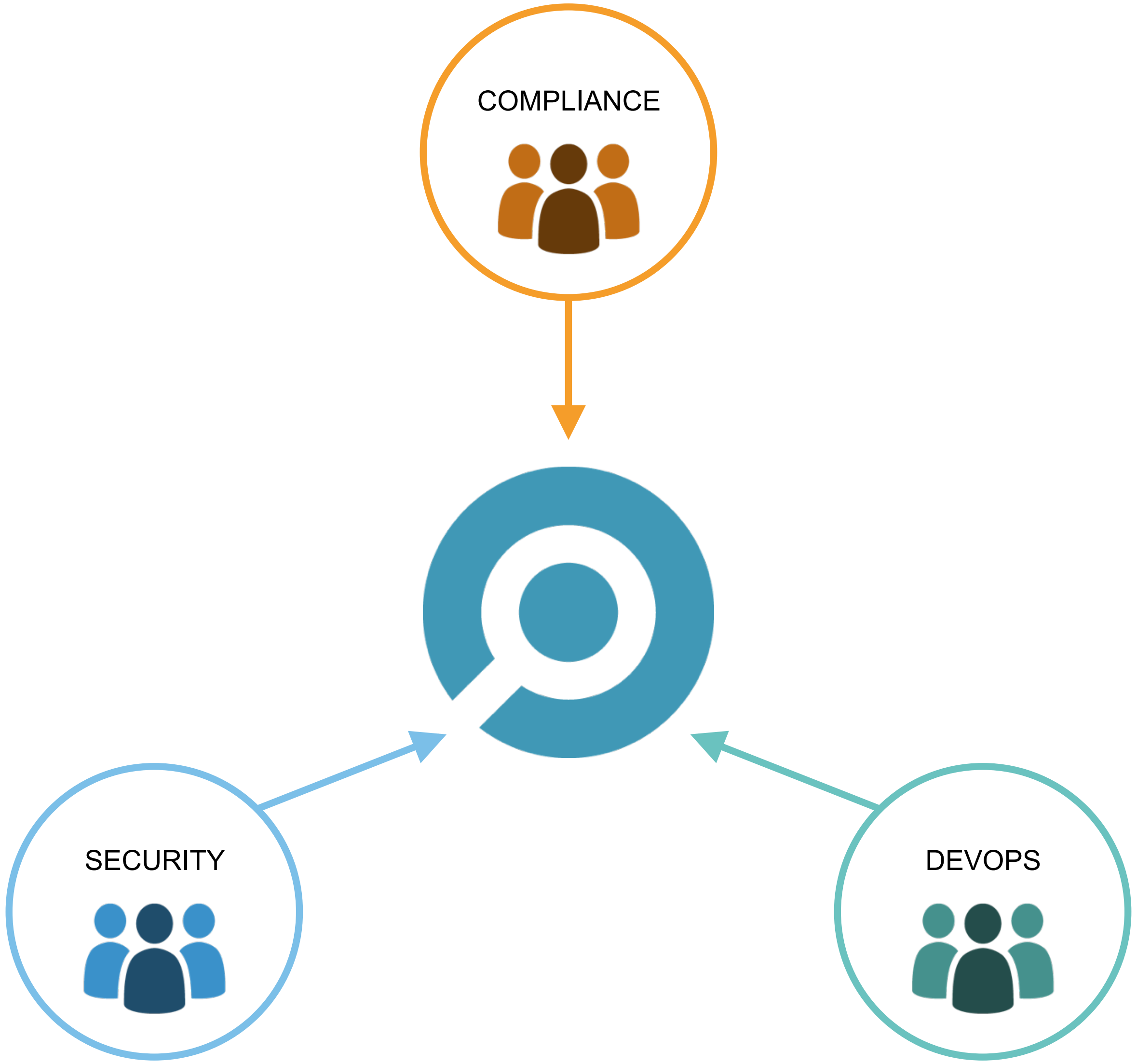
DEVOPS

InSpec

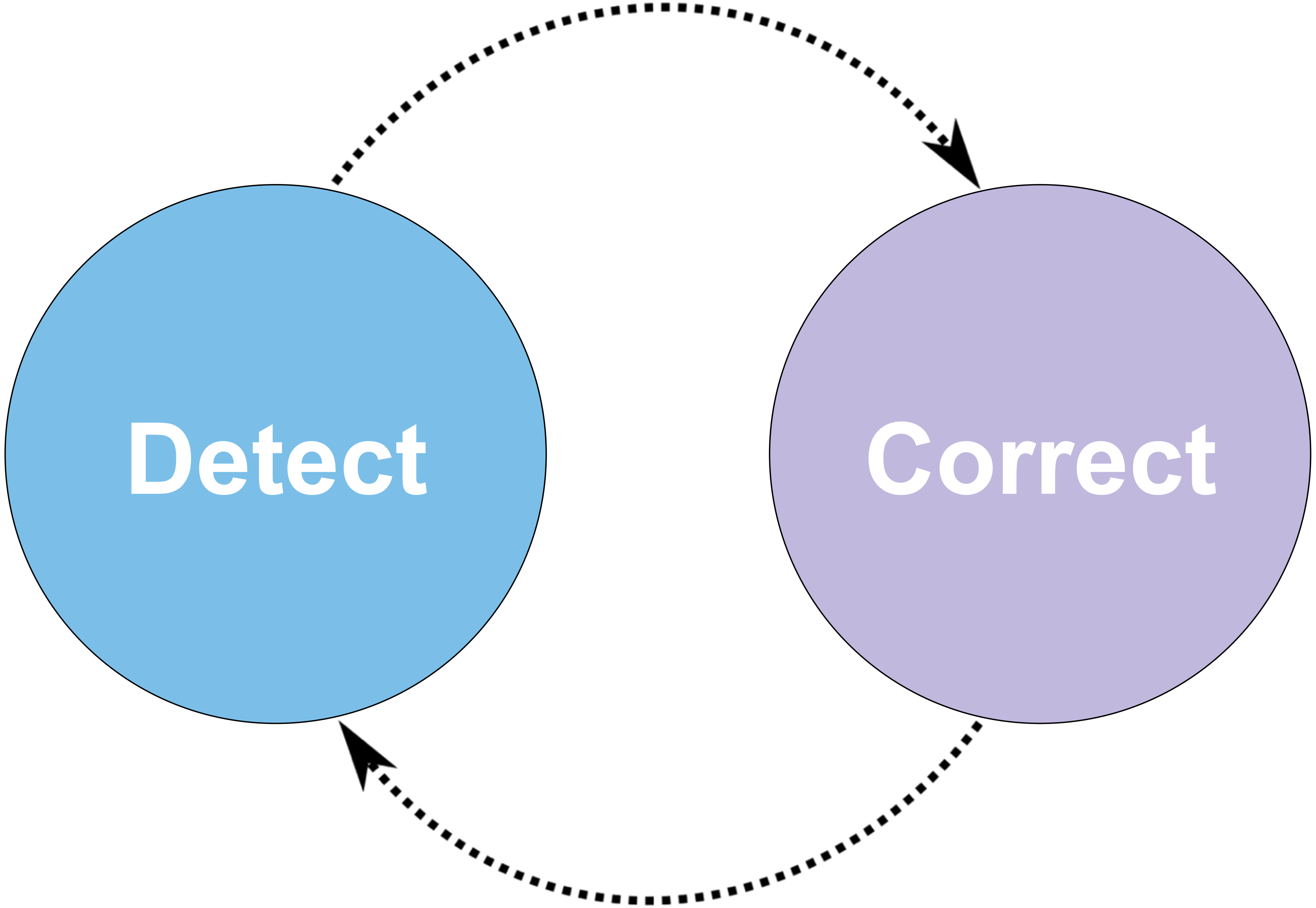
Each team uses separate tools



Unified language



Continuous Workflow



Detect

Correct

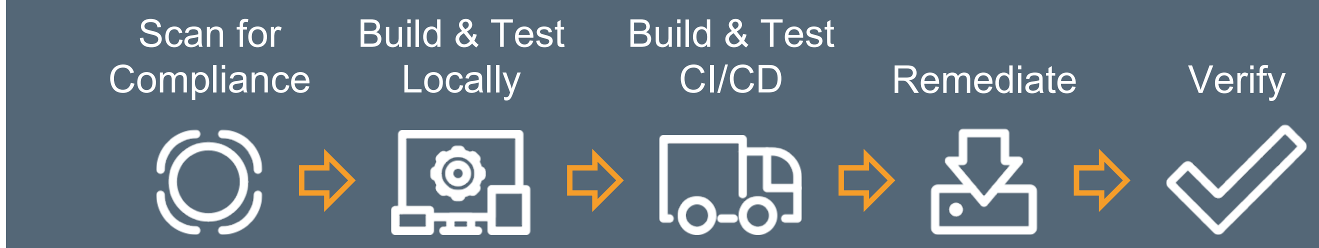
InSpec

Turn security and compliance into code

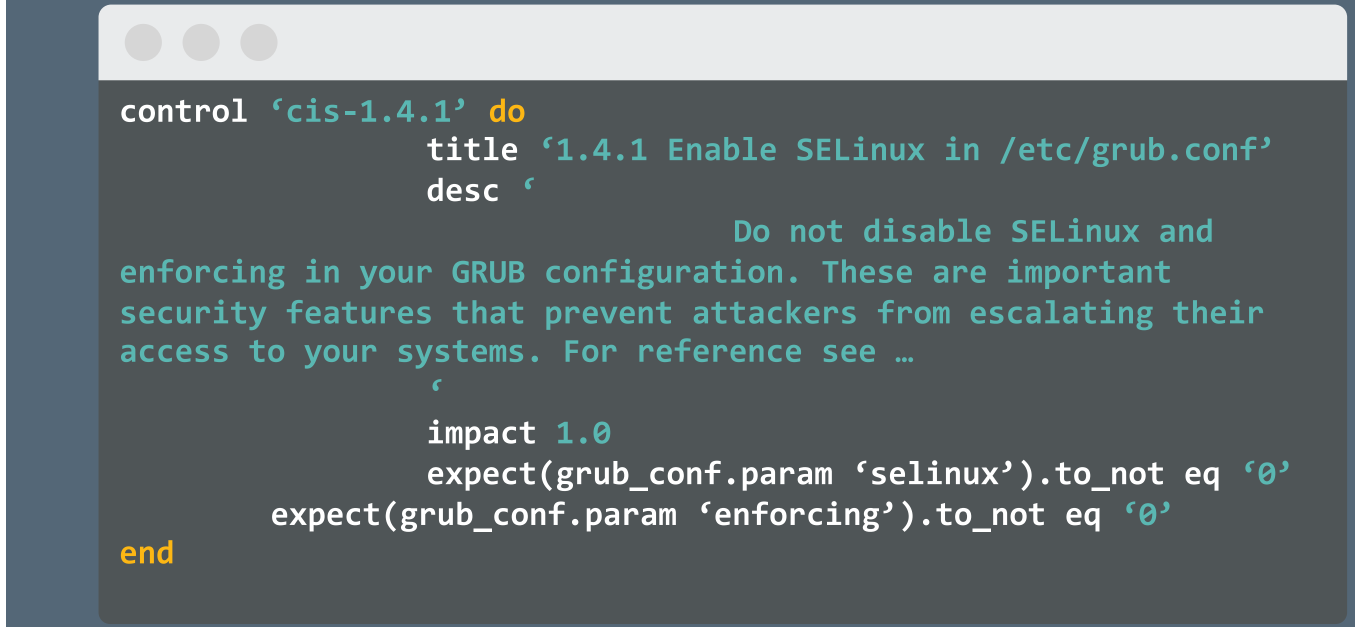
- Translate **compliance** into Code
- **Clearly** express statements of policy
- Move risk to build/test from **runtime**
- Find issues **early**
- Write code **quickly**
- Run code **anywhere**
- **Inspect** machines, data, and APIs

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
```