CHEF™

Chef Training Services

Chef Compliance

Instructor Guide

Course Rev 1.0.0

1: Introduction

This Chef Compliance course provides an understanding of the capabilities of Chef Compliance. This course covers how to install and initially configure the Chef Compliance server, perform compliance scans against Windows and Linux nodes, and remediate compliance issues with Chef, and run Compliance reports.

In addition, you will learn how to use InSpec to create and modify Chef Compliance profiles and learn how to locate CIS (Center for Internet Security) and DoD (Department of Defense) compliance specifications that you can use to write Chef Compliance profiles.

Instructor Note: **Be sure to read Appendix Z at the end of this instructor guide** for training lab set up notes and additional instructor notes.
Instructor Note: This course has been tested on Compliance Server v0.14.5. The labs have been tested against target Linux and Windows nodes that have ChefDK 0.11.2 and inspec 0.14.7. You must use at least inspec 0.14.7 on the target nodes in order for these labs to work.

Slide 2

Introduce Yourselves

- Name
- Current job role
- Previous job roles/background
- Experience with Chef and/or config management

Slide 3

Objectives

After completing this course, you should be able to:

- Describe the capabilities of Chef Compliance.
- Install and initially configure the Chef Compliance server.
- Perform scans with Chef Compliance.
- Remediate compliance issues.
- Use InSpec to create, modify, and test Chef Compliance profiles.
- Schedule and run compliance reports.
- Manage users, organizations, teams and permissions.

Note: You should have attended at least Chef Essentials, Chef Fundamentals or have equivalent Chef experience prior to attending this course.

Instructor Note: You can tell the students that this course covers scanning and remediating both Linux and Windows nodes. For example, module 03 covers scanning and remediating Linux nodes and module 04 covers scanning and remediating Windows nodes. However, the Compliance server runs only on Linux.

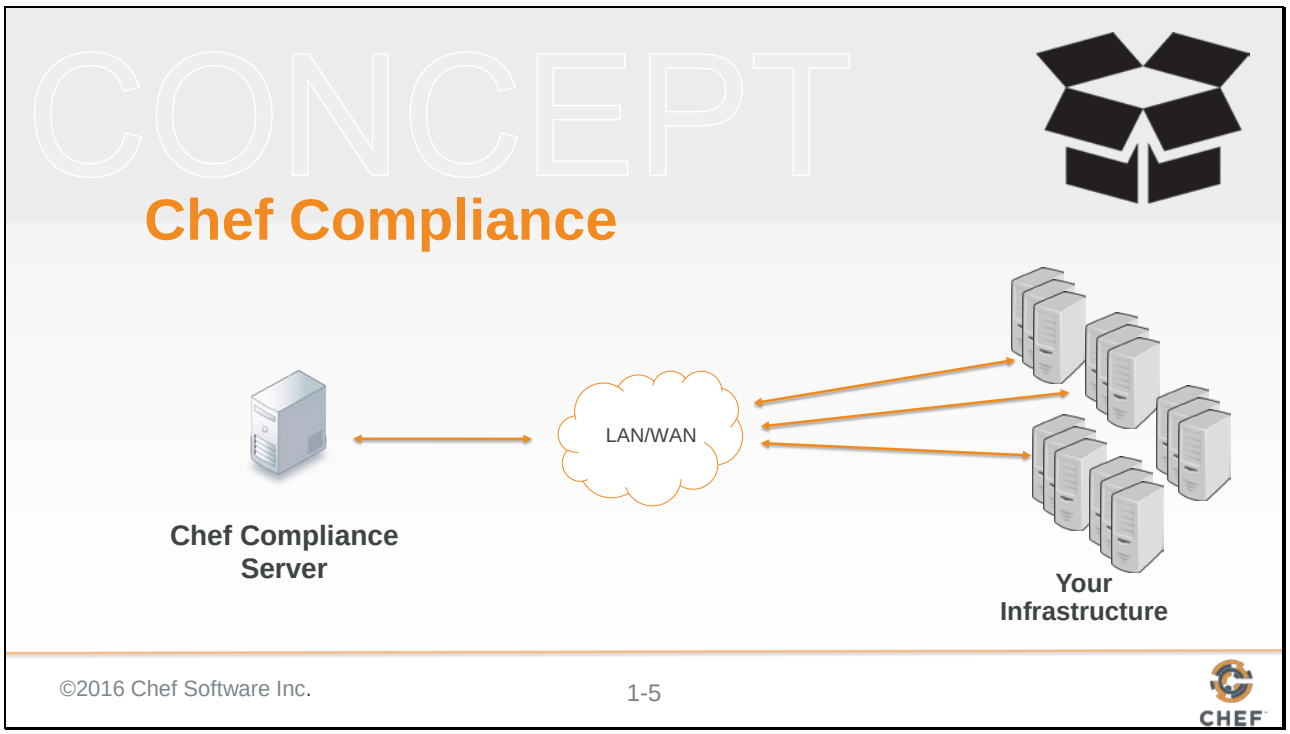
Slide 4

By now you are probably aware of how Chef automates the configuration and management of your infrastructure. But what about risks and compliance issues of your infrastructure?

Regulatory compliance is a fact of life for every enterprise. With Chef Compliance you can scan for risks and compliance issues with easy-to-understand, customizable reports and visualization.

You can then use Chef to automate the remediation of issues and use Chef Compliance to implement a continuous audit of applications and infrastructure.

Slide 5

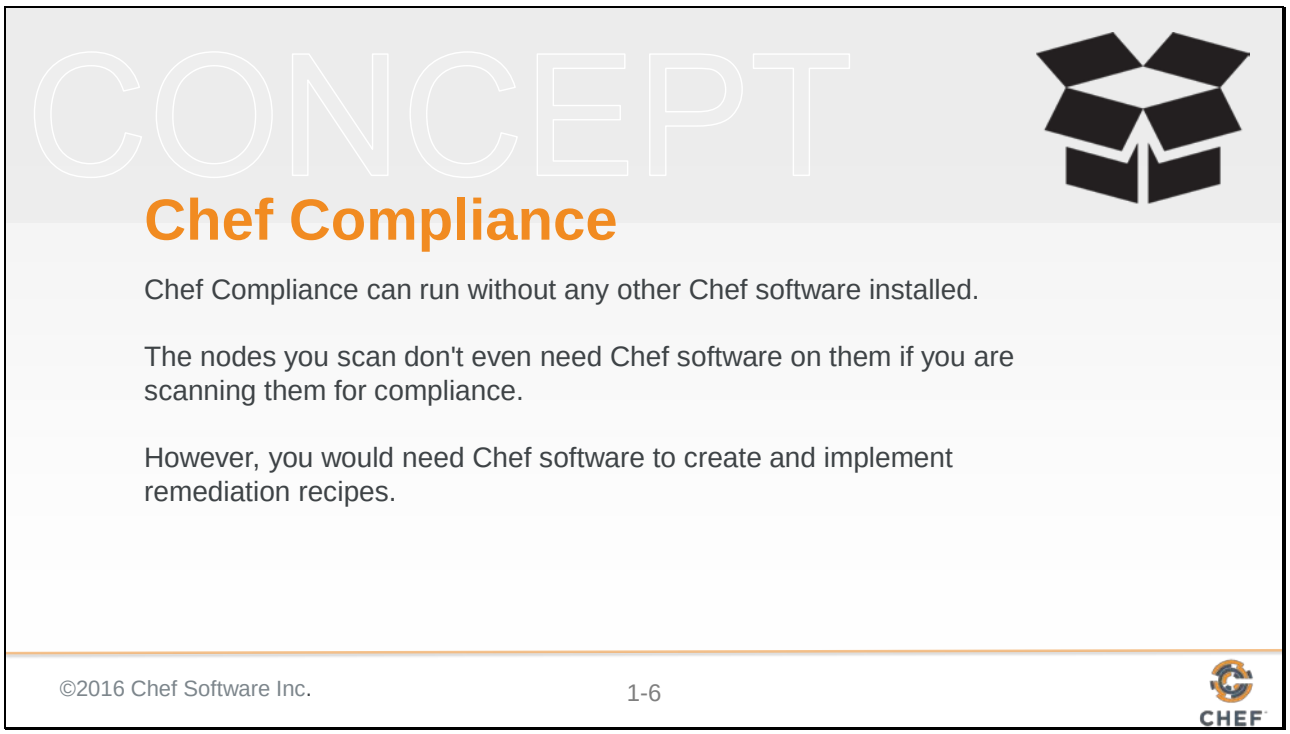


The Chef Compliance server is a centralized location from which all aspects of the state or your infrastructure's compliance can be managed.

With Chef Compliance you can test any node in your infrastructure, including all of the common UNIX and Linux platforms and most versions of Microsoft Windows.

Chef Compliance can continuously test any node against the goals of your organization's security management lifecycle for risks and compliance issues.

Slide 6



Chef Compliance can run without any other Chef software installed on the Chef Compliance server machine.

The nodes you scan don't even need Chef software on them if you are scanning them for compliance.

However, you would need Chef software to create and implement remediation recipes if you choose to use recipes to remediate compliance issues.

Slide 7

CONCEPT

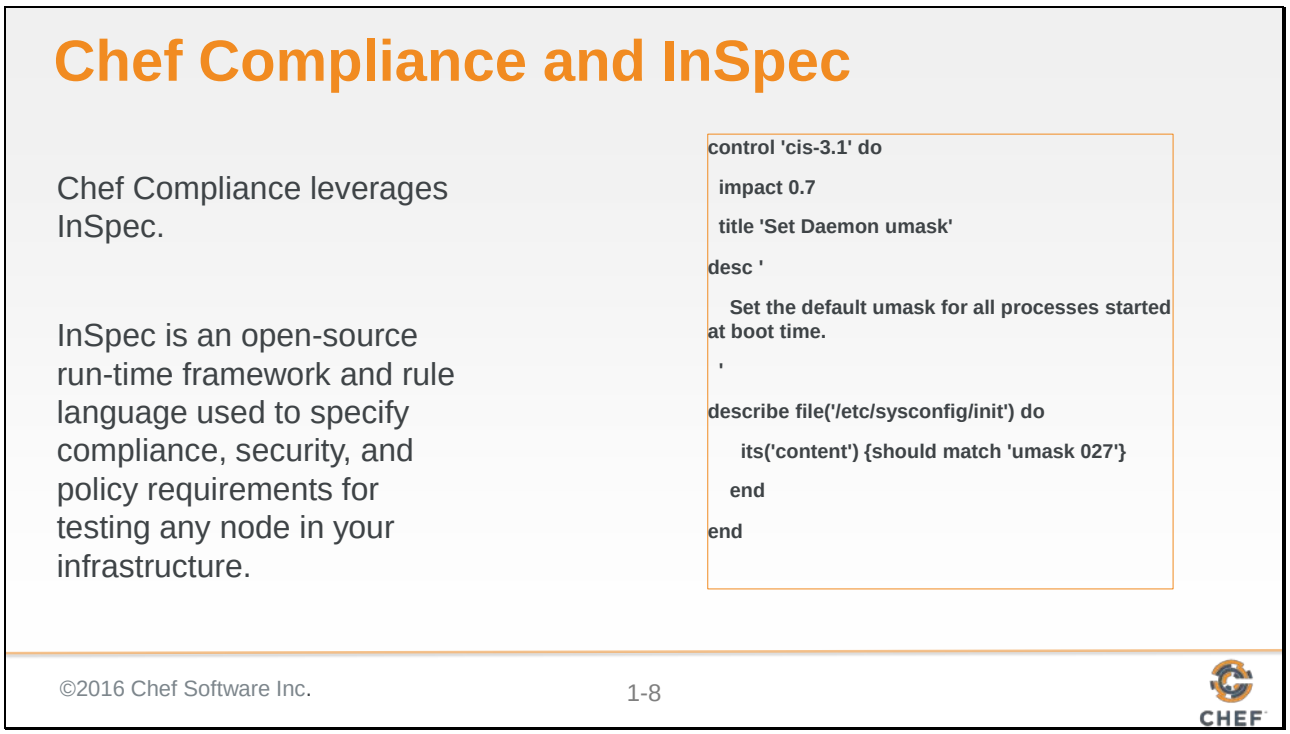
Chef Compliance

Reports: Chef Compliance can produce reports that indicate risks and issues classified by severity and impact levels.

Compliance Profiles: You can get started quickly with pre-built Compliance profiles for scanning Linux and Windows nodes.

CHEF

Slide 8



Chef Compliance leverages InSpec.

InSpec is an open-source run-time framework and rule language used to specify compliance, security, and policy requirements for testing any node in your infrastructure. The InSpec name refers to “infrastructure specification

InSpec includes a collection of resources to help you write auditing rules quickly and easily using the Compliance DSL.

Use InSpec to examine any node in your infrastructure; run the tests locally or remotely.

Any detected security, compliance, or policy issues are flagged in a log and displayed in reports.

The InSpec audit resource framework is fully compatible with Chef Compliance.

Instructor note: InSpec is similar to ServerSpec but learners who have no experience with Serverspec may be confused by the reference.

Slide 9

InSpec DSL

InSpec includes a collection of resources to help you write auditing rules quickly and easily using the Compliance DSL

Use InSpec to examine any node in your infrastructure; run the tests locally or remotely.

Any detected security, compliance, or policy issues are flagged in a log and in Chef Compliance, displayed in a GUI.

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

describe port(443) do
  it { should be_listening }
  its('protocols') {should include 'tcp'}
end
```

Slide 10

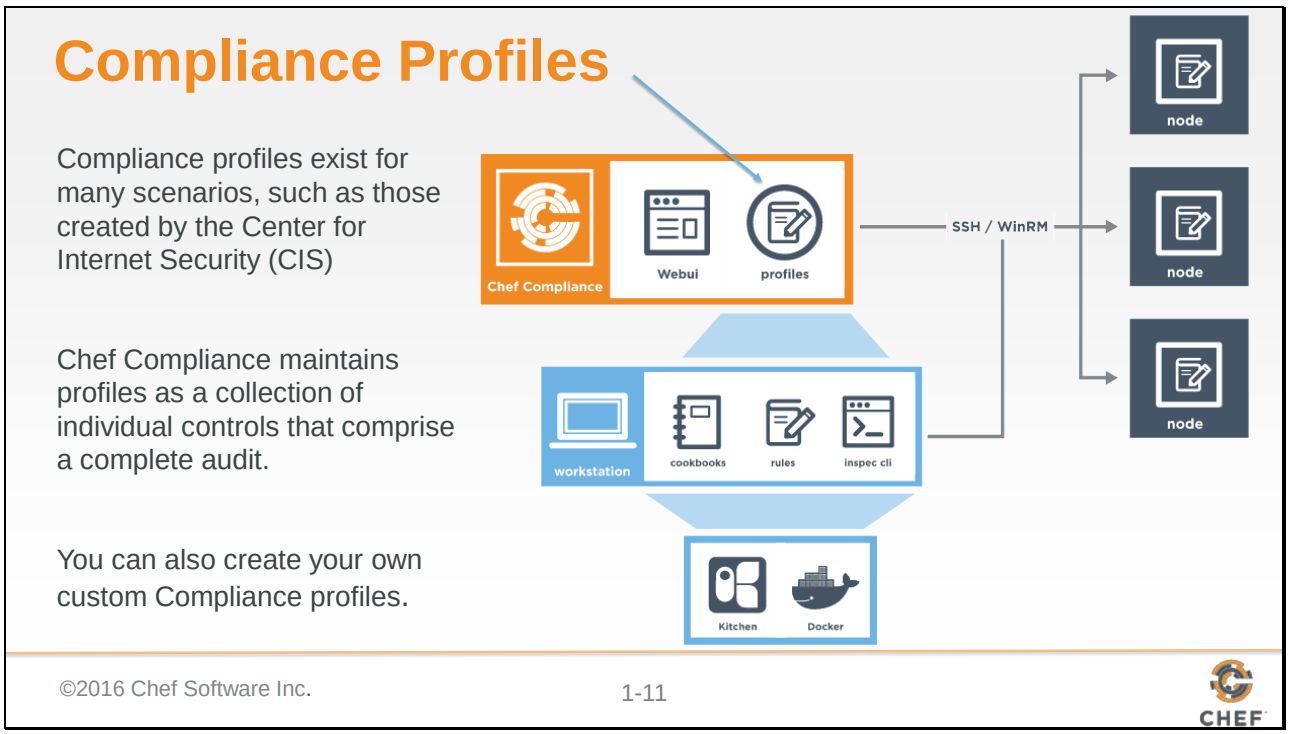
InSpec DSL

The InSpec audit resource framework is fully compatible with Chef Compliance.

The Compliance DSL is a Ruby DSL for writing audit rules, which includes audit resources that you can invoke.

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

describe port(443) do
  it { should be_listening }
  its('protocols') {should include 'tcp'}
end
```

Slide 11



Compliance profiles exist for many scenarios, such as those created by the Center for Internet Security (CIS), a non-profit organization that is focused on enhancing the cyber security readiness and response of public and private sector entities.

Chef Compliance maintains profiles as a collection of individual controls that comprise a complete audit. For example, CIS benchmark 8.1.1.1 recommends testing for the maximum size of the audit log.

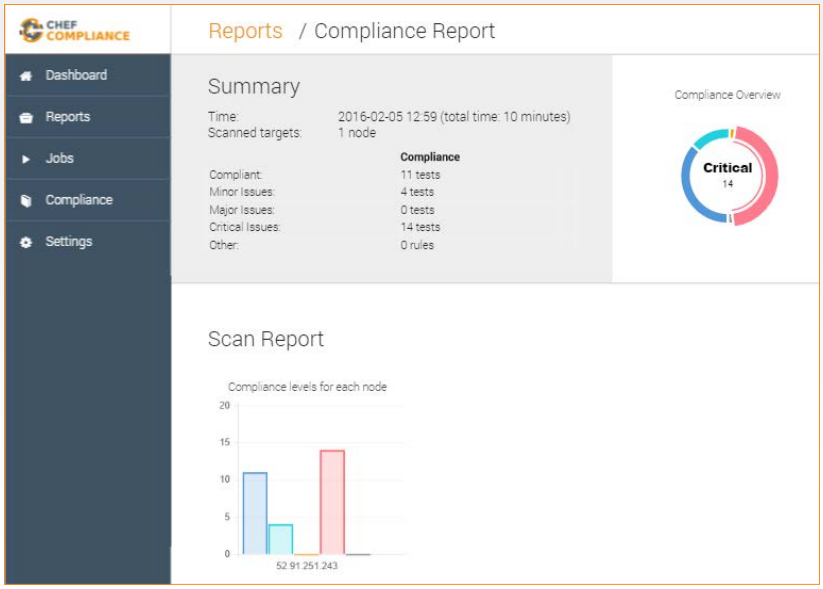
You can also create your own custom Compliance profiles.

Slide 12

Compliance Web UI

The Chef Compliance web UI provides views into compliance scan results as well as views of Chef Compliance profiles.

You execute scans via the Compliance web UI as well.

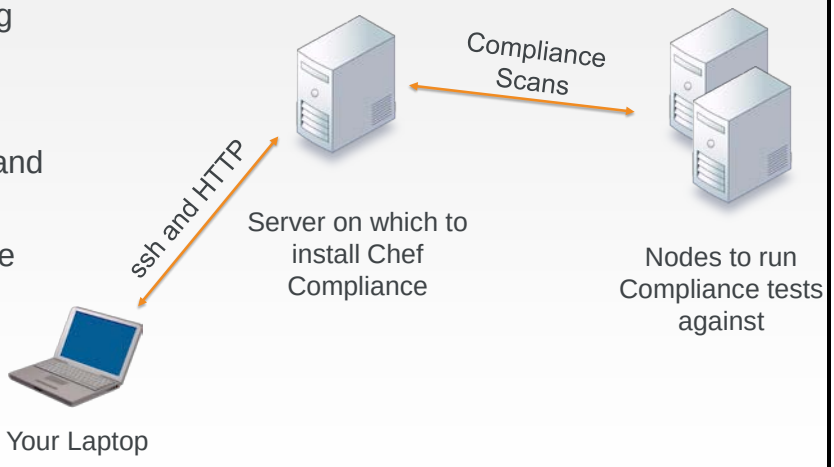


Slide 13

Your Lab Environment for Scanning

We will provide three machines for you to use while performing lab exercises in this course:

- One Linux server to install and run Chef Compliance on.
- One Windows node and one Linux node to perform Chef Compliance scans against.



These are basic AWS AMIs that we use for Chef training. They have ChefDK installed on them although Chef does not actually need to be installed on these instances in order to run scans.

Instructor Note: Now would be a good time to distribute the hostnames of the three nodes you will assign to each student. You should ask the students to note which one they will use as their Compliance Server and which ones they will use as the target nodes for scans.
For example:
ec2-52-91-31-125.compute-1.amazonaws.com = Compliance server.
ec2-54-164-54-218.compute-1.amazonaws.com = Linux Target node.
ec2-54-164-54-210.compute-1.amazonaws.com = Windows Target node.

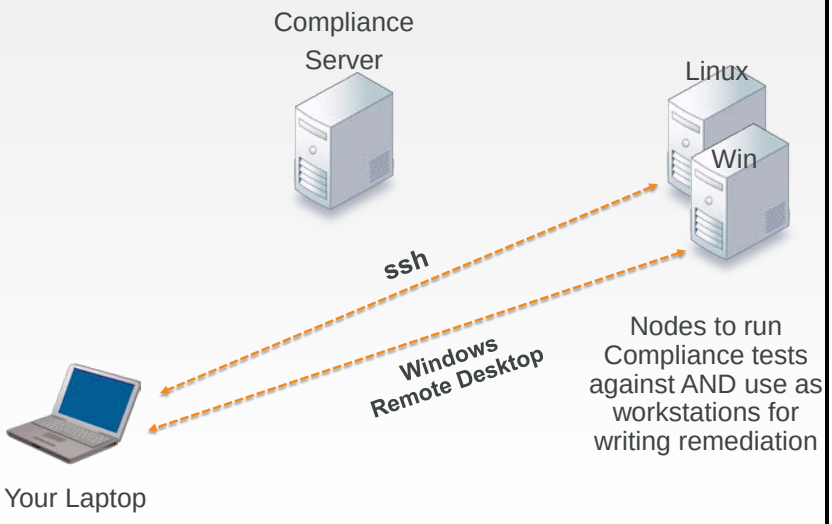
The login credentials for the Linux nodes is chef/chef.
The login credentials for the Windows nodes is Administrator/Cod3Can!

Slide 14

Your Lab Environment for Remediation

You will also log in to your Windows and Linux nodes in order to write remediation and run chef-client in local mode.

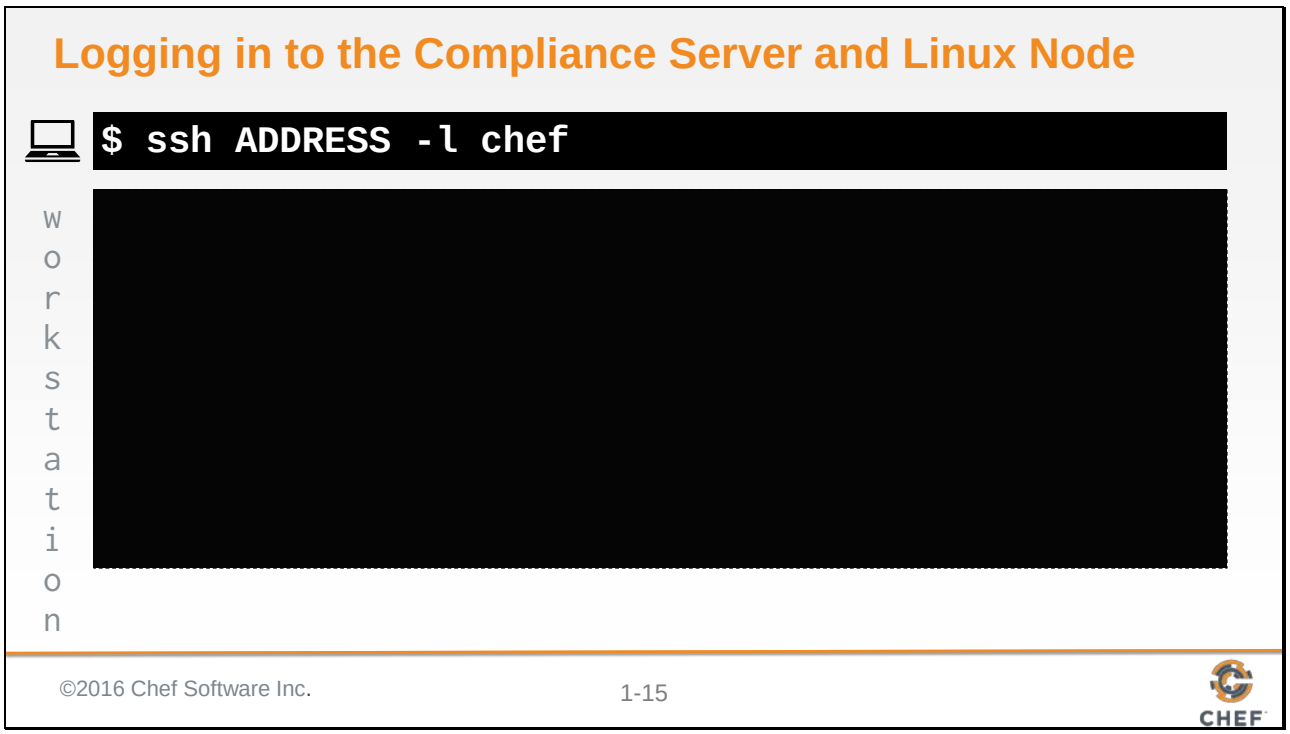
This is so you can use those nodes as virtual workstations while writing remediation.



The dotted lines indicate that those sessions will only be used to write and test remediation. In this scenario, your target nodes will act as virtual workstations.

But all scans will only be run via the Compliance server as indicated in the previous slide.

Slide 15



This is just an explanation. You don't need to log in to these machines at this time.

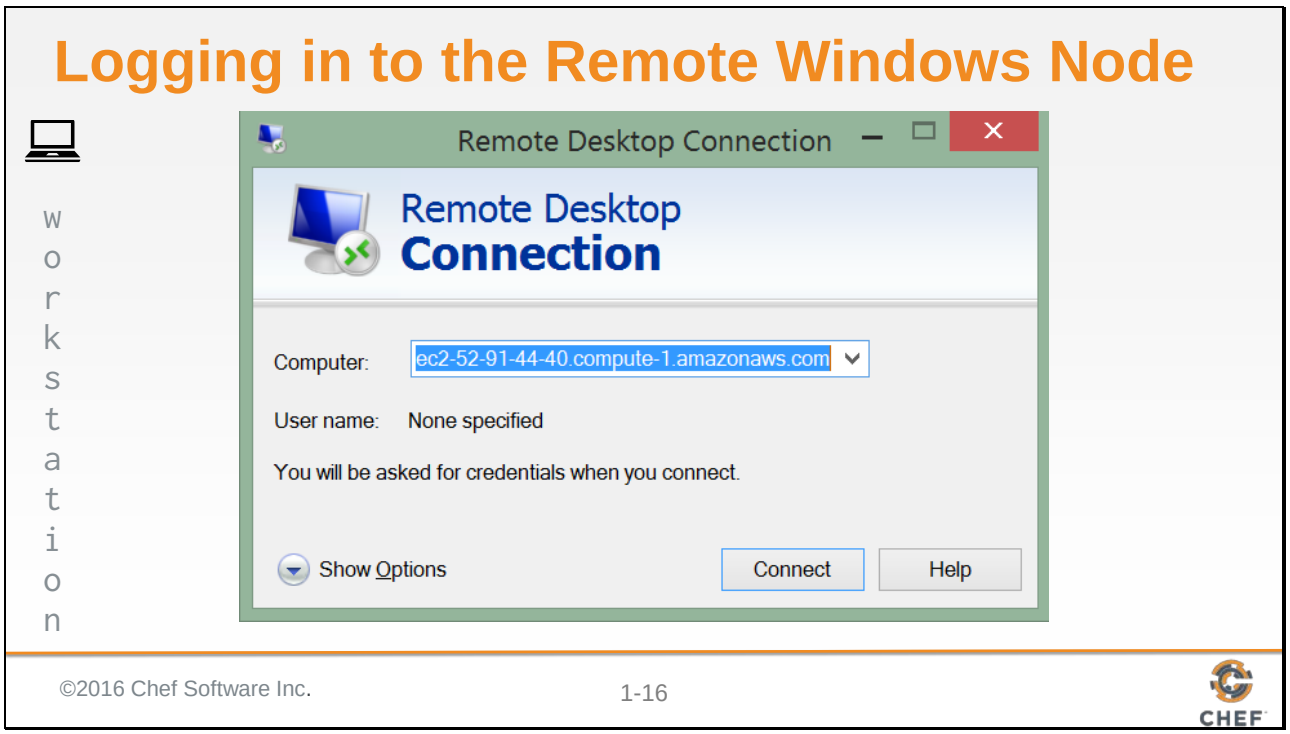
You should use an ssh client like PuTTY or a local command prompt to connect to the remote workstation that we assign to you.

Instead of the command shown in this slide, you could also use this command:

ssh chef@IPADDRESS

For example: ssh chef@52.90.140.22

Slide 16



This is just an explanation. You don't need to log in to these machines at this time.

You should have installed on your laptop a Windows Remote Desktop Connection which you'll only use to write Windows remediation later in this course.

Slide 17

Hands-on Legend

- **GL** or **Group Lab**: All participants and the instructor do this task together with the instructor often leading the way and explaining things as we proceed.
- **Lab**: You perform this task on your own. 

Slide 18

2: Installing Chef Compliance

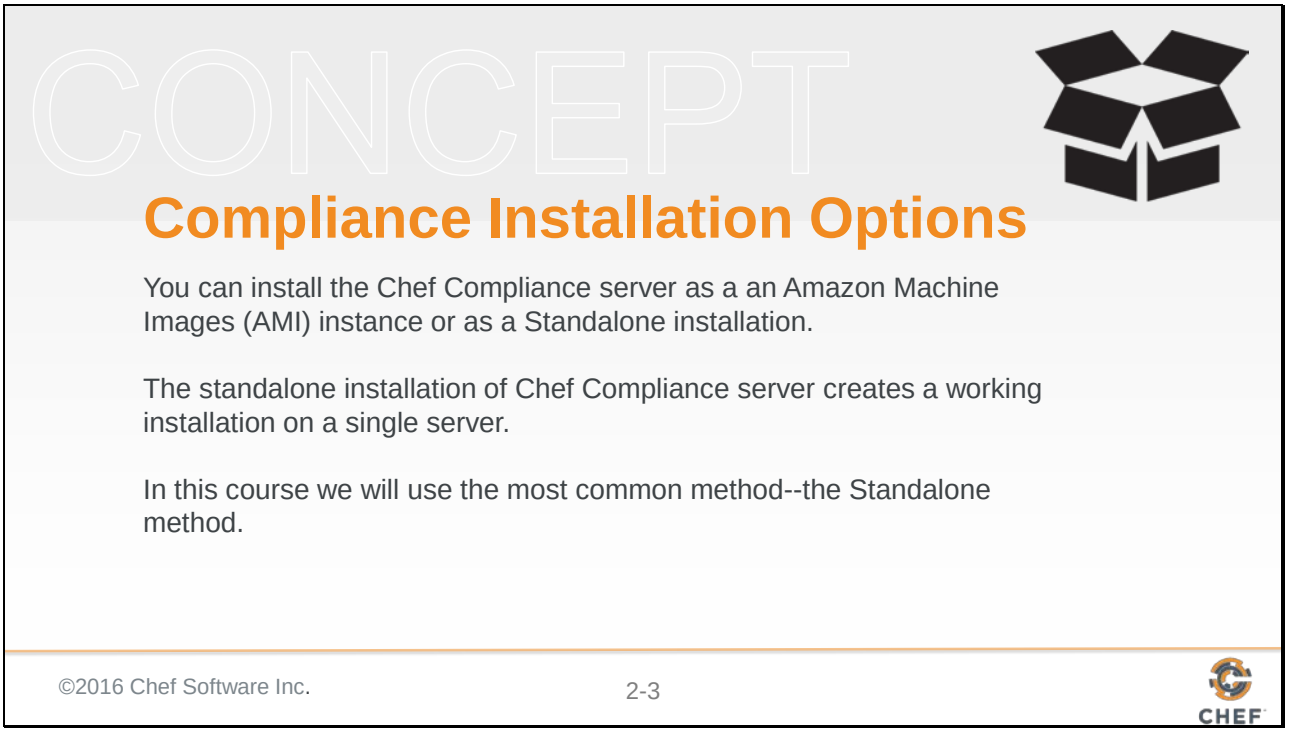
Slide 2

Objectives

After completing this module, you should be able to:

- Install the Chef Compliance server.
- Perform initial configuration of the Compliance server.
- Launch the Compliance Web UI.

Slide 3

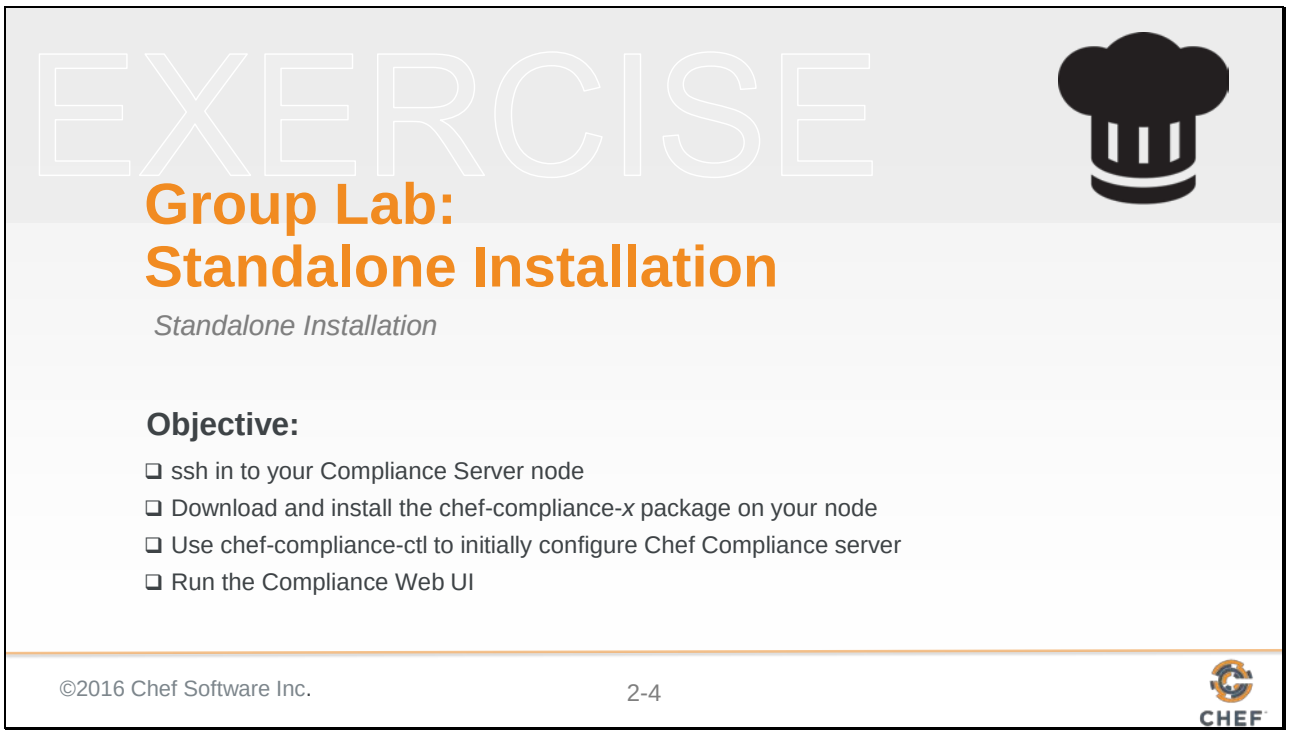


You can install the Chef Compliance server as a an Amazon Machine Images (AMI) instance or as a Standalone installation.

The standalone installation of Chef Compliance server creates a working installation on a single server.

This installation is also useful when you are installing Chef Compliance in a virtual machine, for proof-of-concept deployments, or as a part of a development or testing loop

Slide 4



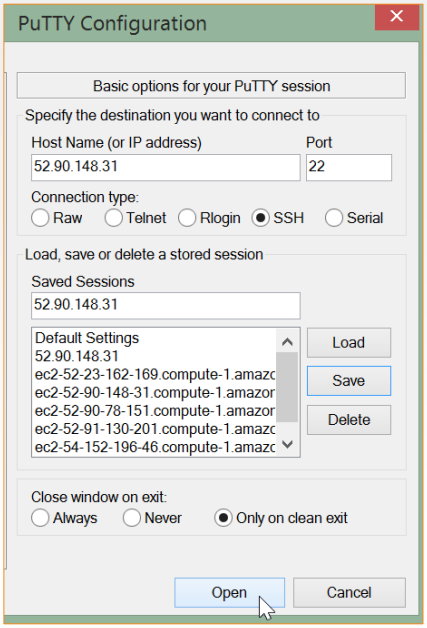
In this group lab you will perform the installation and initial configuration tasks listed in this slide.

Slide 5

GL: Standalone Installation

- ssh in to the node you want to install the Compliance server on.

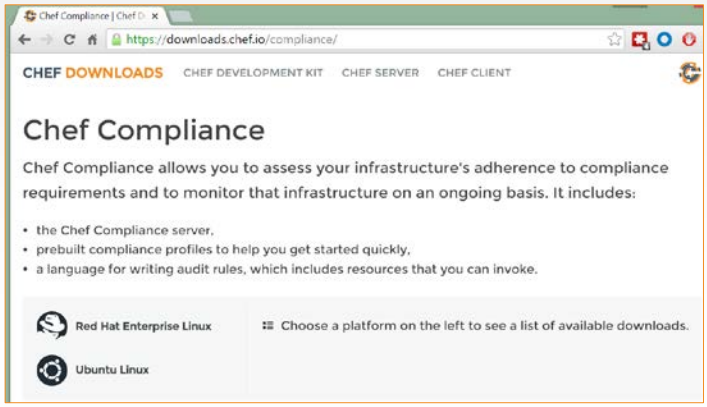
Here is an example of using Putty to do so.



Slide 6

GL: Standalone Installation

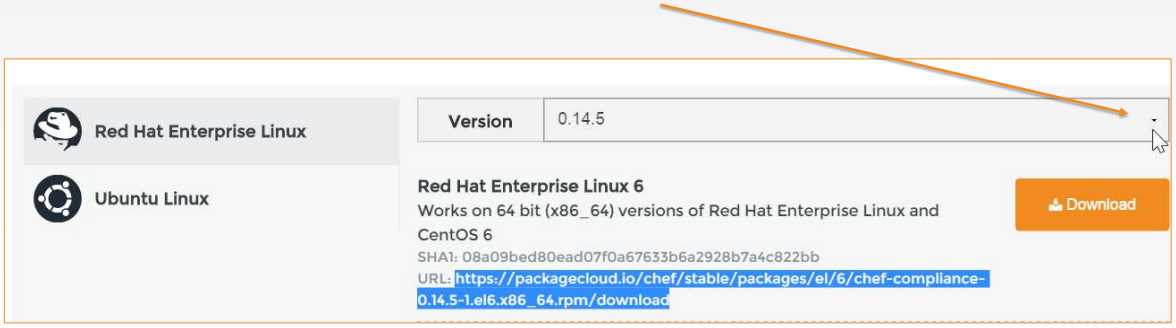
- From your local laptop, use a web browser to navigate to https://downloads.chef.io/compliance/



Slide 7

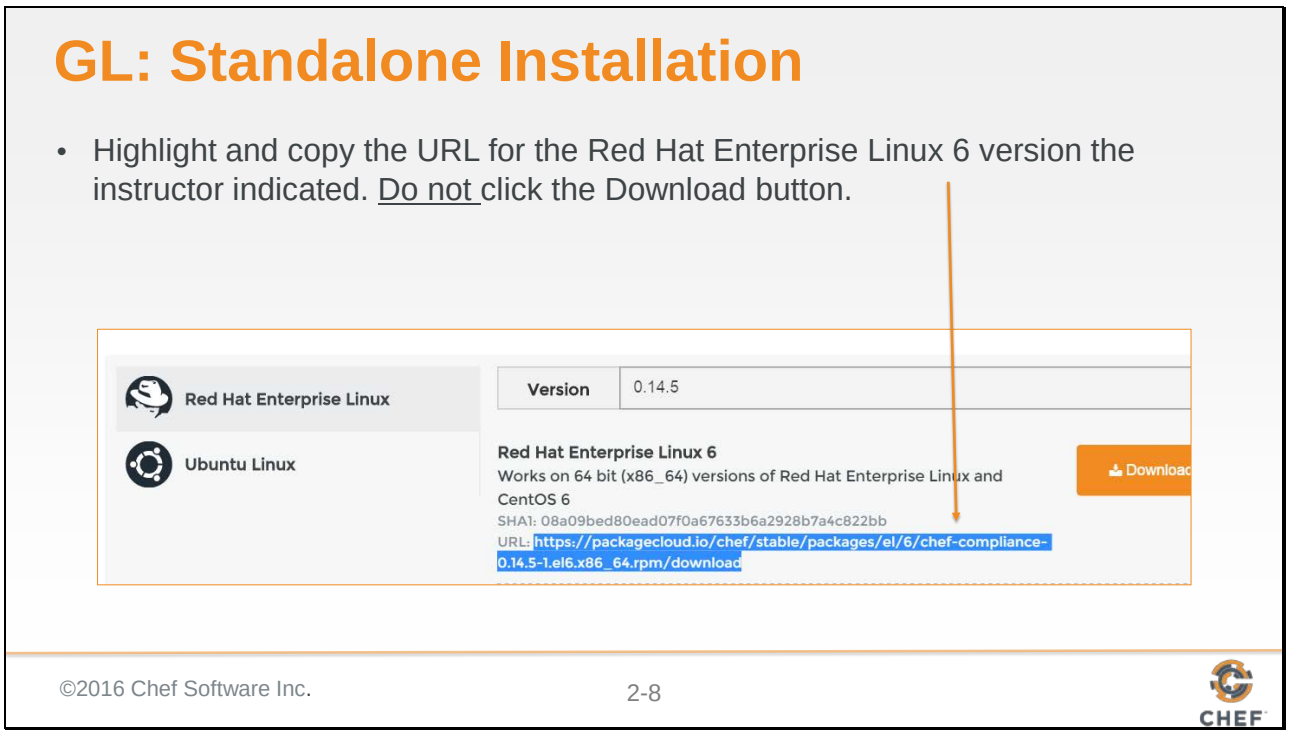
GL: Standalone Installation

- From the resulting web page, click the **Red Had Linux** link and then click the **Version** button.
- Select the version from the **Version** pull-down as indicated by the instructor.



Instructor Note: This course has been tested on Compliance Server v0.14.5.

Slide 8



From the resulting web page, highlight and copy the URL for the **Red Hat Enterprise Linux 6** version Do not click the Download button. We don't want to download this directly to your laptop but instead we will download it directly to our node.

Instructor Note: This course has been tested on Compliance Server v0.14.5.

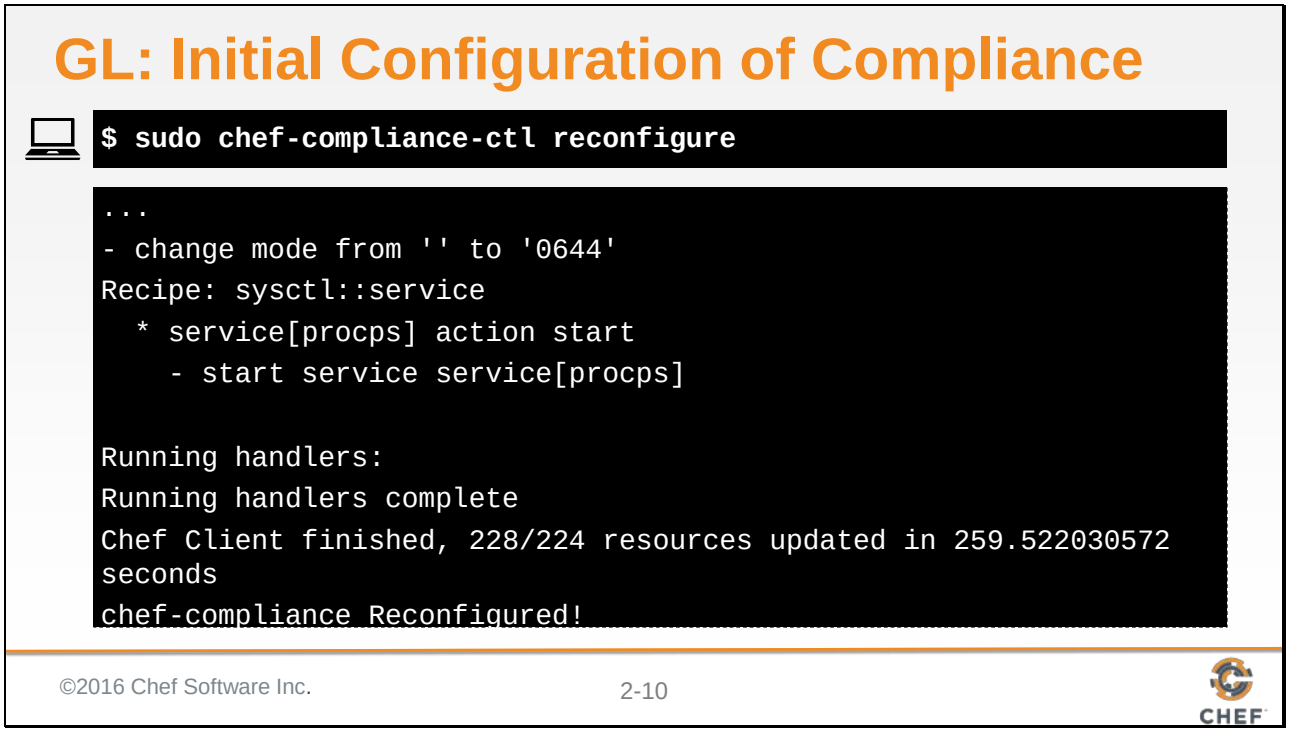
Slide 9

Execute **sudo rpm -Uvh https://packagecloud.io/chef/stable/packages/el/6/chef-compliance-0.9.5-2.el*X*.x86_64.rpm/download** but using the URL you just copied to download and install the Compliance package. This could take some time.

Note: The compliance package will be named **download** at this point.

Instructor Note: The X in this part of the file name - 0.9.2-2.elX - was added so students will paste the latest version and not try to copy/paste the above command verbatim. It could take about a minute to download Compliance server.

Slide 10



This reconfiguration process could take 5 or more minutes to complete.

Important: Do not use Ctrl c while this is running.

Slide 11

GL: Configure the Compliance Server

- From your laptop, open a web browser and point it to:

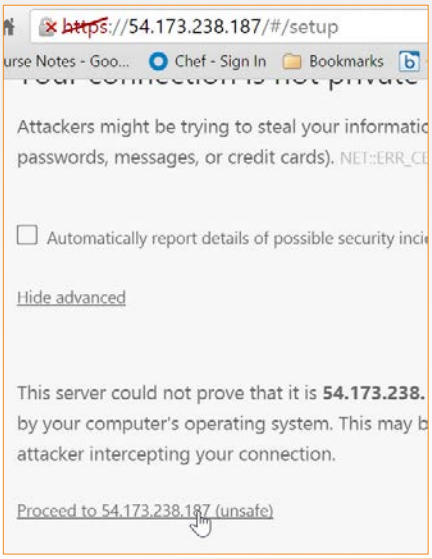
  https://IPADDRESS/#/setup where IPADDRESS is the IP addressed you are using for your Compliance Server.
- For example https://54.173.238.187/#/setup
- Click **Advanced** from the warning page.

Slide 12

GL: Configure the Compliance Server

- Click the **Proceed to 54.173.238.xxx ...** link.

Note: We have not set up SSL so the https strikethrough and warning is fine for now.



We have not set up SSL so the https strikethrough is fine for now. In the workplace you would want to use SSL for connections to your Compliance Web UI.

Slide 13

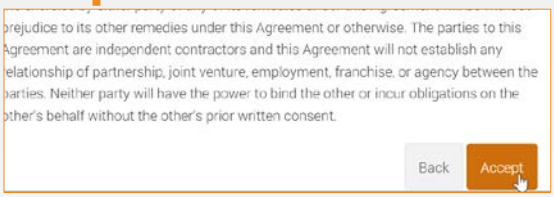
GL: Configure the Compliance Server

- From the resulting page, click the **Setup Chef Compliance** button.

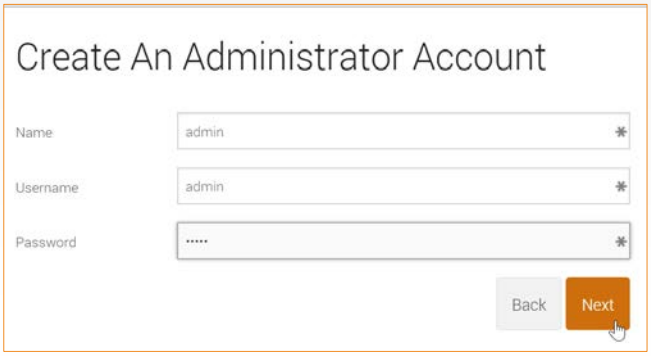
Slide 14

GL: Configure the Compliance Server

- Accept the Online Master License and Services Agreement.



- Create an Administrator Account using **admin/admin** for credentials.

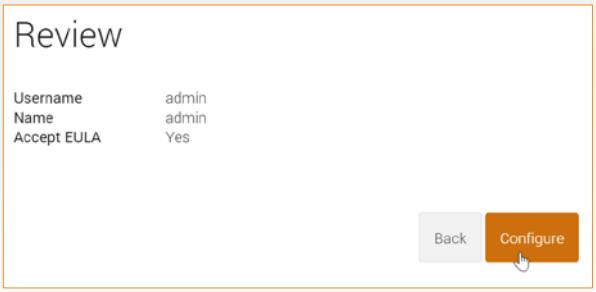


In the workplace you should of course create a more secure password.

Slide 15

GL: Configure the Compliance Server

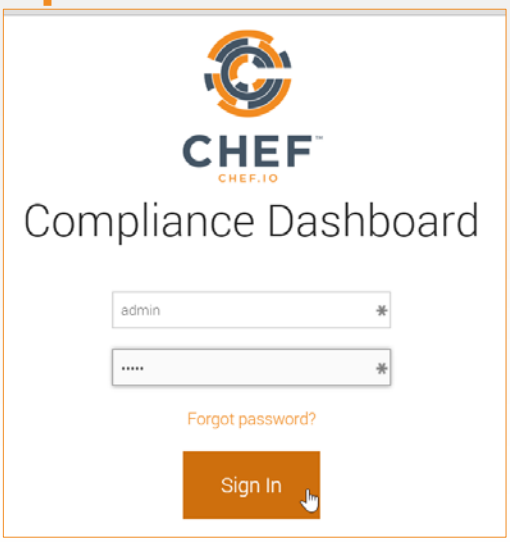
- On the resulting page click the Configure button.



Slide 16

GL: Configure the Compliance Server

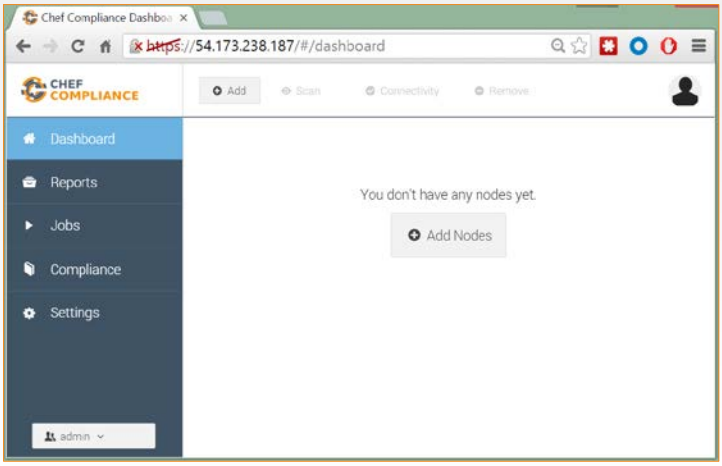
- On the resulting page sign in using admin/admin.



Slide 17

GL: Configure the Compliance Server

You should now see an empty Compliance Dashboard.



In the workplace you would want to use SSL for connections to your Compliance Web UI.

https://docs.chef.io/install_compliance.html > Configure SSL

Slide 18

CONCEPT

Compliance Upgrades

If at some point you want to install a newer version of Chef Compliance, you can simply install the new version over the old version as follows:

1. sudo rpm -Uvh https://packagecloud.io/chef/stable/packages/<*rest of filename*>
2. sudo chef-compliance-ctl reconfigure
3. sudo chef-compliance-ctl restart
4. Relaunch the Compliance Web UI

CHEF

If you needed to downgrade Compliance to a previous version, the `--oldpackage' option will do that for you. In the following example, the `https://package...` would be the version of the old package that you want to downgrade to:

sudo rpm -Uvh **--oldpackage** https://packagecloud.io/chef/stable/packages/el/6/chef-compliance-0.9.6-1.el6.x86_64.rpm/download

...and then run these commands:
sudo chef-compliance-ctl reconfigure
sudo chef-compliance-ctl restart

Slide 19

Review Questions

1. What can this command be used for?

   `` `sudo chef-compliance-ctl reconfigure` ``

2. What does this command do?

   `` `sudo rpm -Uvh https://packagecloud.io/chef/stable/packages/<rest of filename>' ``

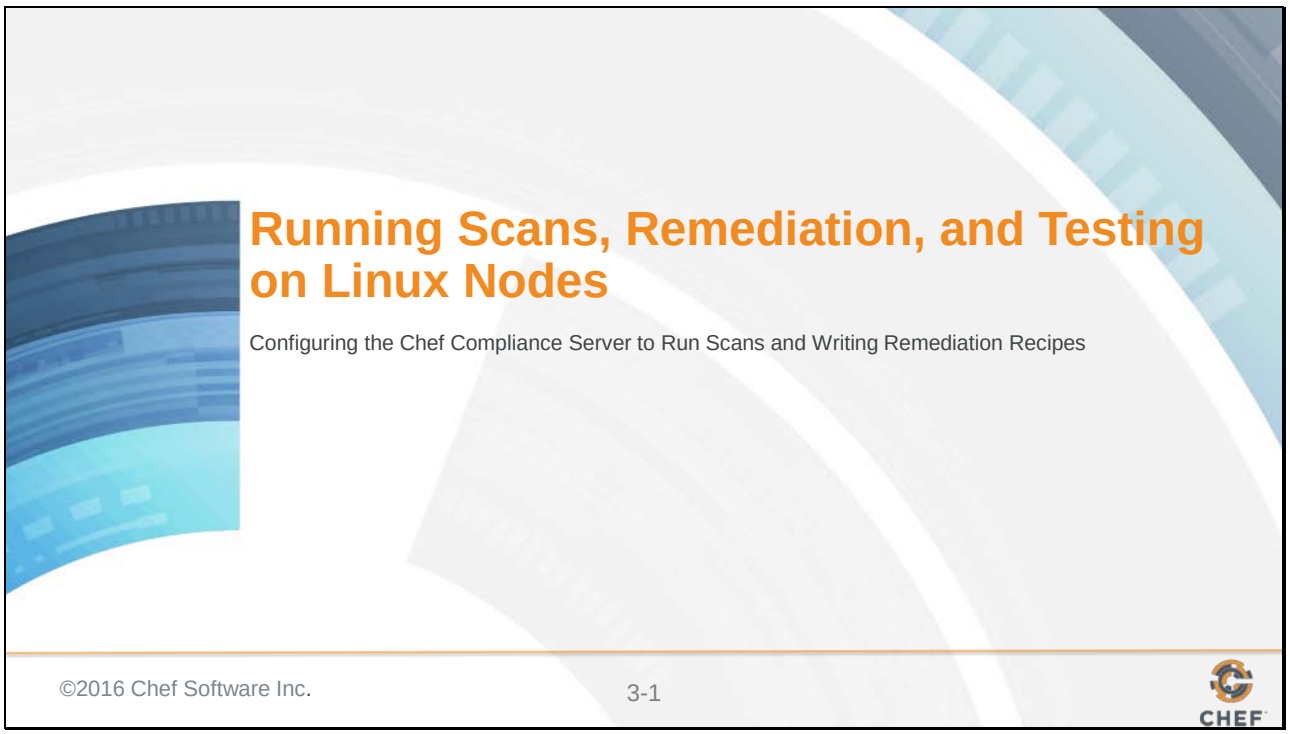
Instructor note answers:

1.It can be used to initially configure a new installation or to reconfigure an upgrade or downgrade of the Compliance Server software.

2. It downloads AND installs the Compliance Server software.

Slide 20

3: Running Scans, Remediation, and Testing on Linux Nodes



Instructor Note: Answers to quizzes are contained in Instructor Notes found below each quiz slide so participants won't see the answers.

Slide 2

Objectives

After completing this module, you should be able to:

- Add a node to test for compliance.
- Run a Compliance scan.
- Test for compliance with InSpec from the command line interface.
- Remediate a compliance issue.
- Use Test Kitchen to test your remediation.
- Test for compliance with InSpec from the CLI
- Rescan the node and ensure compliance.

Slide 3

CONCEPT

Adding a Node to Scan

To add a node you'll need:

- The IP address or FQDN of the nodes to be tested.
- Access configuration (ssh or WinRM).
- The node's username and password OR
- The node's username plus security key pair.

Slide 4

EXERCISE

Group Lab: Adding a Node to Scan

Objective:

- ❑ Add a Linux Node to Scan
- ❑ Test connectivity

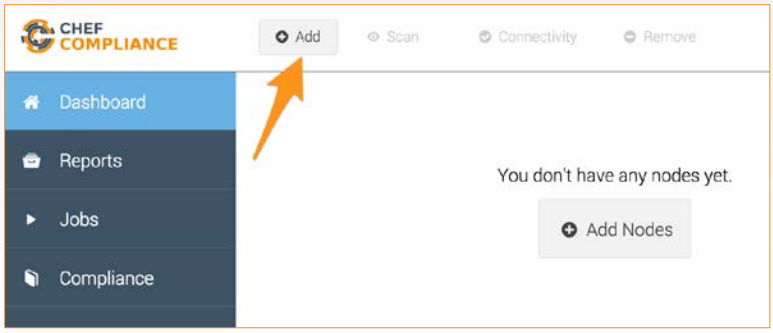
Note: In the next module you will perform the same exercises as in this module but using a Windows node as your target node.

CHEF

Slide 5

GL: Adding a Node to Scan

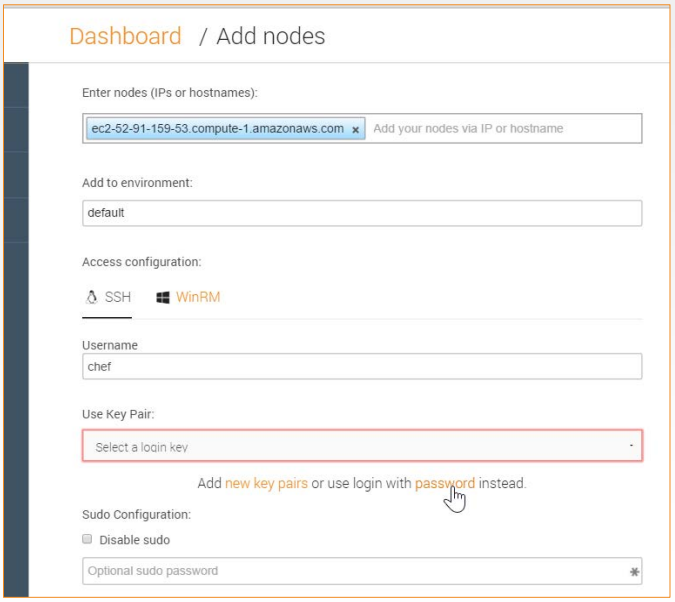
1. From your Chef Compliance Dashboard, click Add Node.



Slide 6

GL: Adding a Node

2. From the resulting page, enter the node's FQDN or IP address.
3. Leave environment blank. A 'default' environment will be used
4. Accept the default **SSH** Access configuration
5. Type **chef** in the **username** field.
6. Click the **password** link as shown in this illustration.



Be sure you are using the hostname of the target node that you noted previously in class.

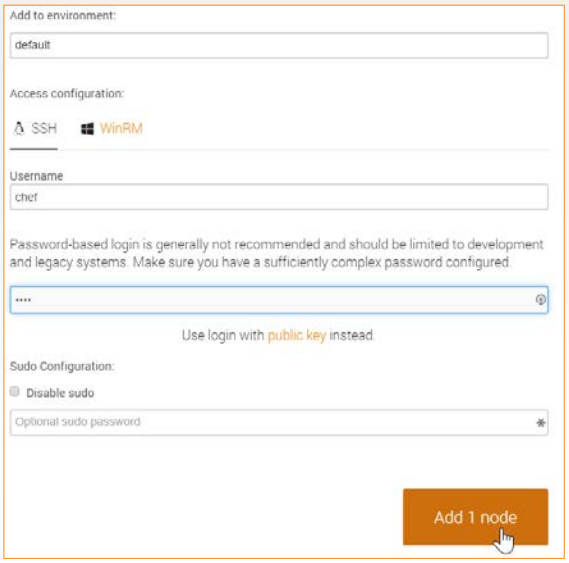
In the workplace, the target node's username and password will likely be different than shown in this example.

We'll discuss using key pair access later in the module.

Slide 7

GL: Adding a Node to Scan

7. Type the password (**chef**) in the password field.
8. Click the **Add 1 node** button as shown in this illustration.

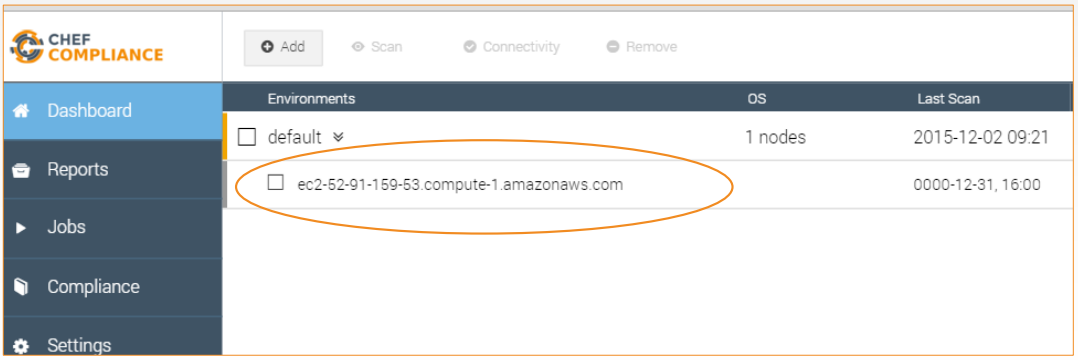


Instructor Note: If a Linux target node's image has /etc/sudoers `Defaults requiretty` uncommented, then the Compliance server won't be able to connect to the target node unless we disable sudo on this page. Once the issue is fixed it should not matter if the target node /etc/sudoers `Defaults requiretty` is uncommented. The Linux AMI used in this course has /etc/sudoers `Defaults requiretty` commented so no worries."

Slide 8

GL: Adding a Node to Scan

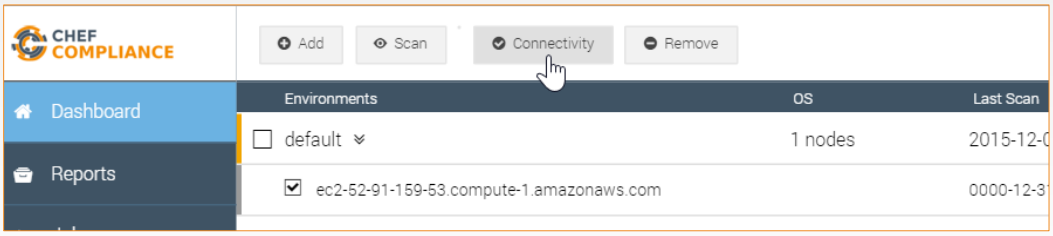
At this point your Compliance Dashboard should list the node you just added.



Slide 9

GL: Testing Connectivity to Your Node

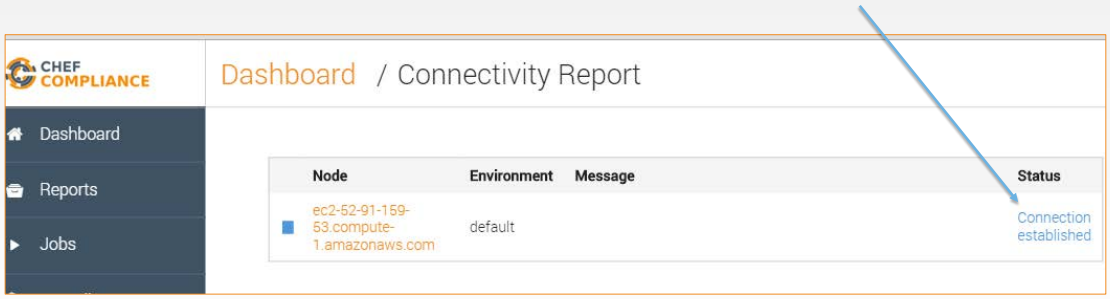
1. Click the **check box** next to your node and then click the **Connectivity** button.



Slide 10

GL: Testing Connectivity to Your Node

The Status column of you node should now indicate **Connection established**.



If your Status column does not indicate `**Connection established**`, please notify the instructor.

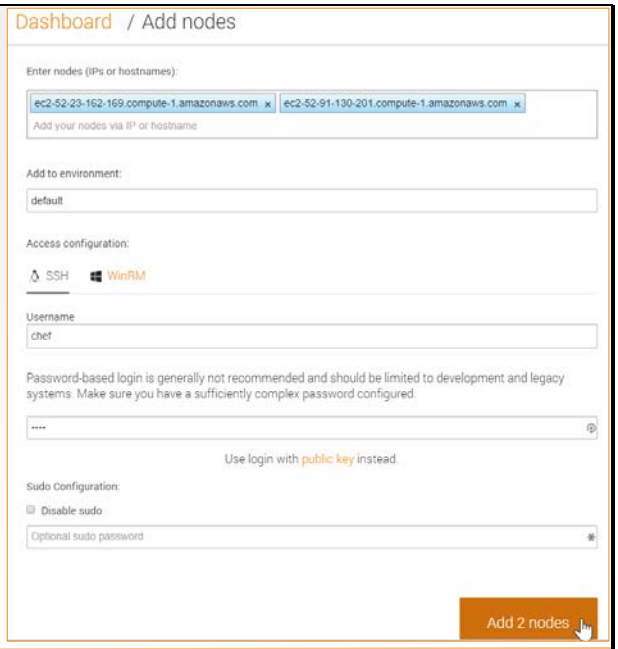
Slide 11

Adding Nodes in Bulk

You could add additional nodes by simply repeating the previous steps.

You could also add a number of nodes at once by separating each hostname or IP address with a comma or a space, as shown in this illustration.

Chef Compliance also supports bulk loading of nodes via API.



CHEF

As you may have noticed, you could add additional nodes by simply repeating the previous steps.

You could also add a number of nodes at once by separating each hostname or IP address with a comma or a space, as shown in this illustration.

Chef Compliance also supports bulk loading of nodes via API.

Slide 12

Adding Nodes in Bulk via API

After class you can go to the following link.

The resulting kitchen_sink.rb will step you through how to upload nodes in bulk.

```
### Script to export Chef Server nodes and import them to Chef Compliance
### Change the 'api_url', 'api_user' and 'api_pass' variables below
### Go to your chef-repo and check Chef Server access first
# cd chef-repo; knife environment list
### Save this Ruby script as kitchen_sink.rb and run it like this:
# cat kitchen_sink.rb | knife exec
### Chef Compliance API docs: https://docs.chef.io/api_compliance.html

require 'json'
require 'uri'
require 'net/http'
require 'openssl'

# This extracts data from the Chef Server. Auth done by `knife exec`
# Change loginKey and any other details that will be posted to the Chef Compliance API:
nodes_array = []
nodes.find('*:*') { |n|
  nodes_array << { id: n.name,
                   name: n.name,
```

https://gist.github.com/alexpop/01b0bba8d259adeee320

CHEF

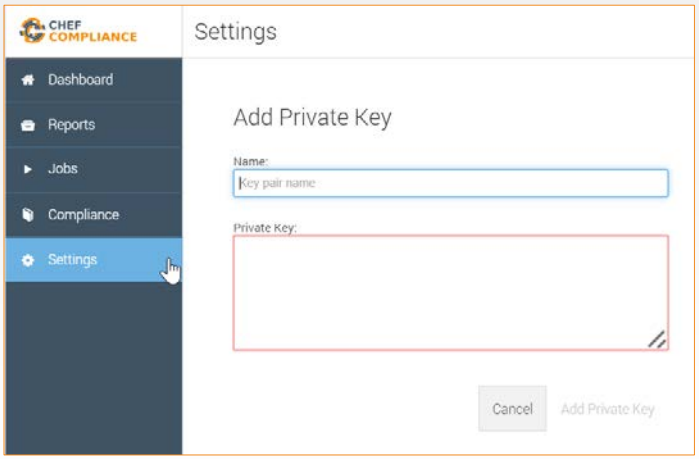
Instructor Note: If you have extra time, you can walk the participants through this file.

Slide 13

Private Keys

In the workplace, using a security key would be a more secure method for connecting to nodes than using the password method.

By clicking **Settings > Add Private Key** you will see where to paste a private key.



In the workplace, using security key pairs would be a more secure method for connecting to nodes than using the password method we are using in class.

By clicking `**Settings > Add Private Key**` you will see where to paste your private key.

Slide 14

CONCEPT

Running Compliance Scans

You can run Compliance scans on demand or schedule them to run at a later time.

Chef Compliance maintains profiles as a collection of individual controls that comprise a complete audit.

As mentioned previously, Chef Compliance comes with a few reference profiles of various compliance policies that you can leverage or use as examples to create your own.

CHEF

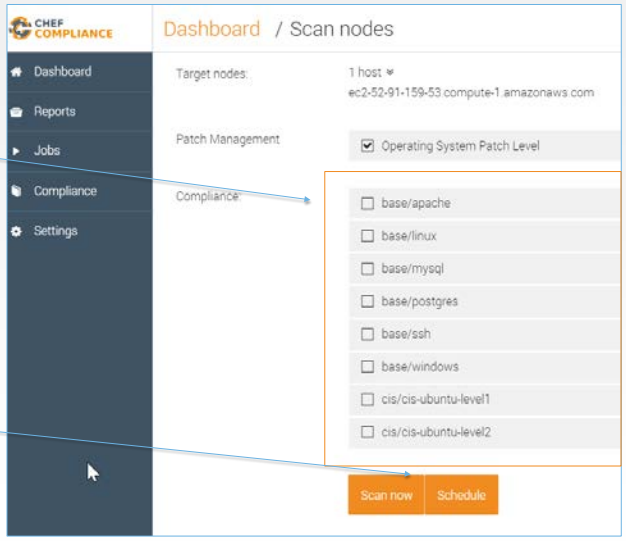
Slide 15

Compliance Profiles Used in Scans

This image shows the default Compliance Profiles as accessed from the Scan Nodes page.

You should be thoughtful with which profiles choose.

Notice how you can also choose to run a scan on demand or schedule a scan.



This image shows the default Compliance Profiles as accessed from the Scan Nodes page. This page displays when you select nodes to scan and then click the Scan button.

You'll access the profiles in a moment. These profiles determine what will be scanned on your nodes.

You should be thoughtful with which profiles choose since the more you choose to run, the longer it will take to execute the scan.

Notice how you can also choose to run a scan on demand (Scan now) or schedule a scan to run at a later time.

Slide 16

EXERCISE

Group Lab: Running a Scan

Objective:

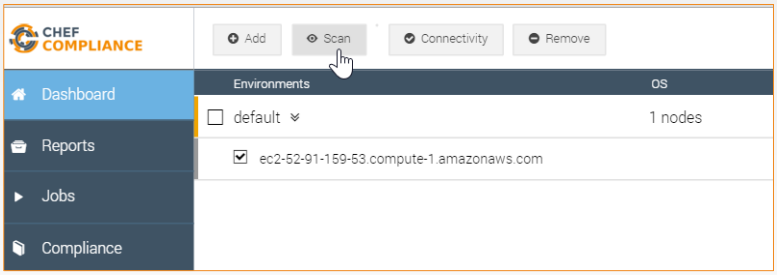
- ❑ Run a Compliance scan.
- ❑ View the output of a scan.

CHEF

Slide 17

GL: Running a Scan

1. Click the **check box** next to your node and then click the **Scan** button.

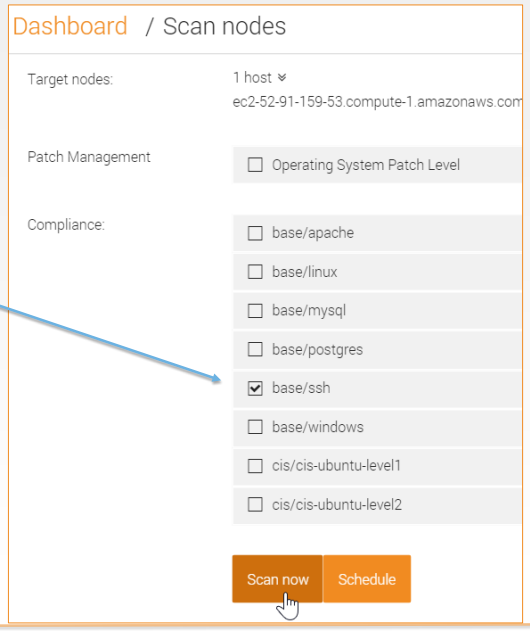


Slide 18

GL: Running a Scan

2. From the resulting page, check the **base/ssh** profile and uncheck any other check boxes.

3. Click the **Scan now** button.

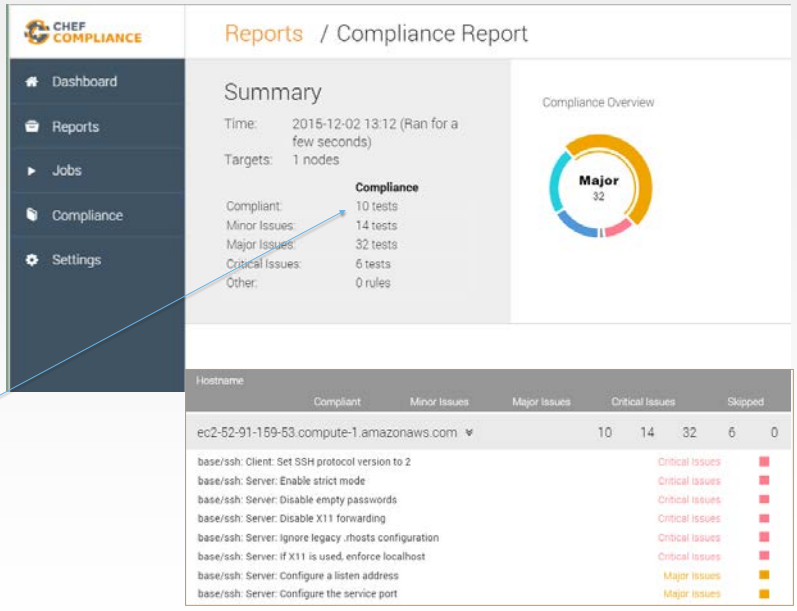


Slide 19

Scan Results

A Compliance Report should now display and your scan results should be similar to that shown here.

Notice how in the upper Summary section in this example, 10 tests were compliant and 6 tests show critical issues with ssh.



There are also 6 critical issues related to ssh on the target node. Your results may be slightly different than this example.

Instructor Note: This and the following slide should be used for a discussion of the scan results. The group exercise continues after that.

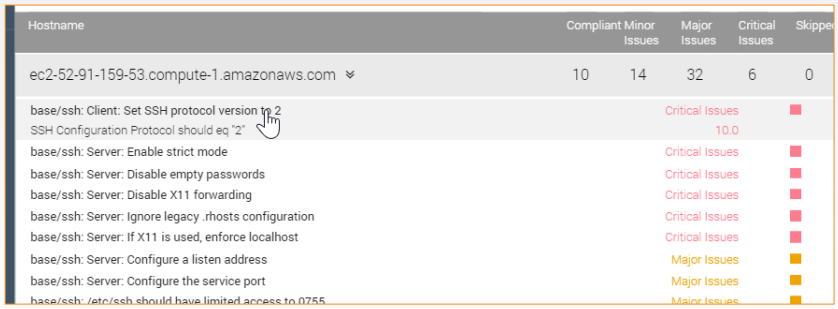
Slide 20

Scan Results

The bottom half of the Compliance Report shown here has a table of details of test results.

These are sorted by severity.

If you click an issue as shown here, a bit more information about the issue displays.



The bottom half of the Compliance Report has a table of details of test results.

These are sorted by severity so the critical issues are listed at the top and the compliant items are at the bottom of the list.

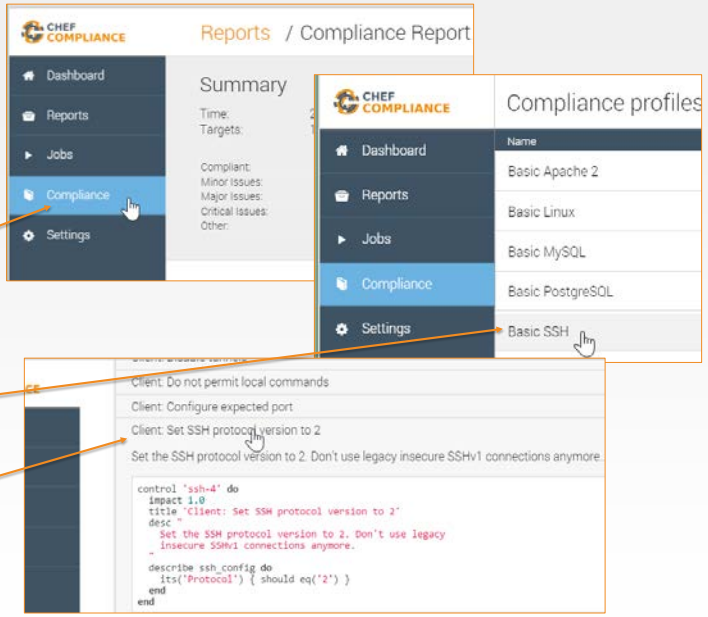
If you click an issue as shown here, a bit more information about the issue displays, but that's not really telling us much.

Slide 21

GL: Profile

To view the InSpec code that comprises this profile, do the following:

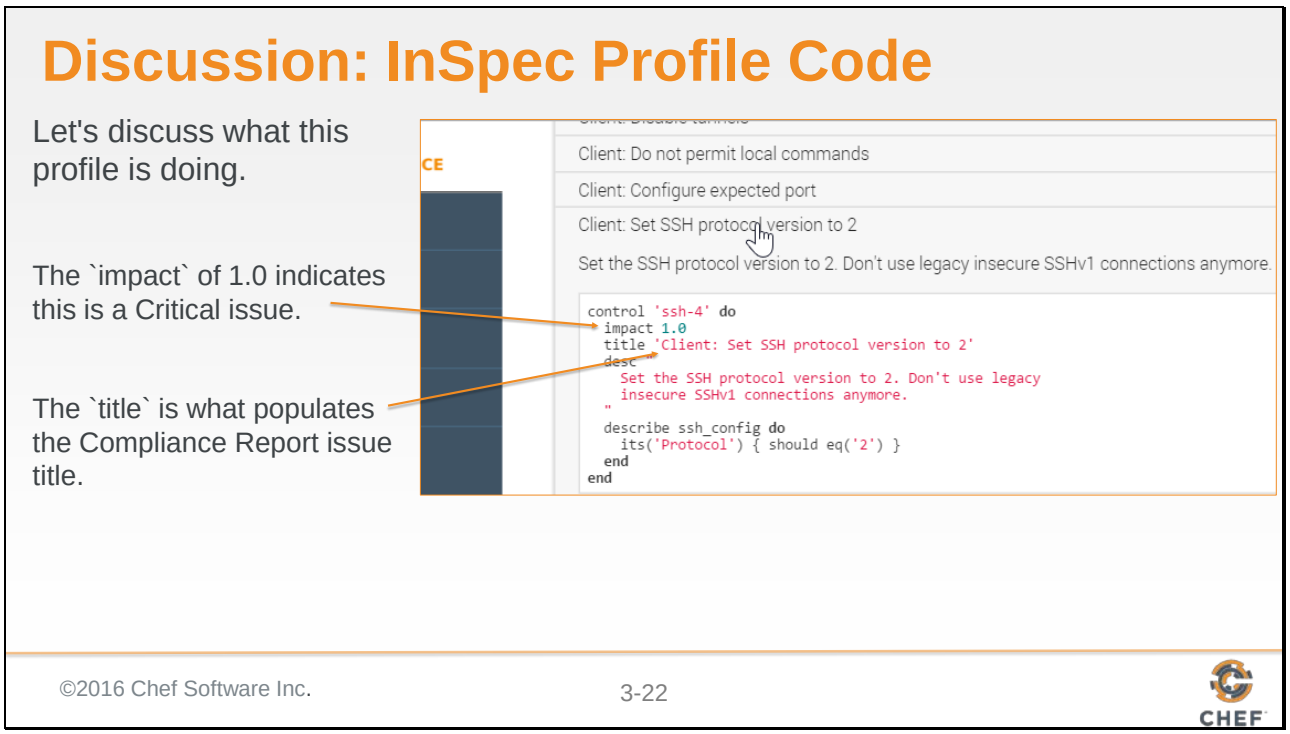
1. Click the **Compliance** button.
2. Click the relevant profile (**Basic SSH**).
3. Scroll down and click the `**Set SSH protocol version to 2**` profile.



CHEF

Instructor Note: Now we continue the group Lab but you should stop as needed to explain what this code means.

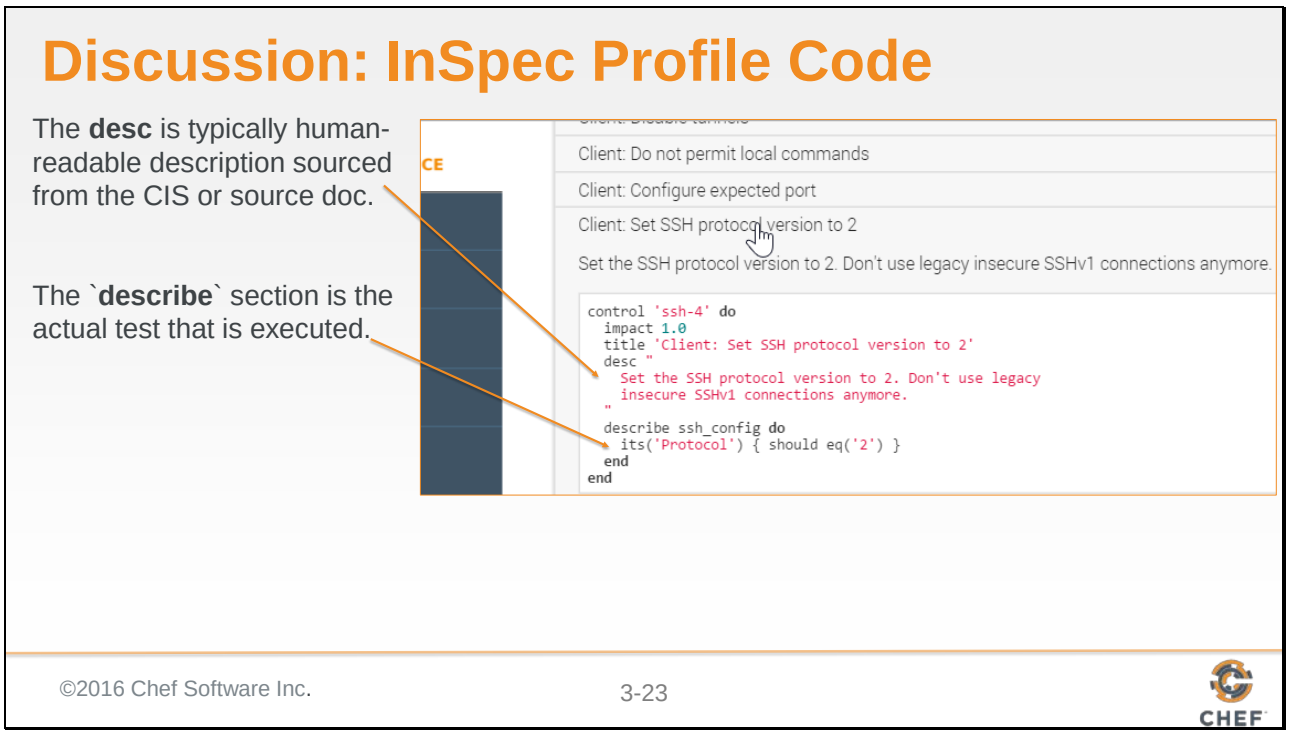
Slide 22



Let's discuss what this profile is doing.

The impact of 1.0 indicates this is a critical issue if it the scanned node violates what is in this code. We'll discuss severity mapping in a moment.

Slide 23



The **desc** is typically human-readable description sourced from the CIS or source doc.

The describe value is the actual test. In this case, this is saying the protocol for `ssh_config` Protocol should be `2`. If the actual value from the node is not Protocol 2, the Critical issue is reported as in this case.

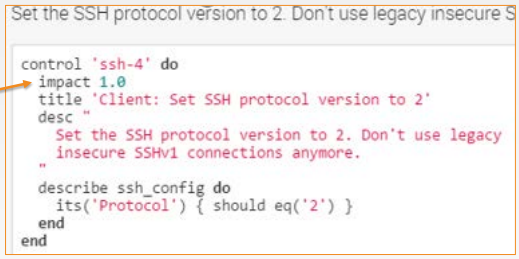
So when you run a scan, the Compliance Server connects to the node using the configuration we specified, in this case ssh, and then it will run this set of code, this InSpec control, on that node. The Compliance Server translates the InSpec code into ssh commands when it transmits across the wire.

No agent or client is required to be running on the target node for this work.

Slide 24

Compliance Profile Severity Mapping

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



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

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

Here is the current mapping of Compliance Profile **impact** numbering to severity.

The image at the top-right shows a Compliance Profile's impact numbering.

The table at the bottom-left shows the current mapping of Compliance Profile impact numbering
to severity.

The image at the bottom-right is an example of the severities listed in the reports in the Compliance web UI.

The mapping is currently analogous to the Common Vulnerability Scoring System (CVSS) framework, which can be viewed via the link at the bottom of this slide.

This mapping will be made configurable in the future.

Slide 25

Example: Node's ssh config

```
$ more /etc/ssh/ssh_config
```

```
#   IdentityFile ~/.ssh/identity
#   IdentityFile ~/.ssh/id_rsa
#   IdentityFile ~/.ssh/id_dsa
#   Port 22
#   Protocol 2,1
#   Cipher 3des
```


CHEF

Slide 26

CONCEPT

Let's Remediate the Issue

Now that we've identified the ssh version issue, let's write a recipe on the target node to remediate the issue.

Then we'll run the compliance scan again to see if we successfully remediated the issue.

Note: In this course we will write a recipe directly on the node that we're running scans on. Of course in a production environment you will likely write such recipes locally and upload them to Chef Server. Then the nodes would convergence the recipes on their next chef-client run.

CHEF

Slide 27

EXERCISE

GL: Remediating the Issue

Objective:

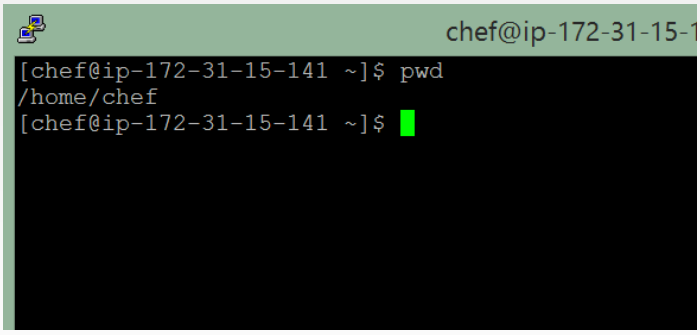
- ❑ Start writing a remediation recipe on that node.
- ❑ Test the recipe with Test Kitchen.
- ❑ Test for compliance with InSpec from the command line interface (CLI)
- ❑ Converge the recipe.
- ❑ Rescan the node and ensure compliance.

Slide 28

GL: Remediating the Issue

Log in to your **target** node (not your compliance server node) using ssh and ensure you are in the **home directory**.

Note: emacs, nano, and vim/vi are installed on your Linux nodes. Some tips for using them can be found below in your participant guide.



`Emacs:` (Emacs is fairly straightforward for editing files.)

OPEN FILE $ emacs FILENAME
WRITE FILE ctrl+x, ctrl+w
EXIT ctrl+x, ctrl+c

`Nano:` (Nano is usually touted as the easiest editor to get started with editing through the command-line.)

OPEN FILE $ nano FILENAME
WRITE (When exiting) ctrl+x, y, ENTER
EXIT ctrl+x

`VIM:` (Vim, like vi, is more complex because of its different modes.)

OPEN FILE $ vim FILENAME
START EDITING i
WRITE FILE ESC, :w
EXIT ESC, :q
EXIT (don't write) ESC, :q!

GL: Create and Change to a ‘cookbooks’ Directory

```
$ mkdir -p cookbooks
$ cd cookbooks
```

From the home directory, create a `**cookbooks**` directory and navigate into it.

Slide 30

GL: Create an SSH Cookbook

```
$ chef generate cookbook ssh
```

```
Compiling Cookbooks...
Recipe: code_generator::cookbook
  * directory[/home/chef/cookbooks/ssh] action create
    - create new directory /home/chef/cookbooks/ssh
...
- create new file /home/chef/cookbooks/ssh/recipes/default.rb
    - update content in file
/home/chef/cookbooks/ssh/recipes/default.rb from none to b702c7
    (diff output suppressed by config)
```

Slide 31

GL: Create an SSH Client Recipe

```
$ chef generate recipe ssh client
```

```
Compiling Cookbooks...
Recipe: code_generator::recipe
  * directory[./ssh/spec/unit/recipes] action create (up to date)
  * cookbook_file[./ssh/spec/spec_helper.rb] action
create_if_missing (up to date)
...
- create new file ./ssh/recipes/client.rb
    - update content in file ./ssh/recipes/client.rb from none to
9c833a
    (diff output suppressed by config)
```


In this step, instead of modifying the default recipe, we will create a new `**ssh client**` recipe.

This is because a default ssh cookbook probably affects an ssh server config and ssh client config and we only want to affect an ssh client.

Slide 32

GL: Create an SSH Config Template

```
$ chef generate template ssh ssh_config.erb -s /etc/ssh/ssh_config
```

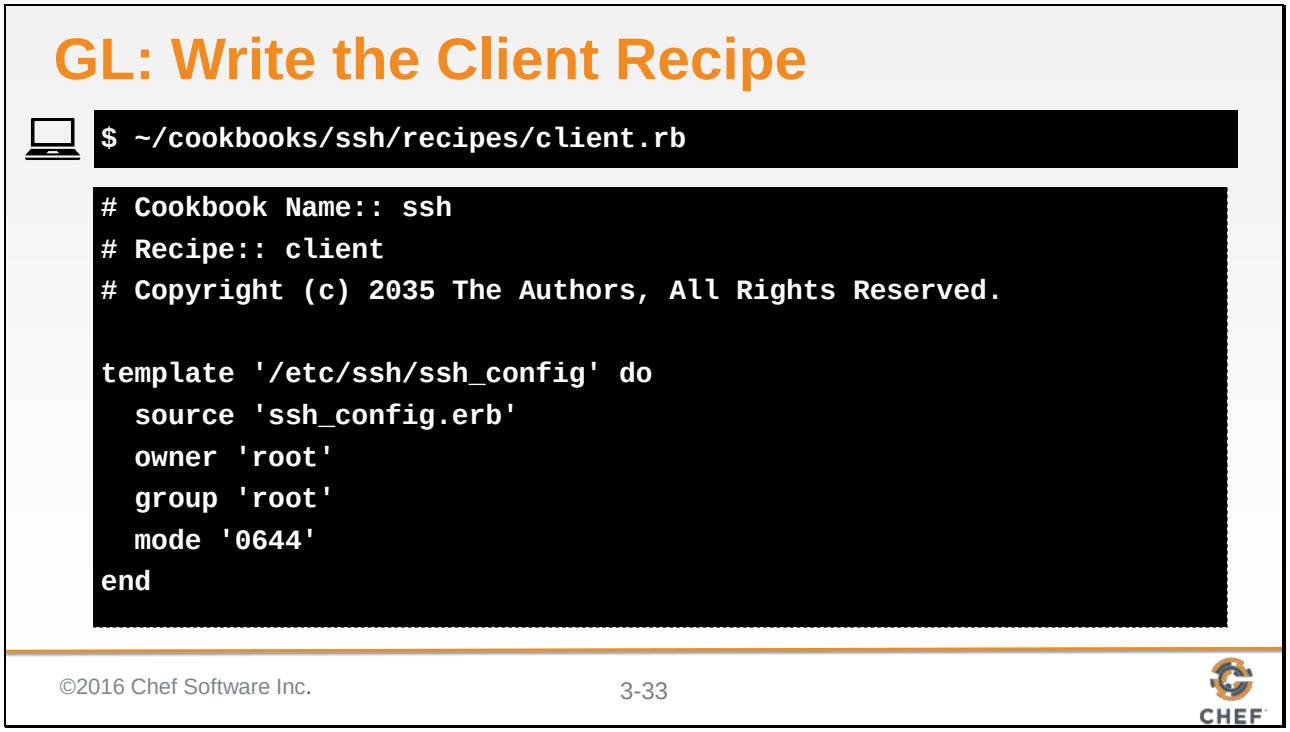
```
Compiling Cookbooks...
Recipe: code_generator::template
  * directory[./ssh/templates/default] action create
    - create new directory ./ssh/templates/default
  * file[./ssh/templates/default/ssh_config.erb] action create
    - create new file ./ssh/templates/default/ssh_config.erb
    - update content in file
./ssh/templates/default/ssh_config.erb from none to 86eb9b
    (diff output suppressed by config)
```


In this step, we want to create a template file to manage our `ssh_config` file.

The `-s` option in this command takes the contents of the current **`/etc/ssh/ssh_config`** file and places it in the **`ssh_config.erb`** file.

Instructor Note: At this time you might want to show the class the contents of **`/home/chef/cookbooks/ssh/templates/default/ssh_config.erb`** to illustrate how the contents of the current **`/etc/ssh/ssh_config`** file is now in the **`ssh_config.erb`** file.

Slide 33



Edit the `**~/cookbooks/ssh/recipes/client.rb**` file and add the contents shown here.

Slide 34

EXERCISE

GL: Testing the Recipe

Objective:

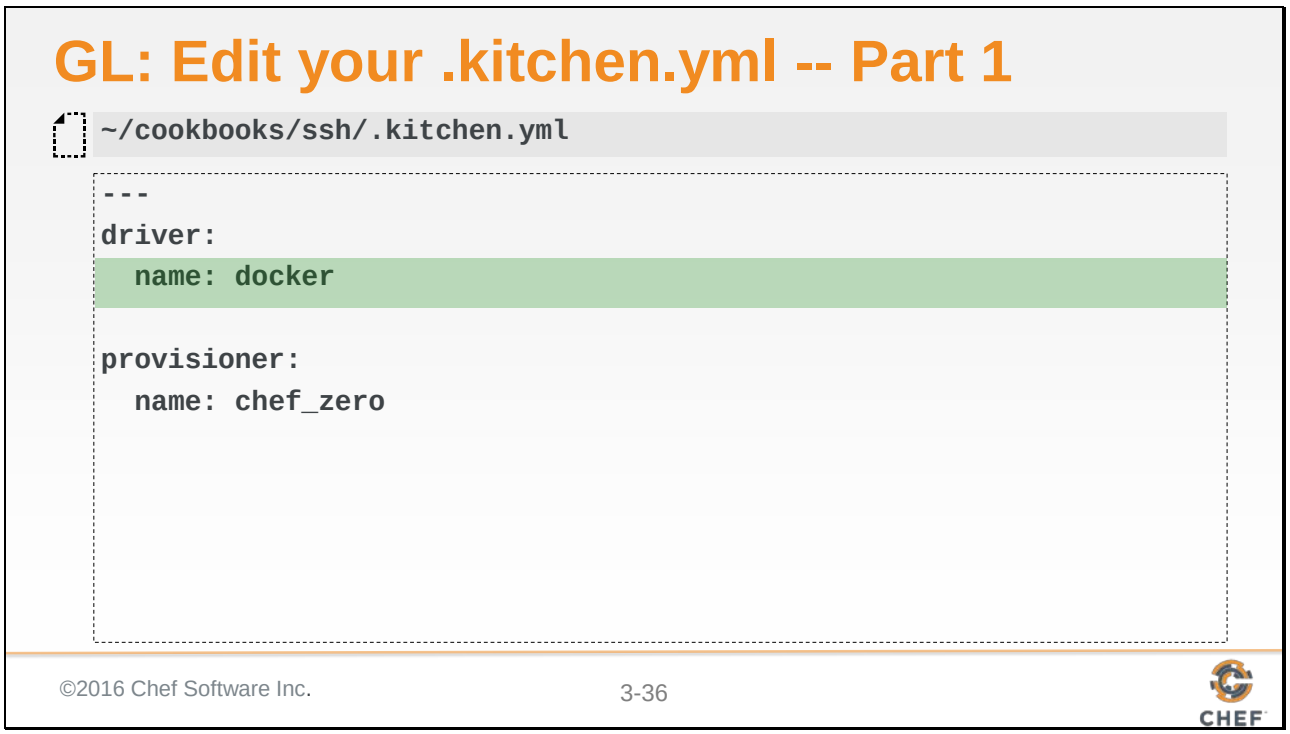
- ✓ Write a remediation recipe on that node.
- ❑ Test the recipe with Test Kitchen.
- ❑ Test for compliance with InSpec from the command line interface (CLI)
- ❑ Converge the recipe.
- ❑ Rescan the node and ensure compliance.

CHEF

Slide 35

To test your recipe, first navigate to where the recipe lives.

Slide 36



Edit your `.kitchen.yml` as shown here and on the following slide.

Because our node is an EC2 AWS instance, we need to change the driver from vagrant to docker. docker should already be installed on the EC2 AWS training instances.

Slide 37

GL: Edit your .kitchen.yml -- Part 2

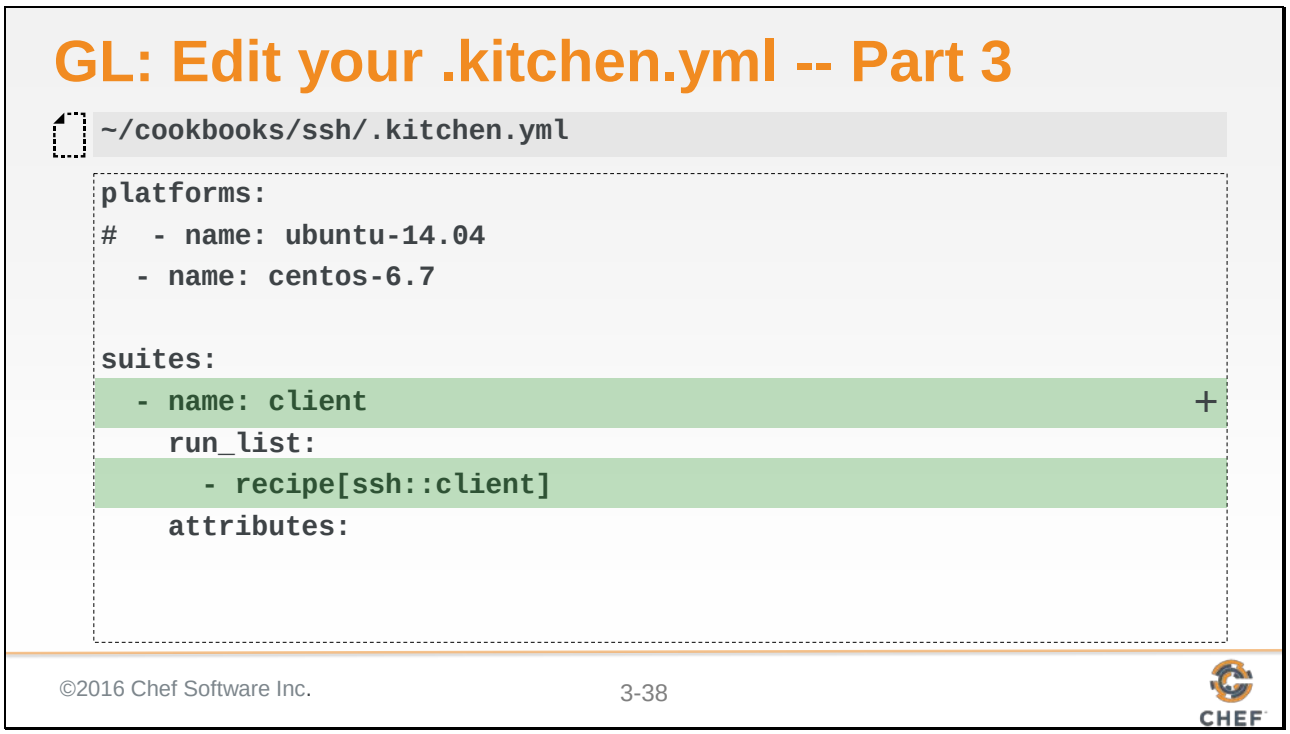
~/cookbooks/ssh/.kitchen.yml

```
platforms:
#  - name: ubuntu-14.04
  - name: centos-6.7

suites:
  - name: default
    run_list:
      - recipe[ssh::default]
    attributes:
```


Also comment the ubuntu platform line and change the centos platform to `centos-6.7', which should be the version running on the training instance.

To confirm the centos release, you could execute `more /etc/*-release`
::::::::::::::
/etc/centos-release
::::::::::::::
CentOS release 6.7 (Final)

Slide 38

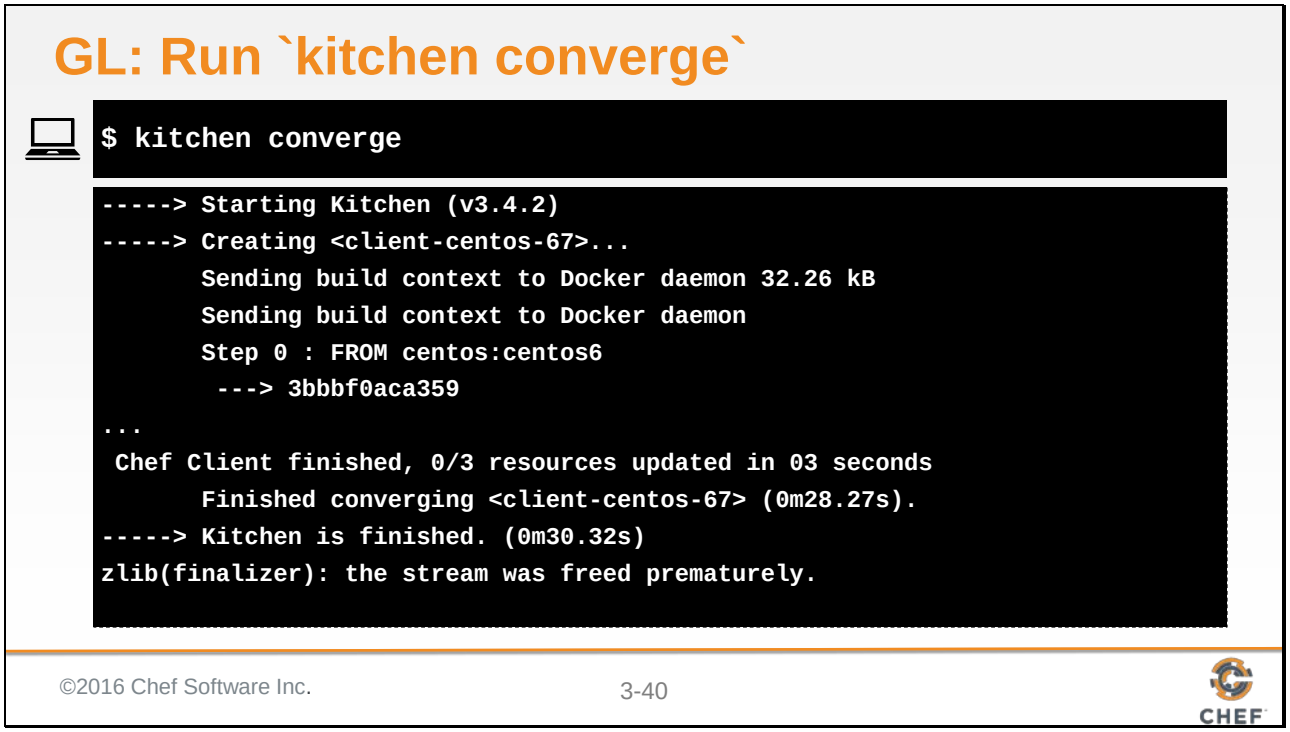


Finally, change the suites name to `client' and the run_list recipe name to `ssh:client`.
run_list:
- recipe[ssh::client]

Slide 39

Now run `kitchen list` from the ~/cookbooks/ssh directory. This command will tell you if you have a typo in your `.kitchen.yml`.

Slide 40



Now run `kitchen converge` from the ~/cookbooks/ssh directory.

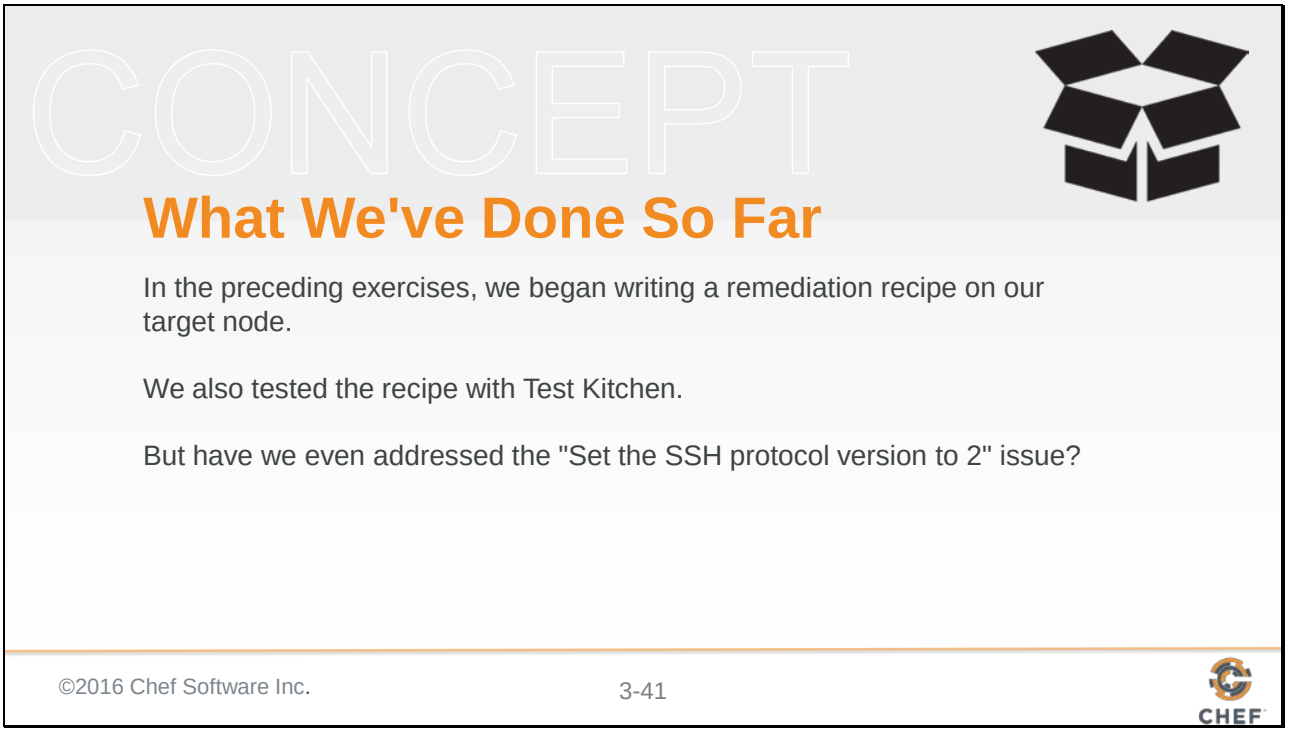
`kitchen converge` will:

- Launch a docker container.
- Place the cookbook into the docker container.
- Install chef-client in the docker container.
- Run chef zero (i.e., chef-client in local mode) across the client recipe.

The end result will be that it should write out the ssh_conf to the appropriate location (i.e., /etc/ssh/ssh_config).

It could take a couple minutes or so for this command to complete.

Slide 41



In the preceding exercises we began writing a remediation recipe on our target node.

We also tested the recipe with Test Kitchen.

But have we even addressed the "Set the SSH protocol version to 2" issue?

If you answered "no", you are correct. In a little while we will modify our recipe to address the "Set the SSH protocol version to 2" issue.

Slide 42

EXERCISE

GL: Using InSpec for Verification

Objective:

- ✓ Write a remediation recipe on that node.
- ✓ Test the recipe with Test Kitchen.
- ❑ Test for compliance with InSpec from the command line interface (CLI)
- ❑ Converge the recipe .
- ❑ Rescan the node and ensure compliance.

CHEF

Slide 43

GL: Create the `inspec` Directory

```
$ mkdir -p ~/cookbooks/ssh/test/integration/client/inspec
```

Slide 44

GL: Create the `client_spec.rb' file

~/cookbooks/ssh/test/integration/client/inspec/client_spec.rb

```
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 SSHv3 connections anymore.
  "
  describe ssh_config do
    its('Protocol') { should eq('2') }
  end
end
```

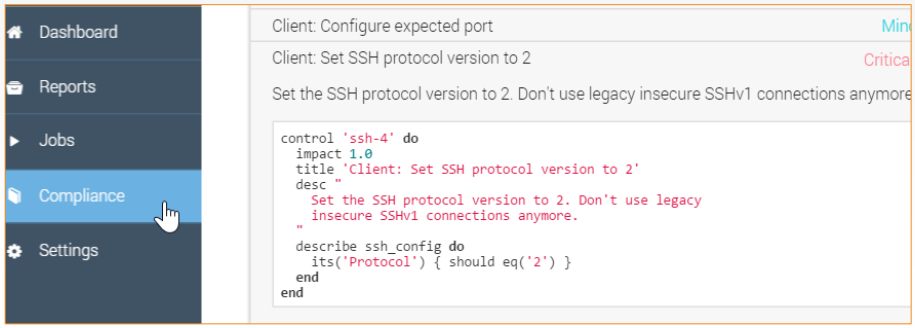

See the following slide for an example of a handy way to populate this file.

Slide 45

Example of Creating the `client_spec.rb' file

One handy way to populate the preceding `client_spec.rb' is to use the Compliance Web UI and copy the InSpec code found in the relevant Compliance profile:

Compliance > Base SSH > Set the SSH protocol version to 2



One handy way to populate the preceding `client_spec.rb' is to use the Compliance Web UI and copy the InSpec control code found in the relevant Compliance profile.

Then you can paste it into the `client_spec.rb' file to save yourself some typing.

Instructor Note: It could also be good for the instructor to demonstrate using the InSpec verifier in test kitchen locally with Vagrant to show the students that it can be done.

Slide 46

CONCEPT

Running InSpec from the Command Line Interface (CLI)

InSpec is an executable application.

InSpec can execute on remote hosts, including docker containers.

You can use 'inspec exec' to run tests at a specified path.

Slide 47

GL: Change Owner of `/var/run/docker.sock`

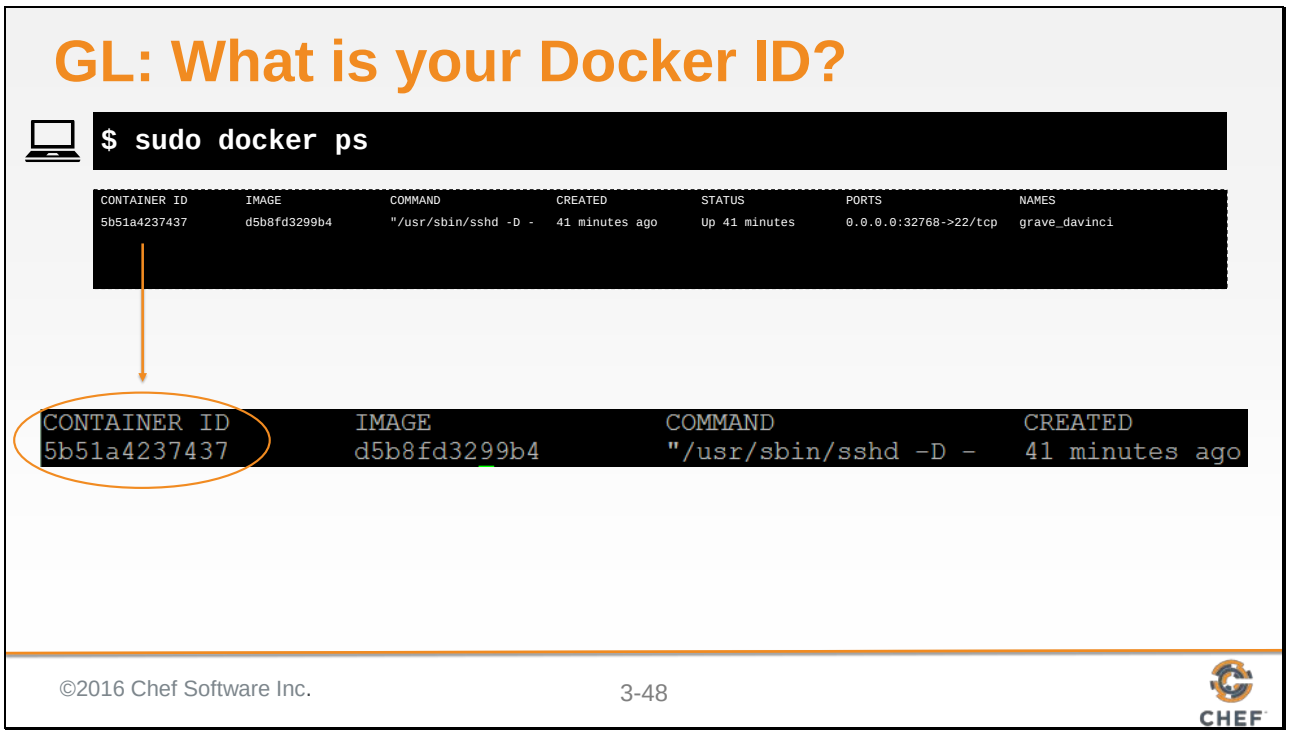
```
$ sudo chown root:dockerroot /var/run/docker.sock
```


First, we need to run this command because we are using Docker solely for testing and placing it in this configuration is not secure. We are doing it here because it is necessary if we do not want to prefix `sudo` in front of the commands we execute.

So it’s done here namely for speed and ease of training so you can focus on Compliance. On your local system you may use vagrant, ec2, or the azure driver and those will not have the same concern that we are experiencing here.

Instructor Note: This command is done in order to put the chef user in the dockerroot group and make /var/run/docker.sock's group dockerroot. This change would not persist when making part of the AMI so we run the command here.

Slide 48



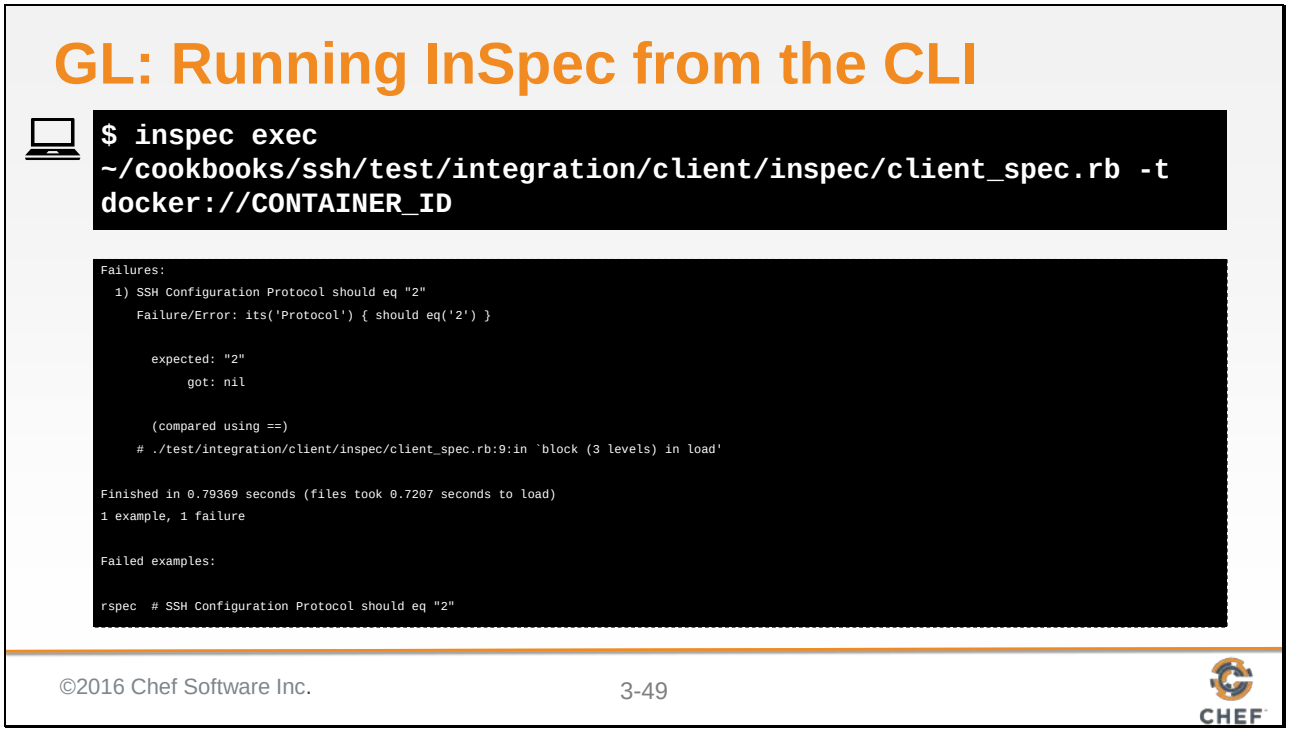
Below is an example of the details of the `sudo docker ps` command. This shows one docker container running.

You should only have only one docker container running too.

You'll need the Container ID for the next step so copy your Container ID, which is the first value that is not a header.

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
5b51a4237437 d5b8fd3299b4 "/usr/sbin/sshd -D - 41 minutes ago Up 41 minutes 0.0.0.0:32768->22/tcp grave_davinci

Slide 49



Run this inspec command using the container ID you just copied, replacing CONTAINER_ID in the example.
`inspec exec ~/cookbooks/ssh/test/integration/client/inspec/client_spec.rb -t docker://CONTAINER_ID`

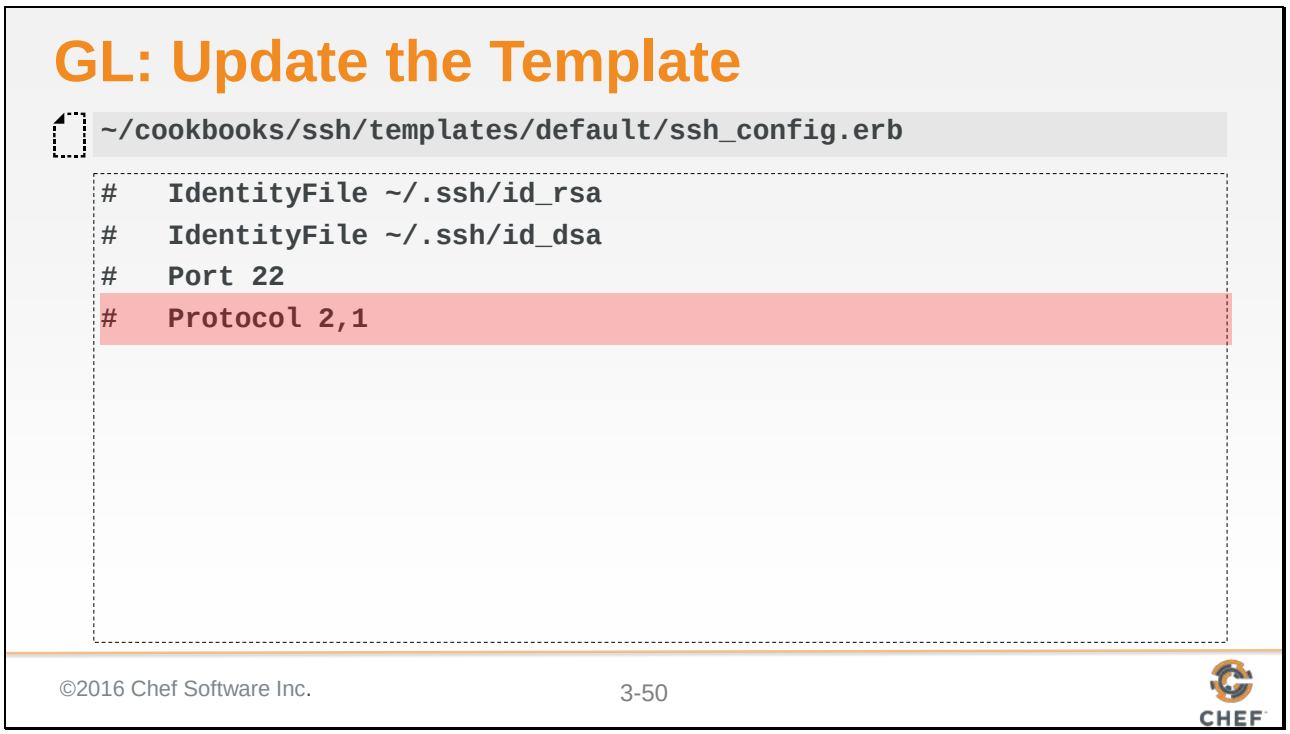
Running InSpec in this way can uncover more complex issues than the basic issue we are remediating in this module.

While the image of the output may be hard to see, key parts of the output is also pasted below. Notice how inspec from the command line also found the "SSH Configuration Protocol should eq "2" non compliance issue.

Key parts of the output is here:
Failures:

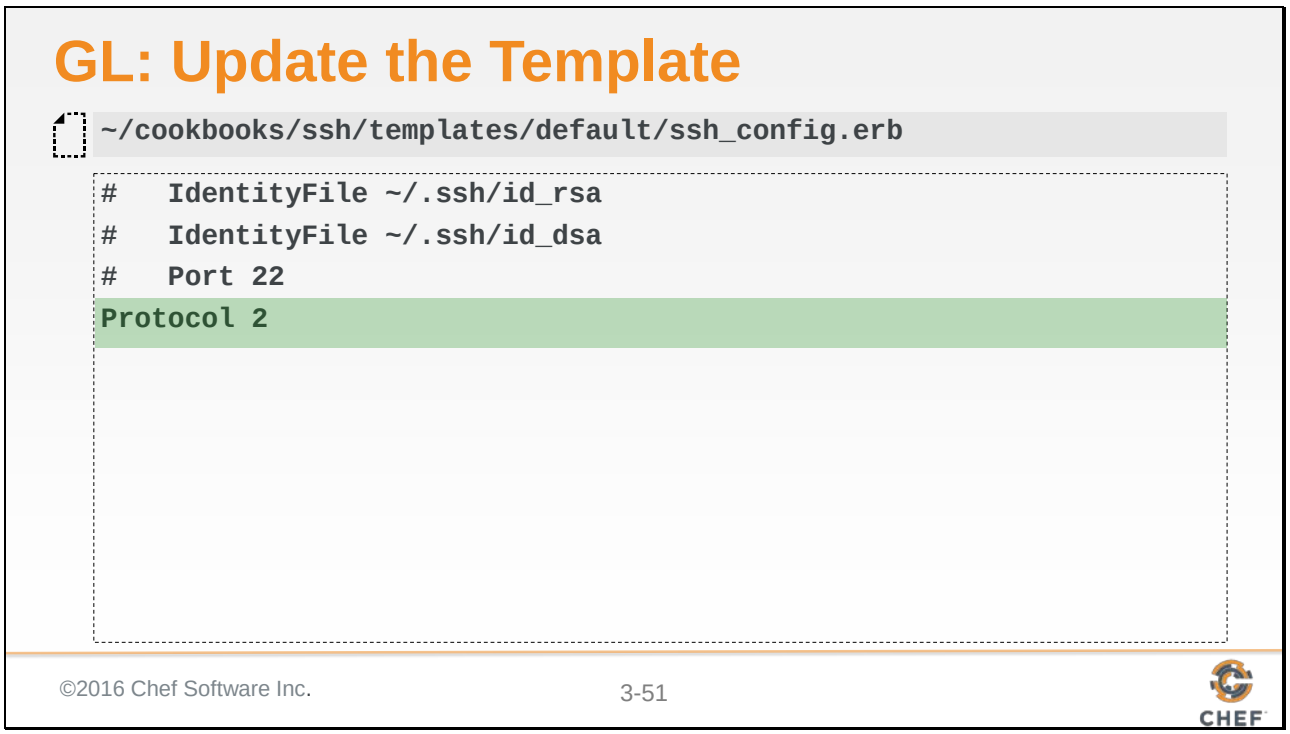
1) SSH Configuration Protocol should eq "2"
Failure/Error: its('Protocol') { should eq('2') }
expected: "2"
got: nil
...
Failed examples:
rspec # SSH Configuration Protocol should eq "2"

Slide 50



- Edit the ~/cookbooks/ssh/templates/default/ssh_config.erb file.
- Uncomment the `#  Protocol 2,1` line.
- Change the protocol version to `2` only.

Slide 51

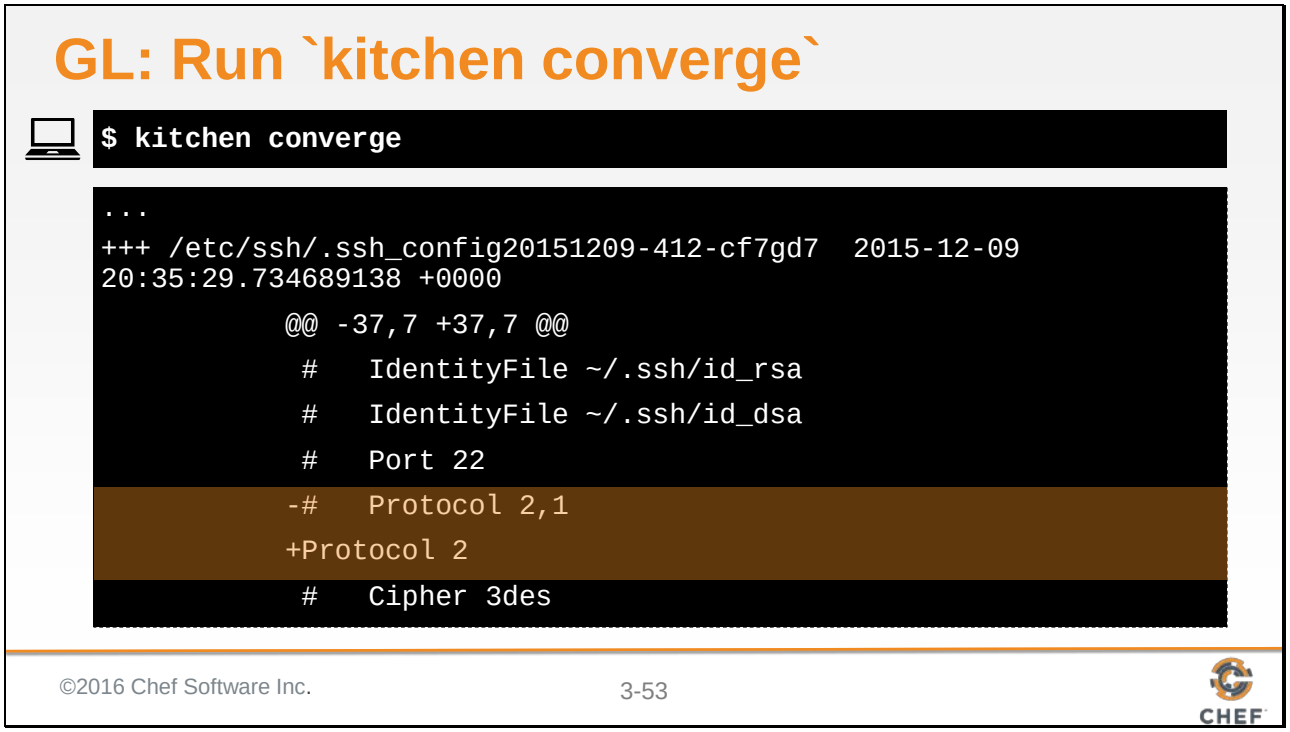


Results: Your ~/cookbooks/ssh/templates/default/ssh_config.erb file's Protocol line should now look like this example.

Slide 52

Change to ~/cookbooks/ssh if not there already.

Slide 53



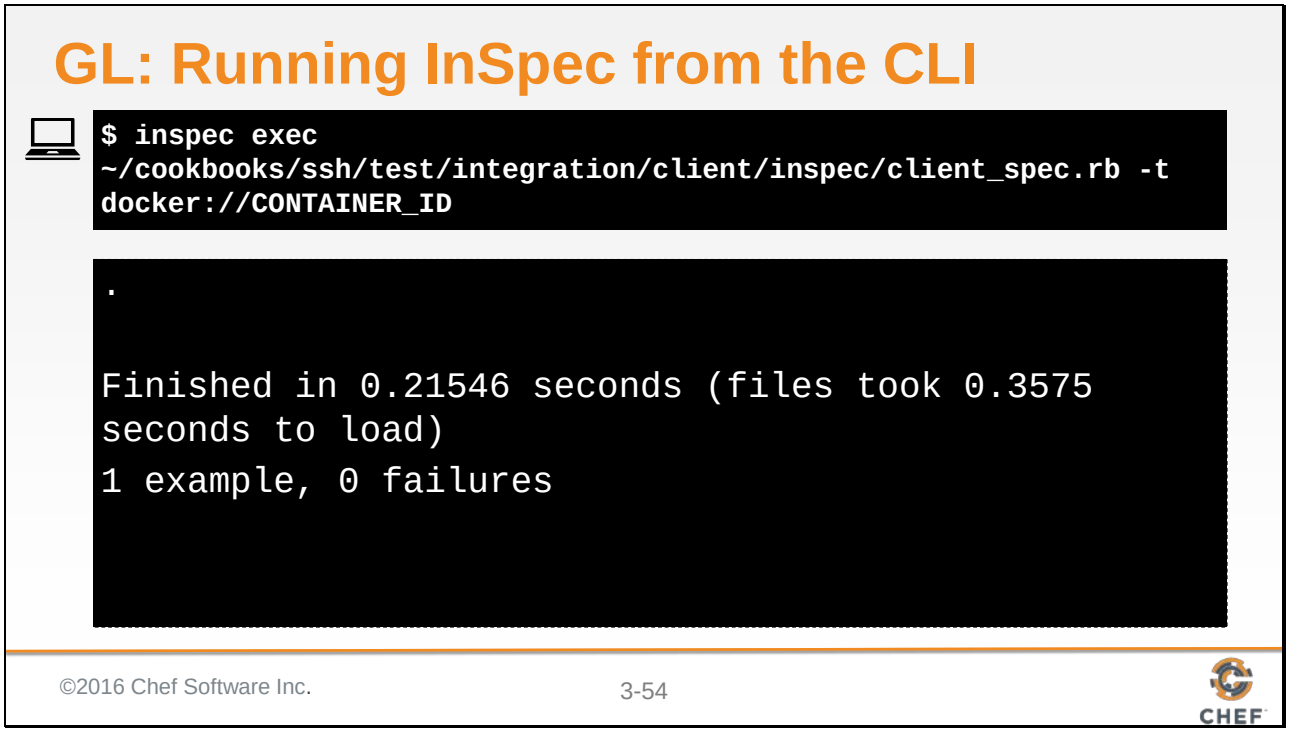
You should now see that only Protocol version 2 is currently set in test kitchen.

- update content in file /etc/ssh/ssh_config from 86eb9b to 065f90
--- /etc/ssh/ssh_config 2015-08-13 09:58:26.000000000 +0000
+++ /etc/ssh/.ssh_config20151209-412-cf7gd7 2015-12-09 20:35:29.734689138 +0000
@@ -37,7 +37,7 @@
# IdentityFile ~/.ssh/id_rsa
# IdentityFile ~/.ssh/id_dsa
# Port 22
-# Protocol 2,1
+Protocol 2
Cipher 3des
Ciphers aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-cbc,3des-cbc

Running handlers:

Chef Client finished, 3/3 resources updated in 03 seconds
Finished converging <client-centos-67> (0m8.22s).

Slide 54

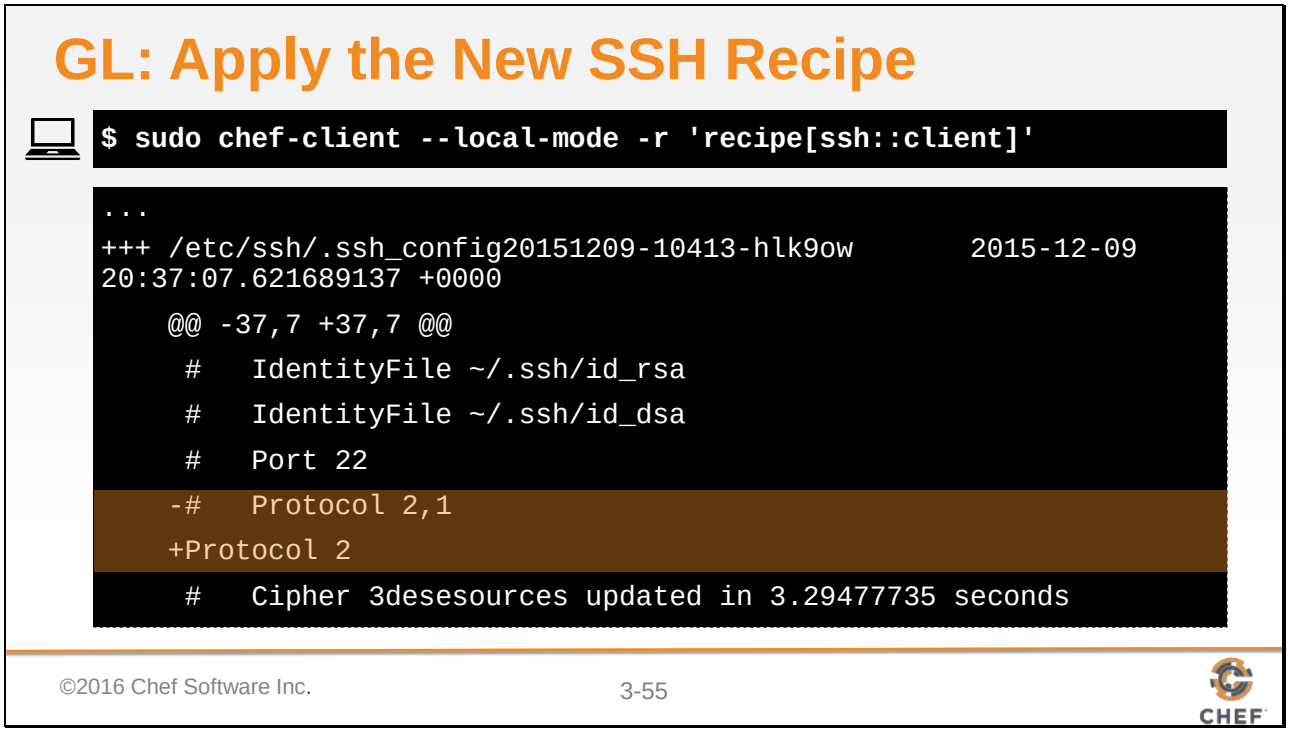


Run this inspec command again using the container ID you copied previously, replacing CONTAINER_ID in the example.

`inspec exec ~/cookbooks/ssh/test/integration/client/inspec/client_spec.rb -t docker://CONTAINER_ID`

You should now see that the test has passed. In addition to the output text that says there were 0 failures, the single dot at the top-left of the output means there was one test made and that it passed.

Slide 55



Now we need to actually apply the change to the node. We'll do this using chef-client in local mode. You should then see that only Protocol version 2 is currently set on the node.

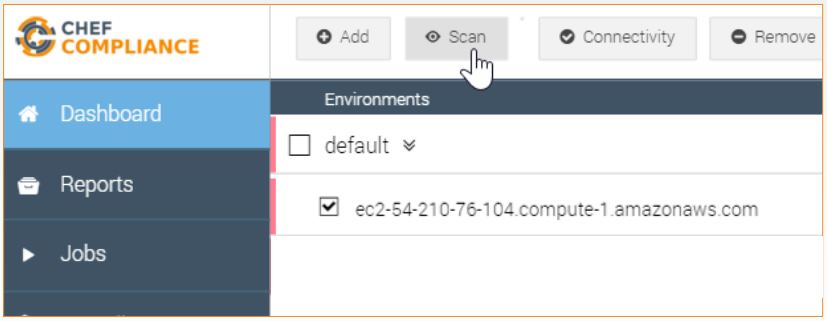
Of course in a production environment chef-client would most likely be set to run automatically to download and converge these changes from Chef Server.

Slide 56

GL: Re-run the Compliance Scan

Return to the Compliance Web UI and re-run the scan on your target node.

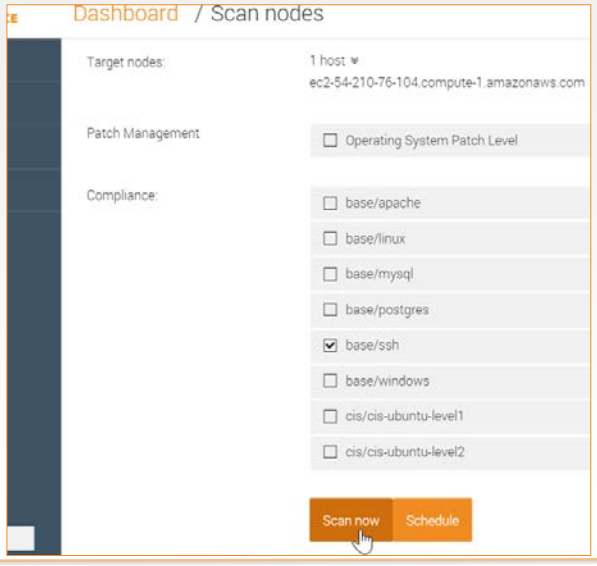
Be sure to run only the base/ssh scan as shown on the next slide.



Slide 57

GL: Re-run the Compliance Scan

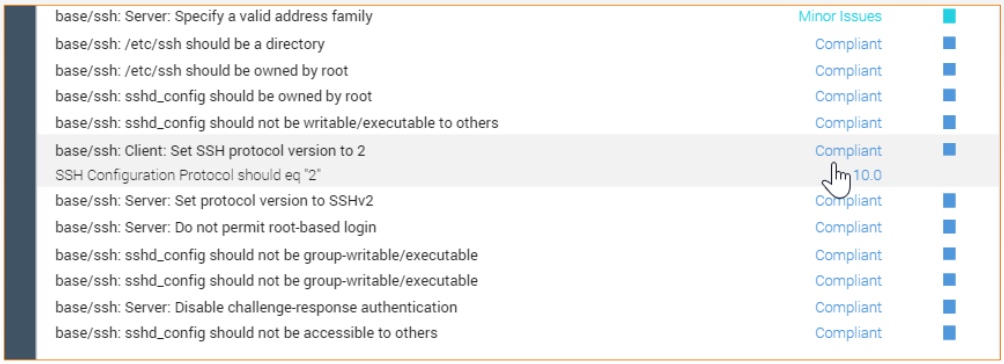
Run only the base/ssh scan.



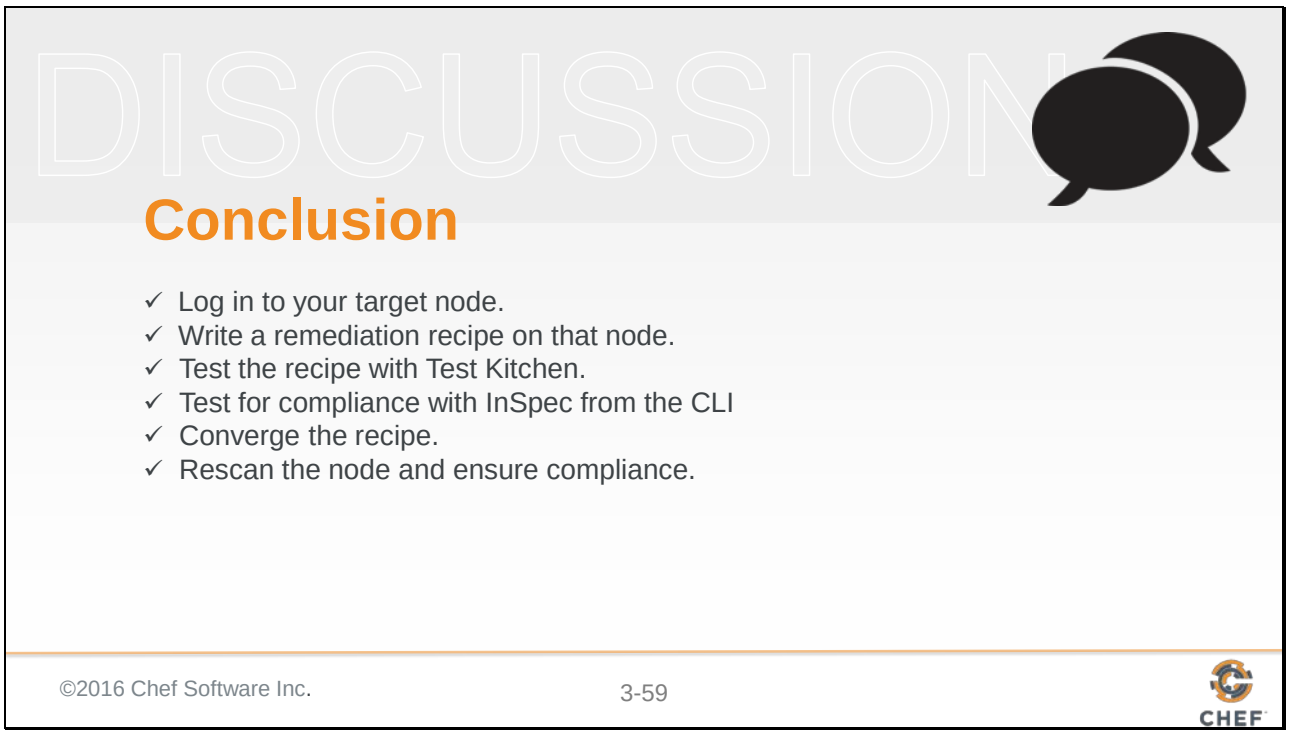
Slide 58

GL: Results of this Exercise

Your scan should show that the ssh protocol issue is now complaint.



Slide 59



In this module we scanned a node for compliance issues. We identified an issue and then wrote a remediation recipe directly on the node scanned. We also tested our recipe with test kitchen.

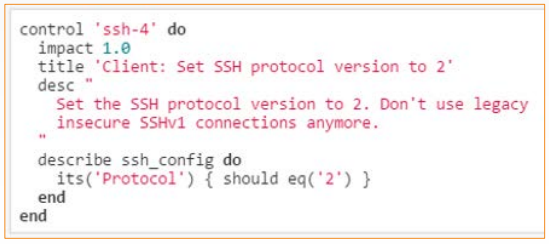
As mentioned previously, in a production environment, you will likely write such recipes locally, add them to the node's run list, and then upload them to Chef Server.

Then the nodes would download the recipes from Chef Server on their next chef-client run and also convergence the recipes.

Slide 60

Review Questions

1. When adding a node to the Compliance server's dashboard, should you use the node's FQDN or just its IP address?
2. What can `inspec exec` be used for?
3. How are compliance severities defined?
4. Using the image on the right, what section is the actual test?



```
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
```


Instructor Note Answers:
1. It doesn't matter...you could use the node's FQDN or just its IP address.
2. `inspec exec` can be used to test a compliance profile against remote hosts, including docker containers.
3. The `impact` value in a Compliance Profile/control defines severity. See slide 3-22 through slide 3-24 for examples.
4. The `describe` section is the actual test.

Slide 61

Review Questions

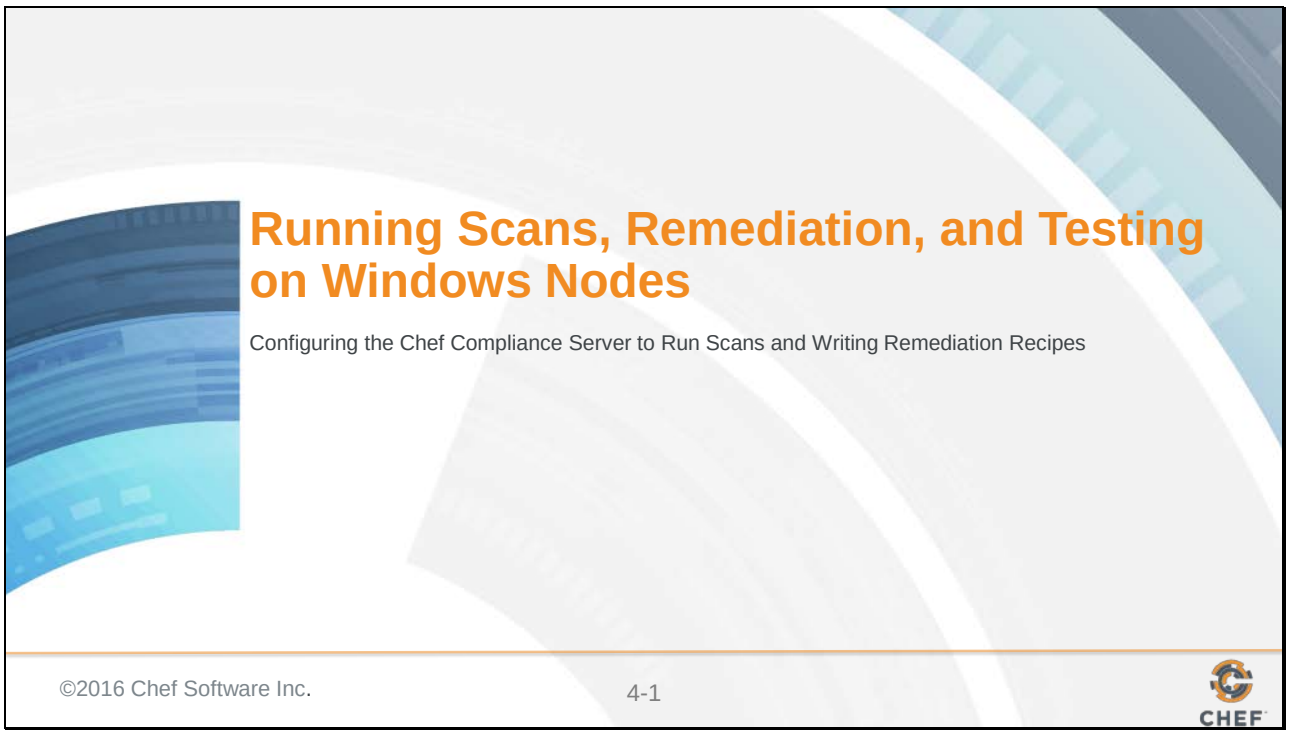
5. If a compliance scan tells you that a node is unreachable, what might you use to troubleshoot the connection?

6. What language is used to define controls?

Instructor Note Answers:
5. You could use the Dashboard's "check the connectivity" test, ssh into the target, and/or check the node's configuration in the Web UI (IP address, login credentials.)
6. InSpec.

Slide 62

4: Running Scans, Remediation, and Testing on Windows Nodes



Instructor Note: The quiz at the end of this module is virtually identical to the quiz in 03-initial-configuration-scans (Linux version). This is because the concepts are the same so you can skip the quiz in the module if you already covered them in the preceding module.

Slide 2

Objectives

After completing this module, you should be able to:

- Add a Windows node to test for compliance.
- Run a Compliance scan.
- Test for compliance with InSpec from the CLI.
- Remediate a compliance issue.
- Use Test Kitchen to test your remediation.
- Rescan the node and ensure compliance.

Slide 3

EXERCISE

Group Lab: Adding a Node to Scan

Objective:

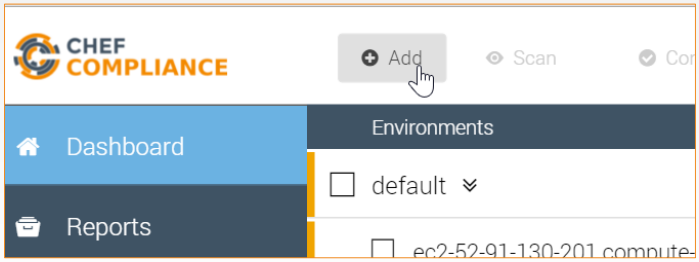
- ❑ Add a Windows Node to Scan
- ❑ Test connectivity

CHEF

Slide 4

GL: Adding a Node to Scan

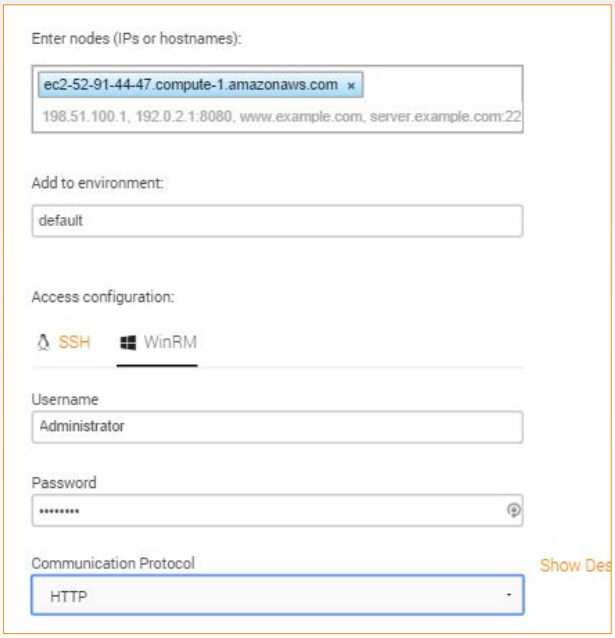
1. From your Chef Compliance Dashboard, click Add Node.



Slide 5

GL: Adding a Node

2. From the resulting page, enter the Windows node's FQDN or IP address.
3. Select the **default** environment.
4. Click the **WinRM** Access configuration.
5. Type **Administrator** in the **Username** field.
6. Type the password (**Cod3Can!**) in the password field.



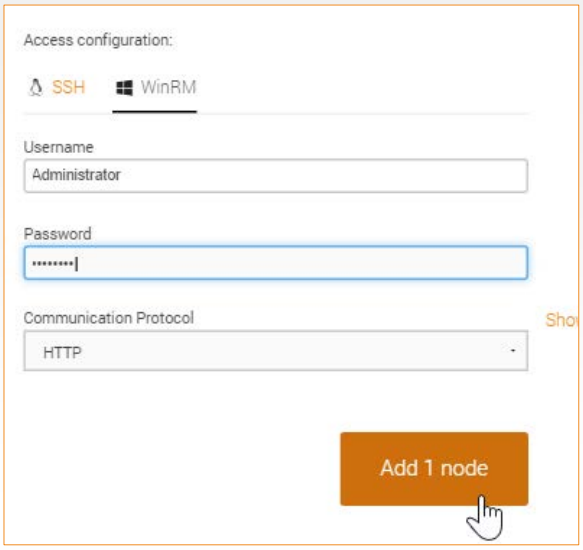
Be sure you are using the hostname of the Windows target node that you noted previously in class.

In the workplace, the target node's username and password will likely be different than shown in this example.

Slide 6

GL: Adding a Node to Scan

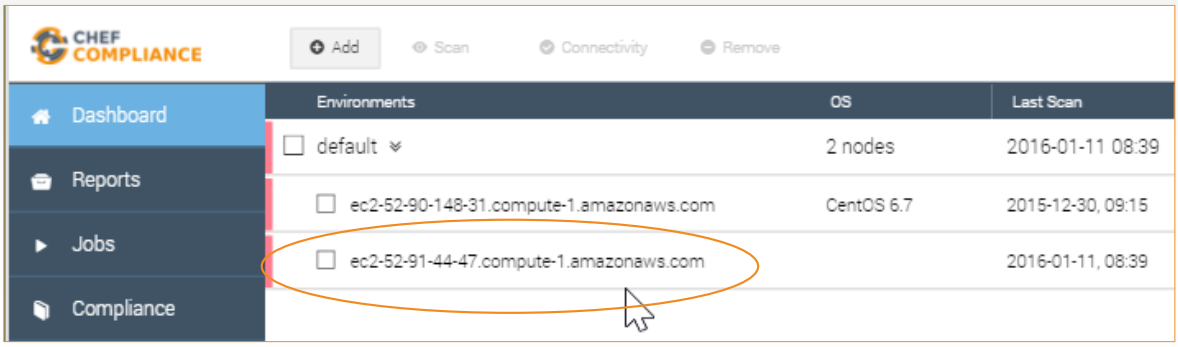
7. Ensure the **HTTP** Communication Protocol is set.
8. Click the **Add 1 node** button.



Slide 7

GL: Adding a Node to Scan

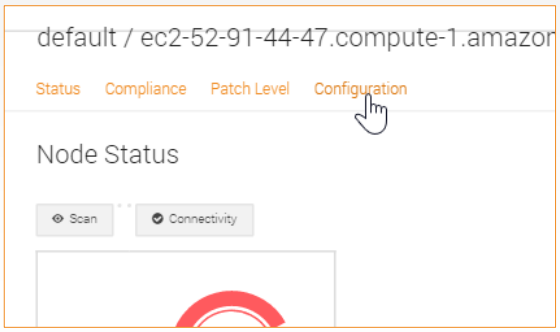
At this point your Compliance Dashboard should list the node you just added. In the next step we'll modify the Windows node name to make it easier to differentiate it from your Linux node.



Slide 8

GL: Modify the Node Name

1. Click the **Windows node**.
2. From the resulting page, click **Configuration**.



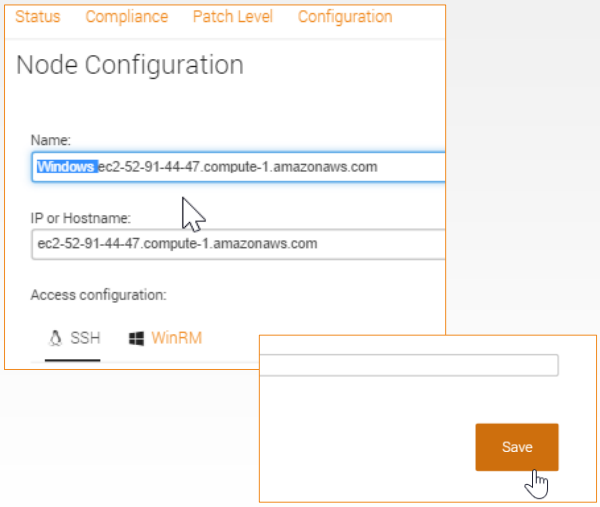
Slide 9

GL: Modify the Node Name

3. Type **Windows** at the beginning of the **Name:** field.

   Important: Do not change the value in the **IP or Hostname** field because that field is used to connect to your node.

4. Click **Save**.



Slide 10

GL: Modify the Node Name

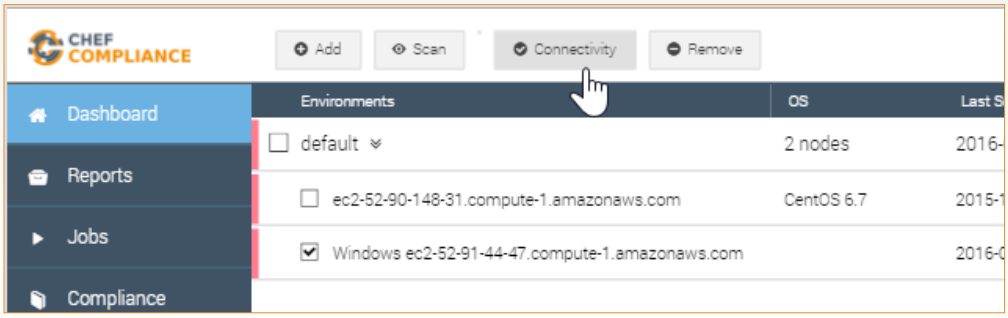
Now you can more easily differentiate your Windows node from your Linux node.

CHEF

Slide 11

GL: Testing Connectivity to your Node

1. Click the **check box** next to your Windows node and then click the **Connectivity** button.

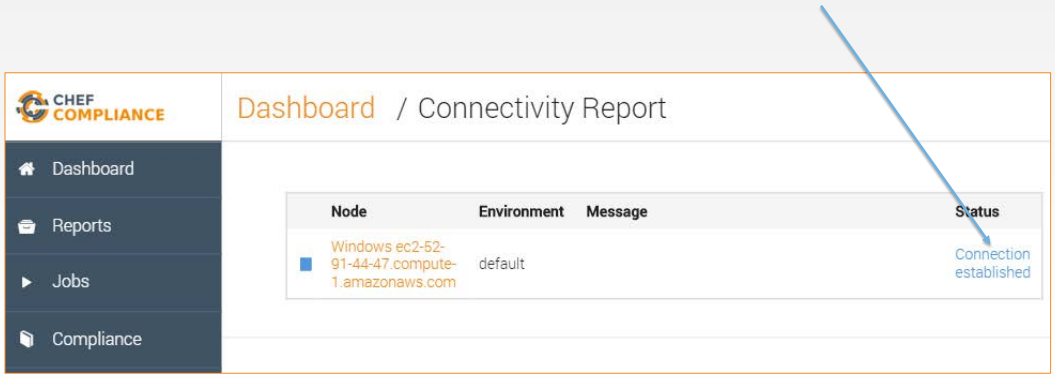


CHEF

Slide 12

GL: Testing Connectivity to your Node

The Status column of your node should now indicate **Connection established**.



If your Status column does not indicate `**Connection established**`, please notify the instructor.

Instructor Note: I have seen where a Windows Connectivity test will fail, but then a scan will work. After the scan is successful, the Connectivity test will start working. Almost like the Windows target was sleepy. Usually happens after the windows node sits idle for a while.

Slide 13

EXERCISE

Group Lab: Running a Scan

Objective:

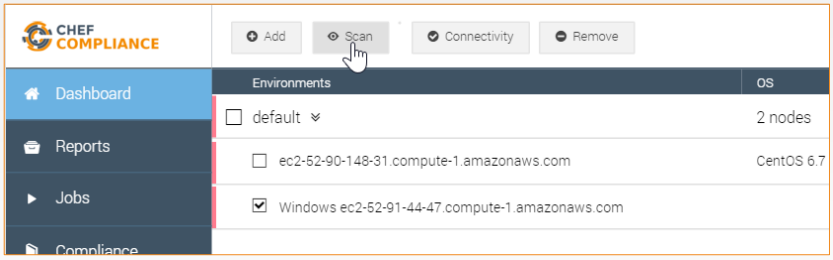
- ❑ Run a Compliance scan.
- ❑ View the output of a scan.

CHEF

Slide 14

GL: Running a Scan

1. Click the **check box** next to your node and then click the **Scan** button.



CHEF

Slide 15

GL: Running a Scan

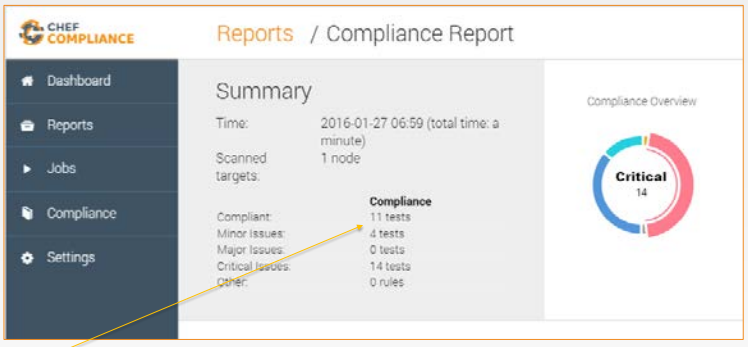
2. From the resulting page, check the **base/windows** profile and uncheck any other check boxes.

3. Click the **Scan now** button.

Slide 16

Scan Results

A Compliance Report should now display and your scan results should be similar to that shown here.



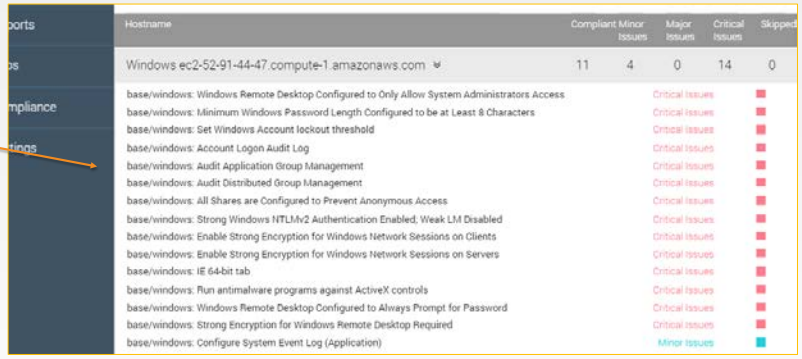
Notice how in the upper Summary section in this example, 11 tests were compliant and 14 tests show critical issues with ssh.

Instructor Note: This and the following slide should be used for a discussion of the scan results. The Group Lab continues after that.

Slide 17

Scan Results

The bottom half of the Compliance Report shown here has a table of details of the scan, similar to what you saw in the Linux example.



Notice how one of the critical issues regards Strong Windows NTMLv2 Authentication Enabled

The bottom half of the Compliance Report has a table of details of test results.

These are sorted by severity so the critical issues are listed at the top and the compliant items are at the bottom of the list.

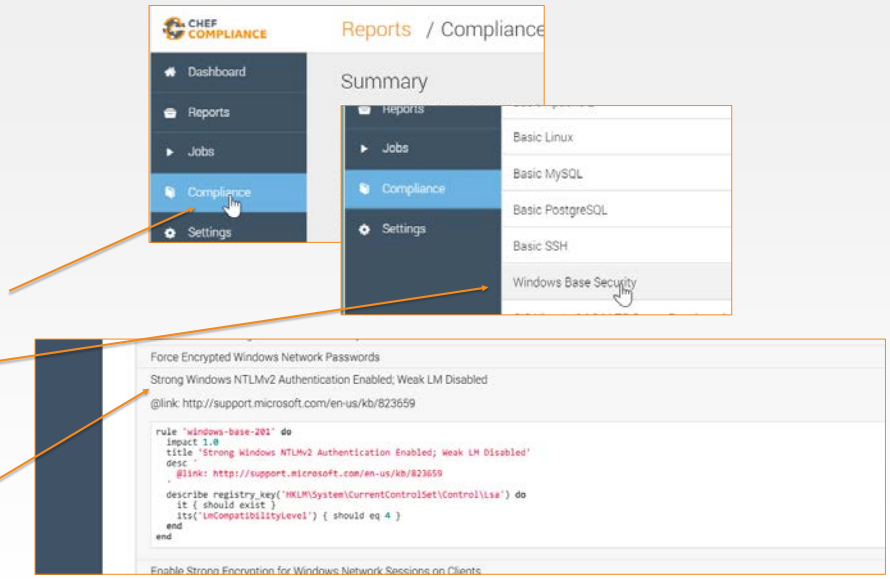
If you click an issue as shown here, a bit more information about the issue displays, but that's not really telling us much.

Slide 18

GL: Profile

To view the InSpec code that comprises this profile, do the following:

1. Click the **Compliance** button.
2. Click the relevant profile category (**Windows Base ...**).
3. Scroll down and click the **Strong Windows NTLMv2...** profile.



Note that the `rule` designation will be changed to `control` in an upcoming release.

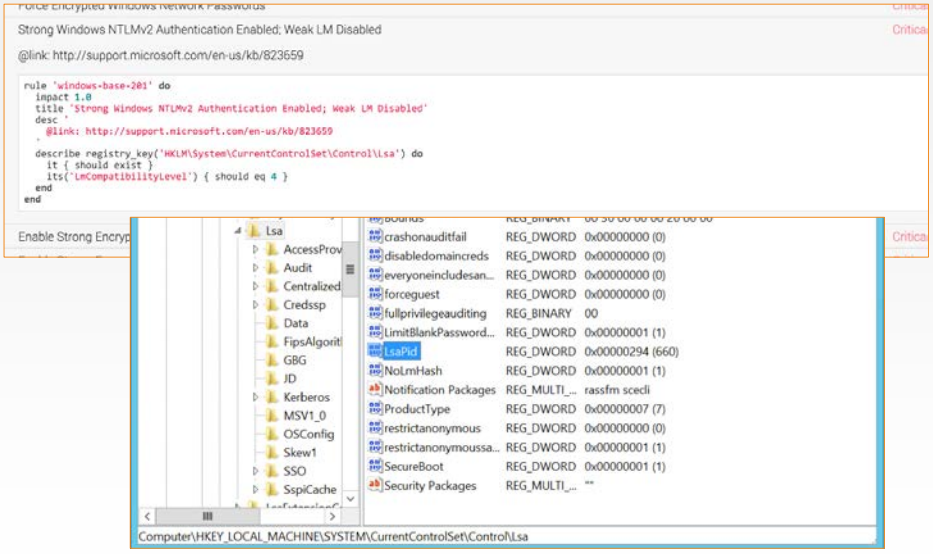
Instructor Note: Now we continue the Group Lab but you should stop as needed to explain what this code means.

Slide 19

Discussion: InSpec Profile Code

You can see in the bottom image where the Registry Key is not set.

In a production environment you'd want to write a Chef recipe to remediate this issue.



Slide 20

PROBLEM

Let's Remediate the Issue

Now that we've identified the issue, let's write a recipe on the target node to remediate the issue.

Then we'll run the compliance scan again to see if we successfully remediated the issue.

Note: In this course we will write a recipe directly on the node that we're running scans on. Of course in a production environment you will likely write such recipes locally and upload them to Chef Server. Then the nodes would converge the recipes on their next chef-client run.

CHEF

Slide 21

EXERCISE

GL: Remediating the Issue

Objective:

- ❑ Write a remediation recipe on that node
- ❑ Test for compliance with InSpec from the command line interface (CLI)
- ❑ Converge the recipe
- ❑ Rescan the node and ensure compliance

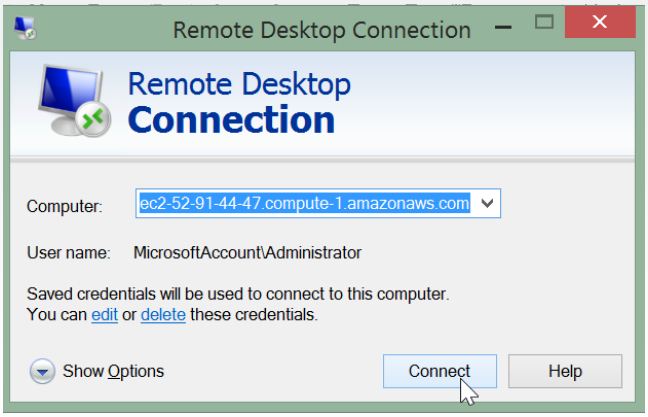
CHEF

Slide 22

GL: Remediating the Issue

Log in to your target node (not your compliance server node) using your RPD client.

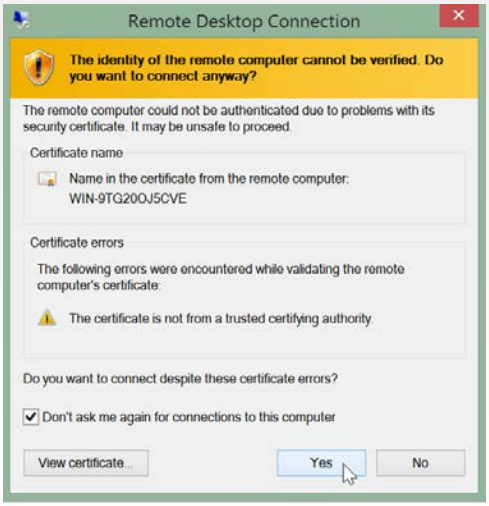
You can use only the IP address instead of the FQDN that this image shows.



Slide 23

GL: Remediating the Issue

If you get a warning like this, click **Yes**.



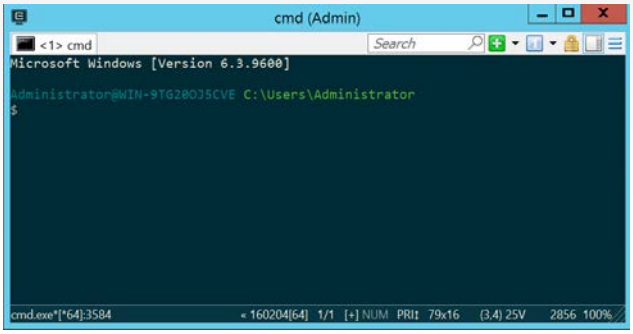
Slide 24

GL: Remediating the Issue

Double-click the ConEmu(x64) icon on the Windows desktop to start that terminal.

At this point the ConEmu terminal should display and you should be in the home directory.

C:\Users\Administrator



CHEF

GL: Create and Change to a 'cookbooks' Directory

```
$ mkdir cookbooks
$ cd cookbooks
```

From the home directory (C:\Users\Administrator\), create a `**cookbooks**` directory and navigate into it.

CHEF

Slide 26

GL: Create a Windows Access Cookbook

```
$ chef generate cookbook windows_access
```

```
Compiling Cookbooks...
Recipe: code_generator::cookbook
  * directory[C:/Users/Administrator/cookbooks/windows_access] action create
    - create new directory C:/Users/Administrator/cookbooks/windows_access
  * template[C:/Users/Administrator/cookbooks/windows_access/metadata.rb] action
create_if_missing
    - create new file C:/Users/Administrator/cookbooks/windows_access/metadata.rb
    - update content in file C:/Users/Administrator/cookbooks/windows_access/metadata.rb from
none to 18be67
    (diff output suppressed by config)
  * template[C:/Users/Administrator/cookbooks/windows_access/README.md] action
create_if_missing
    - create new file C:/Users/Administrator/cookbooks/windows_access/README.md
    - update content in file C:/Users/Administrator/cookbooks/windows_access/README.md from
none to 481e5e
```


CHEF

Slide 27

GL: Create an Authentication Recipe

```
$ chef generate recipe windows_access authentication
```

```
Compiling Cookbooks...
Recipe: code_generator::recipe
  * directory[./windows_access/spec/unit/recipes] action create (up to date)
  * cookbook_file[./windows_access/spec/spec_helper.rb] action create_if_missing (up to date)
  * template[./windows_access/spec/unit/recipes/authentication_spec.rb] action
create_if_missing
    - create new file ./windows_access/spec/unit/recipes/authentication_spec.rb
    - update content in file ./windows_access/spec/unit/recipes/authentication_spec.rb from
none to 021637
    (diff output suppressed by config)
  * template[./windows_access/recipes/authentication.rb] action create
    - create new file ./windows_access/recipes/authentication.rb
    - update content in file ./windows_access/recipes/authentication.rb from none to 11d9b9
    (diff output suppressed by config)
```


CHEF

Slide 28

GL: Write the Authentication Recipe

~/cookbooks/windows_access/recipes/authentication.rb

```
# Cookbook Name:: windows_access
# Recipe:: authentication
# Copyright (c) 2016 The Authors, All Rights Reserved.

lsa_key = 'HKLM\System\CurrentControlSet\Control\Lsa'

registry_key lsa_key do
  values [{
    :name => 'LmCompatibilityLevel',
    :type => :dword,
    :data => 4
  }]
end
```


You can use Atom to edit this file.

Slide 29

EXERCISE

GL: Remediating the Issue

Objective:

- ✓ Write a remediation recipe on that node
- ❑ Test for compliance with InSpec from the command line interface (CLI)
- ❑ Converge the recipe
- ❑ Rescan the node and ensure compliance

CHEF

Slide 30

GL: Create the `inspec` Directory

```
$ mkdir windows_access\test\integration\authentication
$ mkdir windows_access\test\integration\authentication\inspec
```

```
    Directory:

C:\Users\Administrator\cookbooks\windows_access\test\integration\authentica...


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----          2/9/2016   9:06 PM            inspec
```


CHEF

You should still be in the C:\Users\Administrator\cookbooks\ directory prior to running these commands.

Slide 31

GL: Create the `auth` Specification File

~/cookbooks/windows_access/test/integration/authentication/inspec/auth_spec.rb

```
rule '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
```


CHEF

You can use Atom to create this file.

Slide 32

GL: Run InSpec from the CLI

```
$ inspec exec
windows_access\test\integration\authentication\inspec\auth_spec.rb
```

```
.F

Failures:

  1) Registry Key HKLM\System\CurrentControlSet\Control\Lsa
LmCompatibilityLevel should eq 4
     Failure/Error: its('LmCompatibilityLevel') { should eq 4 }

       expected: 4
            got: nil
```


CHEF

This command assumes you are still at C:\Users\Administrator\cookbooks\ when you run it.

Slide 33

EXERCISE

GL: Remediating the Issue

Objective:

- ✓ Write a remediation recipe on that node
- ✓ Test for compliance with InSpec from the command line interface (CLI)
- ❑ Converge the recipe
- ❑ Rescan the node and ensure compliance

CHEF

Slide 34

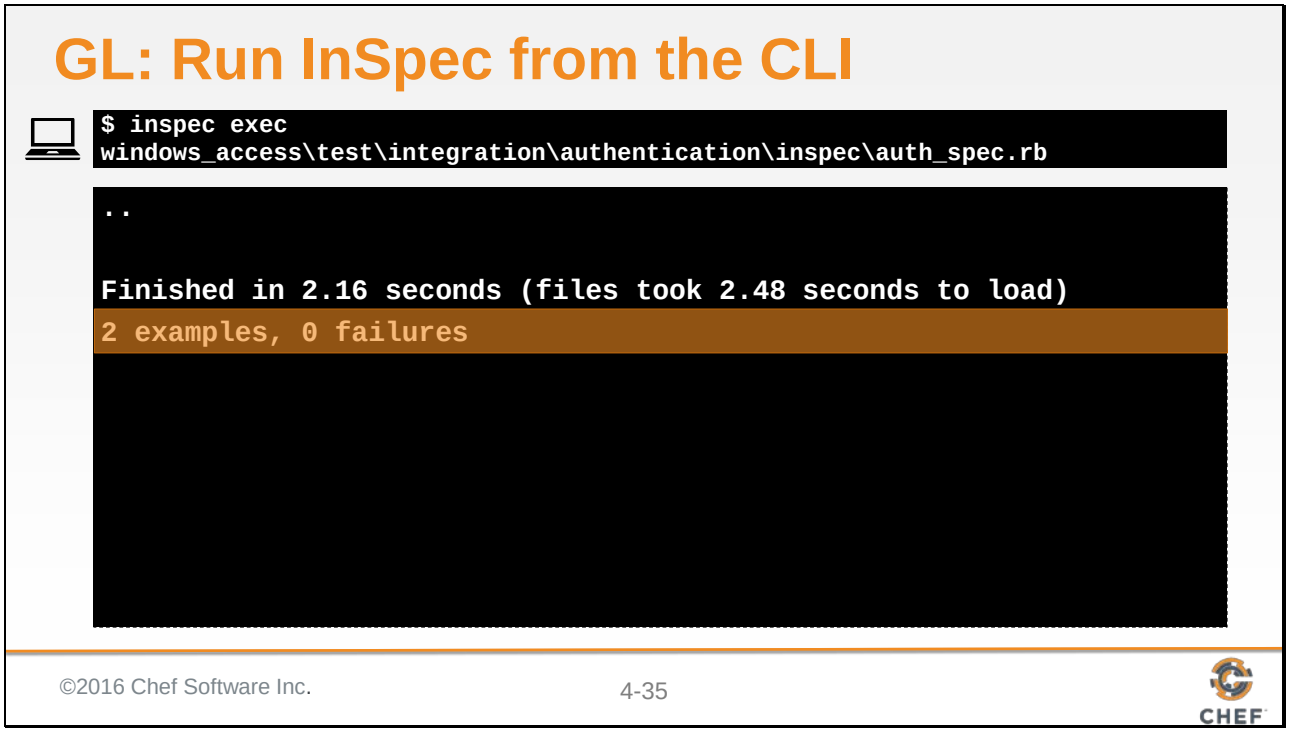
GL: Converge the Recipe

```
$ chef-client --local-mode -r 'recipe[windows_access::authentication]'
```

```
Synchronizing Cookbooks:
  - windows_access (0.1.0)
Compiling Cookbooks...
Converging 1 resources
Recipe: windows_access::authentication
  * registry_key[HKLM\System\CurrentControlSet\Control\Lsa] action create
    - set value {:name=>"LmCompatibilityLevel", :type=>:dword, :data=>4}

Running handlers:
Running handlers complete
Chef Client finished, 1/1 resources updated in 11 seconds
```


This command assumes you are still at C:\Users\Administrator\cookbooks\ when you run it.

Slide 35



This command assumes you are still at C:\Users\Administrator\cookbooks\ when you run it.

Slide 36

EXERCISE

GL: Remediating the Issue

Objective:

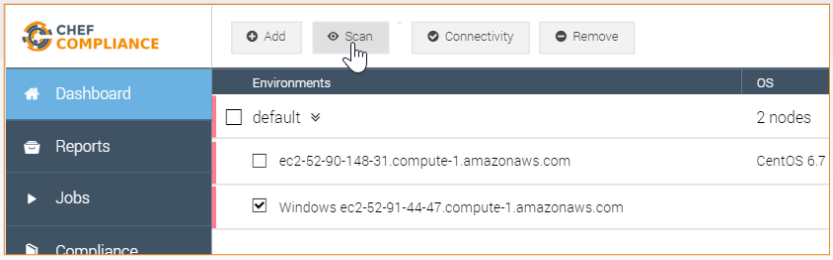
- ✓ Write a remediation recipe on that node
- ✓ Test for compliance with InSpec from the command line interface (CLI)
- ✓ Converge the recipe
- ❑ Rescan the node and ensure compliance

CHEF

Slide 37

GL: Running a Scan

1. Click the **check box** next to your node and then click the **Scan** button.

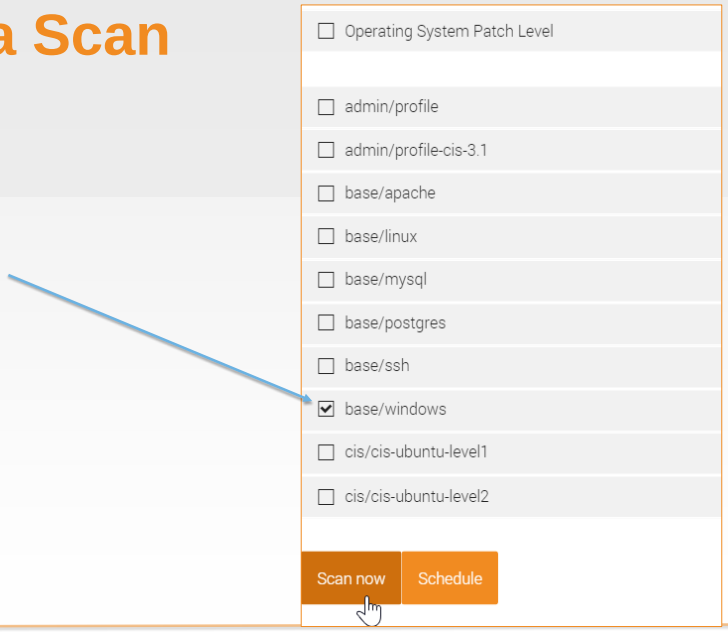


Slide 38

GL: Running a Scan

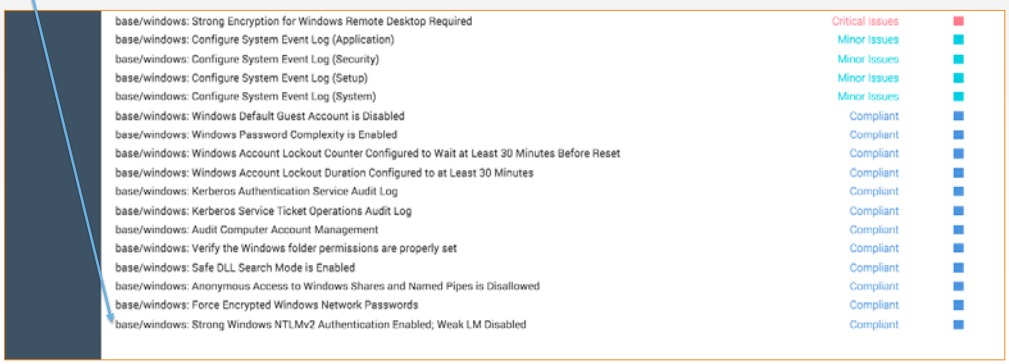
2. From the resulting page, check the **base/windows** profile and uncheck any other check boxes.

3. Click the **Scan now** button.



GL: Results of this Exercise

Your scan should show that the **Strong Windows NTLMv2 Authentication Enabled; Weak LM Disabled** is now compliant.



CHEF

Slide 40

EXERCISE

GL: Remediating the Issue

Objective:

- ✓ Write a remediation recipe on that node.
- ✓ Test for compliance with InSpec from the command line interface (CLI)
- ✓ Converge the recipe.
- ✓ Rescan the node and ensure compliance.

CHEF

Slide 41

Review Questions

1. When adding a node to the Compliance server's dashboard, should you use the node's FQDN or just its IP address?
2. What can `inspec exec` be used for?
3. How are compliance severities defined?
4. Using the image on the right, what section is the actual test?

Instructor Note: This quiz is virtually identical to the quiz in 03-initial-configuration-scans (Linux version). This is because the concepts are the same so you can skip this quiz if you already covered them in the preceding module.
Instructor Note Answers:
1. It doesn't matter...you could use the node's FQDN or just its IP address.
2. `inspec exec` can be used to test a compliance profile against remote hosts, including docker containers.
3. The `impact` value in a Compliance Profile defines severity. See slide 3-22 through slide 3-24 for examples.
4. The `describe` section is the actual test.

Slide 42

Review Questions

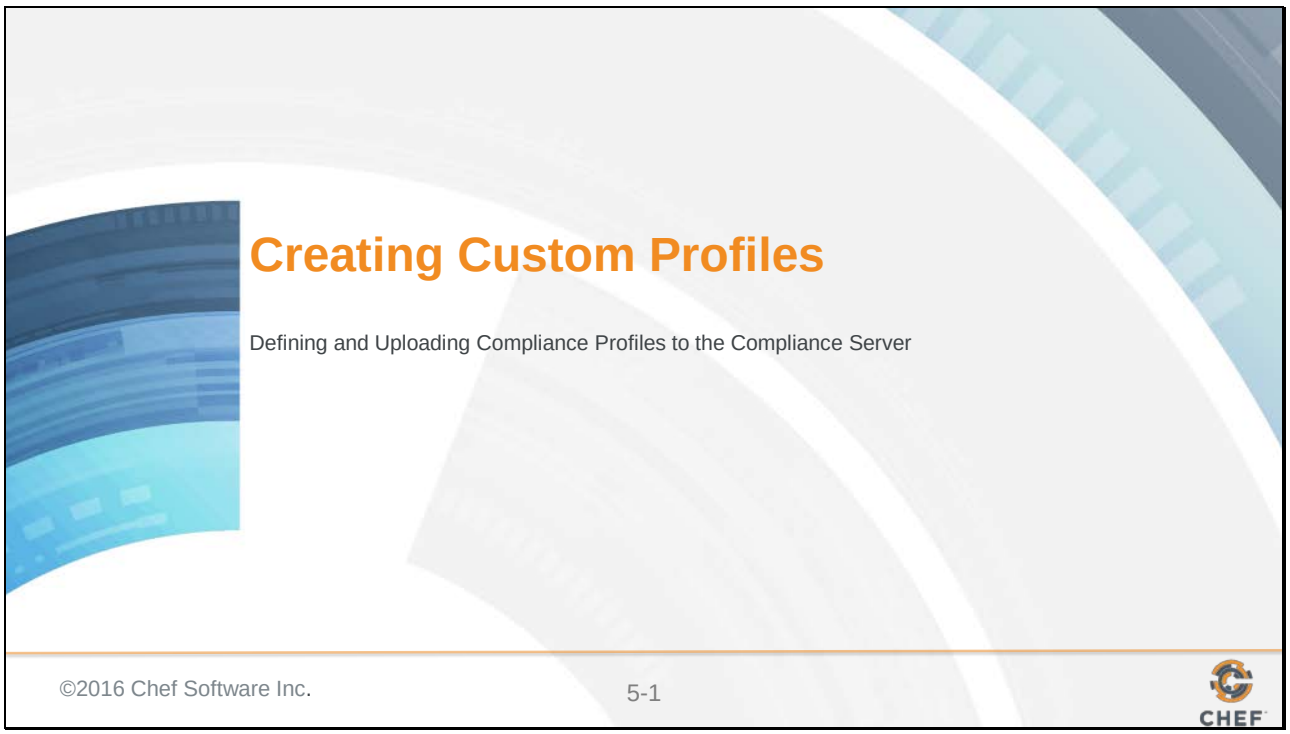
5. If a compliance scan tells you that a node is unreachable, what might you use to troubleshoot the connection?

6. What language is used to define controls?

Instructor Note Answers:
5. You could use the Dashboard's "check the connectivity" test, ssh into the target, and/or check the node's configuration in the Web UI (IP address, login credentials.)
6. InSpec.

Slide 43

5: Creating Custom Profiles



Slide 2

Objectives

After completing this module, you should be able to:

- Write a custom compliance profile.
- Use InSpec to test your code and your custom profile.
- Upload a custom compliance profile to your Chef Compliance server.
- Test your custom profile.

Slide 3

CONCEPT

Creating a Custom Profile

In this section we will create a custom compliance profile.

Custom profiles are created using InSpec, just like the existing profiles were created.

After you have created a custom profile, you'll learn how to upload it to a Compliance Server and then use it to check for compliance issues.

Slide 4

CONCEPT

InSpec Command Line Interface

In this section we will use the InSpec command line interface (CLI) to help us create Compliance profiles and run audit tests against targets.

The InSpec CLI commands can run audit tests against targets using SSH, WinRM, locally, or on Docker containers.

We'll be using `inspec init, `inspec check` and `inspec exec`.

- 'inspec init' streamlines the creation of new Compliance profiles.

CHEF

`inspec init` can create all the directories and basic files that the Compliance Server and `inspec check` and `inspec exec` require.

Slide 5

CONCEPT

InSpec Command Line Interface

We'll be using `inspec init, `inspec check` and `inspec exec`.

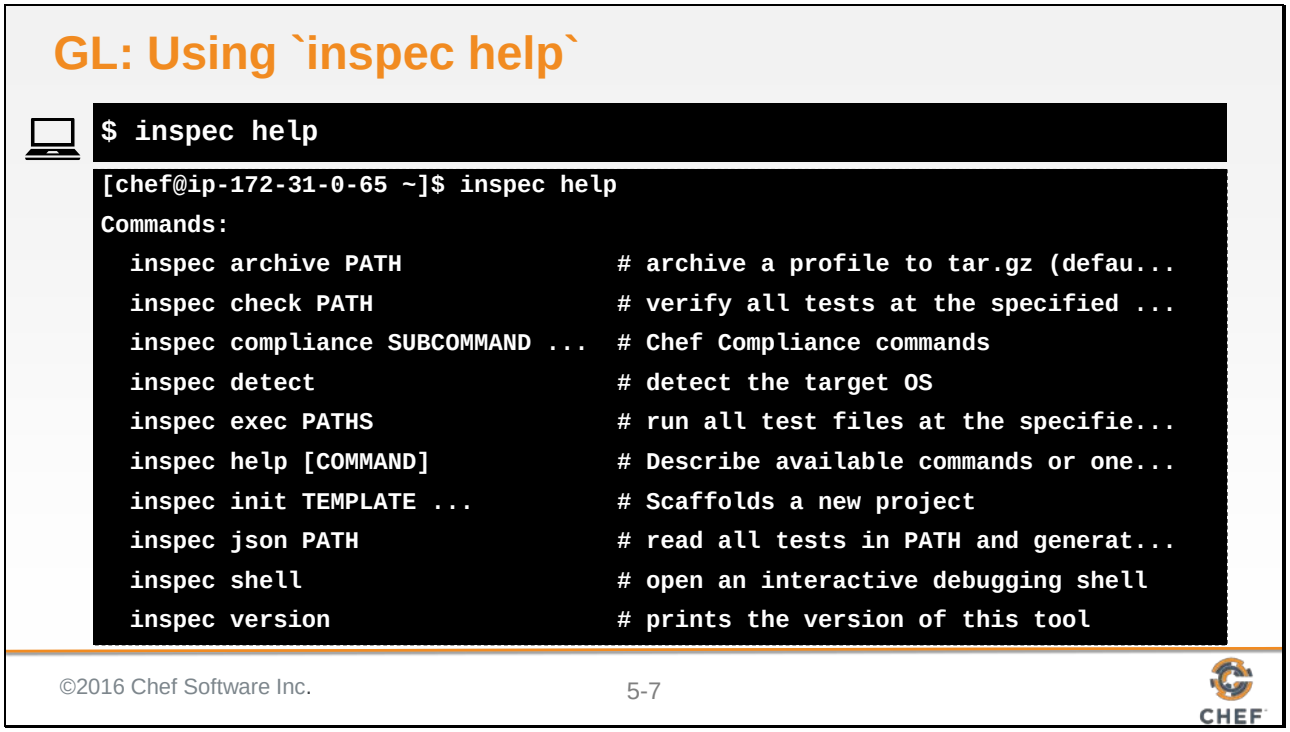
- `inspec check` just verifies the compliance profile code that you write --it doesn't actually test a system.
- `inspec exec' will run the tests against a system.

Slide 6

The custom profile you will create will scan nodes to ensure they have a '/tmp' directory and that directory should be owned by the root user.

Note: In the workplace you would likely perform these custom profile tasks on your local workstation and upload them to the Compliance Server. In this class we'll use our target nodes as a workstation to create the profile on since they already have Chef installed on them. Then we'll ultimately upload the customer profile to your Compliance Server.

Slide 7



From your target Linux node/virtual workstation and from your home directory, run `inspec help`. Notice the `inspec` commands and sub commands that are available.

Slide 8

GL: Using `inspec init` help`

```
$ inspec init help
```

```
[chef@ip-172-31-0-65 ~]$ inspec init help
Commands:
  inspec init help [COMMAND]  # Describe subcommands or one specific subcommand
  inspec init profile NAME    # Create a new profile
# prints the version of this tool
```


The `inspec init` command enables you to create new Compliance profiles with less manual intervention than in previous versions of inspec and Chef Compliance.

Slide 9

GL: Create the Compliance Profile Directories and Files

```
$ inspec init profile secureprofile_01
```

```
Create new profile at /home/chef/secureprofile_01
 * Create file README.md
 * Create directory libraries
 * Create directory controls
 * Create file controls/example.rb
 * Create file inspec.yml
```


Notice the directories and the files that `inspec init profile` creates. The `secureprofile_01` is merely the name of the profile and could be named any way that makes sense in your organization.

Slide 10

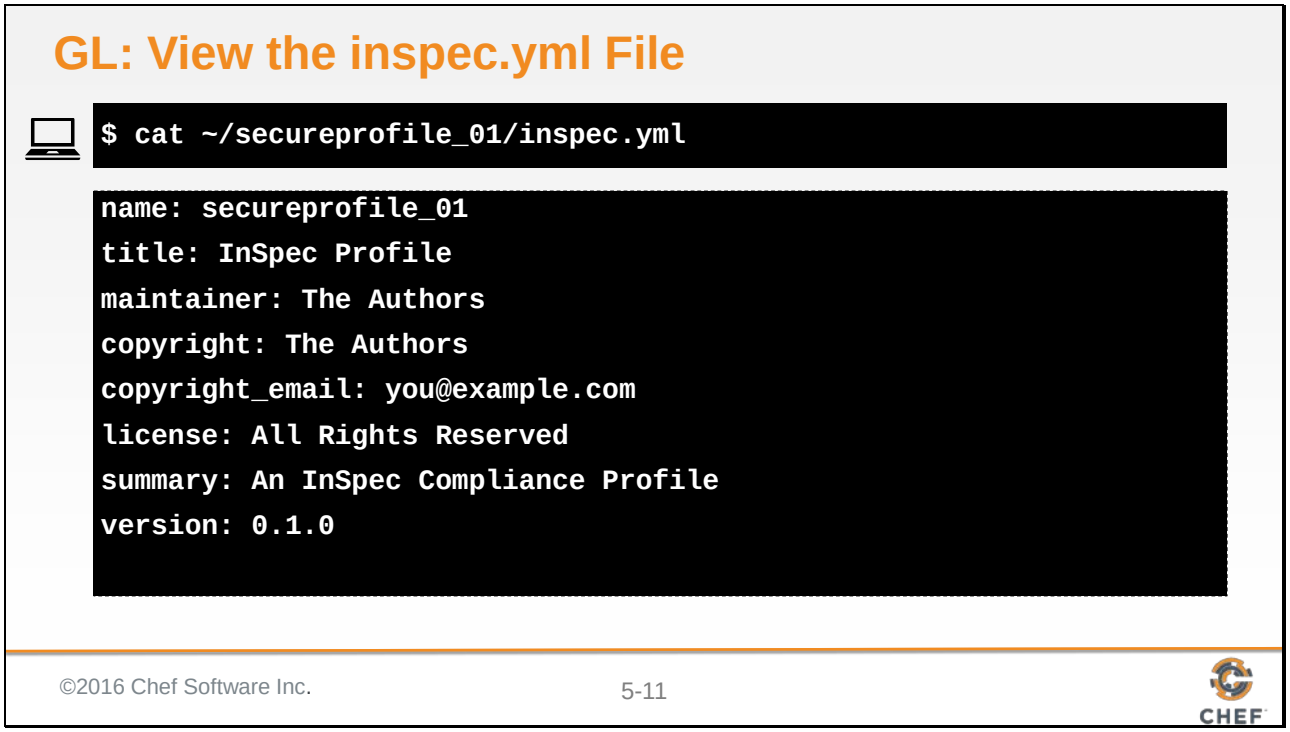
GL: View the Compliance Profile Directories and Files

```
$ tree secureprofile_01
```

```
secureprofile_01
├── controls
│   └── example.rb
├── inspec.yml
├── libraries
└── README.md
```


As you can see, the `tree secureprofile_01 ` command shows the new directories and files that inspec requires.

Slide 11



Let's read the inspec.yml file by issuing ~/secureprofile_01/inspec.yml.

In the workplace you should modify this file but we'll leave it as-is for now.

Slide 12

GL: Writing a Compliance Profile Control

Compliance profiles must be written within the `controls` directory.

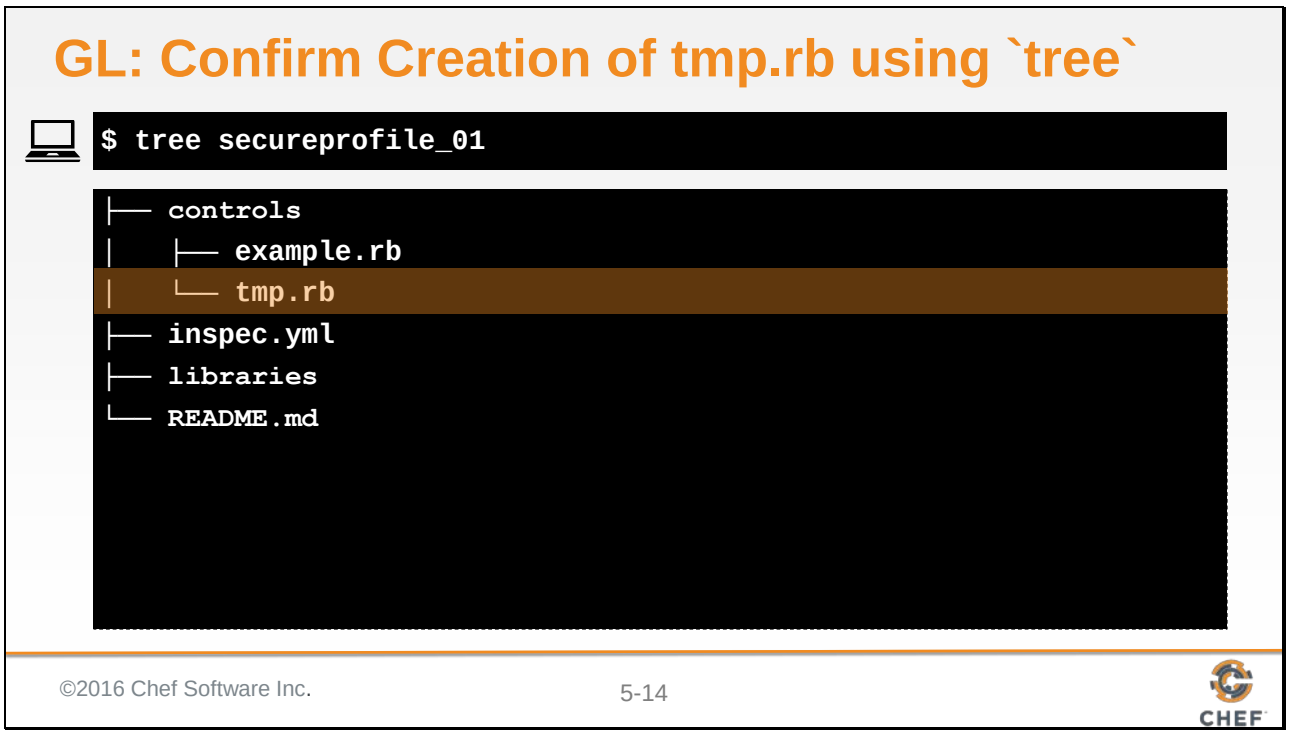
Slide 13

GL: Create the `tmp.rb` Control using the `cp` Command

```
$ cp ~/secureprofile_01/controls/example.rb
~/secureprofile_01/controls/tmp.rb
```

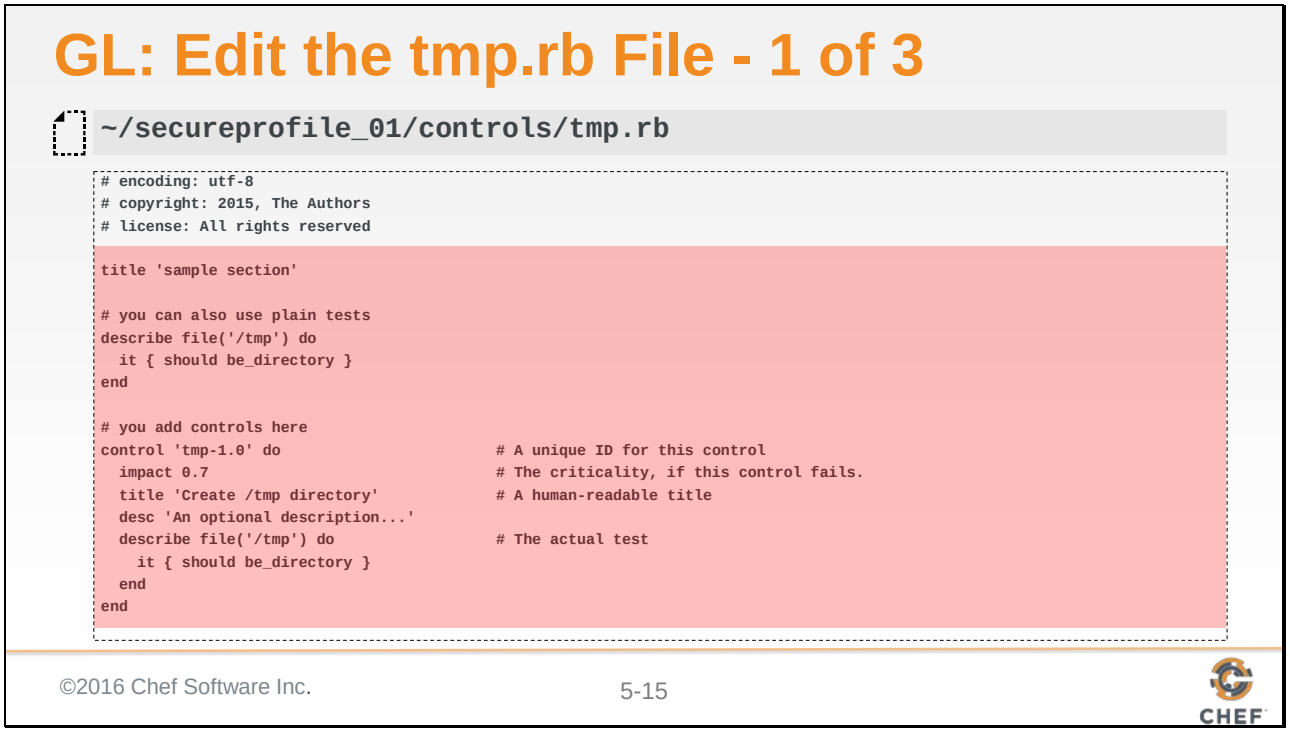

In this example you will use the `cp` command to make a copy of the `example.rb' control and name the copy `tmp/rb`.

Slide 14



You should now see that `tmp.rb` resides next to the default `example.rb`.

Slide 15



Delete everything highlighted in this example.

Slide 16

GL: Edit the tmp.rb File - 2 of 3

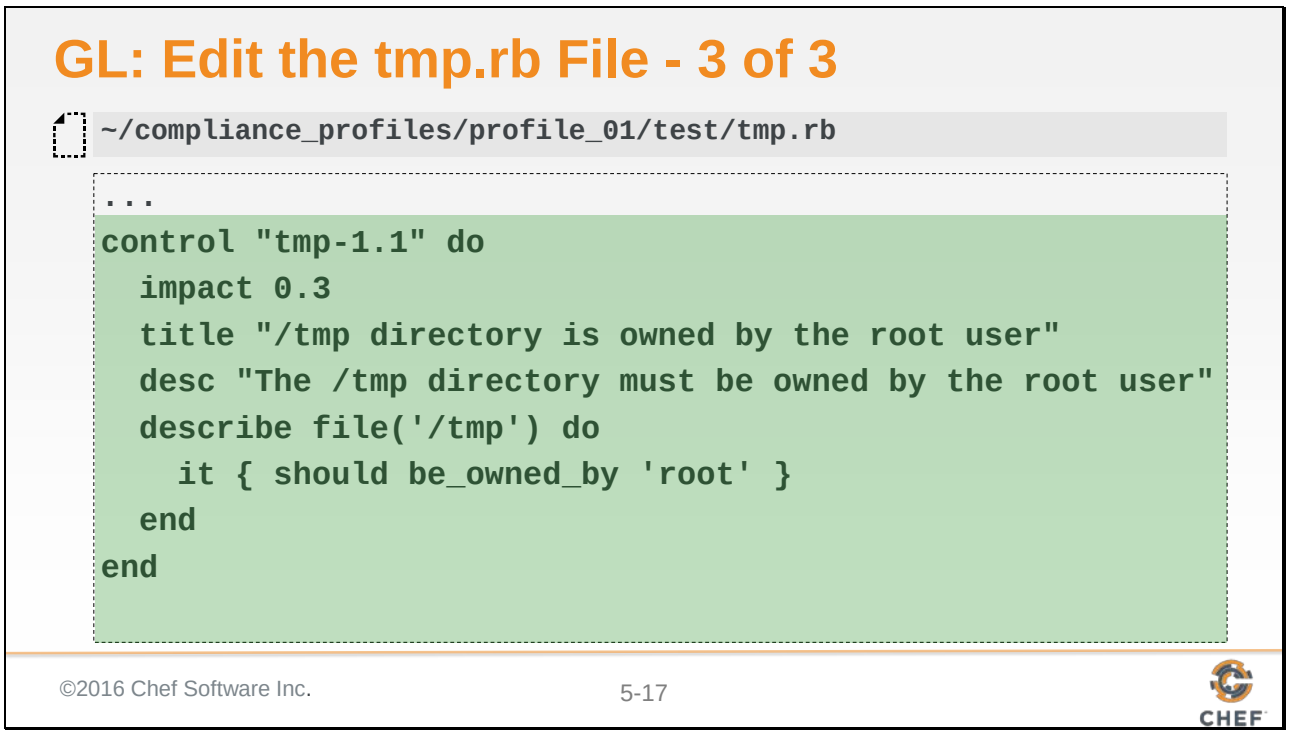
~/secureprofile_01/controls/tmp.rb

```
# encoding: utf-8
# copyright: 2015, The Authors
# license: All rights reserved
title '/tmp profile'

control "tmp-1.0" do
  impact 0.3
  title "Create /tmp directory"
  desc "A /tmp directory must exist"
  describe file('/tmp') do
    it { should be_directory }
  end
end
```


Write the first half of this control as shown here. You'll write the second half below this part in a moment.

Instructor Note: It's generally correct to use single quotes unless string interpolation is used, in which doubles are correct. tbd - Check with inspec team about describe statement correctness. We need to make this example consistent wrt quotes.

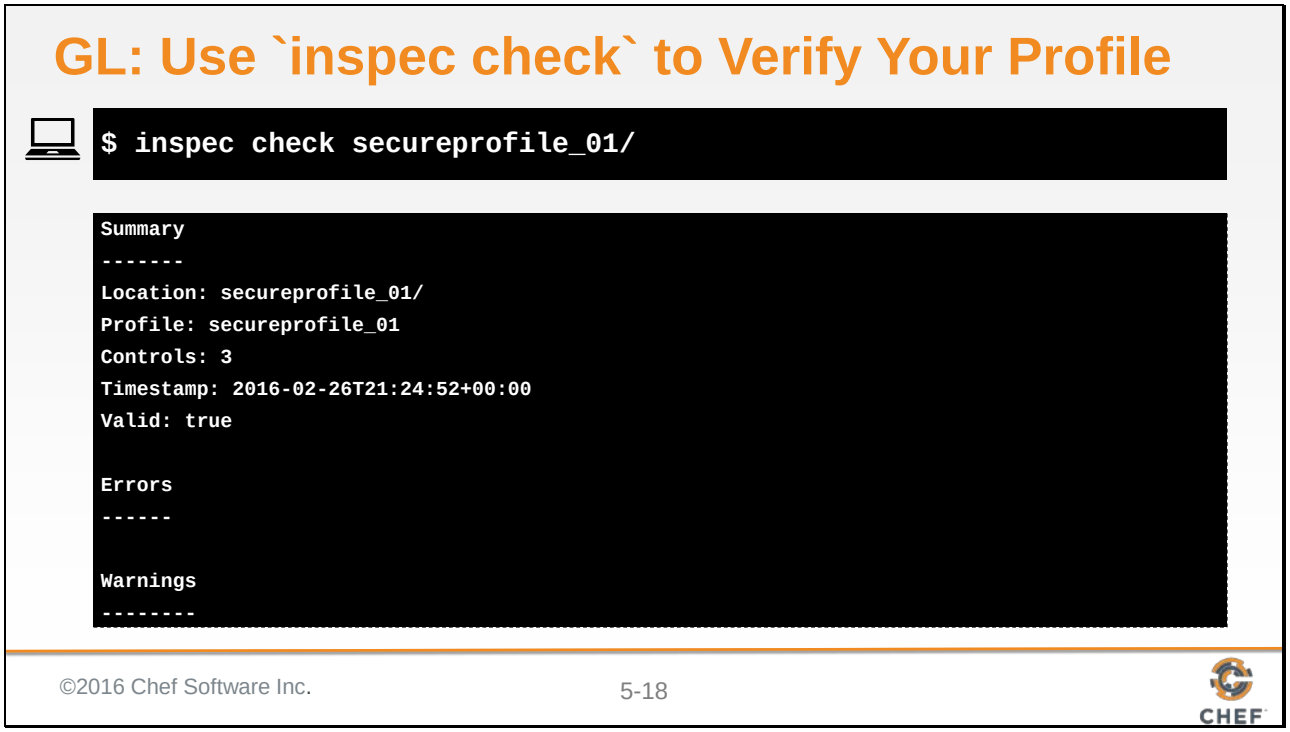
Slide 17



Write the second half of this control as shown here, just below the code you wrote on the preceding slide. We've pasted the entire control code below so you can see it better.

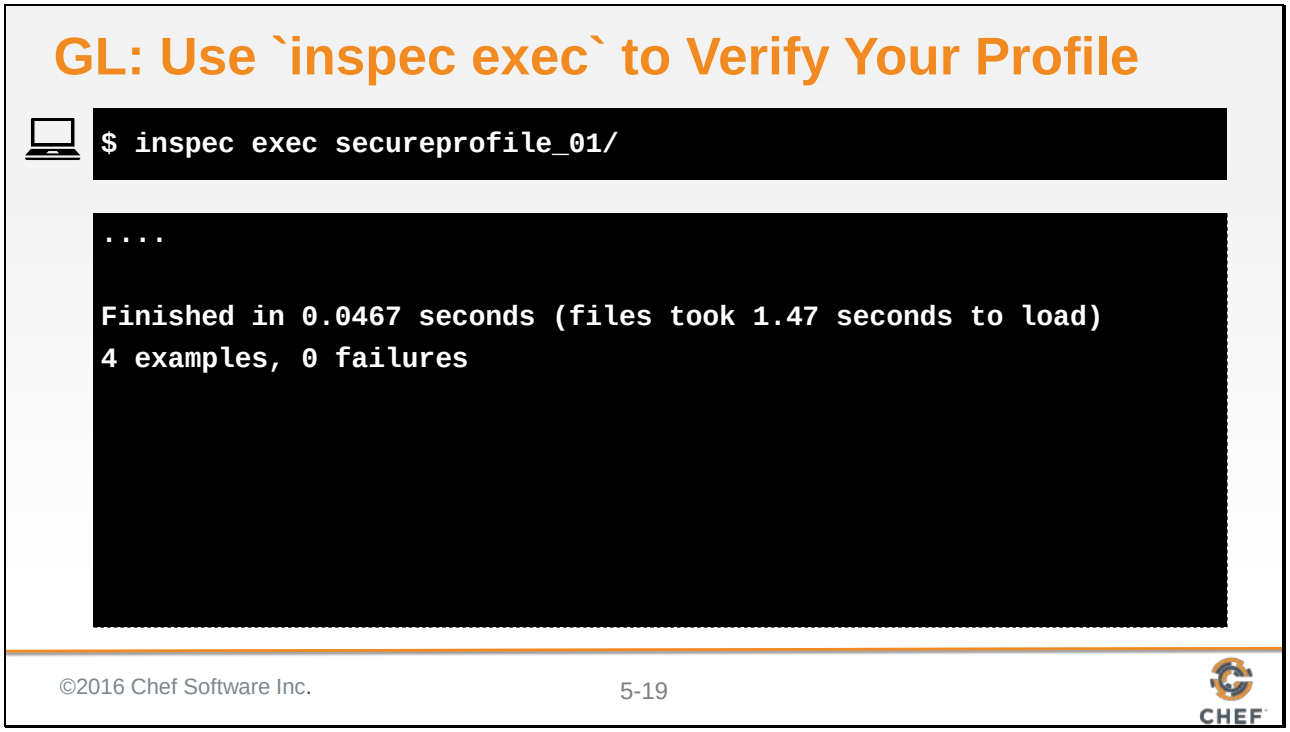
```
# encoding: utf-8
# copyright: 2015, The Authors
# license: All rights reserved
title '/tmp profile'

control "tmp-1.0" do

  impact 0.3
  title "Create /tmp directory"
  desc "A /tmp directory must exist"
  describe file('/tmp') do
    it { should be_directory }
  end
end

control "tmp-1.1" do
  impact 0.3
  title "/tmp directory is owned by the root user"
  desc "The /tmp directory must be owned by the root user"
  describe file('/tmp') do
  it { should be_owned_by 'root' }
  end
end
```

Slide 18



Now use 'inspec check` to verify the compliance profile code that you wrote. You should see no errors or warnings.

Slide 19



Now use 'inspec exec` to test your compliance profile against the node you are working on. You should see no failures.

Since your profile has passed the inspec tests, it is now ready to be uploaded to the Compliance Server. We can assume this new compl

Slide 20

DISCUSSION

Creating a Custom Profile

In the preceding group lab you created a custom Compliance profile and tested your profile with InSpec.

Your code passed the `inspec check` test and your system passed the `inspec exec` test.

But what would an `inspec exec` failure look like?

CHEF

Slide 21

Example of an `inspec exec` Failure

Let's say you modified your

~ secureprofile_01/controls/tmp.rb

and changed **`should be_owned_by root`** to **`should be_owned_by other`**

and then ran `inspec exec` against that file...

```
...
control "tmp-1.1" do
  impact 0.3
  title "/tmp directory is owned by the root user"
  desc "The /tmp directory must be owned by the root user"
  describe file('/tmp') do
    it { should be_owned_by other' }
  end
end
```


CHEF

Slide 22

Example: `inspec exec` Failure

```
$ inspec exec secureprofile_01/
```

```
...
Failures:

  1) File /tmp should be owned by "other"
     Failure/Error: DEFAULT_FAILURE_NOTIFIER = lambda { |failure, _opts| raise failure }
       expected `File /tmp.owned_by?("other")` to return true, got false
     # secureprofile_01/controls/tmp.rb:20:in `block (3 levels) in load'

Finished in 0.06284 seconds (files took 1.4 seconds to load)
4 examples, 1 failure

Failed examples:

rspec  # File /tmp should be owned by "other"
```


...this is an example of running `inspec exec` against the system using the `~/secureprofile_01/controls/tmp.rb' as modified on the preceding slide.

As you can see, based on the modified control, `inspec exec` expected the /tmp directory to be owned by `other` but in fact /tmp is owned by root.

Instructor Note: As an optional lab, you can have the students modify their `~/secureprofile_01/controls/tmp.rb' as shown on the previous slide and then run the command on this slide. If you do, make sure they change the `secureprofile_01/controls/tmp.rb' back so it checks that /tmp is owned by root.

Slide 23

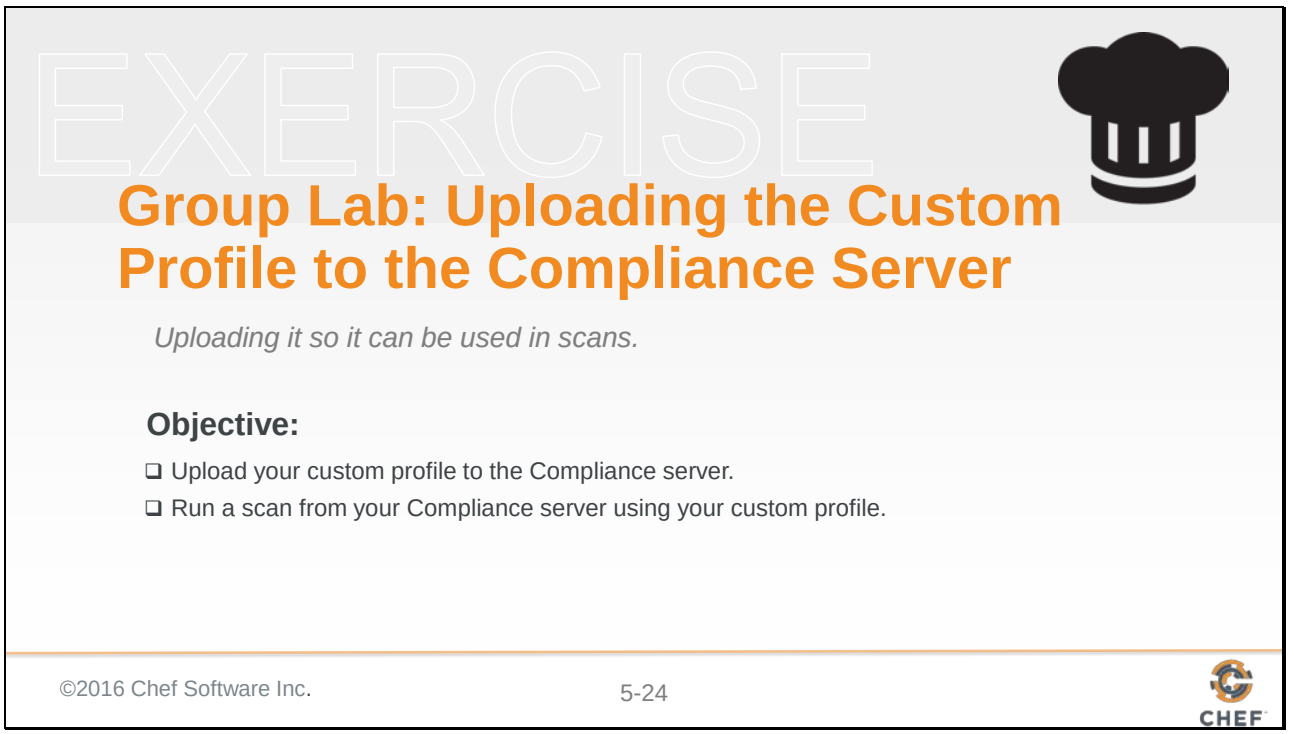
DISCUSSION

Uploading Custom Profiles to Compliance Server

inspec v 0.14.2 and above uses the `inspec compliance upload PATH` command to upload profiles from a workstation to the Compliance Server.

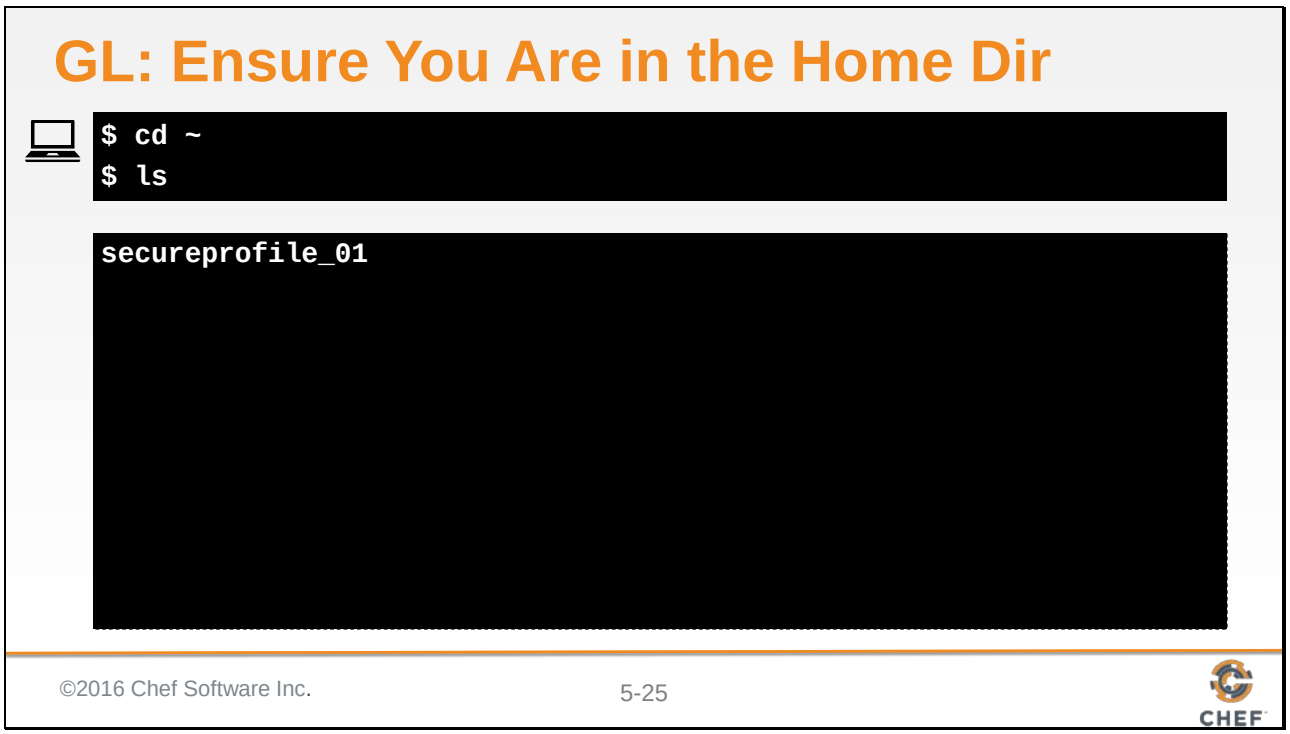
That command should be preceded by the `inspec compliance login SERVER --password=PASSWORD --user=USER` in order to first log in to the Compliance Server.

Slide 24



In addition to the uploading procedure we'll do in the exercise, in the workplace you could also upload custom profiles using an API.

Slide 25



From your Linux node, ensure you are in your Home directory and type `ls` to see your compliance profile

Slide 26

GL: Using 'inspec compliance` Commands

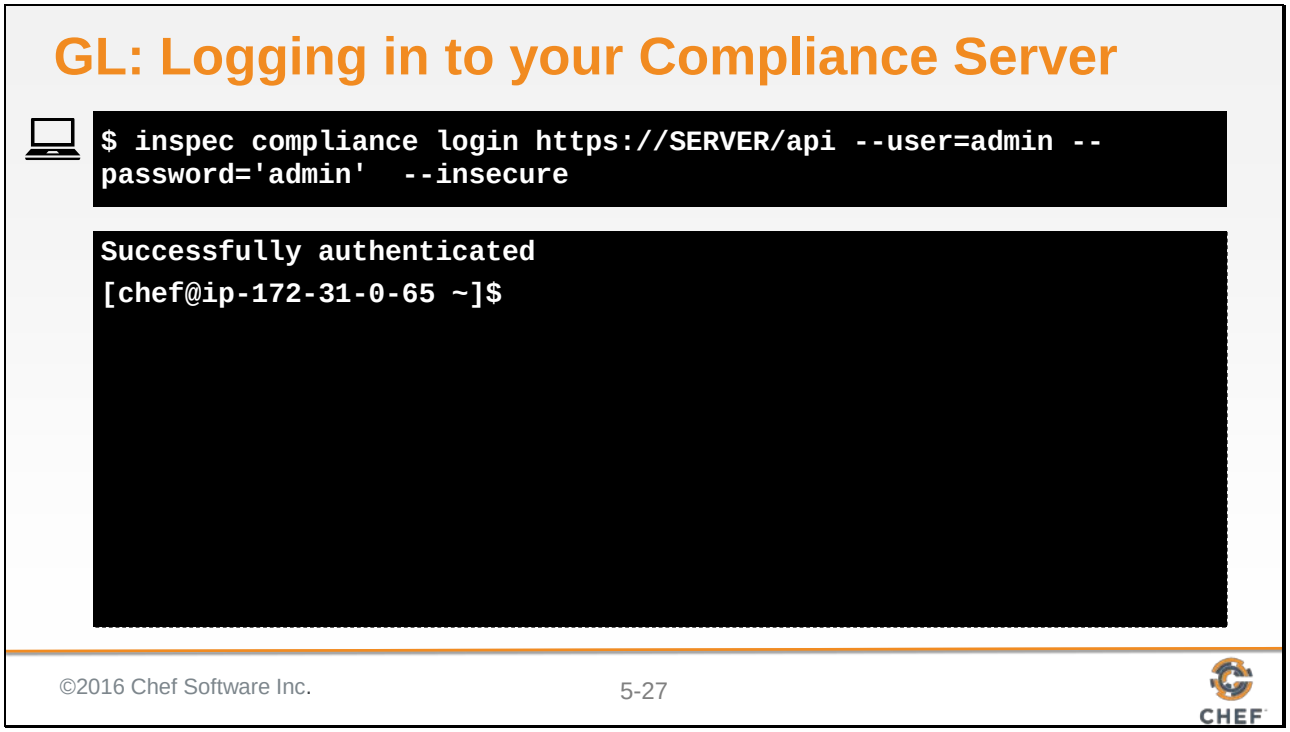
```
$ inspec compliance help
```

```
Commands:
  inspec compliance exec PROFILE                                    # executes ...
  inspec compliance help [COMMAND]                                  # Describe ...
  inspec compliance login SERVER --password=PASSWORD --user=USER    # Log in to...
  inspec compliance logout                                          # user logo...
  inspec compliance profiles                                        # list all ...
  inspec compliance upload PATH                                     # uploads a...
  inspec compliance version                                         # displays ...
```


CHEF

This example shows the options for the `inspec compliance` command. You'll be using the `inspec compliance login SERVER --password=PASSWORD --user=USER` in a moment to first log into your Compliance server.

Slide 27

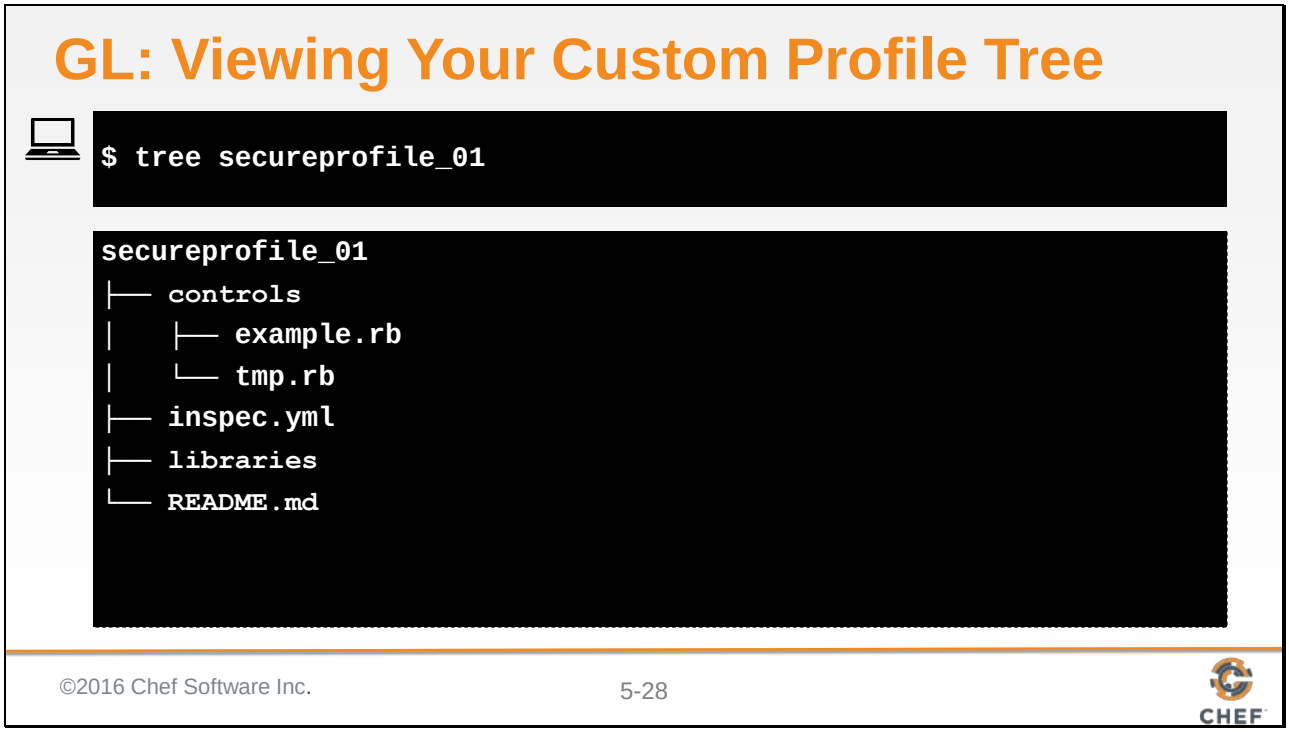


Use your compliance server's IP address in place of SERVER in this example.

Note that the credentials used here are the credentials you created for your Compliance Server UI (admin/admin), not the node's.

Note: We are using the `--insecure option` in this lab because we are not using valid self-signed certificates.

Slide 28



Notice that even though our VM is now logged into the Compliance server, commands such as `tree` are still executed against the VM we are on.

Slide 29

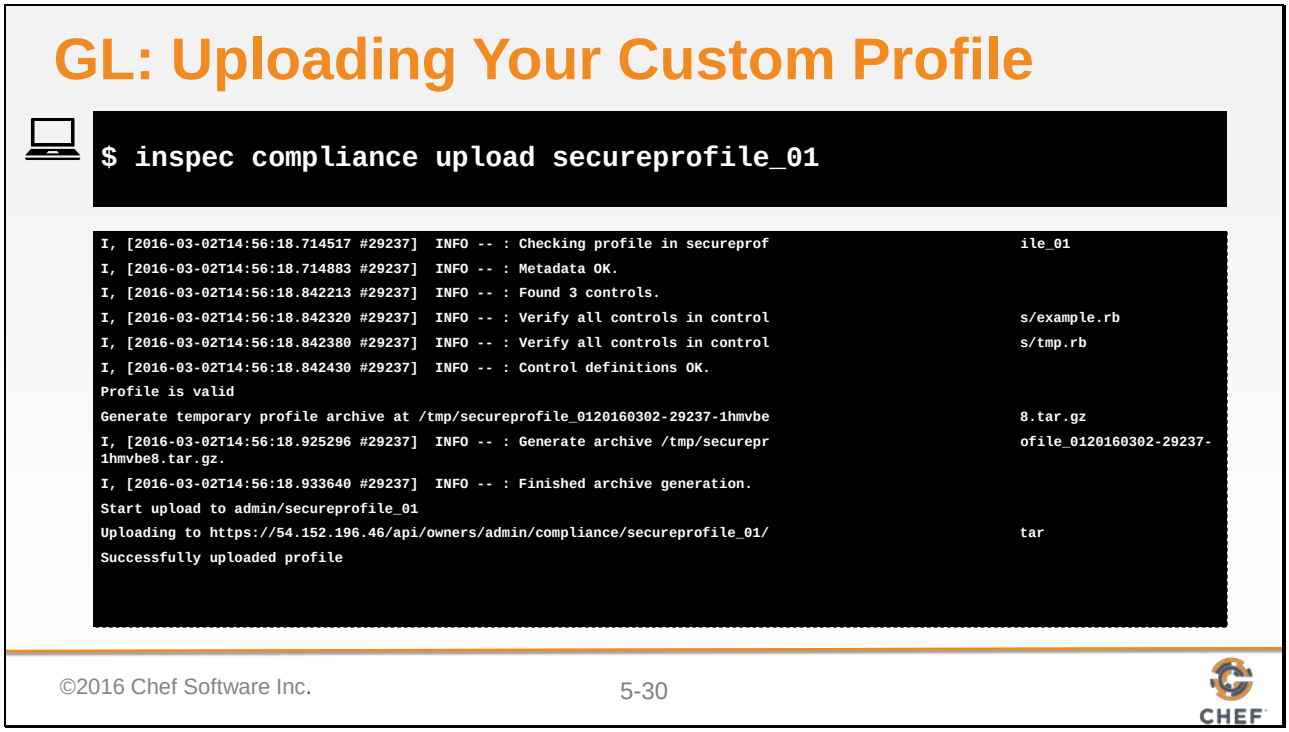
GL: Viewing Compliance Profiles on Your Compliance Server

```
$ inspec compliance profiles
```

```
Available profiles:
------------------
 * admin/profile
 * admin/profile-cis-3.1
 * admin/profile1
 * base/apache
 * base/linux
 * base/mysql
 * base/postgres
 * base/ssh
 * base/windows
 * cis/cis-ubuntu-level1
 * cis/cis-ubuntu-level2
```


Again, this `inspec compliance profiles` command is executed against the VM we are on.

Slide 30



This `inspec compliance upload secureprofile_01` command is now uploading our custom profile to the Compliance server.

Slide 31

GL: Viewing Compliance Profiles on Your Compliance Server

```
$ inspec compliance profiles
```

```
Available profiles:
------------------
 * admin/profile
 * admin/profile-cis-3.1
 * admin/profile1
 * admin/secureprofile_01
 * base/apache
 * base/linux
 * base/mysql
 * base/postgres
 * base/ssh
 * base/windows
 * cis/cis-ubuntu-level1
 * cis/cis-ubuntu-level2
```


This `inspec compliance profiles` is now executing against our Compliance server, thus we are now looking at the Compliance profiles that reside on the Compliance server. Notice that your secureprofile_01 has been uploaded to the compliance server.

Slide 32

GL: Logging out of Your Compliance server

```
$ inspec compliance logout
```

```
Successfully logged out
```

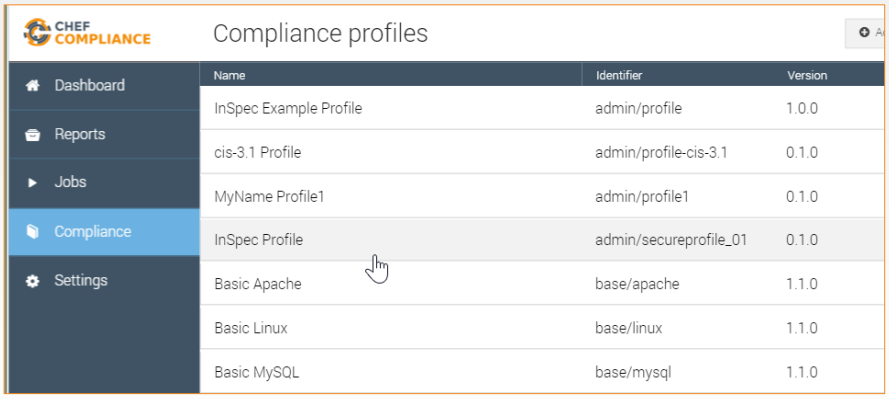

CHEF

Slide 33

GL: Viewing Your Uploaded Custom Profile

Use a web browser to navigate the Compliance tab of your Compliance server.

Notice that your custom profile is present.

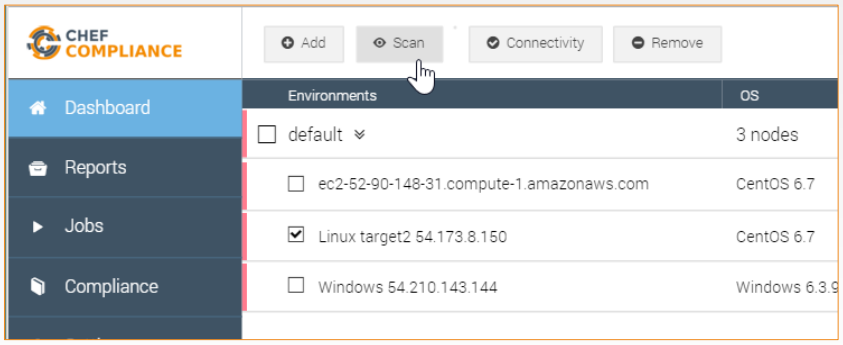


CHEF

Slide 34

GL: Testing Your Uploaded Custom Profile

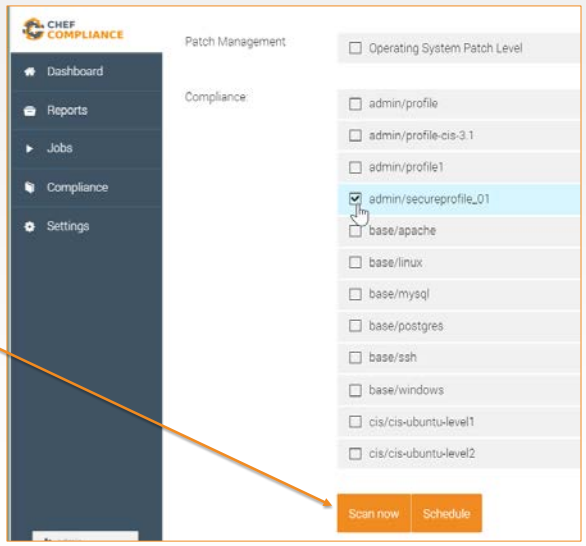
Navigate to the Compliance dashboard, click your Linux target, and then click **Scan**.



Slide 35

GL: Testing Your Uploaded Custom Profile

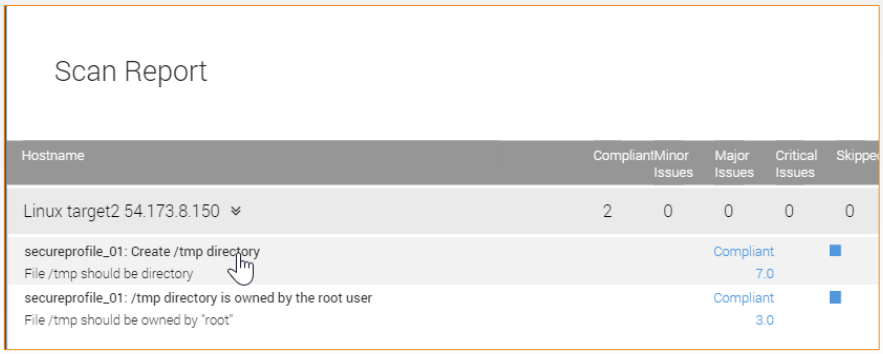
Select only your custom profile and then click **Scan now**.



Slide 36

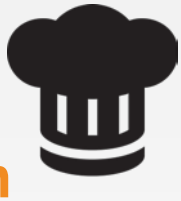
GL: Testing Your Uploaded Custom Profile

You should now see that your custom profile works properly and your Linux target is in compliance.



Slide 37

EXERCISE



Group Lab: Uploading the Custom Profile to the Compliance Server

Uploading it so it can be used in scans.

Objective:

- ✓ Upload your custom profile to the Compliance server.
- ✓ Run a scan from your Compliance server using your custom profile.

CHEF

We have now completed this group lab.

Slide 38

Review Questions

1. What is the difference between `inspec check` and `inspec exec'?
2. What does `inspec init profile` do?

Instructor Note Answers:
1. `inspec check` just verifies the code--it doesn't actually test a system. `inspec exec' will run the tests against a system.
Additional questions will be added when we test this course on the Late February v1.0 Compliance software since inspec.yml will replace the inspec requirement of metadata.rb. Plus the new `generate profile` command will change the way we create profiles too.
2. It creates the directories and baseline files that are required fro creating profiles.

Slide 39

6: Applying Compliance Frameworks Using InSpec

Useful references for this module:

https://docs.chef.io/release/compliance_1-0/dsl_compliance.html

https://docs.chef.io/inspec_reference.html

Slide 2

Objectives

After completing this module, you should be able to:

- Translate CIS (Center for Internet Security) specifications into InSpec tests.
- Translate DoD (Department of Defense) specifications into InSpec tests.

Slide 3

CONCEPT

CIS Compliance Frameworks

The CIS Security Benchmarks program provides well-defined, un-biased and consensus-based industry best practices to help organizations assess and improve their security.

Resources include secure configuration benchmarks, automated configuration assessment tools and content, security metrics and security software product certifications.

https://benchmarks.cisecurity.org/

CHEF

Slide 4

EXERCISE

Group Lab:
Compliance Frameworks - CIS Linux

Translating a CIS benchmark into an InSpec control and Compliance profile.

Objective:

- ❑ Download the benchmark PDF for the platform of your scanning target
- ❑ Implement Section 3 - Specialty Purpose Services as InSpec controls.

CHEF

Slide 5

GL: Downloading the CIS Benchmarks for Linux

1. Go to: https://benchmarks.cisecurity.org/
2. Click the **Products & Services** tab.
3. Click **Benchmarks**.

Slide 6

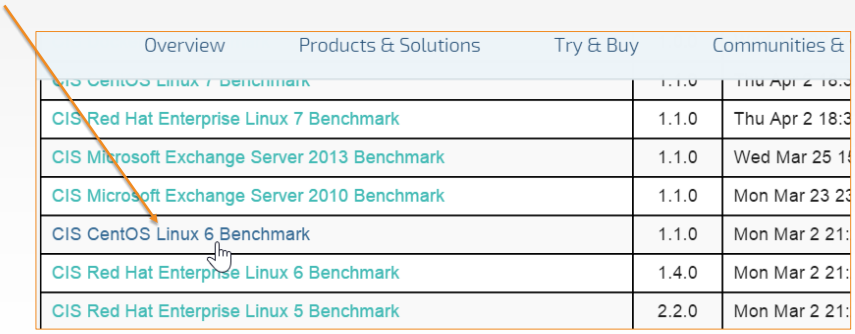
GL: Downloading the CIS Benchmarks

Scroll down to the **Available Free of Charge** section and then click **Current version of CIS Benchmarks.**

Slide 7

GL: Downloading the CIS Linux Benchmarks

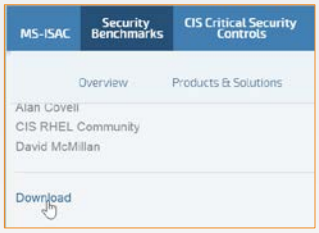
Scroll down to the benchmark for the system you're running. In our case, click the **CIS CentOS Linux 6 Benchmark** link.



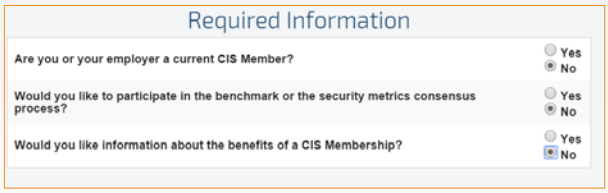
Slide 8

GL: Downloading the PDF

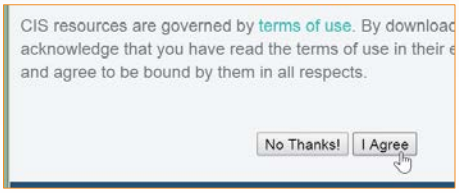
1. Scroll down to the bottom of the page and click **Download**.



2. On the resulting page, click the **Required Information** radio buttons.



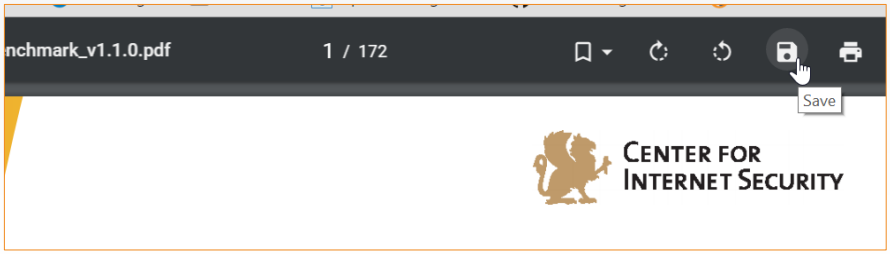
3. Then scroll down and click the **I Agree** button.



Slide 9

GL: Downloading the CIS Benchmarks

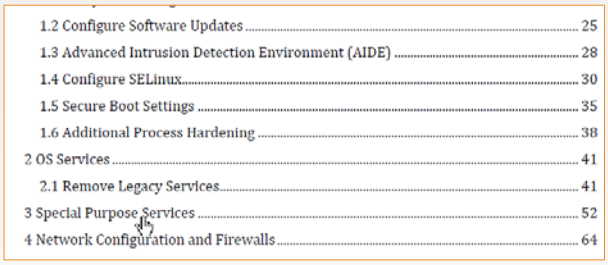
Save the PDF to your laptop. If a Save icon is not obvious, you should be able to save the PDF by right-clicking it.



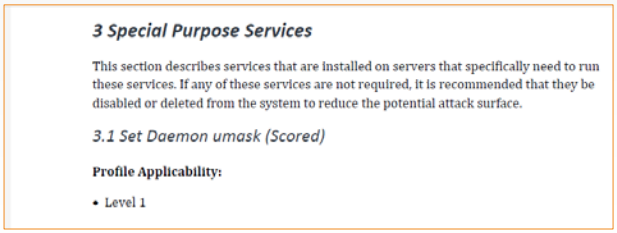
Slide 10

GL: CIS Benchmarks

Open the PDF and then from the Table of Contents, click the **Special Purpose Services** bookmark or otherwise go to that section.

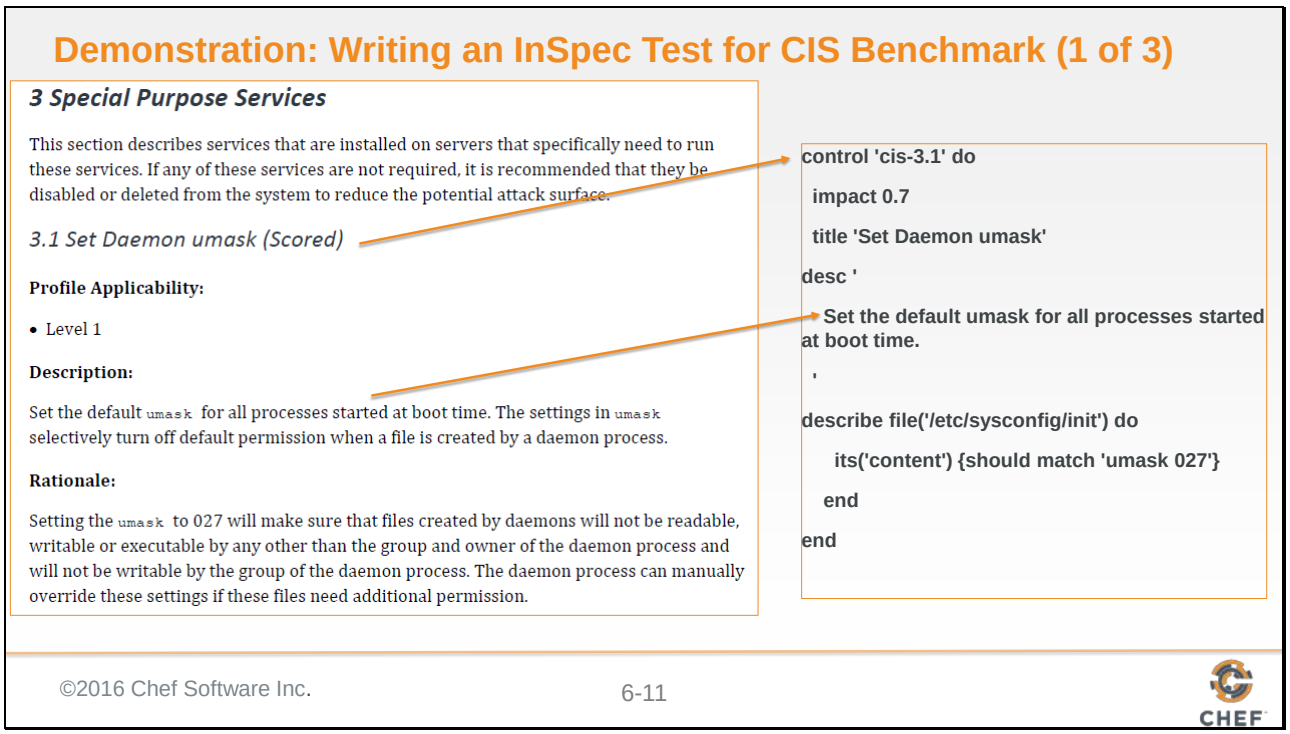


Go to Section 3.1.



CHEF

Slide 11



The text in the example on the left can be used to create a Compliance Profile control (on the right).

In this example, you can name the control the same as the section in the CIS document:
control 'cis-3.1'
The Description text from the CIS document can be used to write the `desc` section.

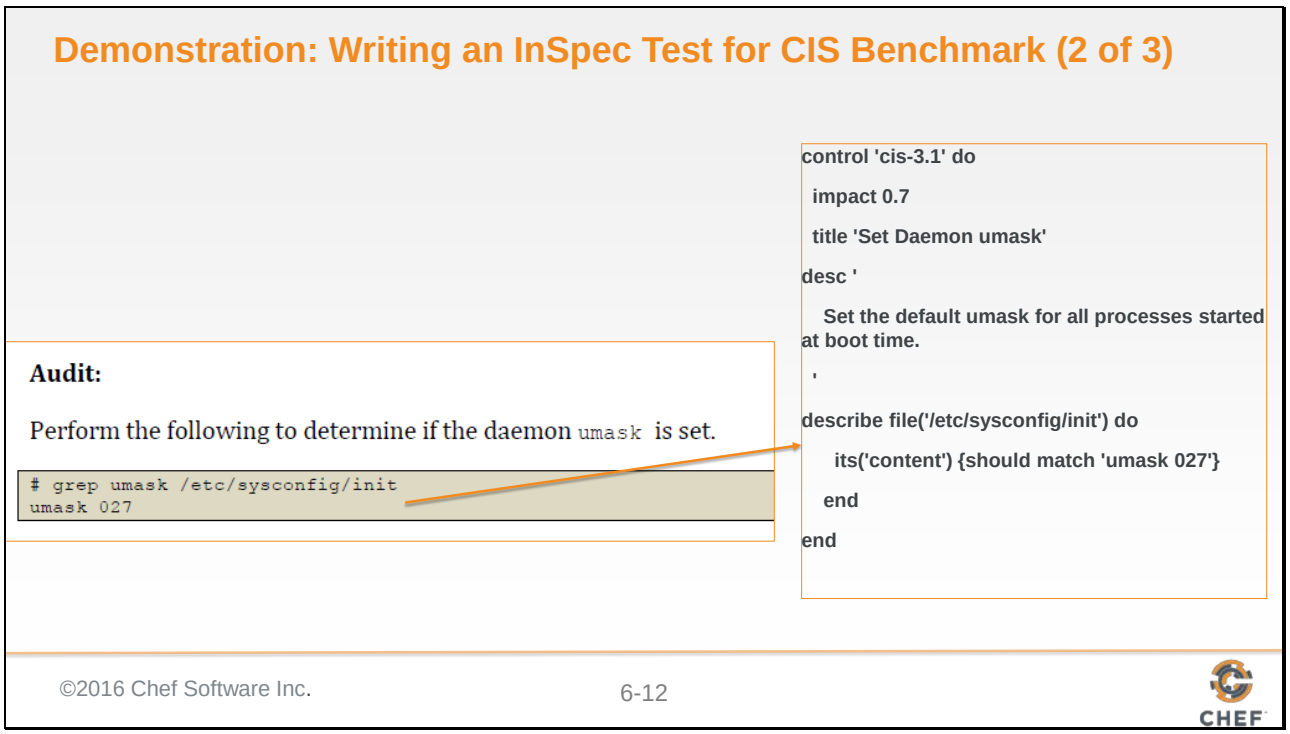
As you can see on that page, the *3.1 Set Daemon umask (Scored)* section says:

Description:

Set the default umask for all processes started at boot time. The settings in umask selectively turn off default permission when a file is created by a daemon process.

Rationale:
Setting the umask to 027 will make sure that files created by daemons will not be readable, writable or executable by any other than the group and owner of the daemon process and will not be writable by the group of the daemon process. The daemon process can manually override these settings if these files need additional permission.

Slide 12



If you scrolled down in that section, you will see an Audit example.

So based on the Audit example on the left, you could write an InSpec test as shown on the right. In this way you can subsequently use this custom profile to scan nodes for umask compliance.

https://docs.chef.io/release/compliance_1-0/dsl_compliance.html

https://docs.chef.io/inspec_reference.html

Slide 13

Demonstration: Writing an InSpec Test for CIS Benchmark (3 of 3)

```
control 'cis-3.1' do
 impact 0.7
 title 'Set Daemon umask'
desc '
   Set the default umask for all processes started
at boot time.
 '
describe file('/etc/sysconfig/init') do
     its('content') {should match 'umask 027'}
   end
end
```

After writing the test, in the workplace you should:

- Use 'inspec exec' to test the control.
- Package the custom compliance profile and upload it to your Compliance server.

After writing the test, in the workplace you should:

Use 'inspec exec' to test the control.
Package the custom compliance profile and upload it to your Compliance server.

Slide 14

EXERCISE

Group Lab:
Compliance Frameworks - CIS Windows

Translating a CIS benchmark into an InSpec control and Compliance profile.

Objective:

- ❑ Download the benchmark PDF for the platform of your scanning target
- ❑ Implement Section 3 - Specialty Purpose Services as InSpec controls.

CHEF

Slide 15

GL: Downloading the CIS Benchmarks for Windows

1. Go to: https://benchmarks.cisecurity.org/
2. Click the **Products & Services** tab.
3. Click **Benchmarks**.

Slide 16

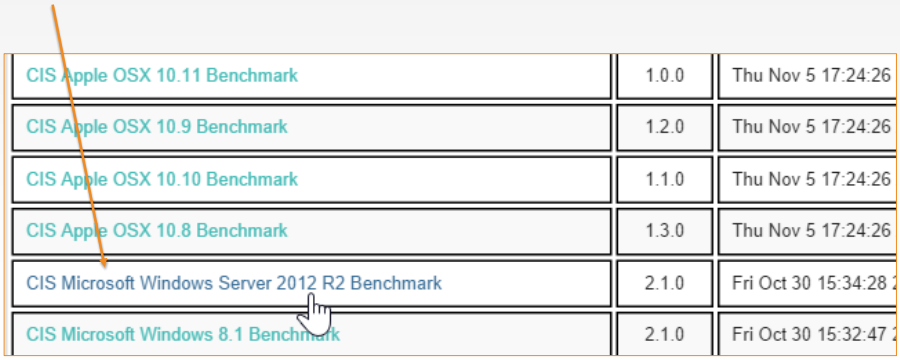
GL: Downloading the CIS Benchmarks for Windows

Scroll down to the **Available Free of Charge** section and then click **Current version of CIS Benchmarks.**

Slide 17

GL: Downloading the CIS Benchmarks for Windows

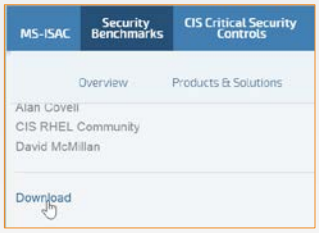
Scroll down to the benchmark for the system you're running. In our case, click the **CIS Microsoft Windows Server 2012 R2 Benchmark** link.



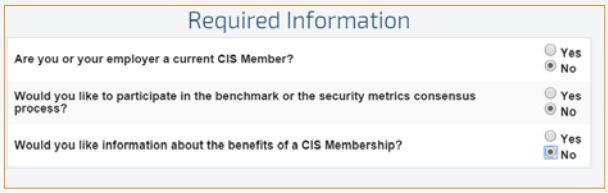
Slide 18

GL: Downloading the PDF

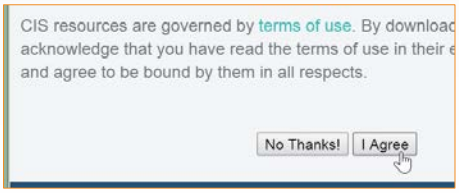
1. Scroll down to the bottom of the page and click **Download**.



2. On the resulting page, click the **Required Information** radio buttons.



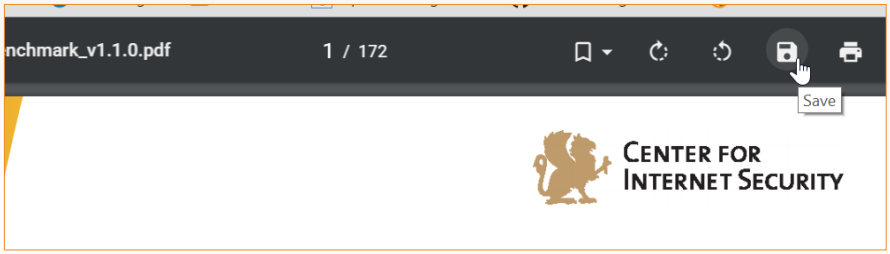
3. Then scroll down and click the **I Agree** button.



Slide 19

GL: Downloading the CIS Benchmarks for Windows

Save the PDF to your laptop. If a Save icon is not obvious, you should be able to save the PDF by right-clicking it.

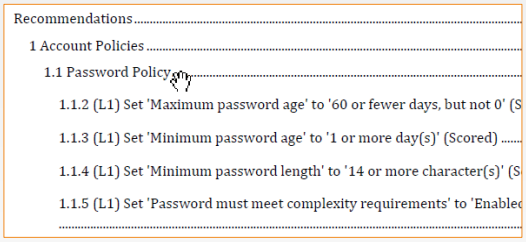


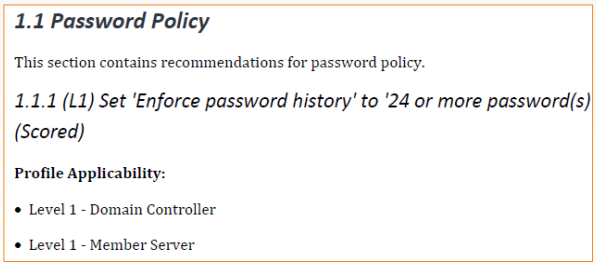
GL: CIS Benchmarks for Windows

Implement Section 1.1 - Password Policy as InSpec controls with the profile of Level 1 - Member Server.

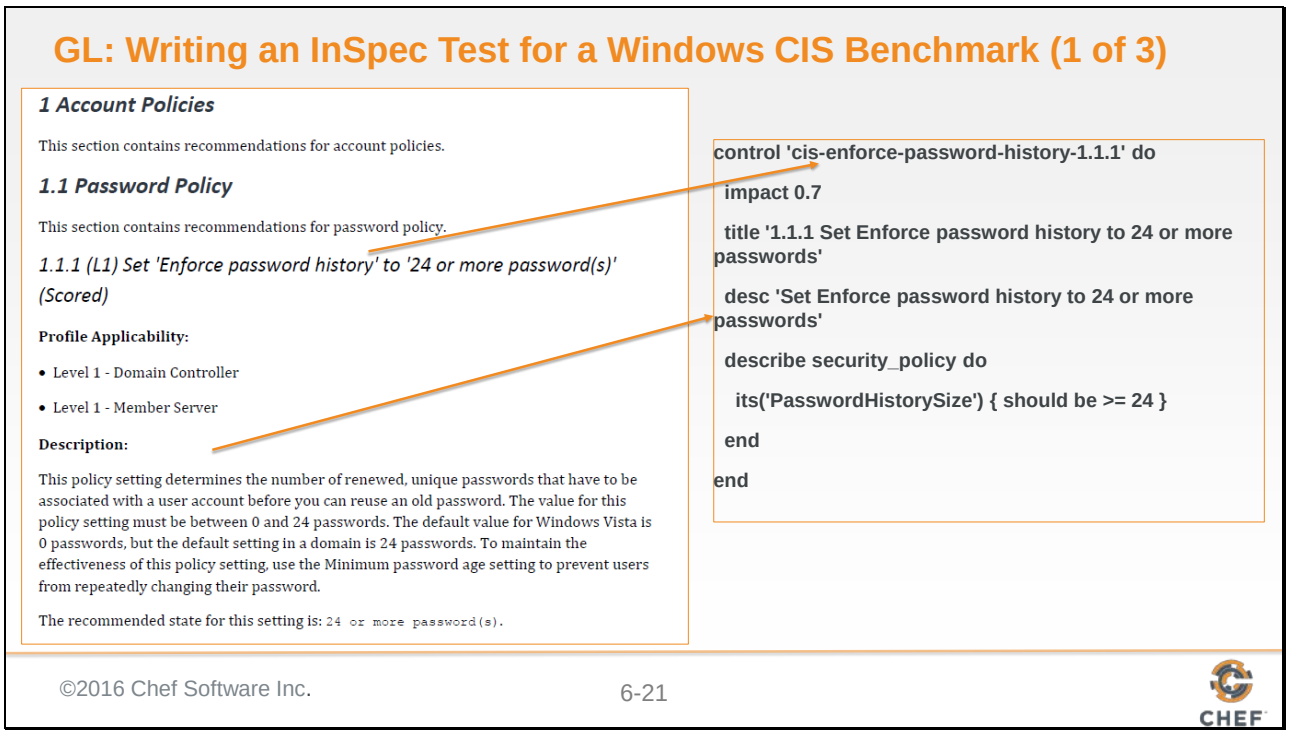
Use 'inspec exec' to test the control.

Package profile and upload to your Compliance server.





Slide 21



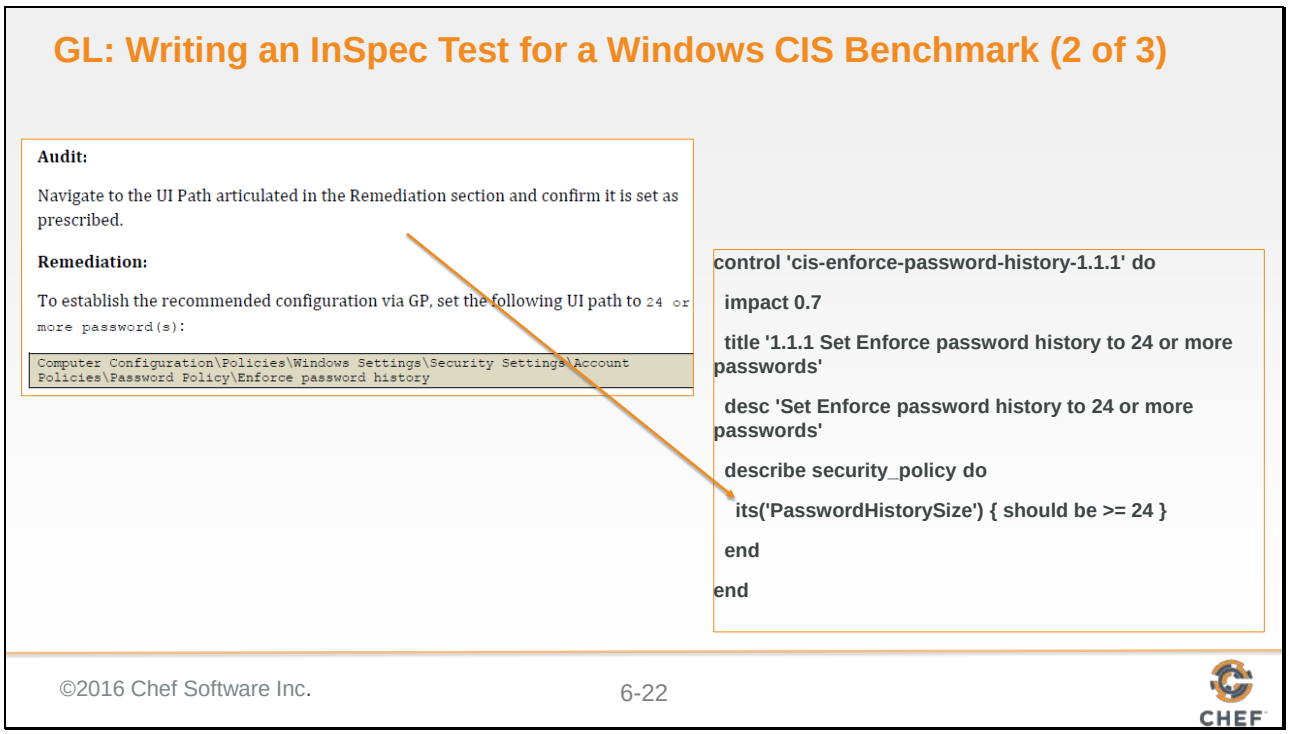
Scroll down to the *1.1 Password Policy* section,

The text in the example on the left can be used to create a Compliance Profile control (on the right).

In this example, you can name the control the same as the section in the CIS document: **cis-enforce-password-history-1.1.1**

The Description text from the CIS document can be used to write the `desc` section.

Slide 22



If you scrolled down in that section, you will see an Audit example.

So based on the Audit example on the left, you could write an InSpec test as shown on the right. In this way you can subsequently use this custom profile to scan nodes for umask compliance.

The path shown in this example could be used if you wanted to manually navigate on the Windows node to see how password history parameter is set. However, when the Compliance Server scans the node, the Compliance server's inspec will use cmd = inspec.command('secedit /export /cfg win_secpol.cfg') to locate the parameter.

https://github.com/chef/inspec/blob/master/lib/resources/security_policy.rb

Slide 23

GL: Writing an InSpec Test for a Windows CIS Benchmark (3 of 3)

This inspec code below shows how inspec can scan for security policy compliance by parsing **`secedit /export /cfg win_secpol.cfg`**.

```
# load security content
  def load
    # export the security policy
    cmd = inspec.command('secedit /export /cfg win_secpol.cfg')
    return nil if cmd.exit_status.to_i != 0
```

```
control 'cis-enforce-password-history-1.1.1' do
 impact 0.7
 title '1.1.1 Set Enforce password history to 24 or more passwords'
 desc 'Set Enforce password history to 24 or more passwords'
 describe security_policy do
  its('PasswordHistorySize') { should be >= 24 }
 end
end
```

https://github.com/chef/inspec/blob/master/lib/resources/security_policy.rb

The Compliance server's inspec will parse cmd = inspec.command('secedit /export /cfg win_secpol.cfg') to locate the parameter.

The following URL shows the full inspec/lib/resources/security_policy.rb code.

https://github.com/chef/inspec/blob/master/lib/resources/security_policy.rb

Slide 24

GL: CIS Benchmarks for Windows

After writing such a Compliance policy, in a production environment you would use 'inspec exec' to test the control.

Then you could package the profile and upload it to your Compliance server.

```
control 'cis-enforce-password-history-1.1.1' do
  impact 0.7
  title '1.1.1 Set Enforce password history to 24 or more passwords'
  desc 'Set Enforce password history to 24 or more passwords'
  describe security_policy do
    its('PasswordHistorySize') { should be >= 24 }
  end
end
```

Slide 25

CONCEPT

DoD Compliance Frameworks

Department of Defense (DoD) STIGs

The Security Technical Implementation Guides (STIGs) and the NSA Guides are the configuration standards for DOD IA and IA-enabled devices/systems.

CHEF

Slide 26

CONCEPT

DoD Compliance Frameworks

Department of Defense (DoD) STIGs

Since 1998, DISA has played a critical role enhancing the security posture of DoD's security systems by providing the Security Technical Implementation Guides (STIGs).

The STIGs contain technical guidance to "lock down" information systems/software that might otherwise be vulnerable to a malicious computer attack.

http://iase.disa.mil/stigs/Pages/index.aspx

CHEF

Slide 27

EXERCISE

GL: Compliance Frameworks - DoD

Translating a CIS benchmark into an InSpec control and Compliance profile.

Objective:

- ❑ Download STIGViewer2.
- ❑ Download DoD Security Rules for RHEL 6 and Windows 2012 MS (Member Server).
- ❑ Explain how to translate the DoD Security Rule into a Chaf Compliance profile control.

CHEF

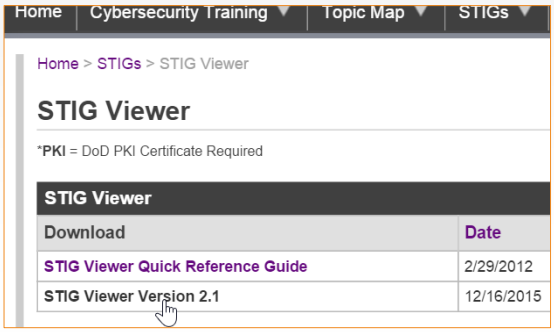
Slide 28

GL: Download STIGViewer2.x

From your local laptop, go to this site...

http://iase.disa.mil/stigs/Pages/stig-viewing-guidance.aspx

...and download the latest version of the STIG Viewer



CHEF

Go to this site...
http://iase.disa.mil/stigs/Pages/stig-viewing-guidance.aspx

...and download the latest version of the STIG Viewer. In this example we are downloading Version 2.1.

Slide 29

GL: Launch STIGViewer2.x

Click the STIGViewer_2.1 shortcut or otherwise launch the STIGViewer_2.1 viewer.

If it doesn't launch, you may need to install the latest Java Runtime Environment (JRE) as indicated on the next slide.

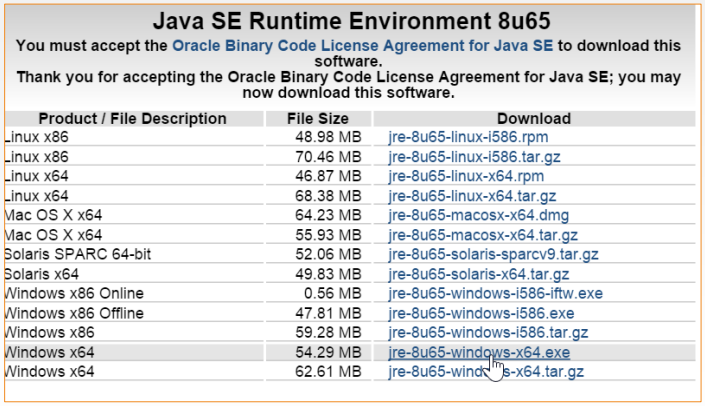
Try to run the STIG Viewer. If it fails due to a Java error, you may need to install the latest Java Runtime Environment (JRE) as indicated on the next slide.

Slide 30

GL: Download Java JRE if Necessary

You may need to install the latest Java Runtime Environment (JRE) if your STIG Viewer doesn't launch when clicked.

http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html



Java SE Runtime Environment 8u65

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

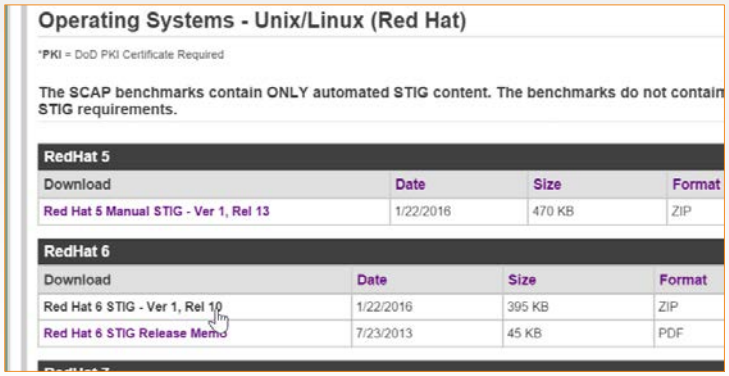
Product / File Description	File Size	Download
Linux x86	48.98 MB	jre-8u65-linux-i586.rpm
Linux x86	70.46 MB	jre-8u65-linux-i586.tar.gz
Linux x64	46.87 MB	jre-8u65-linux-x64.rpm
Linux x64	68.38 MB	jre-8u65-linux-x64.tar.gz
Mac OS X x64	64.23 MB	jre-8u65-macosx-x64.dmg
Mac OS X x64	55.93 MB	jre-8u65-macosx-x64.tar.gz
Solaris SPARC 64-bit	52.06 MB	jre-8u65-solaris-sparcv9.tar.gz
Solaris x64	49.83 MB	jre-8u65-solaris-x64.tar.gz
Windows x86 Online	0.56 MB	jre-8u65-windows-i586-iftw.exe
Windows x86 Offline	47.81 MB	jre-8u65-windows-i586.exe
Windows x86	59.28 MB	jre-8u65-windows-i586.tar.gz
Windows x64	54.29 MB	jre-8u65-windows-x64.exe
Windows x64	62.61 MB	jre-8u65-windows-x64.tar.gz

Slide 31

GL: Download STIG Profiles for Red Hat 6

Download the STIG profiles for RHEL 6 from this site and remember the location to where you downloaded onto your laptop.

http://iase.disa.mil/stigs/os/unix-linux/Pages/red-hat.aspx



Operating Systems - Unix/Linux (Red Hat)

*PKI = DoD PKI Certificate Required

The SCAP benchmarks contain ONLY automated STIG content. The benchmarks do not contain STIG requirements.

RedHat 5			
Download	Date	Size	Format
Red Hat 5 Manual STIG - Ver 1, Rel 13	1/22/2016	470 KB	ZIP

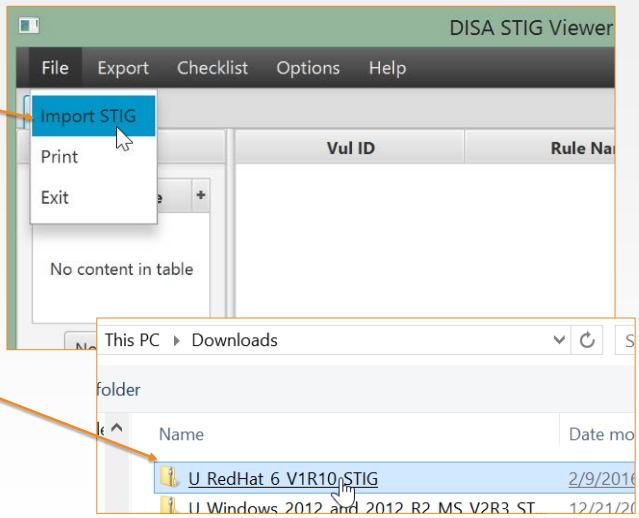
RedHat 6			
Download	Date	Size	Format
Red Hat 6 STIG - Ver 1, Rel 10	1/22/2016	395 KB	ZIP
Red Hat 6 STIG Release Memo	7/23/2013	45 KB	PDF

Slide 32

GL: Import STIG Profiles for Red Hat 6

Click **File > Import STIG**.

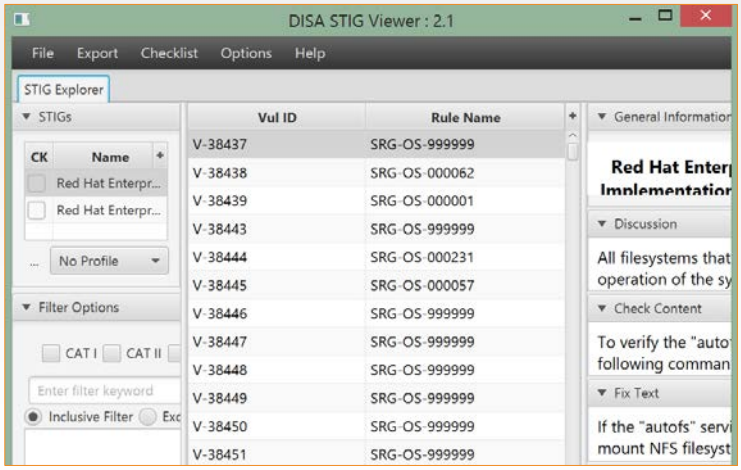
Navigate to the STIG profiles file you just downloaded and click the file.



Slide 33

GL: STIG Profiles for Red Hat 6

Your STIG viewer should now be populated with DoD Security Rule profiles.

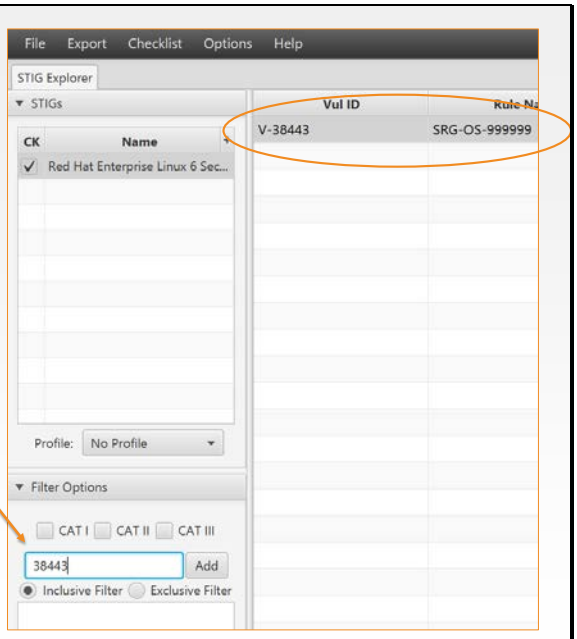


Slide 34

GL: Filter STIG Profiles

Type **38443** in the filter field.

Notice how the center pane now lists only one DoD Security Rule.



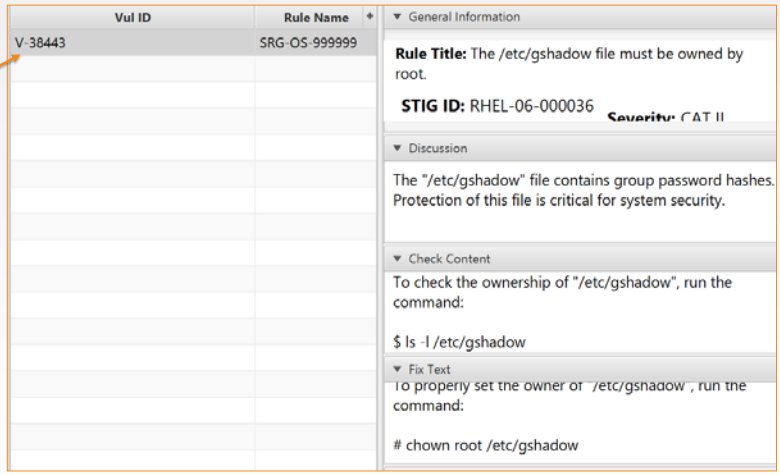
CHEF

TBD SCAP Roadmap.

Slide 35

GL: Writing Compliance Profiles from DoD Rules

```
#The /etc/gshadow file must be owned by
root.

#Severity: Medium

#The "/etc/gshadow" file contains group
password hashes. Protection of this

#file is critical for system security.


control 'v-38443-gshadow' do

  impact 0.5

  title 'v-38443: verify gshadow is
owned by root'

  describe file('/etc/gshadow') do

    it { should be_owned_by 'root' }

  end

end
```



The image on the right shows the right-side pane of the STIG viewer including the details of this DoD Security rule. This rule states that /etc/gshadow must be owned by root.

The image on the left shows a Chef Compliance Profile that was written based on the details of this DoD Security rule. Notice how the Chef Compliance Profile control name reflects the DoD Security rule name. This is a best practice that you should follow when writing Chef Compliance Profiles for DoD Security rules.

Slide 36

GL: Writing Compliance Profiles from DoD Rules

If you have permissions you can access a list of predefined DoD controls at this link:

https://github.com/chef/compliance-profiles/tree/DOD-STIG/stig/rhel6/test

```
#The /etc/gshadow file must be owned by
root.

#Severity: Medium

#The "/etc/gshadow" file contains group
password hashes. Protection of this

#file is critical for system security.


control 'v-38443-gshadow' do

  impact 0.5

  title 'v-38443: verify gshadow is
owned by root'

  describe file('/etc/gshadow') do

    it { should be_owned_by 'root' }

  end

end
```


CHEF

Slide 37

DoD STIG References

Windows 2012 - http://iase.disa.mil/stigs/os/windows/Pages/2012.aspx

Unix/Linux (Red Hat) - http://iase.disa.mil/stigs/os/unix-linux/Pages/red-hat.aspx

All Operating Systems - http://iase.disa.mil/stigs/os/Pages/index.aspx

Slide 38

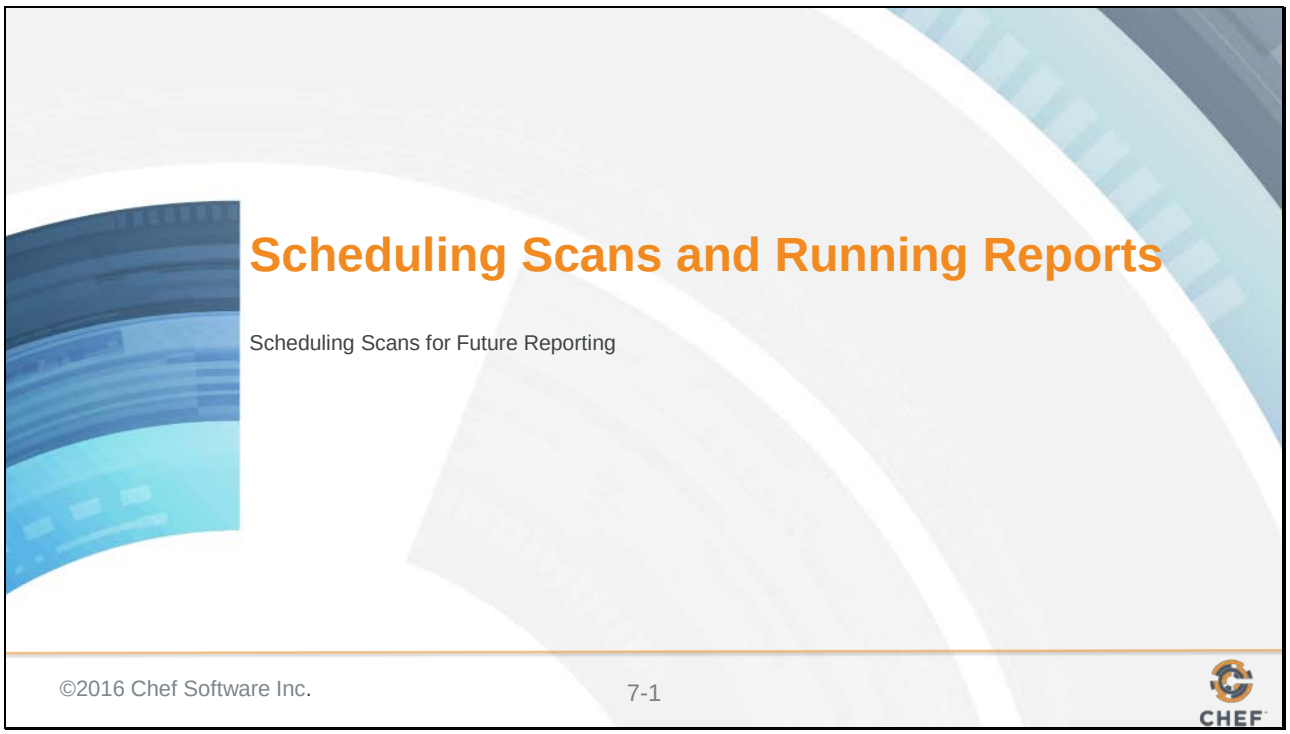
Compliance Profiles for Compliance Premium

Chef Compliance Premium customers can download all CIS profiles in a package that can be directly uploaded to the Chef Compliance server.

In the near future, NIST Security Standards/DoD profiles will be available for Chef Compliance Premium customers in a package that can be directly uploaded to the Chef Compliance server.

Slide 39

7: Scheduling Scans and Running Reports



Slide 2

Objectives

After completing this module, you should be able to:

- Schedule scans.
- View pending jobs.
- View and export reports.

Slide 3

CONCEPT

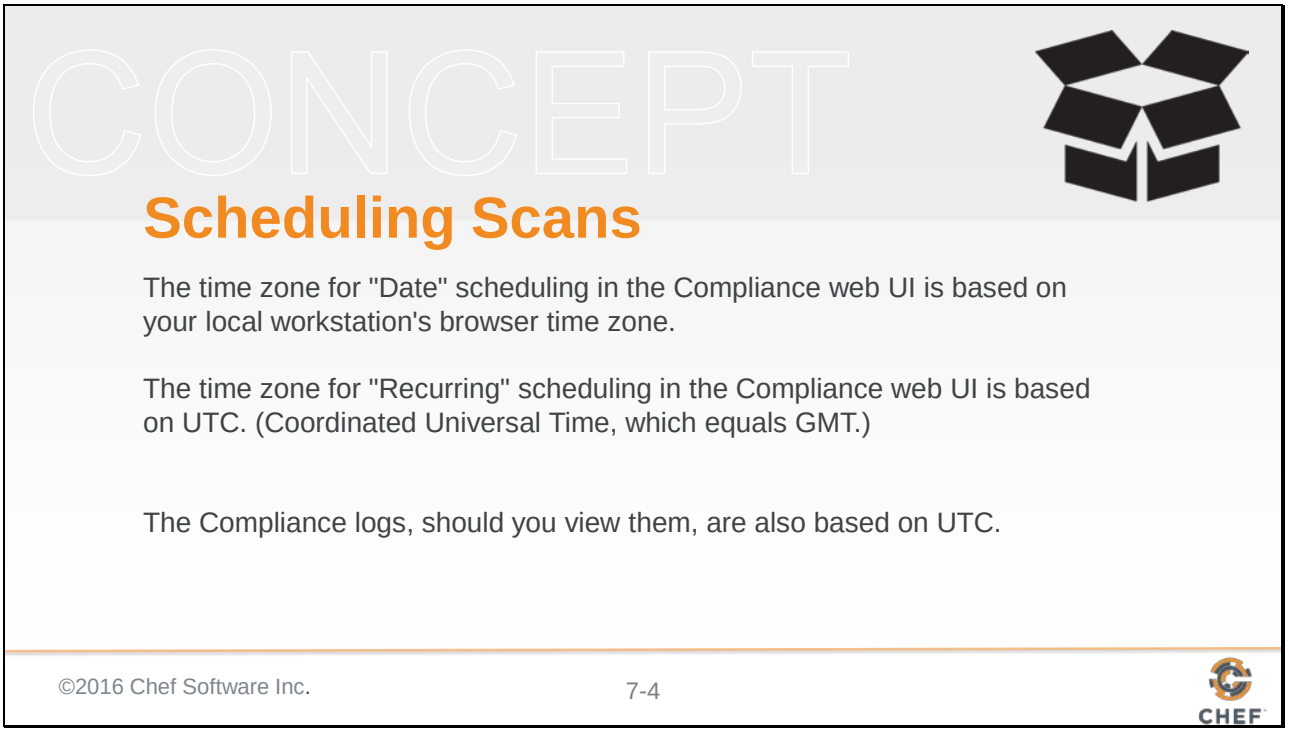
Scheduling Scans

You can schedule scans to run at a later time.

Running a scheduled compliance scan on your infrastructure, say every night, could give you up-to-date compliance information on a daily basis.

In this way, any changes to your infrastructure that may have put some nodes out of compliance can be routinely identified.

Slide 4



This time distinction is important when scheduling scans or if viewing the compliance logs.

Instructor Note: As of this writing this slide is correct but it could be subject to change. Also, here is the `tail` command in case you want to demonstrate the logs as you set a scheduled scan: `sudo tail -f /var/log/chef-compliance/core/current`/

Slide 5

EXERCISE

GE: Scheduling Scans

Objective:

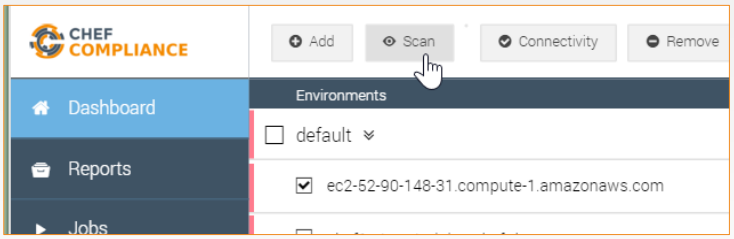
- ❑ Schedule a scan.
- ❑ View the scan output.

CHEF

Slide 6

GE: Scheduling Scans

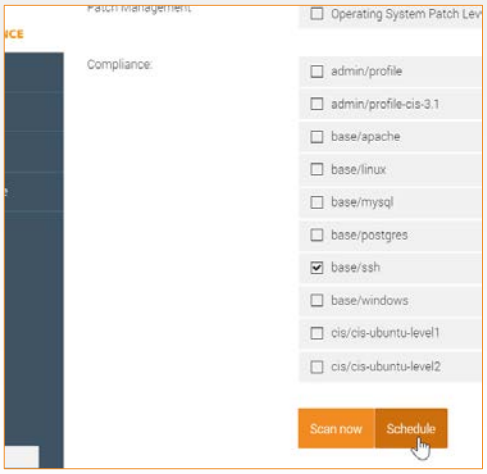
1. Open the Compliance web UI's Dashboard page.
2. Select one of your target nodes.
3. Click **Scan**.



Slide 7

GE: Scheduling Scans

4. Deselect all profiles.
5. Select the **base/ssh** profile for a Linux node or the **base/windows** profile for a Windows node
6. Click the **Schedule** button.



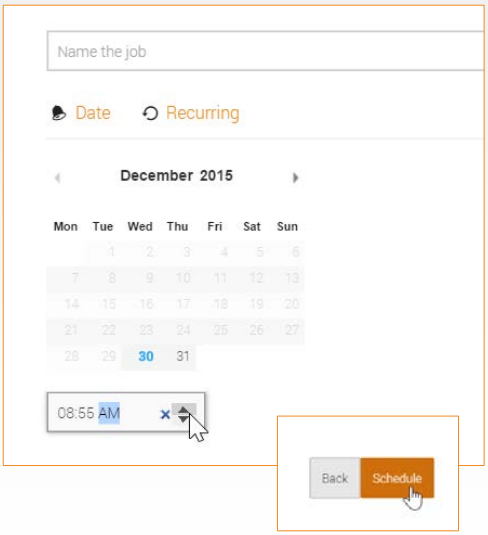
Slide 8

GE: Scheduling Scans

7. Type a name for this scan in the **Name the job** field.

8. Click the up arrow and set the scheduled time to 5 minutes from now and then click **Schedule**.

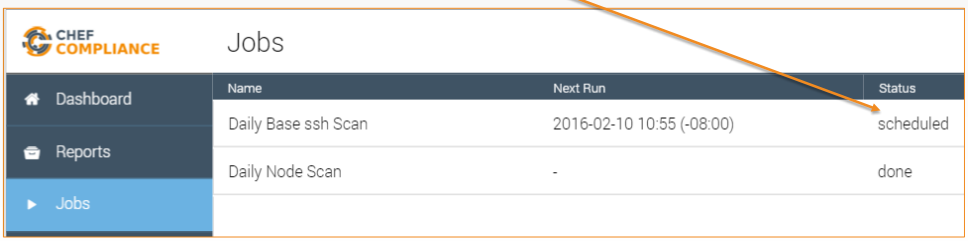
Note: The up/down arrows are also used to set the AM and PM value.



Slide 9

GE: Scheduling Scans

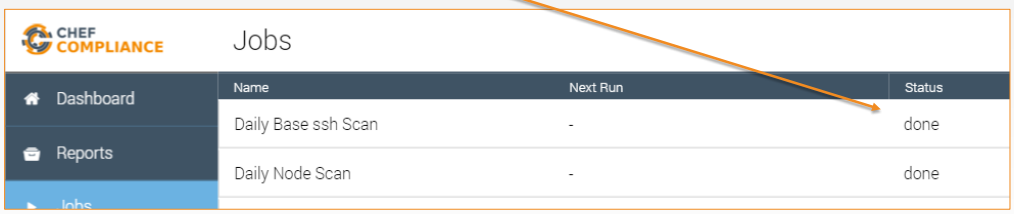
Click the **Jobs** tab and you should see your scheduled scan's details such as its Next Run time and its "scheduled" status.



Slide 10

GE: Scheduling Scans

When 5 minutes have elapsed, refresh your browser and you should see that your Job's status is now "done".

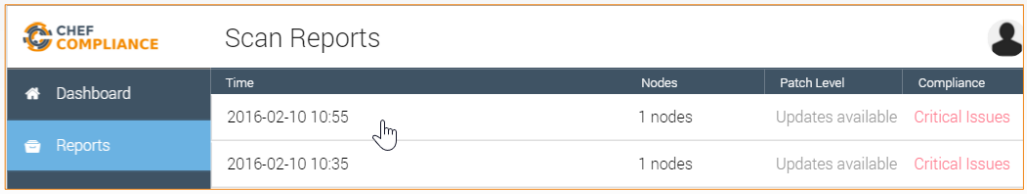


Note: If you like you could click that job to see its original details that you set when you scheduled it.

Slide 11

GE: Scheduling Scans

Click the **Reports** tab and then click the report from your scheduled scan.

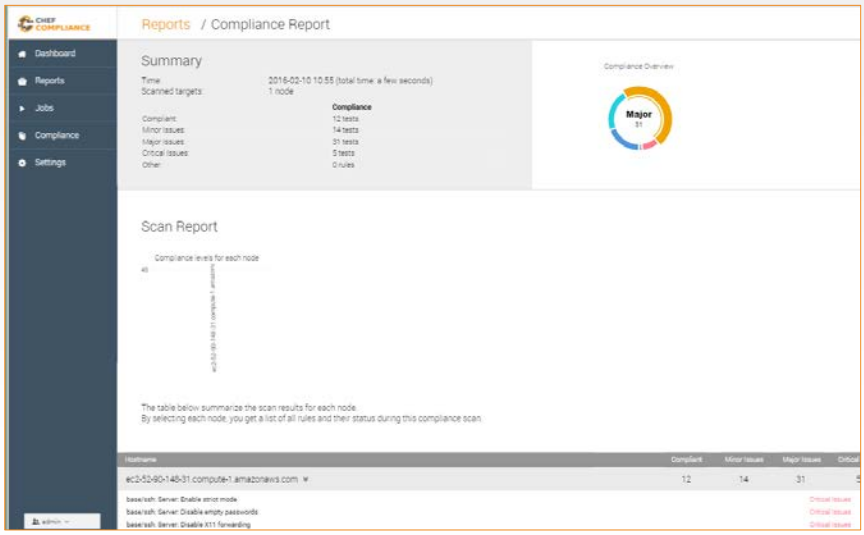


It may take a minute or two after the scan is run for the report to display.

Slide 12

GE: Scheduling Scans

At this point you should be able to view the report from your scheduled scan.

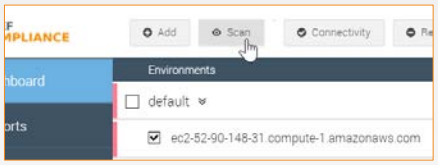


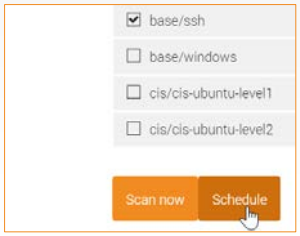
Slide 13

Demonstration: Recurring Scans

You can schedule recurring scans as well.

To do so, set up a scheduled scan like you just did...



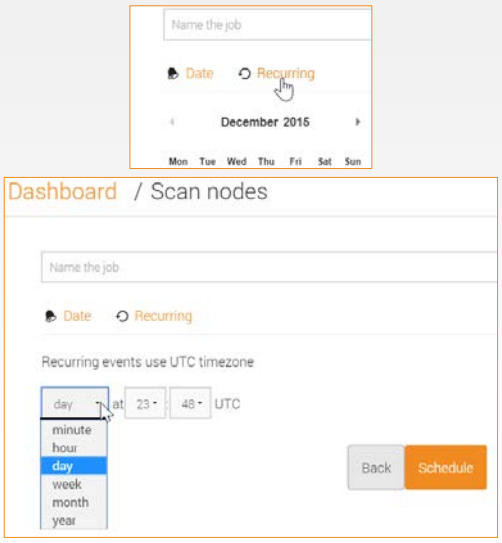


Slide 14

Demonstration: Recurring Scans

...but when you get to the page with the calendar (Scan nodes page), click the **Recurring** link and you'll be able to set the recurrence.

In this example the user is scheduling a scan to run every day at 23:48 UTC.



Slide 15

Demonstration: Scheduled Scan Logs

While not necessary, if you ever want to view the Compliance log files while you are scheduling or running a scan, keep in mind that log files use UTC time instead of your local browser time.

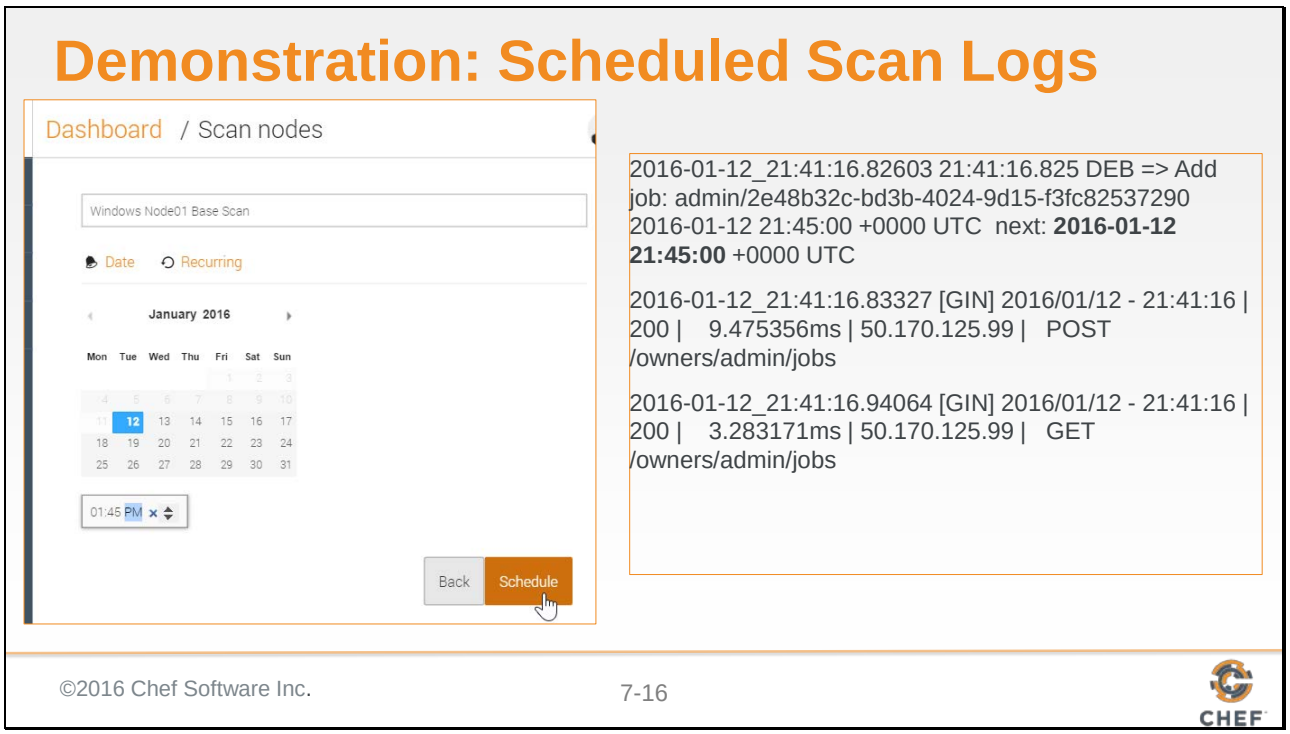
To tail those logs, from your Compliance Server you can run:

`sudo tail -f /var/log/chef-compliance/core/current`

Instructor Note: As this slide indicates, tailing the log files is not required but can be of interest to some users. In one scenario where it came in handy, a browser session to the Compliance Server got stale and would not schedule a scan. After tailing the log files and realizing the scan was not even being scheduled, the user refreshed the Compliance Server web UI, logged in again, and the scan would properly schedule and run.

Instructor Note: If you find yourself ahead of schedule, you can spend some time demonstration the output of this log file as you perform a number of procedures via the Web UI.

Slide 16



The image on the left shows the user setting the next scheduled scan for 1:45 P.M. local browser time.

The image on the right shows the output of `sudo tail -f /var/log/chef-compliance/core/current` taken at the very same time that the scan was scheduled. Notice the log file timestamps are in UTC and the next schedule scan is set for UTC: **2016-01-12 21:45:00** +0000 UTC instead of the local browser time.

Slide 17

CONCEPT

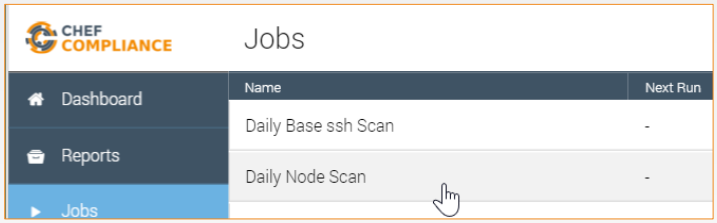
Deleting Old Jobs

The list of old scheduled jobs can grow so you can delete them if you no longer need them.

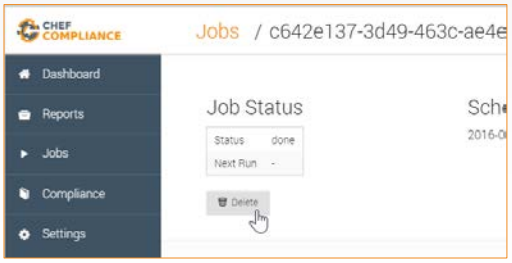
Slide 18

Demonstration: Deleting Jobs

To delete an old job, from the Jobs tab, click a job...



...and from the resulting page, click the **Delete** button.



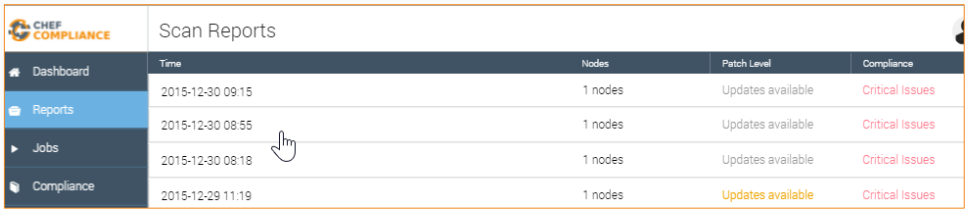
Slide 19

CONCEPT

Compliance Reports

The results of all of your scans are available via the Reports tab.

As of this writing, JSON-based exporting of reports is supported.



CHEF

Slide 20

CONCEPT

Compliance Reports

You can also export a compliance report as a PDF.

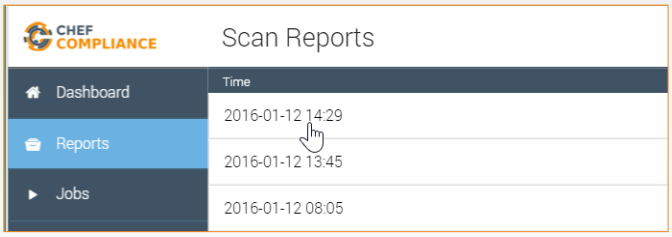
In the near future, Chef Compliance will also support exporting reports to Excel and sending to an email recipient.

Instructor Note: This section is more of a placeholder for exporting Compliance reports after the email and Excel methods are ready for release.

Slide 21

GE: Exporting Reports

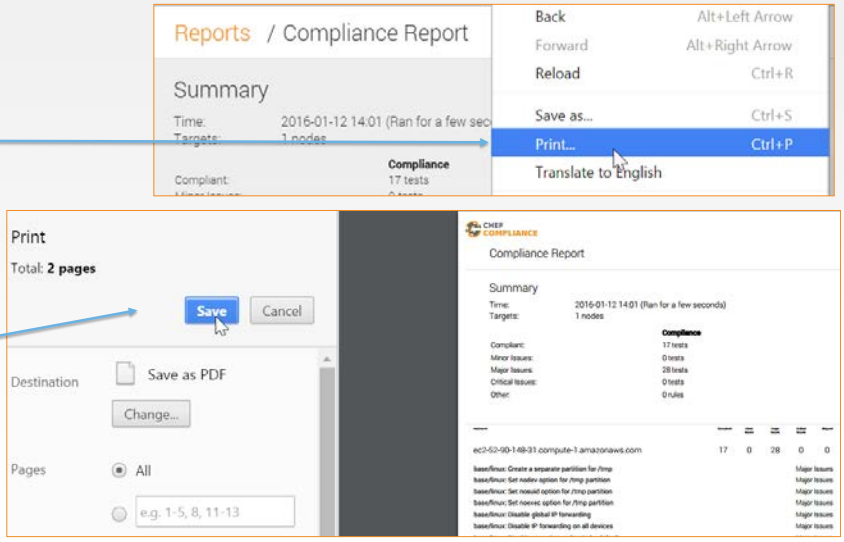
1. From the left column of the web UI, click **Reports** and then click any report that may exist.



Slide 22

GE: Exporting Reports

2. Right-click on the report page and then click **Print**.
3. Ensure your print option is set to PDF and save and then click **Save**, thus saving to you local laptop.

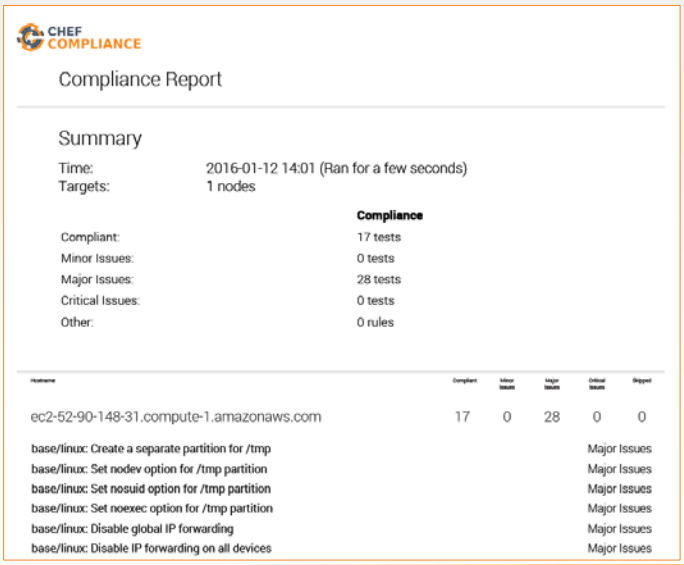


This example is from a Windows user. The exporting reports procedure is similar on a Mac.

Slide 23

GE: Exporting Reports

Your saved report should look similar to this example with the text properly aligned and formatted.



CHEF COMPLIANCE

Compliance Report

Summary

Time: 2016-01-12 14:01 (Ran for a few seconds)
Targets: 1 nodes

	Compliance
Compliant:	17 tests
Minor Issues:	0 tests
Major Issues:	28 tests
Critical Issues:	0 tests
Other:	0 rules

ec2-52-90-148-31.compute-1.amazonaws.com	17	0	28	0	0

base/linux: Create a separate partition for /tmp	Major Issues
base/linux: Set nodev option for /tmp partition	Major Issues
base/linux: Set nosuid option for /tmp partition	Major Issues
base/linux: Set noexec option for /tmp partition	Major Issues
base/linux: Disable global IP forwarding	Major Issues
base/linux: Disable IP forwarding on all devices	Major Issues

CHEF

Additional methods for exporting reports will be available in the near future.

Slide 24

Review Questions

1. When scheduling one time scans, which time zone does the web UI use?
2. When scheduling recurring scans, which time zone does the web UI use? answer
3. In Chef Compliance, what are "Jobs"?

Instructor Note Answers:
1. Your local workstation's browser time zone.
2. The time zone for "Recurring" scheduling in the Compliance web UI is based on UTC. (Coordinated Universal Time, which equals GMT.)
3. They are a list of scheduled scans, past or present.

Slide 25

Review Questions

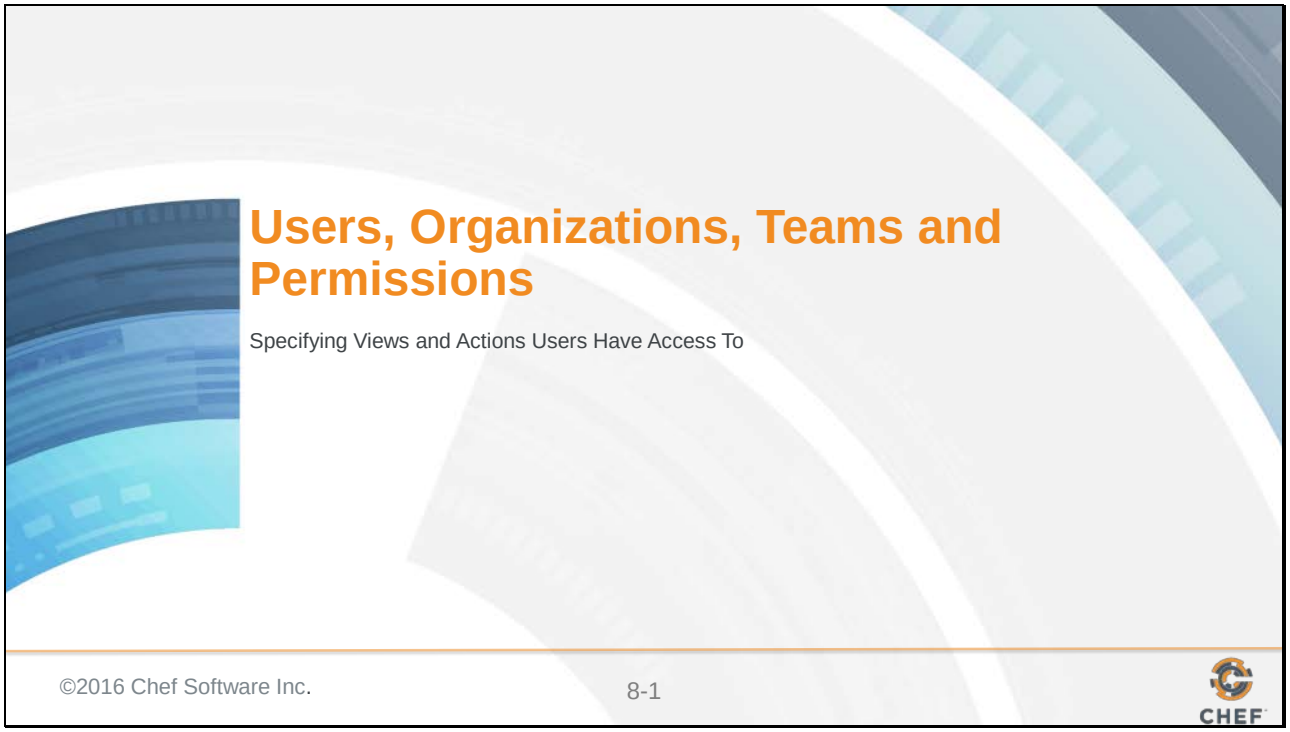
4. Where can you view the results of a scheduled scan?
   a. On the Jobs page.
   b. On the Reports page.
   c. On the Dashboard.
   d. On the Compliance page.

5. What methods are available for exporting reports?

Instructor Note Answers:
4. b. On the Reports page.
5. Print/Save to PDF...(more to come)

Slide 26

8: Users, Organizations, Teams and Permissions



Instructor Note: While managing users and organizations via a web UI may seem like a trivial topic, in Chef Compliance there is some complexity in how nodes are shared via Organizations. So this hands-on module allows the students to manage users, organizations, and teams so they can see the resulting behavior first hand.

Instructor Note: LDAP integration is scheduled for Q2 2016. As of this writing, users have two options for managing manage users, organizations, and teams--The web UI or via API: https://docs.chef.io/release/compliance_1-0/api_compliance.html#users

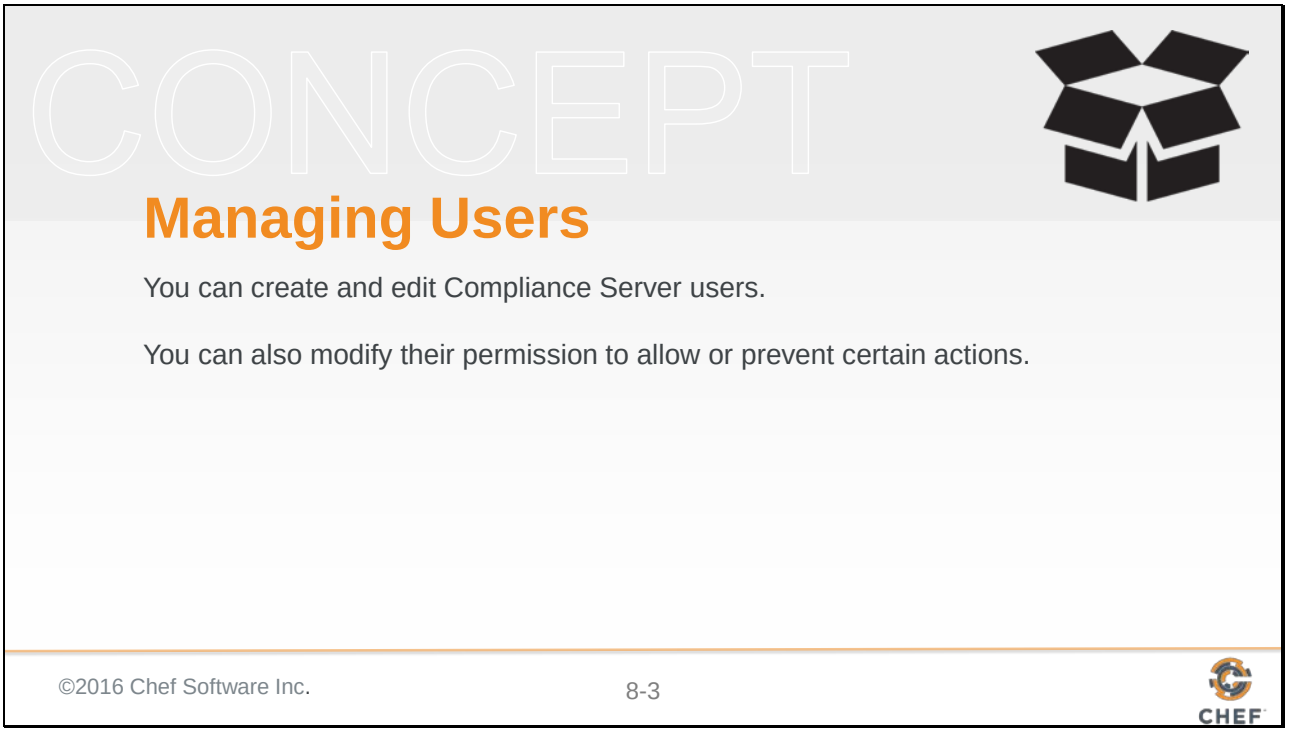
Slide 2

Objectives

After completing this module, you should be able to:

- Create users.
- Set user permissions.
- Create and apply organizations.
- Create Teams.
- Set Team permissions.

Slide 3



Permissions are used to control what users and organizations can do in the context of Chef Compliance. You can effectively enable or disable permissions to scan systems, update packages, or manage various aspects of Chef Compliance including users and organizations.

Slide 4

EXERCISE

Group Lab: Creating Users

Let's learn by doing. We'll stop along the way to explain details.

Objectives:

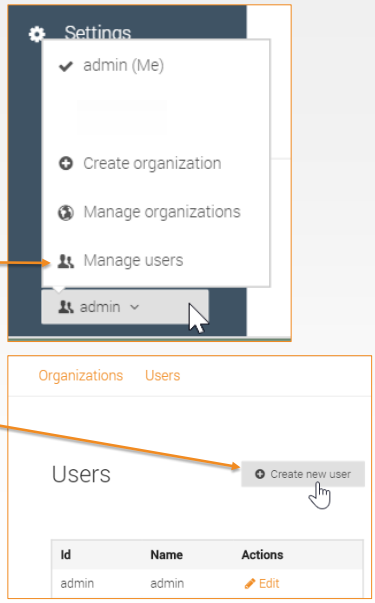
- ❑ Create a new Compliance User.
- ❑ Modify a User's Permissions.

CHEF

Slide 5

GL: Creating Users

1. From the bottom-left, click the **menu** button and then click **Manage user**s.
2. Click **Manage users**.
3. From the resulting screen, click **Create new user**.



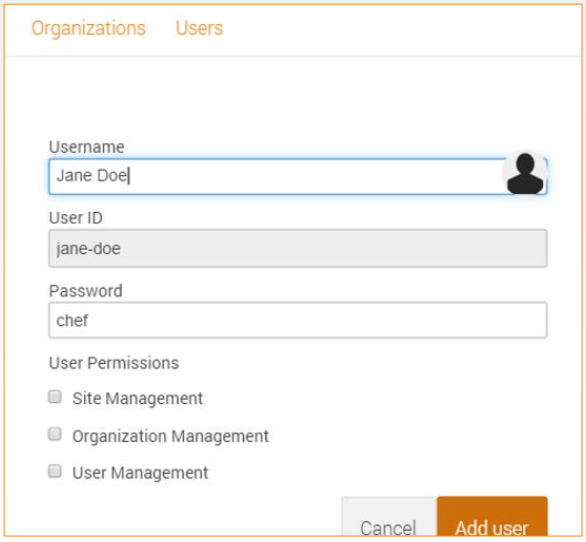
Slide 6

Discussion: Creating Users

The Users/Create screen displays.

A user can have the following permissions:

- **Site Management** users can do everything.
- **Organization Management** users can create, edit or delete Organizations.
- **User Management** users can create, edit or delete users.



The Site Management permission can be thought of as Administrator-level permissions. It is like a superset of the Organization Management and the User Management permissions.

We will discuss Organizations and their purpose later in this module.

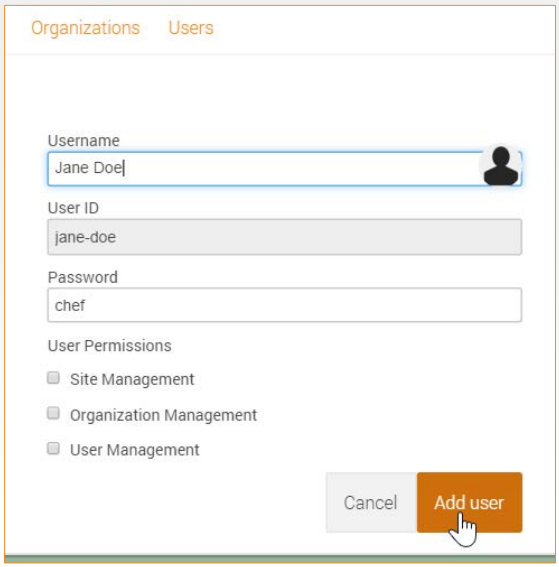
Slide 7

GL: Creating Users

4. From the current Users/Create screen, type a **user name** of your choice and a password you'll remember, like **chef**.

Note: Leave the new user's permissions unchecked for now.

5. Click the **Add user** button.



Slide 8

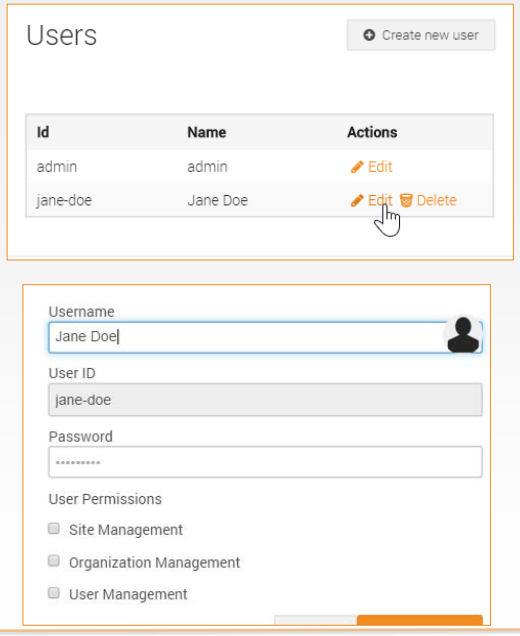
Discussion: Creating Users

Notice from the current **Users** screen you could edit or delete users.

The bottom image shows the permissions that you can edit for a user.

You could also edit the Username and/or Password but not the user ID.

Reminder: Leave the new user's permissions unchecked for now.



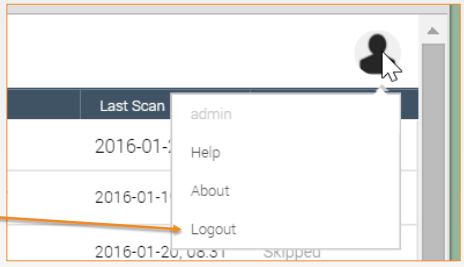
Notice that you can't delete the original admin user, however, since you always need at least an admin user.

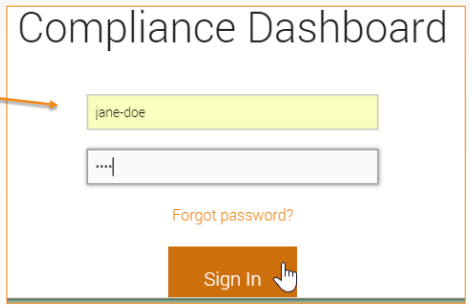
We will use this new user in the up coming lab exercises.

Slide 9

GL: Managing Users

6. From the top-right of the Compliance web UI, click the **User** icon and log out.
7. From the resulting login screen, log in with your **new user ID**.





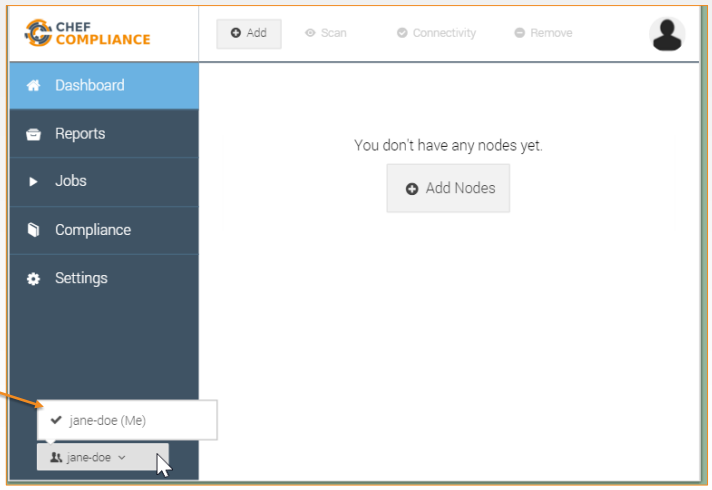
Slide 10

GL: Managing Users

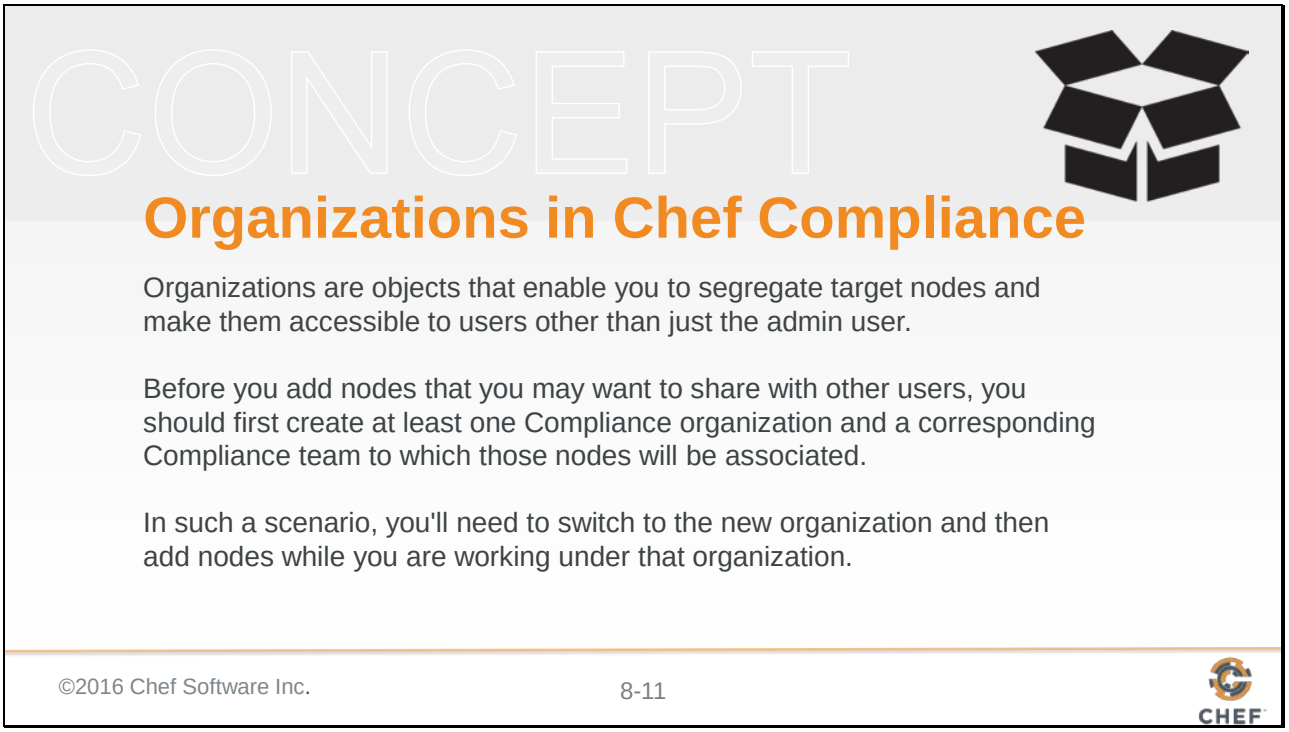
Notice that you cannot access any nodes that were added previously.

Also notice that your current permissions don't allow you to manage users or organizations.

8. Log out and then log back in as the **admin** user.



Slide 11



As of this writing, the only way to share access to nodes between different user accounts via a combination of an Organization object and its Team object.

Slide 12

EXERCISE

Group Lab: Using Organizations and Teams

Let's keep learning by doing. We'll stop along the way to explain details again.

Objectives:

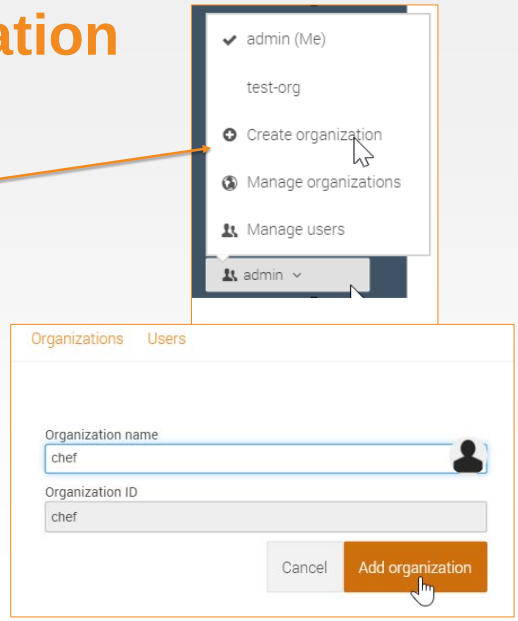
- ❑ Create an Organization.
- ❑ Create a Team.
- ❑ Add a Member to a Team.
- ❑ Add a Node to an Organization.
- ❑ Test a Team Member's Access to a Node.

CHEF

Slide 13

GL: Creating an Organization

1. From the bottom-left of the Dashboard, click **admin > Create Organization**.
2. Type a new organization name (**chef**) in the Organization name field and then click the **Add organization** button.

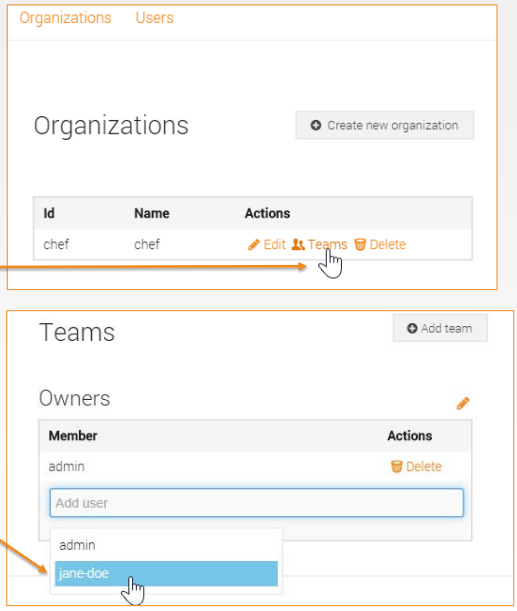


Slide 14

GL: Creating an Organization

3. From the resulting screen, click the **Teams** link to add team member to your new organization.
4. Click the **Add user** field and then select your new user name to add it to this team.

Notice that the original admin user us already part of this and any new organization teams.

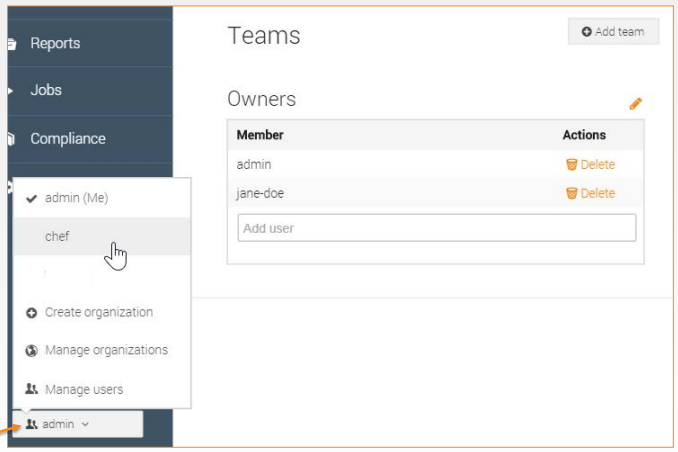


Slide 15

GL: Using Your New Organization

You should now see that your new user name is part of the chef organization's team.

In the next step we'll switch to the new organization so we can add a node to it.



5. From the bottom-left of the web UI, click **admin** and then click your new organization (**chef**) to switch to it.

Slide 16

GL: Using Your New Organization

6. Click the Dashboard link and notice that you can no longer see any nodes.

Question: Where have your previous nodes gone?

7. Click the **Add** button to add a node to your organization.

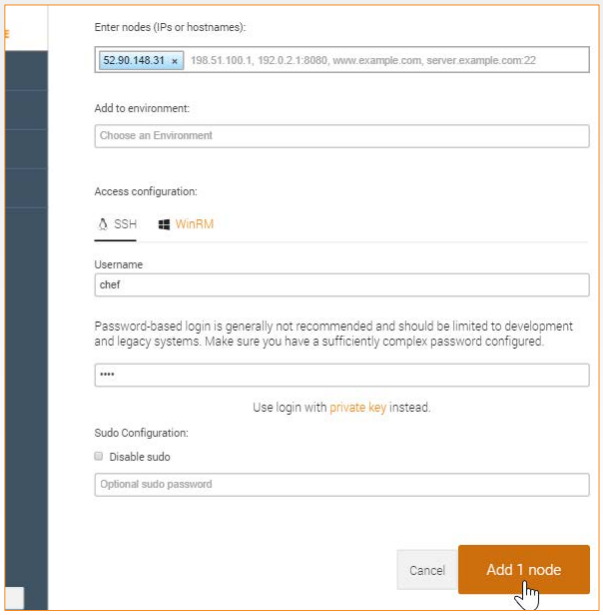
Instructor Note: Answer--The nodes have not gone anywhere. They are only accessible under the admin user's default organization, not under this new `chef` organization. In the next steps we'll add a node to our new organization.

Slide 17

GL: Using Your New Organization

7. Add one of the **target nodes** you were assigned at the beginning of this course and then click the **Add 1 node** button.

Note: Use the same password authentication method as done previously in the course.

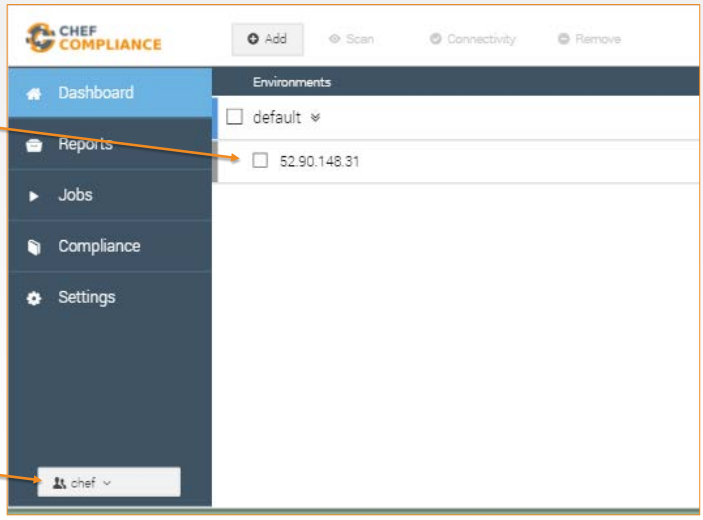


Slide 18

GL: Using Your New Organization

You should now see the node you added to your new **chef** organization.

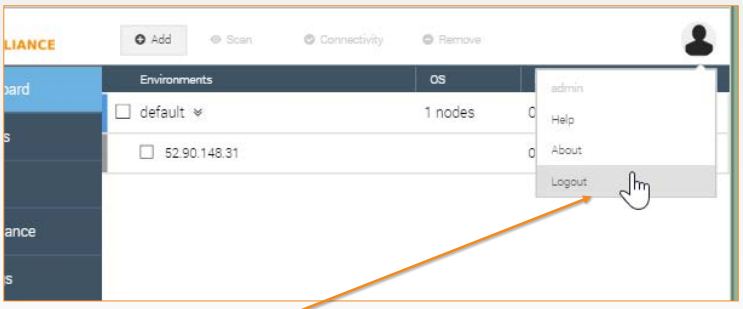
You can also determine which organization you are switched to by looking at the bottom-left of the Chef Compliance Dashboard.



Slide 19

GL: Using Your New Organization

As you may recall, the last time you tried to view nodes while logged in to the Compliance Dashboard with your new user name, you could not see any nodes.



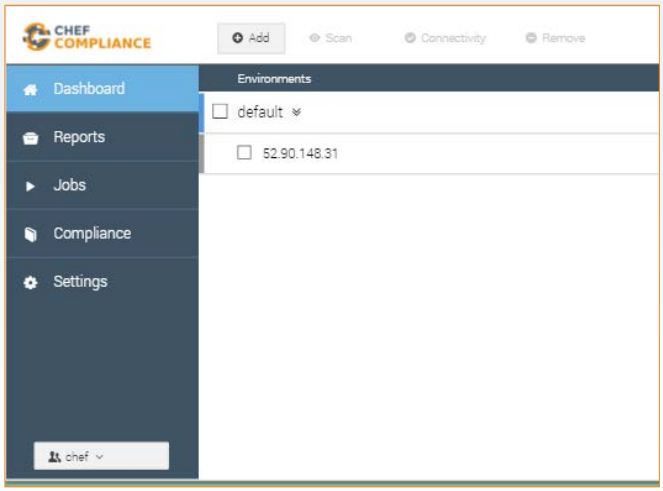
8. Log out of the Compliance Dashboard and log back in with your **new user name**.

Slide 20

GL: Using Your New Organization

While logged in with your new user name, you should now be able to access the node that was created under your new organization.

Question: Do you remember why this is possible?



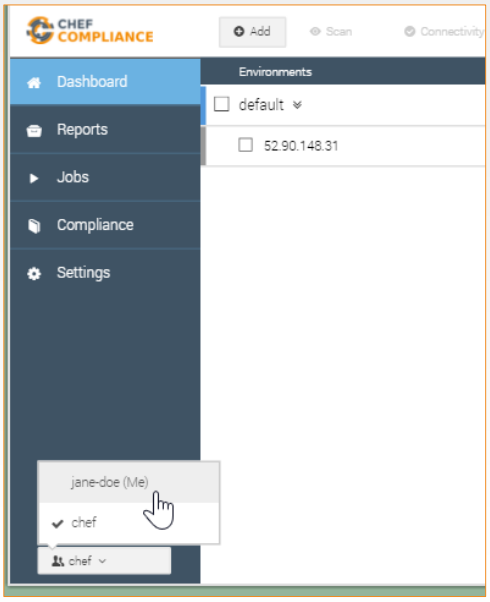
Instructor Note: Answer--Because your new user name is part of the Team object that is resides under the new (chef) organization.

Slide 21

GL: Using Your New Organization

9. From the bottom-left of the Compliance Dashboard, switch back and forth between you new organization and your user name's default organization.

You should only be able to access the node that was created under your new organization.



Slide 22

CONCEPT

Managing Team Permissions

Teams have their own set of permissions.

Team permissions are completely separate and independent of user permissions.

CHEF

Slide 23

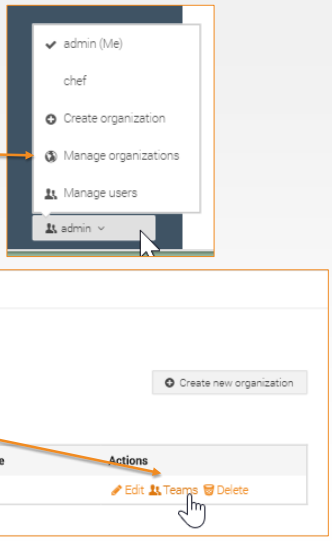
GL: Managing Team Permissions

1. Log out of the Compliance Web UI and log back in as **admin**.
2. If you like you can switch back to the admin organization although it really doesn't matter for the next task.

Slide 24

GL: Managing Team Permissions

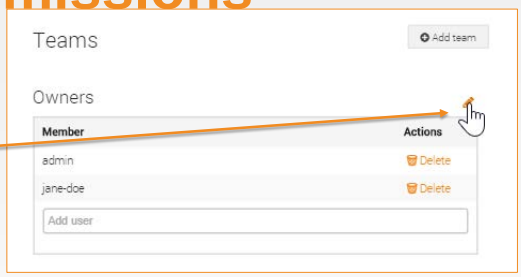
3. Click **Manage Organizations** from the menu.
4. From the resulting Organizations screen, click the **chef** organization's **Teams** link.



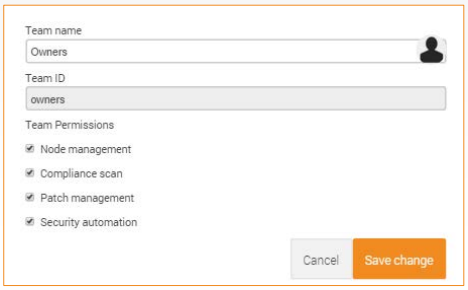
Slide 25

GL: Managing Team Permissions

5. From the resulting Teams screen, click the **pencil** icon.



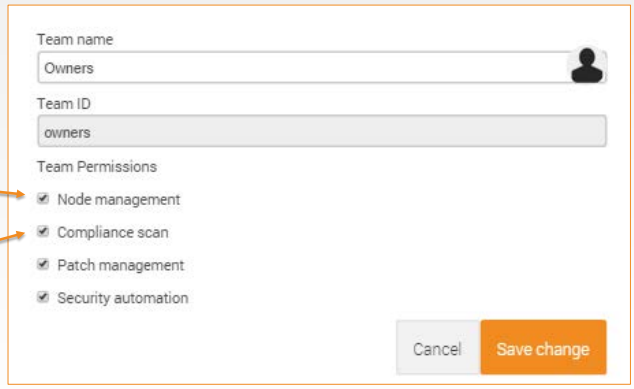
The team permissions screen should display.



Slide 26

Discussion: Managing Team Permissions

- The **Node Management** permissions, when checked, enable the team members to add, edit or delete nodes from the dashboard.
- The **Compliance Scans** permissions, when checked, enable the team members to execute scans.

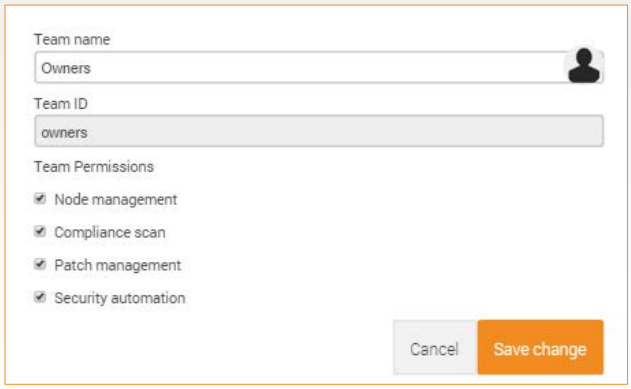


When you create a new team, all these permissions are enabled (checked) so if you want to limit team members' permissions listed here, you will need to disable them.

Slide 27

Discussion: Managing Team Permissions

Note: As of this writing, the Patch Management and Security Automation permissions are not fully functional.



Instructor Note: This is what a Compliance SME wrote about the last two permissions listed above: "patch management - execute patch execution (deactivated by default). Security automation - use cookbooks for remediation (not available yet)

Slide 28

Review Questions

1. How can you share nodes between different user names?

2. Of the following permissions found on the Users/Create screen, which has the highest levels of permissions when enabled?

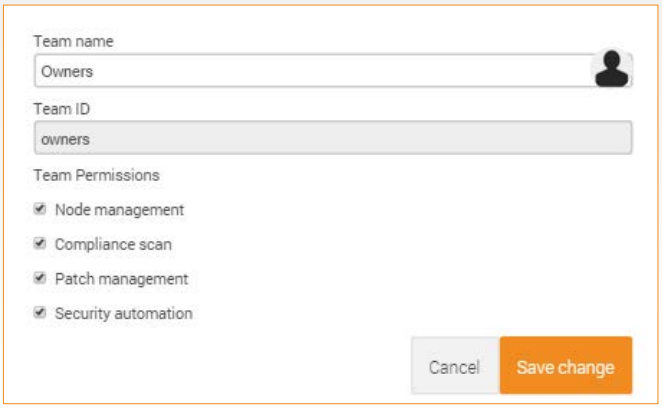
   - ❑ Organization Management.
   - ❑ User Management.
   - ❑ Site Management.

Instructor Note Answers:
1. By using a combination of an Organization object and its Team object. Teams belong to Organizations so members of teams can then view or manage nodes that have been added to their team's organization.
2. Site Management.
A user can have the following permissions:
- ❑ Organization Management users can create, edit or delete Organizations.
- ❑ User Management users can create, edit or delete users.
- ❑ Site Management users can do everything.

Slide 29

Review Questions

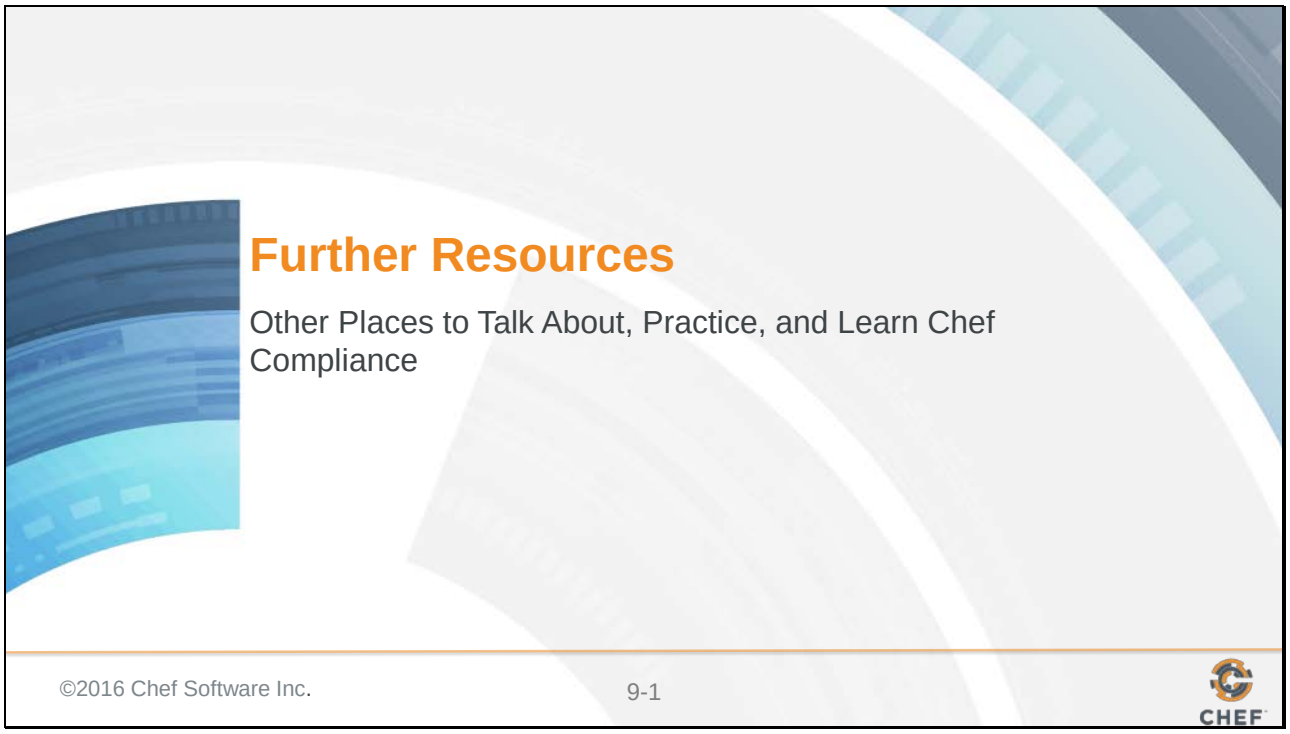
3. Regarding team permissions, what can the **Node management** permission allow?



Instructor Note Answers:
3. The Node Management permissions, when checked, enable the team members to add, edit, or delete nodes from the dashboard.

Slide 30

9: Further Resources



Slide 2

Going Forward

There are many Chef resources available to you outside this class. During this module we will talk about just a few of those resources.

But…remember what we said at the beginning of this class:

The best way to learn Chef is to use Chef

Slide 3

docs.chef.io

Docs are available to you, 24 hours a day, 7 days a week.

Any question you have, you probably will find the answer for on our Docs site.

Slide 4

docs.chef.io

Main Chef Compliance doc link:

https://docs.chef.io/compliance.html

Install Chef Compliance Server:

https://docs.chef.io/install_compliance.html

Slide 5

docs.chef.io

Upgrade Compliance Server doc link:

https://docs.chef.io/upgrade_compliance.html

Slide 6

docs.chef.io

Compliance DSL (domain-specific language):

https://docs.chef.io/dsl_compliance.html

That link contains descriptions and examples of the InSpec syntax used to define controls.

Slide 7

docs.chef.io

Compliance API:

https://docs.chef.io/release/compliance_1-0/api_compliance.html

The Compliance API is a REST-based API that is designed to be easy and predictable.

Slide 8

ChefConf is a gathering of hundreds of Chef community members. We get together to learn about the latest and greatest in the industry (both the hows and the whys), as well as exchange ideas, brainstorm solutions, and give hugs, which has become the calling card of the DevOps community, and the Chef community in particular.

ChefConf 2016 will be held in Austin, Texas during July.

Slide 9

Appendix Z: Course-wide Instructor Notes

1. Training Lab System Setup

Note: You and the students will need to use two different AMIs for this Compliance course. One for Linux and one for Windows. The following steps cover Linux. The Windows AMI steps will follow the Linux steps. Both target AMIs must use inspec 0.14.7. or higher. inspec 0.14.7. is installed on both AMIs.

Linux AMI:

1. Open the AWS site from here: https://aws.amazon.com/
   - Login Credentials for Chef instructors: training-aws@chef.io
   - Password: Contact Chef Training Services if you don't know it or how to obtain it. training@chef.io
   - Partner credentials should be provided by Chef directly to partners.
2. Click the first link in column **EC2 *Virtual Servers in the Cloud***
3. From the navigation pane on the left, select **Images/AMIs**. The "Step 1" page displays with a list of available AMIs.
4. Select **Compliance - CentOS 6.7 - 1.0.6 (ami-740b321e)** from the list of options.
5. Click **Launch**. The "Step 2" page displays.
6. Select the first **Micro Instance** from the list provided and click **Next: Configure Instance Details** at the bottom of the screen. The "Step 3" page displays.
7. Enter the **Number of Instances**.

   Note: You will need 2 instances for each student enrolled in the class and 2 for yourself.

8. Click **Next: Add Storage** at the bottom of the page. The "Step 4" page displays. [Don't change anything on this page].
9. Click **Next: Tag Instance** at the bottom of the page. The "Step 5" page displays.
10. Enter a **Value**.

    Note: A recommended naming convention for the instances: [TRAINER'S INITIALS] - [CLASS NAME] - [CLASS DATE]

11. Click **Next: Configure Security Group**. The "Step 6" page displays.
12. Click the **Select an existing security group** radio button. A list of security groups displays.
13. Select **all-open**.
14. Click **Review and Launch** at the bottom of the screen. The "Step 7" page displays.
15. After you review the instances, click **Launch**. The "Select a key pair" window displays.
16. Confirm that this is set to **Choose an existing key pair** and click the acknowledgement check box.

17. Click **Launch Instances**. The "Launch Status" page displays.
18. Click **View Instances**. The instances list displays.
19. From here, copy all of the instances and create a gist file to share with the class.
20. Use goo.gl to shorten the URL to the gist file.

Note: The login credentials and password for the Linux AMIs used in class are chef/chef. You'll need to tell the students that at the appropriate time.

Windows AMI:

1. Open the AWS site from here: https://aws.amazon.com/
    - Login Credentials for Chef instructors: training-aws@chef.io
    - Password: Contact Chef Training Services if you don't know it or how to obtain it. training@chef.io
    - Partner credentials should be provided by Chef directly to partners.
2. Click the first link in column **EC2 *Virtual Servers in the Cloud***
3. From the navigation pane on the left, select **Images/AMIs**. The "Step 1" page displays with a list of available AMIs.
4. Select **Compliance - Windows 2012 - 1.0.2 – (ami-570c353d)** from the list of options.
5. Click **Launch**. The "Step 2" page displays.
6. Select the first **Micro Instance** from the list provided and click **Next: Configure Instance Details** at the bottom of the screen. The "Step 3" page displays.
7. Enter the **Number of Instances**.

    Note: You will need 1 instance for each student enrolled in the class and 1 for yourself.

8. Click **Next: Add Storage** at the bottom of the page. The "Step 4" page displays. [Don't change anything on this page].
9. Click **Next: Tag Instance** at the bottom of the page. The "Step 5" page displays.
10. Enter a **Value**.

    Note: A recommended naming convention for the instances: [TRAINER'S INITIALS] - [CLASS NAME] - [CLASS DATE]

11. Click **Next: Configure Security Group**. The "Step 6" page displays.
12. Click the **Select an existing security group** radio button. A list of security groups displays.
13. Select **all-open**.
14. Click **Review and Launch** at the bottom of the screen. The "Step 7" page displays.
15. After you review the instances, click **Launch**. The "Select a key pair" window displays.
16. Confirm that this is set to **Choose an existing key pair** and click the acknowledgement check box.
17. Click **Launch Instances**. The "Launch Status" page displays.
18. Click **View Instances**. The instances list displays.
19. From here, copy all of the instances and create a gist file to share with the class.

20. Use goo.gl to shorten the URL to the gist file.

Note: The login credentials and password for the Windows AMIs used in class are Administrator / Cod3Can! You'll need to tell the students that at the appropriate time.

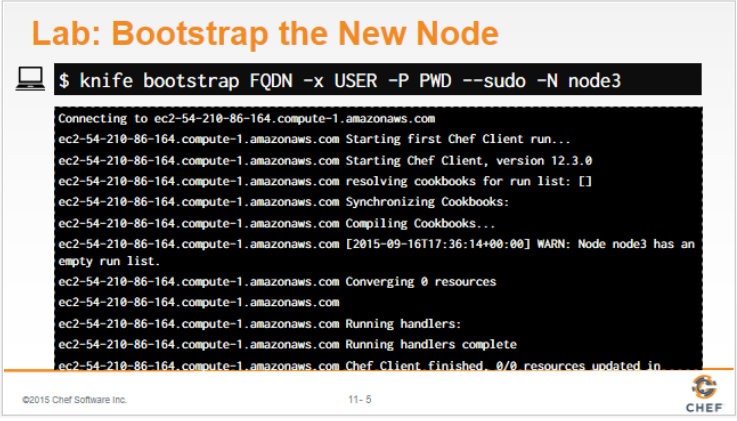
Note: Answers to quizzes are provided as instructor notes below each quiz slide in the instructor guide.

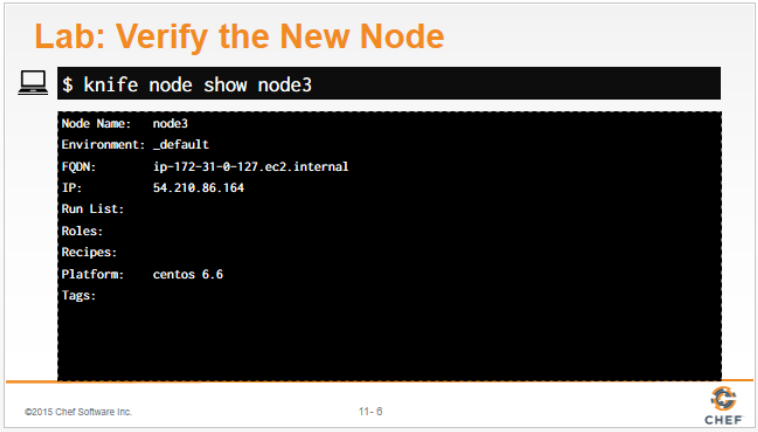
2. How to Use Lab Slides

Regarding the "Lab" exercises if present (not the Group Labs), you should encourage students to use the high-level hammer/wrench "Lab" slide steps first, and then resort to the subsequent detailed step slides if the students need the details to complete the lab. You can still use the subsequent detailed step slides as a vehicle to review each lab. For example:

This is a high-level hammer/wrench "Lab" instruction slide. Encourage students to complete the lab using this high level hammer/wrench "Lab" slide first.

If some students can't complete the lab based on the above slide, they are free to follow the subsequent detailed step slides, such as these:





You can also use the above detailed slides as a vehicle for reviewing the labs.