



CHEFTM

Chef Training Services

Chef Compliance

Participant Guide

1: Introduction

Chef Compliance

Installation, Configuration, and Operation

Introduction

©2016 Chef Software Inc.

1-1

Course v1.0.0

 CHEF

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.

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.

Slide 4

CONCEPT



Chef Compliance Value Proposition

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

Regulatory compliance is a fact of life for every enterprise.

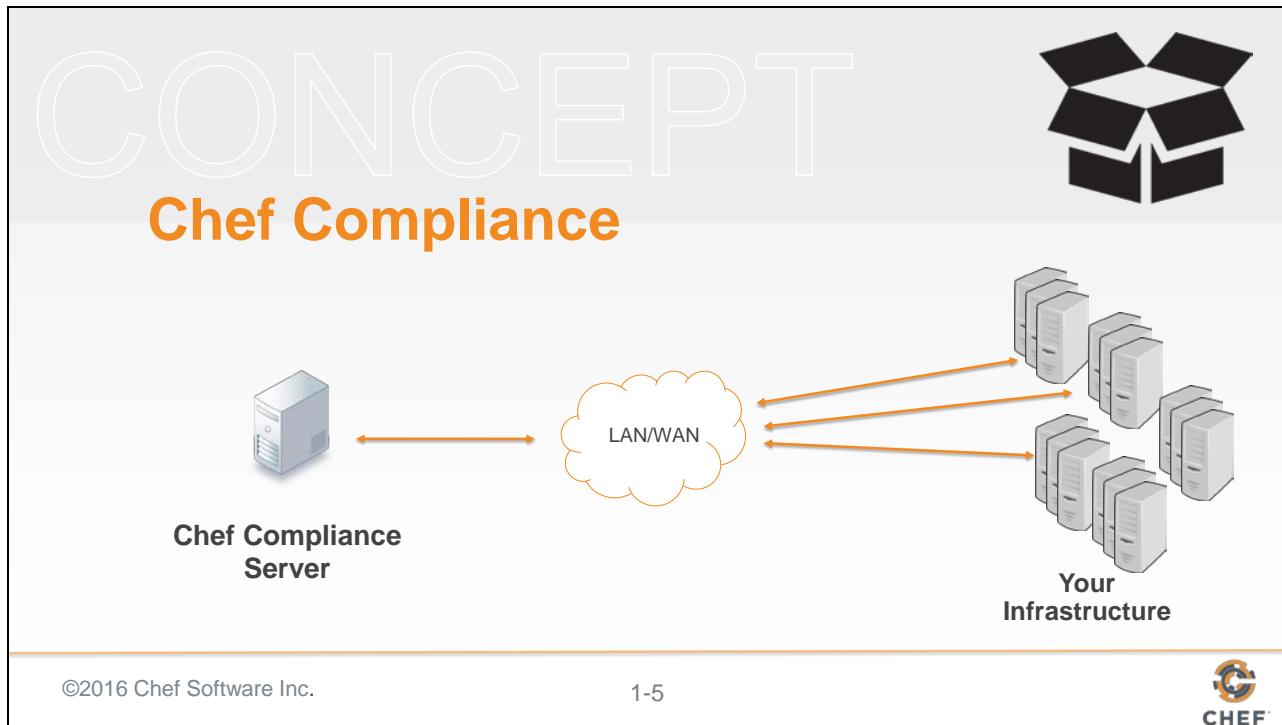
With Chef Compliance you can scan for risks and compliance issues with easy-to-understand, customizable reports and visualization.

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

CONCEPT

Chef Compliance



Chef Compliance can run without any other Chef software installed.

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.

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.

Slide 8

Chef Compliance and InSpec

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.

```
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
```

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.

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
```

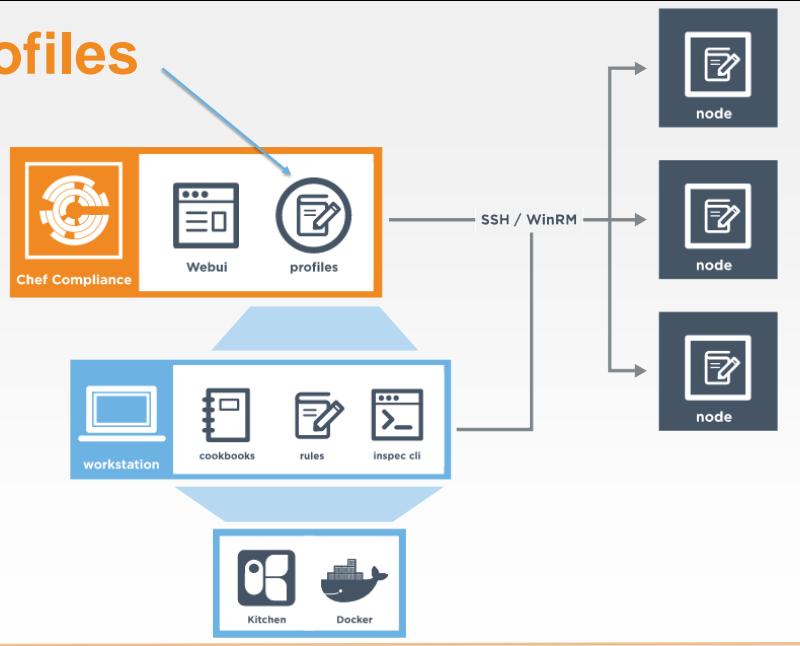
Slide 11

Compliance Profiles

Compliance profiles exist for many scenarios, such as those created by the Center for Internet Security (CIS)

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

You can also create your own custom Compliance profiles.



©2016 Chef Software Inc.

1-11



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.

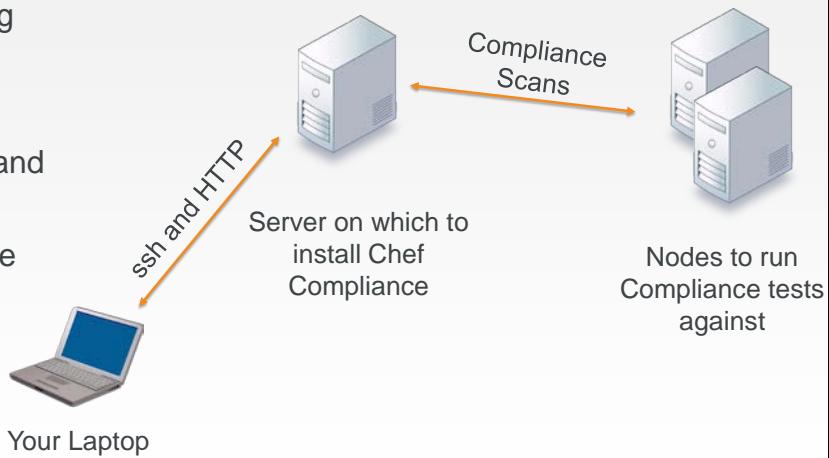
The screenshot shows the Chef Compliance web UI interface. On the left is a dark sidebar with navigation links: Dashboard, Reports, Jobs, Compliance, and Settings. The main area has a header "Reports / Compliance Report". Below the header are two sections: "Summary" and "Scan Report". The "Summary" section displays a timestamp (2016-02-05 12:59), the number of scanned targets (1 node), and a breakdown of issues: Compliant (11 tests), Minor Issues (4 tests), Major Issues (0 tests), Critical Issues (14 tests), and Other (0 rules). To the right of the summary is a "Compliance Overview" donut chart with a red segment labeled "Critical" and the number 14. The "Scan Report" section contains a bar chart titled "Compliance levels for each node". The y-axis ranges from 0 to 20. The x-axis lists nodes with their corresponding bars: 52 (blue bar at ~11), 91 (cyan bar at ~4), 251 (orange bar at ~1), and 243 (red bar at ~14).

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.



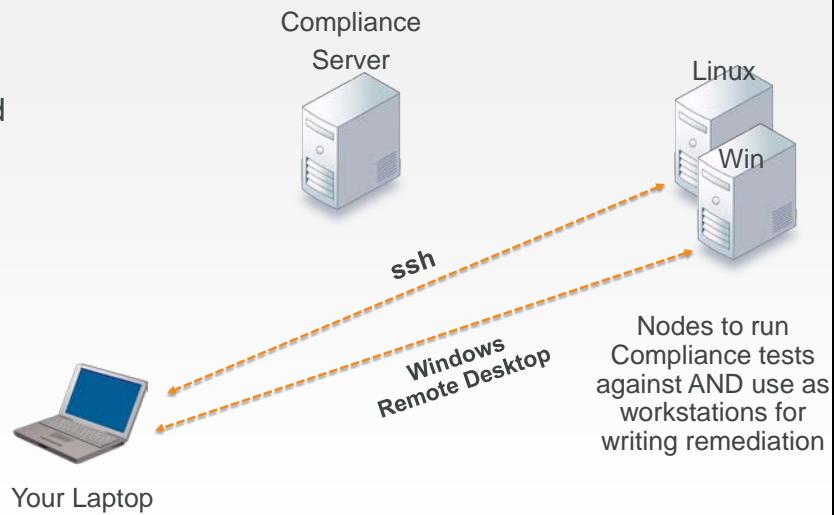
These are basic AWS AMIs that we use for Chef training. They have ChefDK installed on them although Chef does not actually need to be installed on these instances in order to run scans.

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

Logging in to the Compliance Server and Linux Node



```
$ ssh ADDRESS -l chef
```

w
o
r
k
s
t
a
t
i
o
n

©2016 Chef Software Inc.

1-15



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

Logging in to the Remote Windows Node

w
o
r
k
s
t
a
t
i
o
n

©2016 Chef Software Inc. 1-16 

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



©2016 Chef Software Inc.

2: Installing Chef Compliance

Installing Chef Compliance

Installing a Standalone Chef Compliance Server

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

CONCEPT



Compliance Installation Options

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.

In this course we will use the most common method--the Standalone method.

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

EXERCISE



Group Lab: Standalone Installation

Standalone Installation

Objective:

- ssh in to your Compliance Server node
- Download and install the chef-compliance-x package on your node
- Use chef-compliance-ctl to initially configure Chef Compliance server
- Run the Compliance Web UI

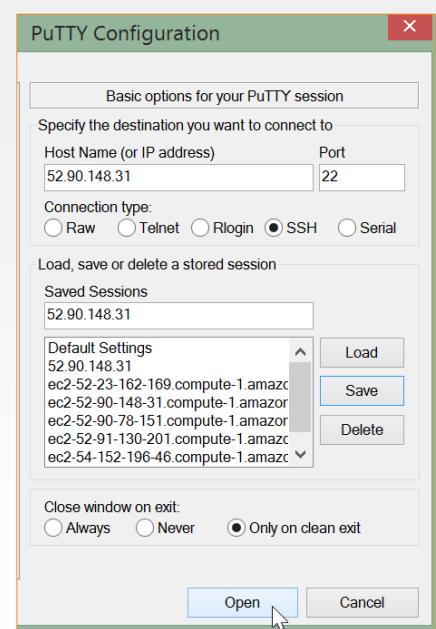
In this group lab you will perform the installation and initial configuration tasks listed in this slide.

Slide 5

GL: Standalone Installation

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

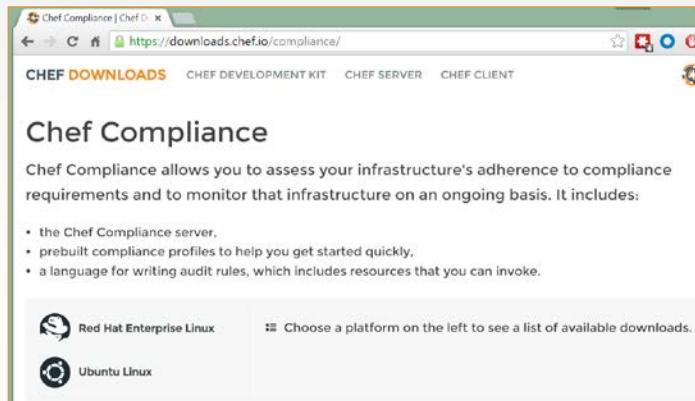
Here is an example of using Putty to do so.



Slide 6

GL: Standalone Installation

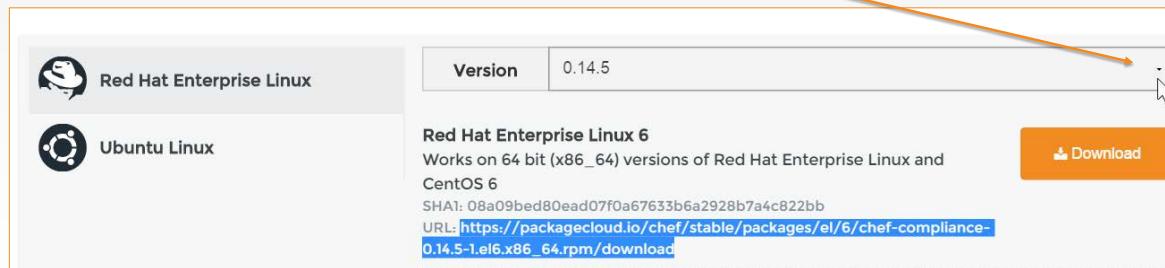
- From your local laptop, use a web browser to navigate to <https://downloads.chef.io/compliance/>



Slide 7

GL: Standalone Installation

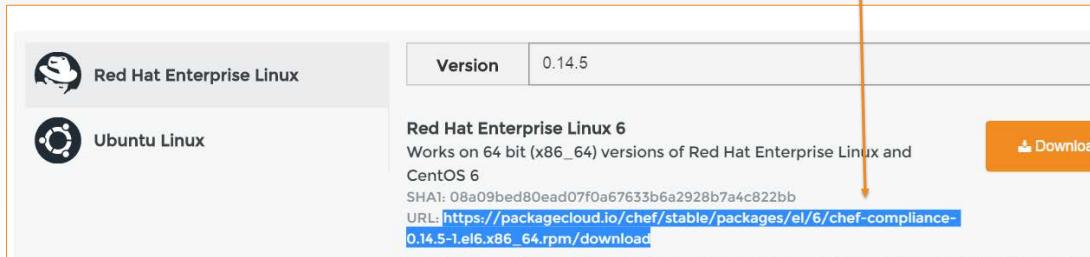
- From the resulting web page, click the **Red Hat Linux** link and then click the **Version** button.
- Select the version from the **Version** pull-down as indicated by the instructor.



Slide 8

GL: Standalone Installation

- Highlight and copy the URL for the Red Hat Enterprise Linux 6 version the instructor indicated. Do not click the Download button.



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.

Slide 9

GL: Install and Install Compliance Package



```
$ sudo rpm -Uvh  
https://packagecloud.io/chef/stable/packages/el/6/chef-compliance-  
0.9.5-2.el6.x86_64.rpm/download
```

```
Retrieving https://packagecloud.io/chef/stable/packages/el/6/chef-  
compliance-0.9.5-2.el6.x86_64.rpm/download
```

```
warning: /var/tmp/rpm-tmp.4Mgmgw: Header V4 DSA/SHA2 Signature,  
key ID 83ef826a: NOKEY
```

```
Preparing...  
#####
[ 200%]
```

```
You're about to install chef-compliance!
```

```
2:chef-compliance  
#####
[ 200%]
```

Execute **sudo rpm -Uvh https://packagecloud.io/chef/stable/packages/el/6/chef-compliance-0.9.5-2.el6.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.

Slide 10

GL: Initial Configuration of Compliance



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

```
...
- change mode from '' to '0644'
Recipe: sysctl::service
  * service[procps] action start
    - start service service[procps]

Running handlers:
Running handlers complete
Chef Client finished, 228/224 resources updated in 259.522030572
seconds
chef-compliance Reconfigured!
```

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 address you are using for your Compliance Server.

- For example
`https://54.173.238.187/#/setup`
- Click **Advanced** from the warning page.

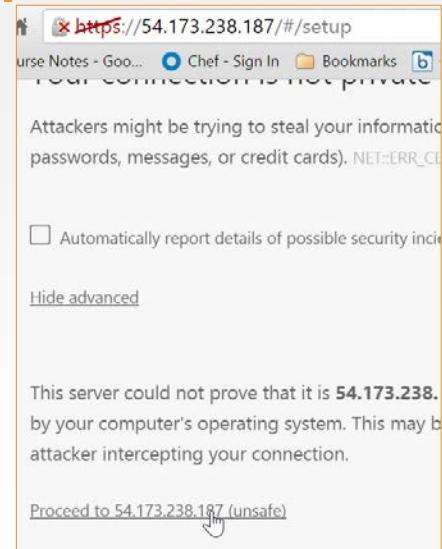


Slide 12

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.

prejudice to its other remedies under this Agreement or otherwise. The parties to this Agreement are independent contractors and this Agreement will not establish any relationship of partnership, joint venture, employment, franchise, or agency between the parties. Neither party will have the power to bind the other or incur obligations on the other's behalf without the other's prior written consent.

Back **Accept**

Create An Administrator Account

Name	<input type="text" value="admin"/>	*
Username	<input type="text" value="admin"/>	*
Password	<input type="password" value="....."/>	*

Back **Next**

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.

Review

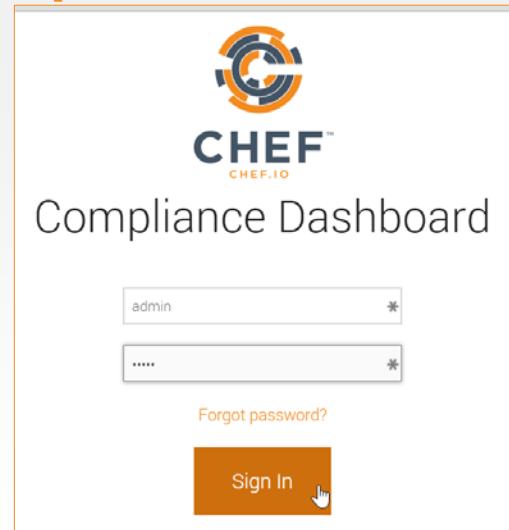
Username	admin
Name	admin
Accept EULA	Yes

Back  Configure

Slide 16

GL: Configure the Compliance Server

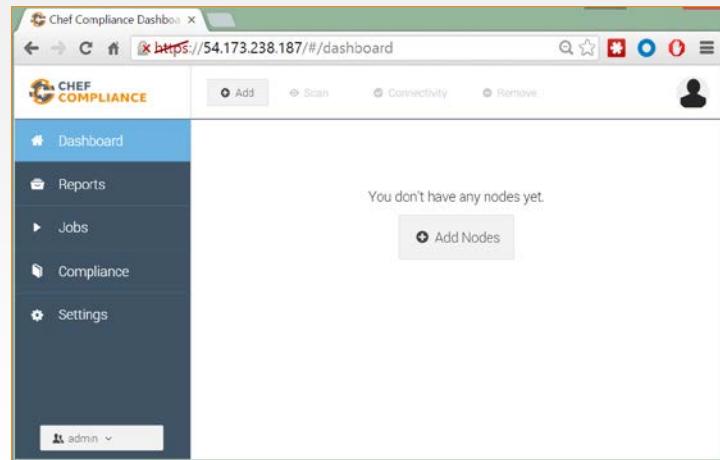
- On the resulting page sign in using admin/admin.



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

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>'
```

Slide 20



©2016 Chef Software Inc.

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

Running Scans, Remediation, and Testing on Linux Nodes

Configuring the Chef Compliance Server to Run Scans and Writing Remediation Recipes

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:

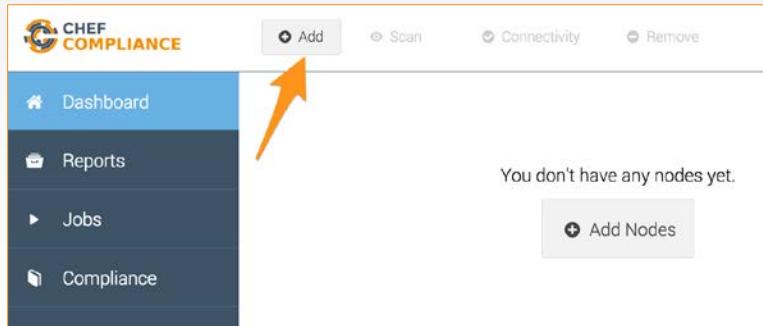
- Add a Linux Node to Scan
- Test connectivity

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

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.

Dashboard / Add nodes

Enter nodes (IPs or hostnames):
ec2-52-91-159-53.compute-1.amazonaws.com [x](#) [Add your nodes via IP or hostname](#)

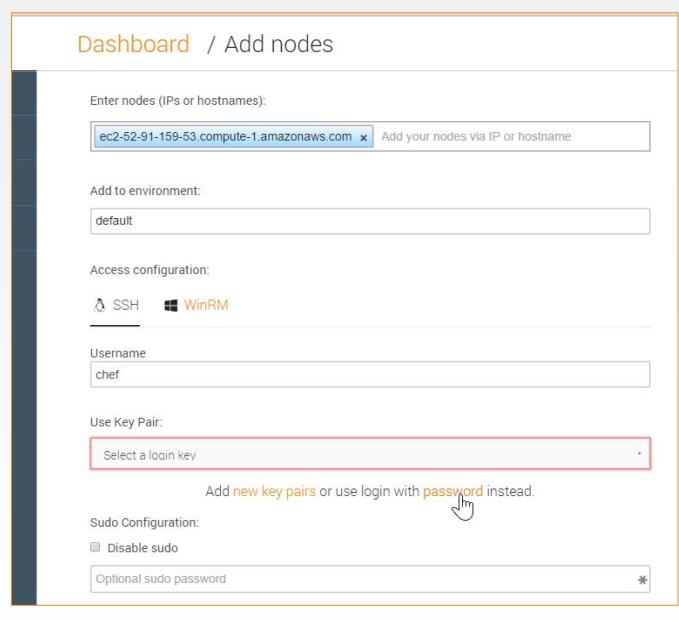
Add to environment:
 default

Access configuration:
 SSH WinRM

Username
chef

Use Key Pair:
[Select a login key](#) Add new key pairs or use login with password instead.

Sudo Configuration:
 Disable sudo
Optional sudo password *



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.

Add to environment:
default

Access configuration:
SSH WinRM

Username:
chef

Password-based login is generally not recommended and should be limited to development and legacy systems. Make sure you have a sufficiently complex password configured.

Use login with [public key](#) instead

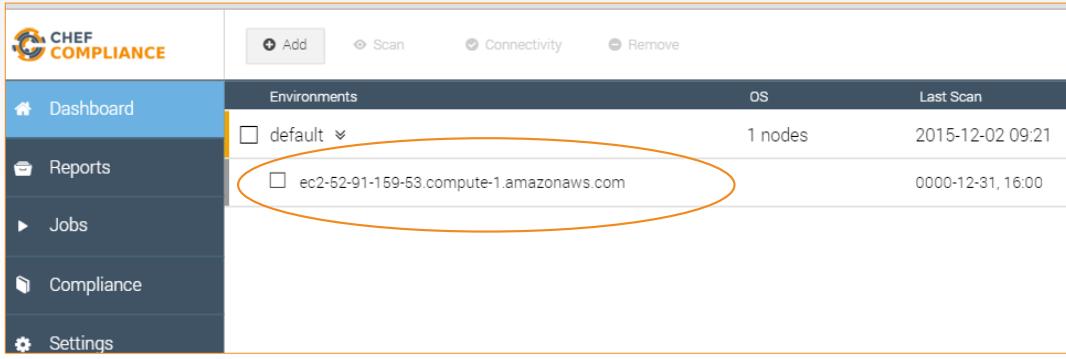
Sudo Configuration:
 Disable sudo
 Optional sudo password *

Add 1 node

Slide 8

GL: Adding a Node to Scan

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



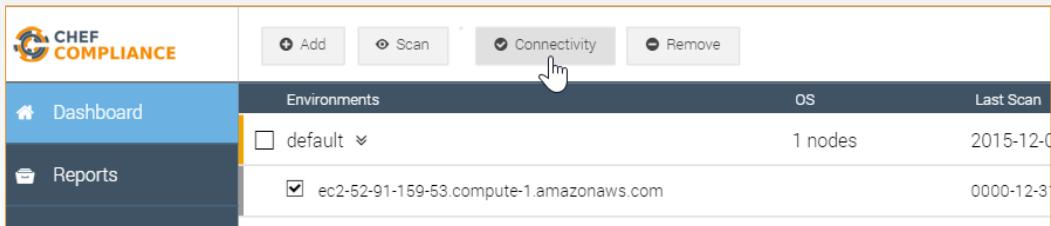
The screenshot shows the Chef Compliance web interface. On the left is a sidebar with links: Dashboard (which is selected and highlighted in blue), Reports, Jobs, Compliance, and Settings. At the top right are four buttons: Add (highlighted in grey), Scan, Connectivity, and Remove. Below these is a table titled 'Environments'. The table has three columns: 'Environments', 'OS', and 'Last Scan'. There are two rows. The first row shows 'default' with a dropdown arrow, '1 nodes', and '2015-12-02 09:21'. The second row shows 'ec2-52-91-159-53.compute-1.amazonaws.com' with a dropdown arrow, '0000-12-31, 16:00'. An orange oval highlights the second row.

Environments	OS	Last Scan
default ▾	1 nodes	2015-12-02 09:21
ec2-52-91-159-53.compute-1.amazonaws.com		0000-12-31, 16:00

Slide 9

GL: Testing Connectivity to Your Node

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



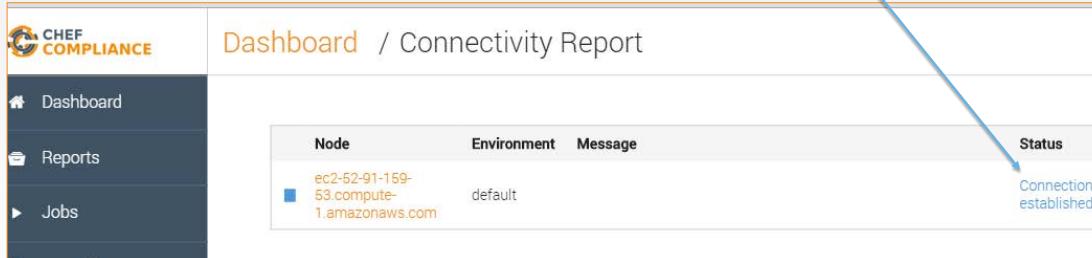
The screenshot shows the Chef Compliance dashboard. On the left, there's a sidebar with 'Dashboard' and 'Reports'. The main area has tabs for 'Environments', 'OS', and 'Last Scan'. Under 'Environments', there are two rows: 'default' (unchecked) and 'ec2-52-91-159-53.compute-1.amazonaws.com' (checked). At the top right, there are buttons for 'Add', 'Scan', 'Connectivity' (which has a hand cursor over it), and 'Remove'. The 'Connectivity' button is highlighted with a yellow border.

Environment	OS	Last Scan
default	1 nodes	2015-12-0
ec2-52-91-159-53.compute-1.amazonaws.com	0000-12-3	

Slide 10

GL: Testing Connectivity to Your Node

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



Node	Environment	Message	Status
ec2-52-91-159-53.compute-1.amazonaws.com	default		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.

The screenshot shows the 'Add nodes' page of the Chef Compliance web interface. At the top, it says 'Dashboard / Add nodes'. Below that, there's a field labeled 'Enter nodes (IPs or hostnames):' containing 'ec2-52-23-162-169.compute-1.amazonaws.com' and 'ec2-52-91-130-201.compute-1.amazonaws.com'. There's also a link 'Add your nodes via IP or hostname'. Under 'Add to environment:', the 'default' environment is selected. In the 'Access configuration:' section, 'SSH' is chosen over 'WinRM'. The 'Username' field contains 'chef'. A note below says 'Password-based login is generally not recommended and should be limited to development and legacy systems. Make sure you have a sufficiently complex password configured.' There's a placeholder '...' and a link 'Use login with public key instead.'. In the 'Sudo Configuration:' section, there's a checked checkbox 'Disable sudo' and an optional 'Optional sudo password' field. At the bottom right is a large orange button labeled 'Add 2 nodes' with a hand cursor icon pointing to it. The Chef logo is in the bottom right corner of the form area.

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.

```
1  ### Script to export Chef Server nodes and import them to Chef Compliance
2  ### Change the 'api_url', 'api_user' and 'api_pass' variables below
3  ### Go to your chef-repo and check Chef Server access first
4  # cd chef-repo; knife environment list
5  ### Save this Ruby script as kitchen_sink.rb and run it like this:
6  # cat kitchen_sink.rb | knife exec
7  ### Chef Compliance API docs: https://docs.chef.io/api_compliance.html
8
9  require 'json'
10 require 'uri'
11 require 'net/http'
12 require 'openssl'
13
14 # This extracts data from the Chef Server. Auth done by 'knife exec'
15 # Change loginKey and any other details that will be posted to the Chef Compliance API:
16 nodes_array = []
17 nodes.find('*:*') { |n|
18   nodes_array << { id: n.name,
19     name: n.name,
```

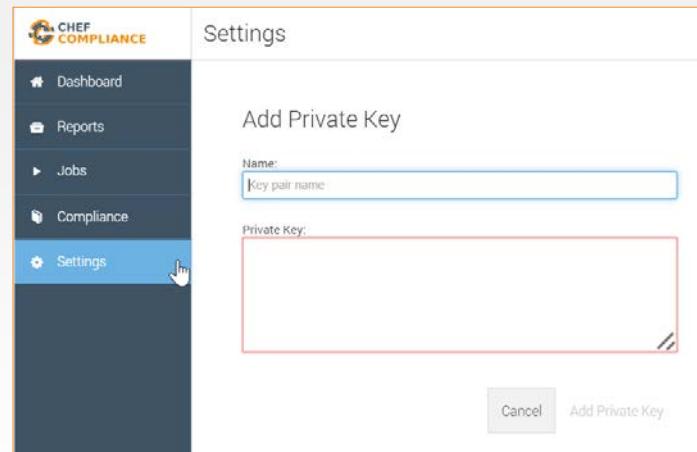
<https://gist.github.com/alexpop/01b0bba8d259adeee320>

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.

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.

The screenshot shows the Chef Compliance web interface. On the left is a dark sidebar with navigation links: Dashboard, Reports, Jobs, Compliance (which is currently selected), and Settings. The main content area has a light background. At the top, it says "Dashboard / Scan nodes". Below that, "Target nodes:" is listed with "1 host *" and the IP address "ec2-52-91-159-53.compute-1.amazonaws.com". Under "Patch Management", there is a checkbox labeled "Operating System Patch Level" which is checked. To its right is a dropdown menu titled "Compliance:" containing a list of profiles: "base/apache", "base/linux", "base/mysql", "base/postgres", "base/ssh", "base/windows", "cis/cis-ubuntu-level1", and "cis/cis-ubuntu-level2". At the bottom right of the main area are two buttons: "Scan now" and "Schedule".

©2016 Chef Software Inc.

3-15



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

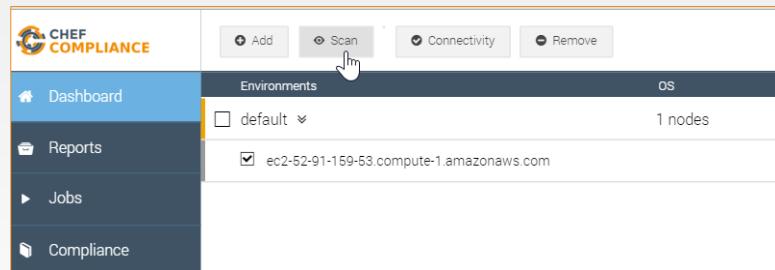
Objective:

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

Slide 17

GL: Running a Scan

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



The screenshot shows the Chef Compliance dashboard. On the left is a sidebar with 'Dashboard' (selected), 'Reports', 'Jobs', and 'Compliance'. At the top right are buttons for 'Add', 'Scan' (highlighted in orange), 'Connectivity', and 'Remove'. Below these are tabs for 'Environments' and 'OS'. Under 'Environments', there's a dropdown set to 'default' and a table showing '1 nodes' with one entry: 'ec2-52-91-159-53.compute-1.amazonaws.com' with a checked checkbox next to it.

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.

Dashboard / Scan nodes

Target nodes: 1 host ▾
ec2-52-91-159-53.compute-1.amazonaws.com

Patch Management

Operating System Patch Level

Compliance:

base/apache
 base/linux
 base/mysql
 base/postgres
 base/ssh
 base/windows
 cis/cis-ubuntu-level1
 cis/cis-ubuntu-level2

Scan now **Schedule**



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.

Hostname	Compliant	Minor Issues	Major Issues	Critical Issues	Skipped
ec2-52-91-159-53.compute-1.amazonaws.com	10	14	32	6	0
base/ssh: Client: Set SSH protocol version to 2				Critical Issues	
base/ssh: Server: Enable strict mode				Critical Issues	
base/ssh: Server: Disable empty passwords				Critical Issues	
base/ssh: Server: Disable X11 forwarding				Critical Issues	
base/ssh: Server: Ignore legacy .hosts configuration				Critical Issues	
base/ssh: Server: If X11 is used, enforce localhost				Critical Issues	
base/ssh: Server: Configure a listen address				Major Issues	
base/ssh: Server: Configure the service port				Major Issues	

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

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.

Hostname	Compliant	Minor Issues	Major Issues	Critical Issues	Skipped
ec2-52-91-159-53.compute-1.amazonaws.com	10	14	32	6	0
base/ssh: Client: Set SSH protocol version to 2				Critical Issues	
SSH Configuration Protocol should eq "2"				10.0	
base/ssh: Server: Enable strict mode				Critical Issues	
base/ssh: Server: Disable empty passwords				Critical Issues	
base/ssh: Server: Disable X11 forwarding				Critical Issues	
base/ssh: Server: Ignore legacy .rhosts configuration				Critical Issues	
base/ssh: Server: If X11 is used, enforce localhost				Critical Issues	
base/ssh: Server: Configure a listen address				Critical Issues	
base/ssh: Server: Configure the service port				Major Issues	
base/ssh: /etc/ssh should have limited access to 0755				Major Issues	
				Major Issues	

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.

The screenshot shows the Chef Compliance web interface. It consists of three main panels:

- Left Panel:** A sidebar with navigation links: Dashboard, Reports, Jobs, **Compliance** (highlighted with a blue background), and Settings.
- Middle Panel:** A "Reports / Compliance Report" section. It includes a "Summary" table with columns for Time, Targets, Compliant, Minor Issues, Major Issues, Critical Issues, and Other. Below the summary is a list of profiles:
 - Basic Apache 2
 - Basic Linux
 - Basic MySQL
 - Basic PostgresOL
 - Basic SSH** (highlighted with a blue background)
- Right Panel:** A detailed view of the "Basic SSH" profile. It shows the profile name, a list of checks (Client: Do not permit local commands, Client: Configure expected port, Client: Set SSH protocol version to 2), and the corresponding InSpec code:

```
control 'sshv4' 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
```

Slide 22

Discussion: InSpec Profile Code

Let's discuss what this profile is doing.

The `impact` of 1.0 indicates this is a Critical issue.

The `title` is what populates the Compliance Report issue title.

```
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
```

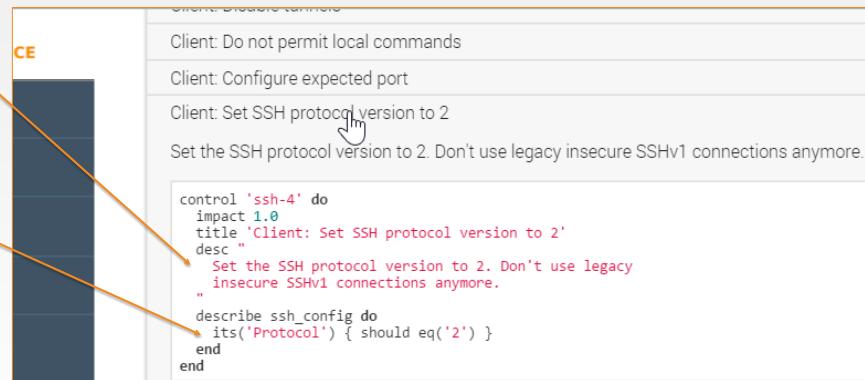
Let's discuss what this profile is doing.

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

Discussion: InSpec Profile Code

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

The `describe` section is the actual test that is executed.



```
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
```

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

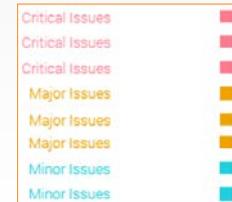
Compliance Profile Severity Mapping

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

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

```
control 'ssh-4' do
  impact 1.0
  title 'Client: Set SSH protocol version to 2'
  desc "
    Set the SSH protocol version to 2. Don't use legacy
    insecure SSHv1 connections anymore.
  "
  describe ssh_config do
    its('Protocol') { should eq('2') }
  end
end
```

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



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

©2016 Chef Software Inc.

3-24



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
```

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 converge the recipes on their next chef-client run.

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.



```
chef@ip-172-31-15-1: ~]$ pwd  
/home/chef  
[chef@ip-172-31-15-1: ~]$ █
```

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!

Slide 29

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.

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.

Slide 33

GL: Write the Client Recipe



```
$ ~/cookbooks/ssh/recipes/client.rb
```

```
# Cookbook Name:: ssh
# Recipe:: client
# Copyright (c) 2035 The Authors, All Rights Reserved.

template '/etc/ssh/ssh_config' do
  source 'ssh_config.erb'
  owner 'root'
  group 'root'
  mode '0644'
end
```

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.

Slide 35

GL: Navigate to your SSH Cookbook



```
$ cd ~/cookbooks/ssh/
```

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

Slide 36

GL: Edit your .kitchen.yml -- Part 1

~/cookbooks/ssh/.kitchen.yml

```
---
```

```
driver:
```

```
  name: docker
```



```
provisioner:
```

```
  name: chef_zero
```

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

GL: Edit your .kitchen.yml -- Part 3

```
~/cookbooks/ssh/.kitchen.yml
```

```
platforms:  
# - name: ubuntu-14.04  
- name: centos-6.7  
  
suites:  
- name: client  
  run_list:  
    - recipe[ssh::client]  
  attributes:
```

Finally, change the suites name to `client` and the run_list recipe name to `ssh:client`.
run_list:

```
- recipe[ssh::client]
```

Slide 39

GL: Run `kitchen list` from ~/cookbooks/ssh/



```
$ kitchen list
```

Instance	Driver	Provisioner	Verifier	Transport	Last Action
client-centos-67	Docker	ChefZero	Busser	Ssh	<Not Created> :

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

GL: Run `kitchen converge`



```
$ kitchen converge
```

```
-----> Starting Kitchen (v3.4.2)
-----> Creating <client-centos-67>...
      Sending build context to Docker daemon 32.26 kB
      Sending build context to Docker daemon
      Step 0 : FROM centos:centos6
      ---> 3bbbf0aca359
...
Chef Client finished, 0/3 resources updated in 03 seconds
      Finished converging <client-centos-67> (0m28.27s).
-----> Kitchen is finished. (0m30.32s)
zlib(finalizer): the stream was freed prematurely.
```

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

CONCEPT



What We've Done So Far

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?

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.

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/inpec/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
```

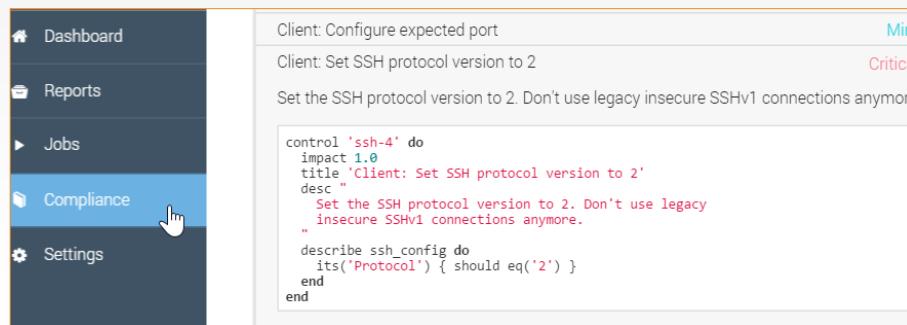
See the following slide for an example of a handy way to populate this file.

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



```
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
```

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.

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`



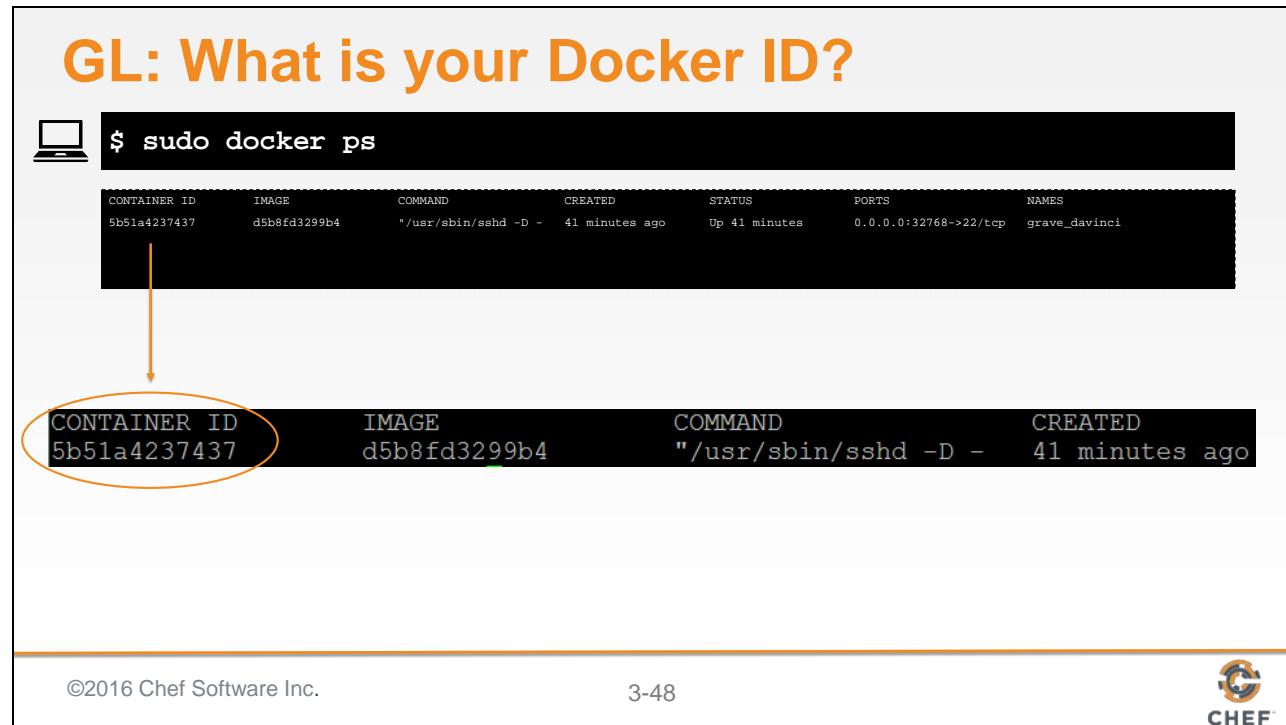
```
$ sudo chown root:dockerroot /var/run/docker.sock
```

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.

Slide 48

GL: What is your Docker ID?



```
$ sudo docker ps
```

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

©2016 Chef Software Inc. 3-48 

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

GL: Running InSpec from the CLI



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

```
Failures:  
1) SSH Configuration Protocol should eq "2"  
Failure/Error: its('Protocol') { should eq('2') }  
  
expected: "2"  
got: nil  
  
(compared using ==)  
# ./test/integration/client/inspec/client_spec.rb:9:in `block (3 levels) in load'  
  
Finished in 0.79369 seconds (files took 0.7207 seconds to load)  
1 example, 1 failure  
  
Failed examples:  
  
rspec # SSH Configuration Protocol should eq "2"
```

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"
```

GL: Update the Template



~/cookbooks/ssh/templates/default/ssh_config.erb

```
# IdentityFile ~/.ssh/id_rsa
# IdentityFile ~/.ssh/id_dsa
# Port 22
# Protocol 2,1
```

- 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

GL: Update the Template



~/cookbooks/ssh/templates/default/ssh_config.erb

```
# IdentityFile ~/.ssh/id_rsa  
# IdentityFile ~/.ssh/id_dsa  
# Port 22  
Protocol 2
```

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

Slide 52

GL: Ensure you are in ~/cookbooks/ssh



```
$ cd ~/cookbooks/ssh
```

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

Slide 53

GL: Run `kitchen converge`



```
$ kitchen converge
```

```
...
+++ /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
```

©2016 Chef Software Inc.

3-53



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).
```

GL: Running InSpec from the CLI



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

```
.
```

```
Finished in 0.21546 seconds (files took 0.3575  
seconds to load)  
1 example, 0 failures
```

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

GL: Apply the New SSH Recipe



```
$ sudo chef-client --local-mode -r 'recipe[ssh::client]'
```

```
...
+++ /etc/ssh/.ssh_config20151209-10413-hlk9ow      2015-12-09
20:37:07.621689137 +0000
@@ -37,7 +37,7 @@
 #   IdentityFile ~/.ssh/id_rsa
 #   IdentityFile ~/.ssh/id_dsa
 #   Port 22
-#   Protocol 2,1
+Protocol 2
 #   Cipher 3des
resources updated in 3.29477735 seconds
```

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.

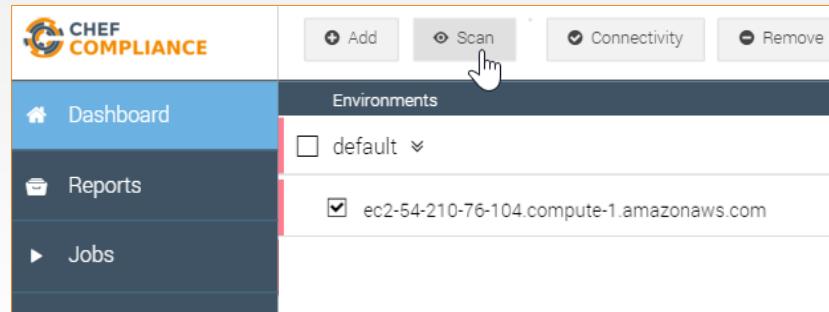
Of course in a production environment chef-client would most likely be set to run automatically to download and converge these changes from Chef Server.

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.

The screenshot shows the 'Scan nodes' page of the Chef Compliance interface. At the top, it says 'Dashboard / Scan nodes'. Below that, 'Target nodes:' is listed as '1 host: ec2-54-210-76-104.compute-1.amazonaws.com'. Under 'Patch Management', there is an unchecked checkbox for 'Operating System Patch Level'. Under 'Compliance', there are several checkboxes: 'base/apache', 'base/linux', 'base/mysql', 'base/postgres', 'base/ssh' (which is checked), 'base/windows', 'cis/cis-ubuntu-level1', and 'cis/cis-ubuntu-level2'. At the bottom right, there are two buttons: 'Scan now' (highlighted with a mouse cursor) and 'Schedule'.

Slide 58

GL: Results of this Exercise

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

	Minor Issues	
base/ssh: Server: Specify a valid address family	Compliant	
base/ssh: /etc/ssh should be a directory	Compliant	
base/ssh: /etc/ssh should be owned by root	Compliant	
base/ssh: sshd_config should be owned by root	Compliant	
base/ssh: sshd_config should not be writable/executable to others	Compliant	
base/ssh: Client: Set SSH protocol version to 2	Compliant	
SSH Configuration Protocol should eq "2"	10.0	
base/ssh: Server: Set protocol version to SSHv2	Compliant	
base/ssh: Server: Do not permit root-based login	Compliant	
base/ssh: sshd_config should not be group-writable/executable	Compliant	
base/ssh: sshd_config should not be group-writable/executable	Compliant	
base/ssh: Server: Disable challenge-response authentication	Compliant	
base/ssh: sshd_config should not be accessible to others	Compliant	

Slide 59

DISCUSSION



Conclusion

- ✓ Log in to your target node.
- ✓ Write a remediation recipe on that node.
- ✓ Test the recipe with Test Kitchen.
- ✓ Test for compliance with InSpec from the CLI
- ✓ Converge the recipe.
- ✓ Rescan the node and ensure compliance.

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 converge 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
```

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?

Slide 62



©2016 Chef Software Inc.

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

Running Scans, Remediation, and Testing on Windows Nodes

Configuring the Chef Compliance Server to Run Scans and Writing Remediation Recipes

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

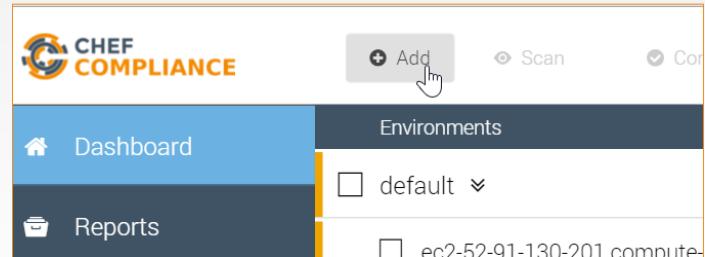
Objective:

- Add a Windows Node to Scan
- Test connectivity

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.

Enter nodes (IPs or hostnames):
 x
198.51.100.1, 192.0.2.1:8080, www.example.com, server.example.com:22

Add to environment:

Access configuration:
 SSH WinRM

Username

Password
 ?

Communication Protocol Show Des

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.

Access configuration:

SSH WinRM

Username
Administrator

Password
.....

Communication Protocol
HTTP

Show

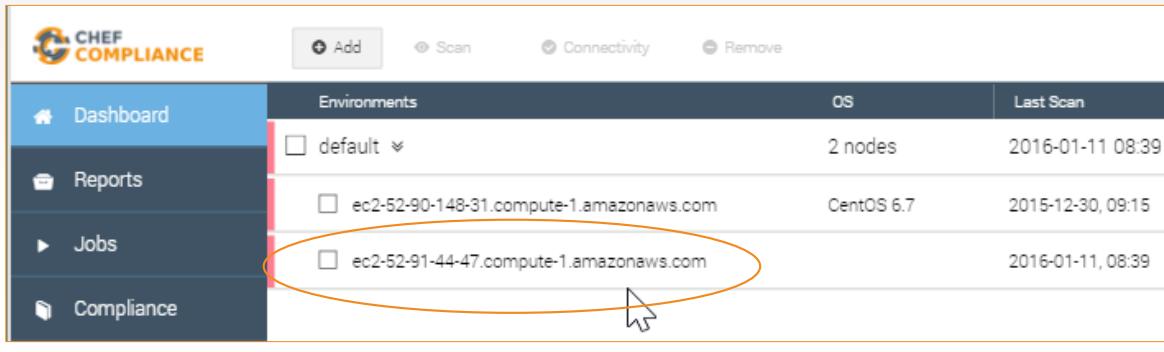
Add 1 node



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.



Environments	OS	Last Scan
default	2 nodes	2016-01-11 08:39
ec2-52-90-148-31.compute-1.amazonaws.com	CentOS 6.7	2015-12-30, 09:15
ec2-52-91-44-47.compute-1.amazonaws.com		2016-01-11, 08:39

©2016 Chef Software Inc. 4-7 

Slide 8

GL: Modify the Node Name

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

Environments	OS
default	2 nodes
ec2-52-90-148-31.compute-1.amazonaws.com	CentOS 6.7
ec2-52-91-44-47.compute-1.amazonaws.com	

The screenshot shows the Chef Compliance web interface. At the top, there's a navigation bar with 'Environments' and 'OS' dropdowns. Below that is a table listing environments and their nodes. The 'default' environment has two nodes: 'ec2-52-90-148-31.compute-1.amazonaws.com' (CentOS 6.7) and 'ec2-52-91-44-47.compute-1.amazonaws.com'. A hand cursor is hovering over the second node. The main content area shows a card for the node 'ec2-52-91-44-47.compute-1.amazonaws.com'. The card has tabs for 'Status', 'Compliance', 'Patch Level', and 'Configuration', with 'Configuration' being the active tab. Below the tabs is a section titled 'Node Status' with 'Scan' and 'Connectivity' buttons. A progress bar at the bottom of the card is red.

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**.

The screenshot shows the 'Node Configuration' page in the Chef web interface. At the top, there are tabs for Status, Compliance, Patch Level, and Configuration, with Configuration selected. Below the tabs, the title 'Node Configuration' is displayed. There are two input fields: 'Name:' containing 'Windows' and 'IP or Hostname:' containing 'ec2-52-91-44-47.compute-1.amazonaws.com'. Under 'Access configuration:', there are two options: 'SSH' (selected) and 'WinRM'. A large orange rectangular box highlights the 'Save' button at the bottom right, which has a hand cursor icon pointing to it.

Slide 10

GL: Modify the Node Name

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



The screenshot shows the Chef Compliance dashboard. On the left, there's a sidebar with links for Dashboard, Reports, Jobs, and Compliance. The main area is titled "Environments" and lists three entries:

	Environments	OS	Last Scan
<input type="checkbox"/>	default	2 nodes	2016-01-11 08:39
<input type="checkbox"/>	ec2-52-90-148-31.compute-1.amazonaws.com	CentOS 6.7	2015-12-30, 09:15
<input type="checkbox"/>	Windows ec2-52-91-44-47.compute-1.amazonaws.com		2016-01-11, 08:39

A hand cursor icon is positioned over the third row, indicating it can be selected or modified.

Slide 11

GL: Testing Connectivity to your Node

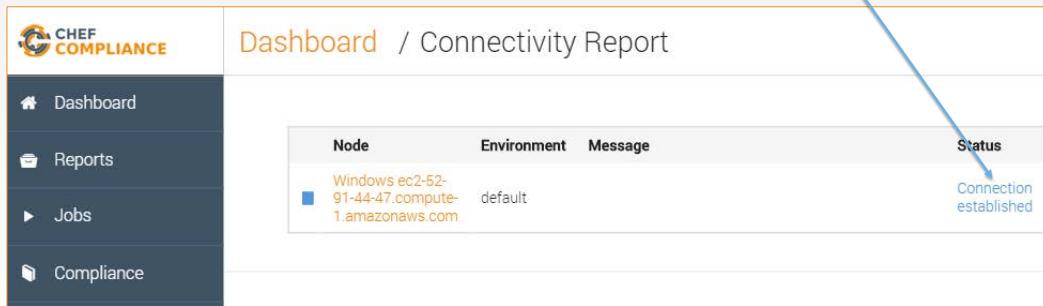
1. Click the **check box** next to your Windows node and then click the **Connectivity** button.

Environments	OS	Last S
<input type="checkbox"/> default	2 nodes	2016-01-01
<input type="checkbox"/> ec2-52-90-148-31.compute-1.amazonaws.com	CentOS 6.7	2015-12-31
<input checked="" type="checkbox"/> Windows ec2-52-91-44-47.compute-1.amazonaws.com		2016-01-01

Slide 12

GL: Testing Connectivity to your Node

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



Node	Environment	Message	Status
Windows ec2-52-91-44-47.compute-1.amazonaws.com	default		Connection established

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

Slide 13

EXERCISE

Group Lab: Running a Scan



Objective:

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

Slide 14

GL: Running a Scan

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

Environments	os
<input type="checkbox"/> default	2 nodes
<input type="checkbox"/> ec2-52-90-148-31.compute-1.amazonaws.com	CentOS 6.7
<input checked="" type="checkbox"/> Windows ec2-52-91-44-47.compute-1.amazonaws.com	

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.

The screenshot shows a list of compliance profiles on the right side of a web interface. A blue arrow points from the third step in the list to the 'base/windows' checkbox. The 'base/windows' checkbox is checked, while all other checkboxes are unchecked. At the bottom of the list are two buttons: 'Scan now' (highlighted with a mouse cursor) and 'Schedule'.

<input type="checkbox"/> Operating System Patch Level
<input type="checkbox"/> admin/profile
<input type="checkbox"/> admin/profile-cis-3.1
<input type="checkbox"/> base/apache
<input type="checkbox"/> base/linux
<input type="checkbox"/> base/mysql
<input type="checkbox"/> base/postgres
<input type="checkbox"/> base/ssh
<input checked="" type="checkbox"/> base/windows
<input type="checkbox"/> cis/cis-ubuntu-level1
<input type="checkbox"/> cis/cis-ubuntu-level2

Scan now Schedule

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.

Reports / Compliance Report

Summary

Time: 2016-01-27 06:59 (total time: a minute)
Scanned targets: 1 node

Compliance

Category	Value
Compliant	11 tests
Minor Issues	4 tests
Major Issues	0 tests
Critical Issues	14 tests
Other	0 rules

Compliance Overview

Critical 14

©2016 Chef Software Inc. 4-16

CHEF

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 NTLMv2 Authentication Enabled

Hostnames	Compliant	Minor Issues	Major Issues	Critical Issues	Skipped
Windows ec2-52-91-44-47.compute-1.amazonaws.com	11	4	0	14	0
base/windows: Windows Remote Desktop Configured to Only Allow System Administrators Access	Critical Issues				
base/windows: Minimum Windows Password Length Configured to be at Least 8 Characters	Critical Issues				
base/windows: Set Windows Account Lockout threshold	Critical Issues				
base/windows: Account Login Audit Log	Critical Issues				
base/windows: Audit Application Group Management	Critical Issues				
base/windows: Audit Distributed Group Management	Critical Issues				
base/windows: All Shares are Configured to Prevent Anonymous Access	Critical Issues				
base/windows: Strong Windows NTLMv2 Authentication Enabled; Weak LM Disabled	Critical Issues				
base/windows: Enable Strong Encryption for Windows Network Sessions on Clients	Critical Issues				
base/windows: Enable Strong Encryption for Windows Network Sessions on Servers	Critical Issues				
base/windows: IE 64-bit tab	Critical Issues				
base/windows: Run antimalware programs against ActiveX controls	Critical Issues				
base/windows: Windows Remote Desktop Configured to Always Prompt for Password	Critical Issues				
base/windows: Strong Encryption for Windows Remote Desktop Required	Critical Issues				
base/windows: Configure System Event Log (Application)	Minor Issues				

©2016 Chef Software Inc. 4-17 

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.

The screenshot shows the Chef Compliance web interface. On the left, a sidebar has 'Compliance' highlighted with a mouse cursor. The main area is titled 'Reports / Compliance' and 'Summary'. A list of compliance profiles includes 'Basic Linux', 'Basic MySQL', 'Basic PostgreSQL', 'Basic SSH', and 'Windows Base Security', with 'Windows Base Security' also having a cursor over it. Below this, a detailed view of the 'Windows Base Security' profile shows InSpec code for 'Force Encrypted Windows Network Passwords' and 'Strong Windows NTLMv2 Authentication Enabled; Weak LM Disabled'. The code includes a rule named 'windows-base-201' with its impact level set to 1. It describes a registry key under 'HKLM\System\CurrentControlSet\Control\Lsa' and checks if its 'LMCompatibilityLevel' is equal to 4. At the bottom, there is a note about enabling strong encryption for Windows Network Sessions on Clients.

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

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.

The screenshot shows the Windows Registry Editor with the path `Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa`. The `Lsa` key is expanded, and the `LsaPid` value is highlighted with a blue selection bar, indicating it is missing or not set. The registry table lists various keys and their values:

Name	Type	Value
crashonauditfail	REG_DWORD	0x00000000 (0)
disabledomaincreds	REG_DWORD	0x00000000 (0)
everyoneincludesan...	REG_DWORD	0x00000000 (0)
forceguest	REG_DWORD	0x00000000 (0)
fullprivilegeauditing	REG_BINARY	00
LimitBlankPassword...	REG_DWORD	0x00000001 (1)
LsaPid	REG_DWORD	0x00000294 (660)
NoLMHash	REG_DWORD	0x00000001 (1)
Notification Packages	REG_MULTI_SZ	rassfm scedi
ProductType	REG_DWORD	0x00000007 (7)
restrictanonymous	REG_DWORD	0x00000000 (0)
restrictanonymoussa...	REG_DWORD	0x00000001 (1)
SecureBoot	REG_DWORD	0x00000001 (1)
Security Packages	REG_MULTI_SZ	" "

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.

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

Slide 22

GL: Remediating the Issue

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

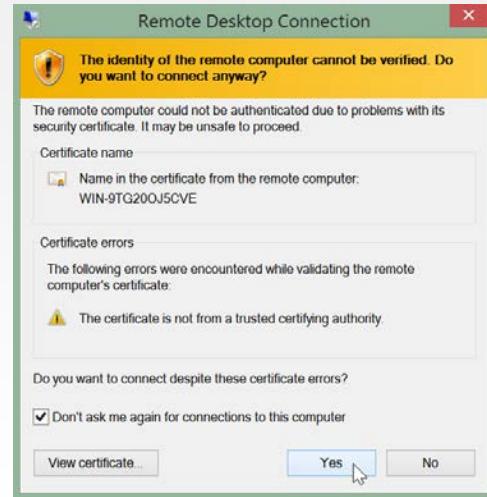
You can use only the IP address instead of the FQDN that this image shows.



Slide 23

GL: Remediating the Issue

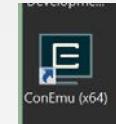
If you get a warning like this,
click Yes.



Slide 24

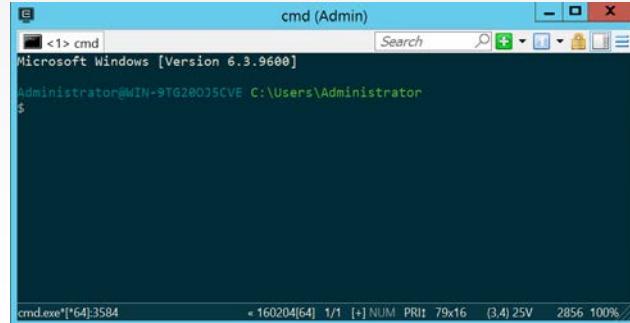
GL: Remediating the Issue

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



Slide 25

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.

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
```

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)
```

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

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

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
```

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
```

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

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[HKEY_LOCAL_MACHINE\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

GL: Run InSpec from the CLI



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

```
..
```

```
Finished in 2.16 seconds (files took 2.48 seconds to load)
```

```
2 examples, 0 failures
```

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

Slide 37

GL: Running a Scan

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

Environments	os
<input type="checkbox"/> default	2 nodes
<input type="checkbox"/> ec2-52-90-148-31.compute-1.amazonaws.com	CentOS 6.7
<input checked="" type="checkbox"/> Windows ec2-52-91-44-47.compute-1.amazonaws.com	

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.



<input type="checkbox"/> Operating System Patch Level
<input type="checkbox"/> admin/profile
<input type="checkbox"/> admin/profile-cis-3.1
<input type="checkbox"/> base/apache
<input type="checkbox"/> base/linux
<input type="checkbox"/> base/mysql
<input type="checkbox"/> base/postgres
<input type="checkbox"/> base/ssh
<input checked="" type="checkbox"/> base/windows
<input type="checkbox"/> cis/cis-ubuntu-level1
<input type="checkbox"/> cis/cis-ubuntu-level2

Scan now **Schedule**

Slide 39

GL: Results of this Exercise

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

	Critical issues	Minor issues
base/windows: Strong Encryption for Windows Remote Desktop Required		
base/windows: Configure System Event Log (Application)		
base/windows: Configure System Event Log (Security)		
base/windows: Configure System Event Log (Setup)		
base/windows: Configure System Event Log (System)		
base/windows: Windows Default Guest Account is Disabled		
base/windows: Windows Password Complexity is Enabled		
base/windows: Windows Account Lockout Counter Configured to Wait at Least 30 Minutes Before Reset		
base/windows: Windows Account Lockout Duration Configured to at Least 30 Minutes		
base/windows: Kerberos Authentication Service Audit Log		
base/windows: Kerberos Service Ticket Operations Audit Log		
base/windows: Audit Computer Account Management		
base/windows: Verify the Windows folder permissions are properly set		
base/windows: Safe DLL Search Mode is Enabled		
base/windows: Anonymous Access to Windows Shares and Named Pipes is Disallowed		
base/windows: Force Encrypted Windows Network Passwords		
base/windows: Strong Windows NTLMv2 Authentication Enabled; Weak LM Disabled		

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.

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?

```
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
```

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?

Slide 43



©2016 Chef Software Inc.

5: Creating Custom Profiles

Creating Custom Profiles

Defining and Uploading Compliance Profiles to the Compliance Server

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.

`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

EXERCISE

Group Lab: Creating a Custom Profile



Creating custom profiles to fit your business needs.

Objective:

- Create a custom profile.
- Test your profile with InSpec

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

GL: Using `inspec help`



```
$ inspec help  
[chef@ip-172-31-0-65 ~]$ inspec help  
Commands:  
  inspec archive PATH          # archive a profile to tar.gz (defau...  
  inspec check PATH           # verify all tests at the specified ...  
  inspec compliance SUBCOMMAND ... # Chef Compliance commands  
  inspec detect                # detect the target OS  
  inspec exec PATHS            # run all test files at the specifie...  
  inspec help [COMMAND]        # Describe available commands or one...  
  inspec init TEMPLATE ...    # Scaffolds a new project  
  inspec json PATH             # read all tests in PATH and generat...  
  inspec shell                 # open an interactive debugging shell  
  inspec version               # prints the version of this tool
```

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

GL: View the inspec.yml File



```
$ cat ~/secureprofile_01/inspec.yml
```

```
name: secureprofile_01
title: InSpec Profile
maintainer: The Authors
copyright: The Authors
copyright_email: you@example.com
license: All Rights Reserved
summary: An InSpec Compliance Profile
version: 0.1.0
```

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.

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

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

GL: Confirm Creation of tmp.rb using `tree`



```
$ tree secureprofile_01
```

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

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

Slide 15

GL: Edit the tmp.rb File - 1 of 3

~/secureprofile_01/controls/tmp.rb

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

title 'sample section'

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

# you add controls here
control 'tmp-1.0' do
  impact 0.7
  title 'Create /tmp directory'
  desc 'An optional description...'
  describe file('/tmp') do
    it { should be_directory }
  end
end
```

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.

Slide 17

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

```
~/compliance_profiles/profile_01/test/tmp.rb

...
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
```

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

GL: Use `inspec check` to Verify Your Profile



```
$ inspec check secureprofile_01/
```

```
Summary
```

```
-----
```

```
Location: secureprofile_01/
```

```
Profile: secureprofile_01
```

```
Controls: 3
```

```
Timestamp: 2016-02-26T21:24:52+00:00
```

```
Valid: true
```

```
Errors
```

```
-----
```

```
Warnings
```

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

Slide 19

GL: Use `inspec exec` to Verify Your Profile



```
$ inspec exec secureprofile_01/
```

```
....
```

```
Finished in 0.0467 seconds (files took 1.47 seconds to load)
4 examples, 0 failures
```

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?

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
```

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.

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

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.

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

GL: Ensure You Are in the Home Dir



```
$ cd ~  
$ ls
```

```
secureprofile_01
```

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 ...
```

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

GL: Logging in to your Compliance Server



```
$ inspec compliance login https://SERVER/api --user=admin --password='admin' --insecure
```

```
Successfully authenticated  
[chef@ip-172-31-0-65 ~]$
```

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

GL: Viewing Your Custom Profile Tree



```
$ tree secureprofile_01
```

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

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

GL: Uploading Your Custom Profile



```
$ inspec compliance upload secureprofile_01
```

```
I, [2016-03-02T14:56:18.714517 #29237] INFO -- : Checking profile in secureprof          ile_01
I, [2016-03-02T14:56:18.714883 #29237] INFO -- : Metadata OK.
I, [2016-03-02T14:56:18.842213 #29237] INFO -- : Found 3 controls.
I, [2016-03-02T14:56:18.842320 #29237] INFO -- : Verify all controls in control           s/example.rb
I, [2016-03-02T14:56:18.842380 #29237] INFO -- : Verify all controls in control           s/tmp.rb
I, [2016-03-02T14:56:18.842430 #29237] INFO -- : Control definitions OK.

Profile is valid
Generate temporary profile archive at /tmp/secureprofile_0120160302-29237-lhvbe             8.tar.gz
I, [2016-03-02T14:56:18.925296 #29237] INFO -- : Generate archive /tmp/securepr          ofile_0120160302-29237-
lhvbe8.tar.gz.
I, [2016-03-02T14:56:18.933640 #29237] INFO -- : Finished archive generation.

Start upload to admin/secureprofile_01
Uploading to https://54.152.196.46/api/owners/admin/compliance/secureprofile_01/          tar
Successfully uploaded profile
```

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
```

Slide 33

GL: Viewing Your Uploaded Custom Profile

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

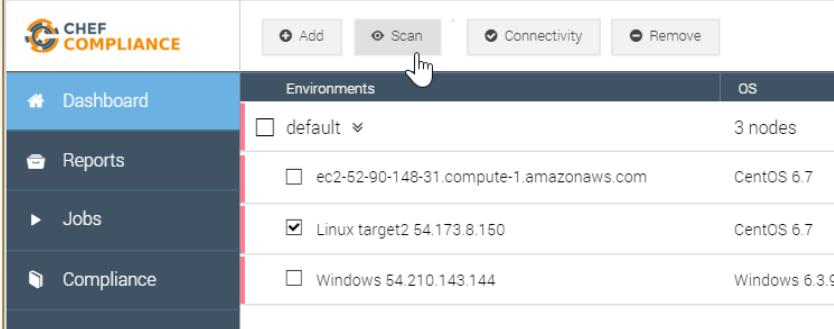
Notice that your custom profile is present.

Compliance profiles			
	Name	Identifier	Version
Dashboard	InSpec Example Profile	admin/profile	1.0.0
Reports	cis-3.1 Profile	admin/profile-cis-3.1	0.1.0
Jobs	MyName Profile1	admin/profile1	0.1.0
Compliance	InSpec Profile	admin/secureprofile_01	0.1.0
Settings	Basic Apache	base/apache	1.1.0
	Basic Linux	base/linux	1.1.0
	Basic MySQL	base/mysql	1.1.0

Slide 34

GL: Testing Your Uploaded Custom Profile

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



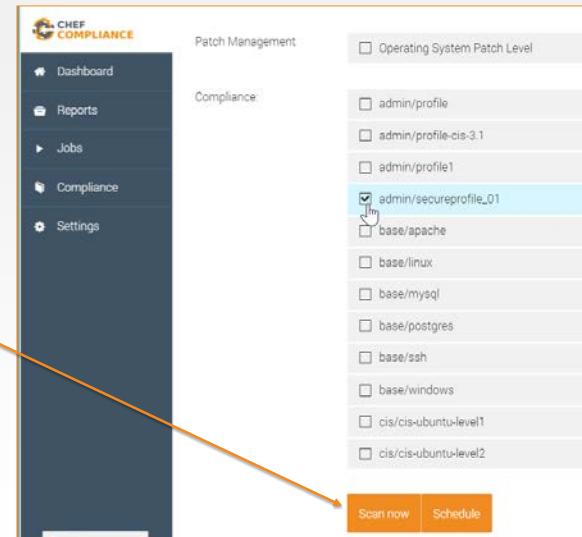
The screenshot shows the Chef Compliance interface. On the left is a sidebar with 'Dashboard' (selected), 'Reports', 'Jobs', and 'Compliance'. At the top right are buttons for 'Add', 'Scan' (which has a hand cursor over it), 'Connectivity', and 'Remove'. Below these are tabs for 'Environments' and 'OS'. The main table lists environments: 'default' (3 nodes), 'ec2-52-90-148-31.compute-1.amazonaws.com' (CentOS 6.7), 'Linux target2 54.173.8.150' (CentOS 6.7, checked), and 'Windows 54.210.143.144' (Windows 6.3.9). A red vertical bar highlights the 'Linux target2' row.

Environments	OS
<input type="checkbox"/> default	3 nodes
<input type="checkbox"/> ec2-52-90-148-31.compute-1.amazonaws.com	CentOS 6.7
<input checked="" type="checkbox"/> Linux target2 54.173.8.150	CentOS 6.7
<input type="checkbox"/> Windows 54.210.143.144	Windows 6.3.9

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.

Scan Report					
Hostname	Compliant	Minor Issues	Major Issues	Critical Issues	Skipped
Linux target2 54.173.8.150 ▾	2	0	0	0	0
secureprofile_01: Create /tmp directory	Compliant	7.0			
File /tmp should be directory					
secureprofile_01: /tmp directory is owned by the root user	Compliant	3.0			
File /tmp should be owned by "root"					

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.

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?

Slide 39



©2016 Chef Software Inc.

6: Applying Compliance Frameworks Using InSpec

Applying Compliance Frameworks Using InSpec

Translating CIS and DoD Specifications into InSpec Tests

©2016 Chef Software Inc.

6-1



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/>

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.

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**.

Available Free of Charge

On this web site, you'll find:

Information about the [Benchmarks](#), [Metrics](#), and [Assessment Tools](#)

111 (Auto-populates via J-SON) Benchmark documents in PDF

[Download Form](#)

[Browse Downloads](#)

Current version of CIS Benchmarks

28 [Security Metric Definitions](#) in PDF can be used across organizations to collect and analyze performance.

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.



Overview	Products & Solutions	Try & Buy	Communities &
CIS CentOS Linux 7 Benchmark	1.1.0	Thu Apr 2 18:3	
CIS Red Hat Enterprise Linux 7 Benchmark	1.1.0	Thu Apr 2 18:3	
CIS Microsoft Exchange Server 2013 Benchmark	1.1.0	Wed Mar 25 19:3	
CIS Microsoft Exchange Server 2010 Benchmark	1.1.0	Mon Mar 23 23:3	
CIS CentOS Linux 6 Benchmark	1.1.0	Mon Mar 2 21:3	
CIS Red Hat Enterprise Linux 6 Benchmark	1.4.0	Mon Mar 2 21:3	
CIS Red Hat Enterprise Linux 5 Benchmark	2.2.0	Mon Mar 2 21:3	

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.

MS-ISAC	Security Benchmarks	CIS Critical Security Controls
Overview Products & Solutions		
Alan Covell CIS RHEL Community David McMillan		
Download		

Required Information

Are you or your employer a current CIS Member? Yes No

Would you like to participate in the benchmark or the security metrics consensus process? Yes No

Would you like information about the benefits of a CIS Membership? Yes No

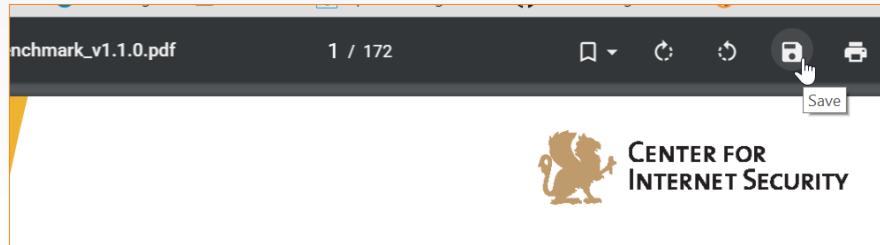
CIS resources are governed by [terms of use](#). By download acknowledge that you have read the terms of use in their e and agree to be bound by them in all respects.

[No Thanks!](#) [I Agree](#)

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.

1.2 Configure Software Updates	25
1.3 Advanced Intrusion Detection Environment (AIDE)	28
1.4 Configure SELinux.....	30
1.5 Secure Boot Settings	35
1.6 Additional Process Hardening	38
2 OS Services	41
2.1 Remove Legacy Services.....	41
3 Special Purpose Services	52
4 Network Configuration and Firewalls	64

3 Special Purpose Services

This section describes services that are installed on servers that specifically need to run these services. If any of these services are not required, it is recommended that they be disabled or deleted from the system to reduce the potential attack surface.

3.1 Set Daemon umask (Scored)

Profile Applicability:

- Level 1

Slide 11

Demonstration: Writing an InSpec Test for CIS Benchmark (1 of 3)**3 Special Purpose Services**

This section describes services that are installed on servers that specifically need to run these services. If any of these services are not required, it is recommended that they be disabled or deleted from the system to reduce the potential attack surface.

3.1 Set Daemon umask (Scored)**Profile Applicability:**

- Level 1

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.

```
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
```

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

Demonstration: Writing an InSpec Test for CIS Benchmark (2 of 3)**Audit:**

Perform the following to determine if the daemon umask is set.

```
# grep umask /etc/sysconfig/init  
umask 027
```

```
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
```

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.

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**.

Available Free of Charge

On this web site, you'll find:

Information about the [Benchmarks](#), [Metrics](#), and [Assessment Tools](#)

111 (Auto-populates via J-SON) Benchmark documents in PDF

[Download Form](#)

[Browse Downloads](#)

Current version of CIS Benchmarks

28 [Security Metric Definitions](#) in PDF can be used across organizations to collect and analyze performance.

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.

CIS Apple OSX 10.11 Benchmark	1.0.0	Thu Nov 5 17:24:26
CIS Apple OSX 10.9 Benchmark	1.2.0	Thu Nov 5 17:24:26
CIS Apple OSX 10.10 Benchmark	1.1.0	Thu Nov 5 17:24:26
CIS Apple OSX 10.8 Benchmark	1.3.0	Thu Nov 5 17:24:26
CIS Microsoft Windows Server 2012 R2 Benchmark	2.1.0	Fri Oct 30 15:34:28 2014
CIS Microsoft Windows 8.1 Benchmark	2.1.0	Fri Oct 30 15:32:47 2014

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.

MS-ISAC	Security Benchmarks	CIS Critical Security Controls
Overview Products & Solutions		
Alan Covell CIS RHEL Community David McMillan		
Download		

Required Information

Are you or your employer a current CIS Member? Yes No

Would you like to participate in the benchmark or the security metrics consensus process? Yes No

Would you like information about the benefits of a CIS Membership? Yes No

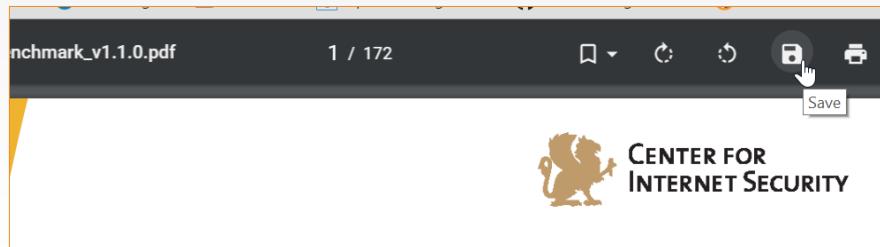
CIS resources are governed by [terms of use](#). By download acknowledge that you have read the terms of use in their e and agree to be bound by them in all respects.

[No Thanks!](#) [I Agree](#)

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.



Slide 20

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.

Recommendations.....
1 Account Policies
1.1 Password Policy
1.1.2 (L1) Set 'Maximum password age' to '60 or fewer days, but not 0' (Scored)
1.1.3 (L1) Set 'Minimum password age' to '1 or more day(s)' (Scored)
1.1.4 (L1) Set 'Minimum password length' to '14 or more character(s)' (Scored)
1.1.5 (L1) Set 'Password must meet complexity requirements' to 'Enabled'

1.1 Password Policy

This section contains recommendations for password policy.

1.1.1 (L1) Set 'Enforce password history' to '24 or more password(s) (Scored)

Profile Applicability:

- Level 1 - Domain Controller
- Level 1 - Member Server

Slide 21

GL: Writing an InSpec Test for a Windows CIS Benchmark (1 of 3)

1 Account Policies

This section contains recommendations for account policies.

1.1 Password Policy

This section contains recommendations for password policy.

1.1.1 (L1) Set 'Enforce password history' to '24 or more password(s)' (Scored)

Profile Applicability:

- Level 1 - Domain Controller
- Level 1 - Member Server

Description:

This policy setting determines the number of renewed, unique passwords that have to be associated with a user account before you can reuse an old password. The value for this policy setting must be between 0 and 24 passwords. The default value for Windows Vista is 0 passwords, but the default setting in a domain is 24 passwords. To maintain the effectiveness of this policy setting, use the Minimum password age setting to prevent users from repeatedly changing their password.

The recommended state for this setting is: 24 or more password(s).

```
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
```

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

GL: Writing an InSpec Test for a Windows CIS Benchmark (2 of 3)

Audit:

Navigate to the UI Path articulated in the Remediation section and confirm it is set as prescribed.

Remediation:

To establish the recommended configuration via GP, set the following UI path to 24 or more password(s):

Computer Configuration\Policies\Windows Settings\Security Settings\Account Policies>Password Policy\Enforce password history

```
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
```

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.

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>

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 Chef Compliance profile control.

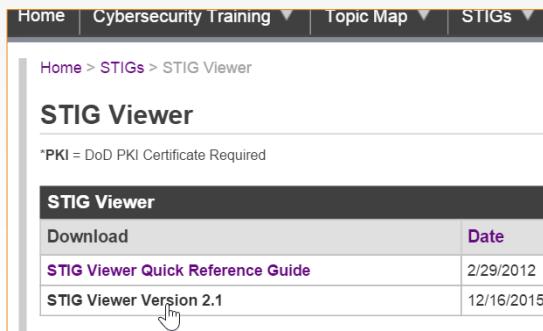
Slide 28

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



The screenshot shows a web page titled "STIG Viewer". At the top, there is a navigation bar with links for "Home", "Cybersecurity Training", "Topic Map", and "STIGs". Below the navigation bar, the URL "Home > STIGs > STIG Viewer" is displayed. The main content area has a heading "STIG Viewer" and a note "*PKI = DoD PKI Certificate Required". Below this, there is a table with two rows:

STIG Viewer	Date
Download	2/29/2012
STIG Viewer Quick Reference Guide	12/16/2015
STIG Viewer Version 2.1	

©2016 Chef Software Inc.

6-28



Go to this site...

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

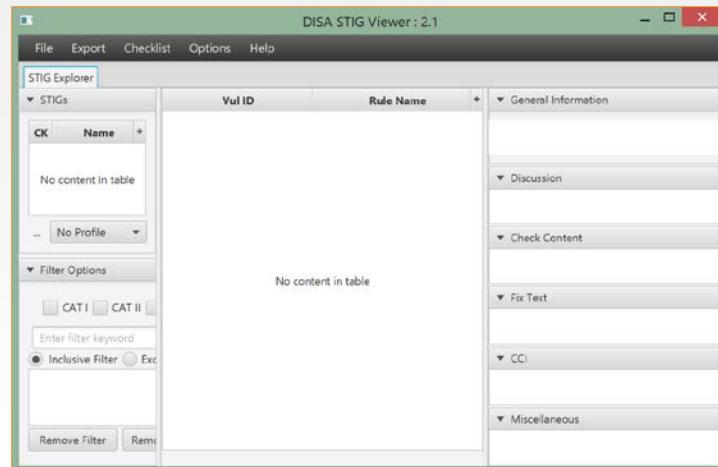
...and download the latest version of the STIG Viewer. In this example we are downloading Version 2.1.

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.

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-ifw.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>

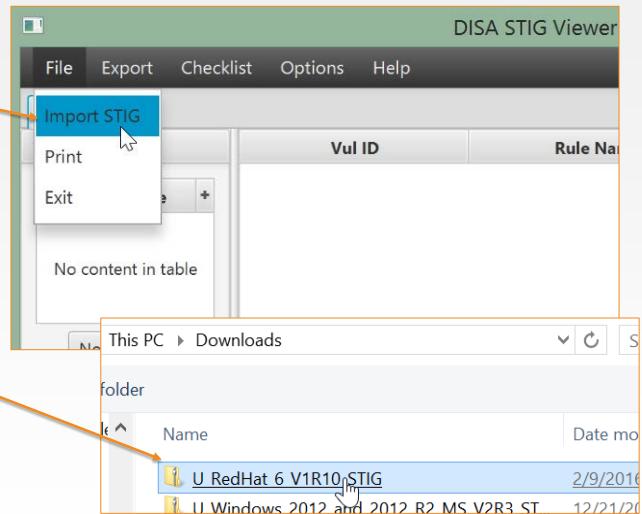
Operating Systems - Unix/Linux (Red Hat)			
*PKI = DoD PKI Certificate Required			
The SCAP benchmarks contain ONLY automated STIG content. The benchmarks do not contain STIG requirements.			
RedHat 5			
Download	Date	Size	Format
Red Hat 5 Manual STIG - Ver 1, Rel 13	1/22/2016	470 KB	ZIP
RedHat 6			
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.

The screenshot shows the DISA STIG Viewer 2.1 application window. The main pane displays a table of vulnerabilities (Vul ID) and their corresponding rule names (Rule Name). The table includes rows for V-38437 through V-38451, all of which have SRG-OS-999999 as the rule name. The left sidebar contains a 'STIG Explorer' section with dropdown menus for 'CK' and 'Name', and a 'Filter Options' section with checkboxes for 'CAT I' and 'CAT II'. The right sidebar provides 'General Information' about 'Red Hat Enterprise Implementation', including sections for 'Discussion', 'Check Content', and 'Fix Text'.

Vul ID	Rule Name
V-38437	SRG-OS-999999
V-38438	SRG-OS-000062
V-38439	SRG-OS-000001
V-38443	SRG-OS-999999
V-38444	SRG-OS-000231
V-38445	SRG-OS-000057
V-38446	SRG-OS-999999
V-38447	SRG-OS-999999
V-38448	SRG-OS-999999
V-38449	SRG-OS-999999
V-38450	SRG-OS-999999
V-38451	SRG-OS-999999

Slide 34

GL: Filter STIG Profiles

Type **38443** in the filter field.

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

The screenshot shows the STIG Explorer interface. At the top, there's a menu bar with File, Export, Checklist, Options, and Help. Below the menu is a toolbar with a 'STIG Explorer' button. The main area has two panes: a left pane labeled 'STIGs' containing a table with columns CK and Name, and a right pane labeled 'Vul ID' and 'Rule No.' containing a table with a single row V-38443 SRG-OS-999999. A large orange arrow points from the text 'Type 38443 in the filter field.' towards the filter input field in the bottom-left pane. Another orange circle highlights the 'Vul ID' column header in the right pane. The bottom-left pane also contains a 'Profile' dropdown set to 'No Profile', a 'Filter Options' section with checkboxes for CAT I, CAT II, and CAT III, and a search input field containing '38443' with an 'Add' button next to it. Radio buttons at the bottom of the filter options indicate 'Inclusive Filter' is selected.

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
```

Vul ID	Rule Name
V-38443	SRG-OS-999999

General Information

Rule Title: The /etc/gshadow file must be owned by root.

STIG ID: RHEL-06-000036 Severity: CAT II

Discussion

The "/etc/gshadow" file contains group password hashes. Protection of this file is critical for system security.

Check Content

To check the ownership of "/etc/gshadow", run the command:

```
$ ls -l /etc/gshadow
```

Fix Text

To properly set the owner of "/etc/gshadow", run the command:

```
# chown root /etc/gshadow
```

©2016 Chef Software Inc. 6-35 

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
```

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



©2016 Chef Software Inc.

7: Scheduling Scans and Running Reports

Scheduling Scans and Running Reports

Scheduling Scans for Future Reporting

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

CONCEPT

Scheduling Scans



The time zone for "Date" scheduling in the Compliance web UI is based on your local workstation's browser time zone.

The time zone for "Recurring" scheduling in the Compliance web UI is based on UTC. (Coordinated Universal Time, which equals GMT.)

The Compliance logs, should you view them, are also based on UTC.

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

Slide 5

EXERCISE

GE: Scheduling Scans



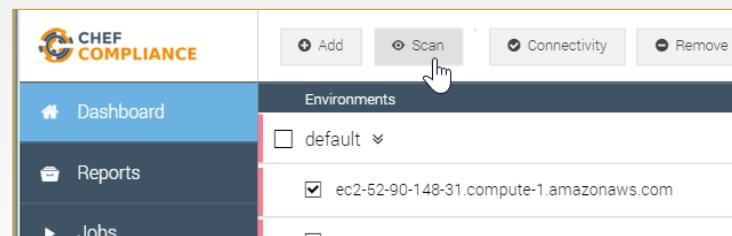
Objective:

- Schedule a scan.
- View the scan output.

Slide 6

GE: Scheduling Scans

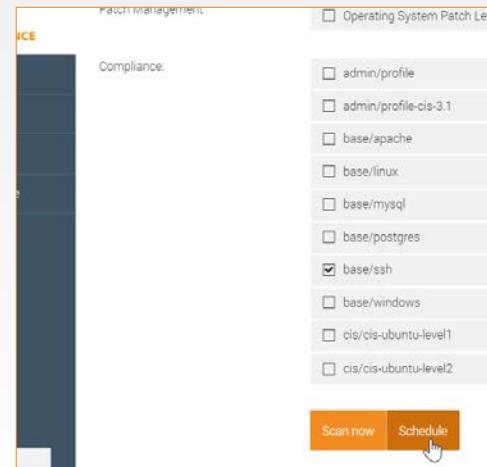
1. Open the Compliance web UI's Dashboard page.
2. Select one of your target nodes.
3. Click **Scan**.



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.

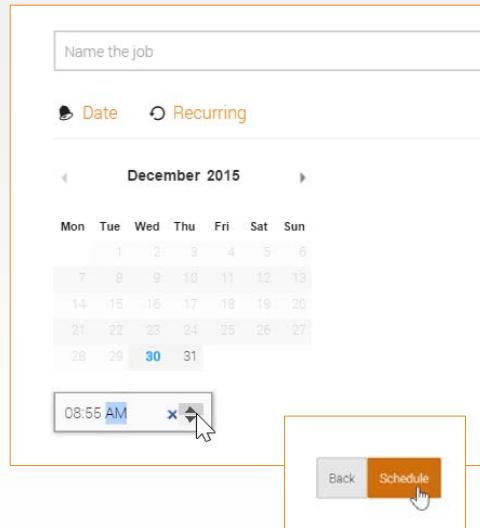


Slide 8

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**.

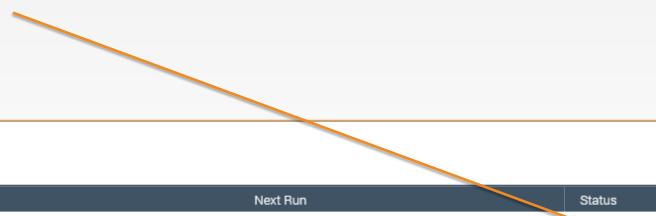
Note: The up/down arrows are also used to set the AM and PM value.



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.



Jobs			
	Name	Next Run	Status
Dashboard	Daily Base ssh Scan	2016-02-10 10:55 (-08:00)	scheduled
Reports	Daily Node Scan	-	done
Jobs			

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".



	Name	Next Run	Status
	Daily Base ssh Scan	-	done
	Daily Node Scan	-	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.



Scan Reports				
	Time	Nodes	Patch Level	Compliance
Dashboard	2016-02-10 10:55	1 nodes	Updates available	Critical Issues
Reports	2016-02-10 10:35	1 nodes	Updates available	Critical Issues

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.

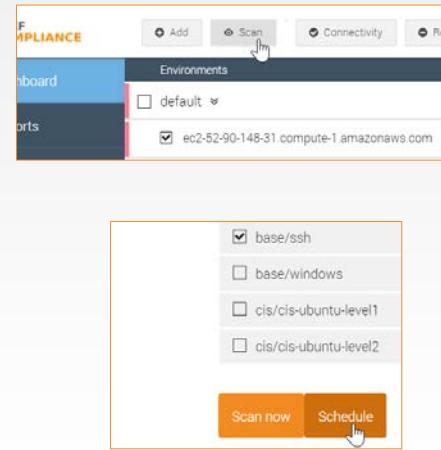
The screenshot shows the Chef Compliance web application. On the left is a sidebar with links: Dashboard, Reports, Jobs, Compliance, and Settings. The main area has a title "Reports / Compliance Report". Under "Summary", it shows the date "2016-02-10 10:55 (total time: a few seconds)" and "1 node". It lists "Compliant" (12 tests), "Minor issues" (14 tests), "Major issues" (31 tests), "Critical issues" (2 tests), and "Other" (0 issues). A "Compliance Overview" donut chart indicates 81% Major issues. Below this is a "Scan Report" section with a table of scan results for a single node. The table has columns: Hostname, Compliant, Minor Issues, Major Issues, and Critical. The row for the node "ec2-52-90-148-31.compute-1.amazonaws.com" shows: Compliant (12), Minor Issues (14), Major Issues (31), and Critical (2). Each critical issue is listed as "base[ssh]: Server: Enable root mode", "base[ssh]: Server: Disable empty passwords", and "base[ssh]: Server: Disable x11 forwarding".

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.

Name the job

Date Recurring

December 2015

Mon Tue Wed Thu Fri Sat Sun

Dashboard / Scan nodes

Name the job

Date Recurring

Recurring events use UTC timezone

day at 23 : 48 UTC

minute
hour
day
week
month
year

Back Schedule

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`
```

Slide 16

Demonstration: Scheduled Scan Logs

The screenshot shows the Chef Compliance web interface. On the left, there's a calendar for January 2016 with the 12th highlighted. Below the calendar, a time selector shows "01:45 PM". To the right of the calendar are two buttons: "Back" and "Schedule". A hand cursor is hovering over the "Schedule" button. On the right side of the interface, there's a log entry with the following details:

```
2016-01-12_21:41:16.82603 21:41:16.825 DEB => Add  
job: admin/2e48b32c-bd3b-4024-9d15-f3fc82537290  
2016-01-12 21:45:00 +0000 UTC next: 2016-01-12  
21:45:00 +0000 UTC  
2016-01-12_21:41:16.83327 [GIN] 2016/01/12 - 21:41:16 |  
200 | 9.475356ms | 50.170.125.99 | POST  
/owners/admin/jobs  
2016-01-12_21:41:16.94064 [GIN] 2016/01/12 - 21:41:16 |  
200 | 3.283171ms | 50.170.125.99 | GET  
/owners/admin/jobs
```

At the bottom of the interface, there are copyright information and a page number:

©2016 Chef Software Inc. 7-16 

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.

The screenshot shows the 'Jobs' section of the Chef Compliance interface. On the left is a sidebar with 'Dashboard', 'Reports', and 'Jobs' (which is highlighted in blue). The main area has a table titled 'Jobs' with columns 'Name' and 'Next Run'. Two entries are listed: 'Daily Base ssh Scan' and 'Daily Node Scan'. To the right of the 'Daily Node Scan' entry is a small circular icon containing a hand cursor, indicating it's a clickable element.

This screenshot shows a detailed view of a specific job. The top bar says 'Jobs / c642e137-3d49-463c-ae4e'. The sidebar includes 'Dashboard', 'Reports', 'Jobs' (selected), 'Compliance', and 'Settings'. The main content area shows 'Job Status' with 'Status: done' and 'Next Run: -'. Below this is a large 'Delete' button with a circular cursor icon over it.

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.

Scan Reports				
	Time	Nodes	Patch Level	Compliance
Dashboard	2015-12-30 09:15	1 nodes	Updates available	Critical Issues
Reports	2015-12-30 08:55	1 nodes	Updates available	Critical Issues
Jobs	2015-12-30 08:18	1 nodes	Updates available	Critical Issues
Compliance	2015-12-29 11:19	1 nodes	Updates available	Critical Issues

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.

The screenshot shows a sidebar menu with options: Dashboard, Reports, Jobs, Compliance, and Settings. The 'Reports' option is selected. The main area displays a summary table for a node named 'ec2-52-90-148-91.compute-1.amazonaws.com'. The table has columns: Compliant, Minor Issues, Major Issues, Critical Issues, and None. The values are: 12, 14, 31, 5, and 0 respectively. Below the table, a list of specific issues is shown:

- base::ssh: Server::Enable sshd mode
- base::ssh: Enable empty passwords
- base::ssh: Ensure the port number should be '22'
- base::ssh: Ensure the port number should be '22'
- base::ssh: Ensure legacy sshd configuration
- base::ssh: Ensure 'ifconfig' is used, enforce localhost
- base::ssh: Ensure a listen address

Slide 21

GE: Exporting Reports

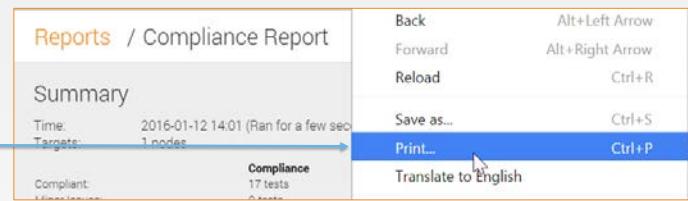
1. From the left column of the web UI, click **Reports** and then click any report that may exist.

Scan Reports	
	Time
	2016-01-12 14:29
	2016-01-12 13:45
	2016-01-12 08:05

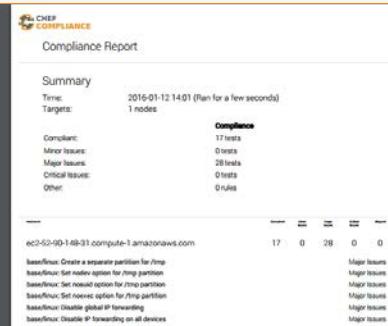
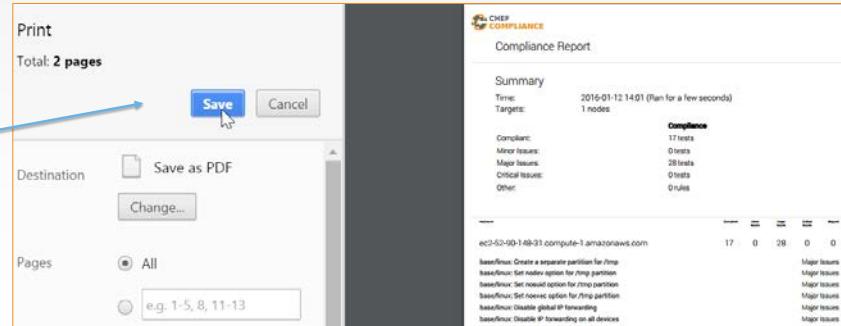
Slide 22

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 your 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

Resource	Compliant	Minor Issues	Major Issues	Critical Issues	Skipped
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		

©2016 Chef Software Inc.

7-23



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"?

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?

Slide 26



©2016 Chef Software Inc.

8: Users, Organizations, Teams and Permissions

Users, Organizations, Teams and Permissions

Specifying Views and Actions Users Have Access To

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

CONCEPT

Managing Users



You can create and edit Compliance Server users.

You can also modify their permission to allow or prevent certain actions.

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.

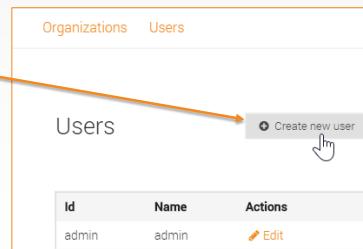
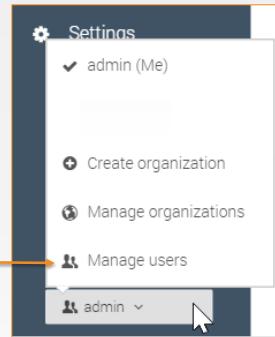
Objectives:

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

Slide 5

GL: Creating Users

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



The screenshot shows a table with one row:

Id	Name	Actions
admin	admin	 Edit

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 screenshot shows the 'Users' tab selected in the top navigation bar. The 'Username' field contains 'Jane Doe'. The 'User ID' field contains 'jane-doe'. The 'Password' field contains 'chef'. Under 'User Permissions', there are three checkboxes: 'Site Management' (unchecked), 'Organization Management' (unchecked), and 'User Management' (unchecked). At the bottom right are 'Cancel' and 'Add user' buttons, with 'Add user' being orange.

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.

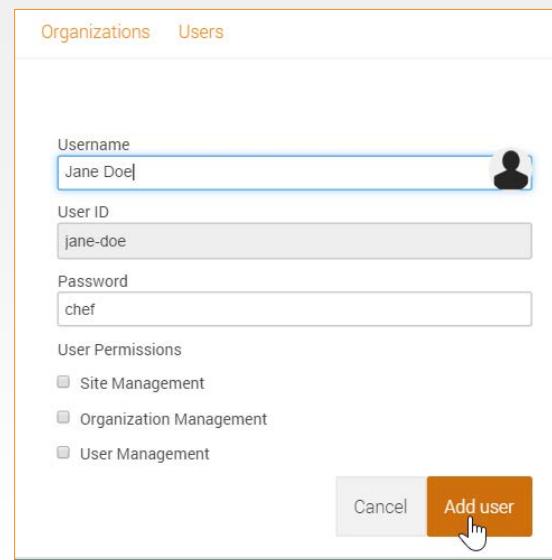
Slide 7

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.



The screenshot shows a user creation form with the following fields:

- Username: Jane Doe
- User ID: jane-doe
- Password: chef
- User Permissions:
 - Site Management
 - Organization Management
 - User Management

At the bottom right, there are "Cancel" and "Add user" buttons. The "Add user" button is highlighted with a mouse cursor icon pointing to it.

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.

Users		
Id	Name	Actions
admin	admin	Edit
jane-doe	Jane Doe	Edit Delete

Username
Jane Doe

User ID
jane-doe

Password

User Permissions

Site Management

Organization Management

User Management

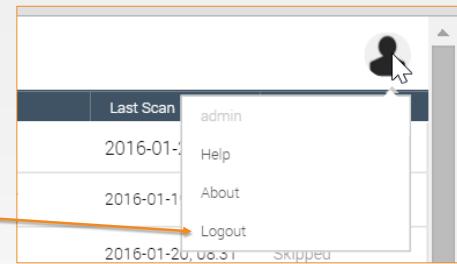
Notice that you can't delete the original admin user, however, since you always need at least an admin user.

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**.

A screenshot of the Chef Compliance login page. It features a title 'Compliance Dashboard'. There are two input fields: one yellow for 'jane-doe' and one white for a password. Below the fields are links for 'Forgot password?' and 'Sign In'. A mouse cursor is hovering over the 'Sign In' button.

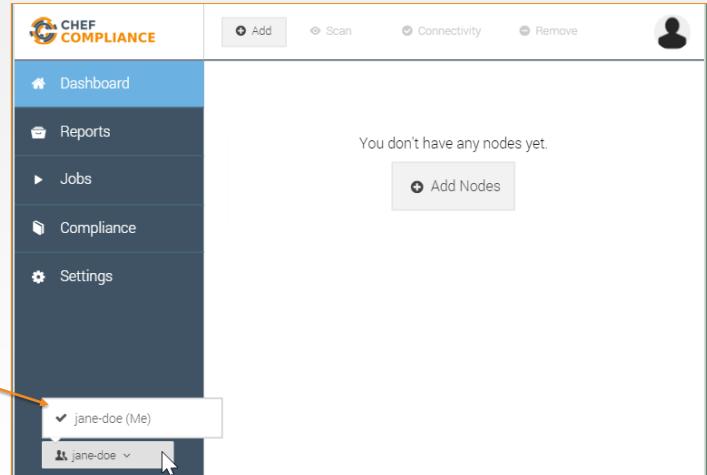
Slide 10

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

CONCEPT



Organizations in Chef Compliance

Organizations are objects that enable you to segregate target nodes and make them accessible to users other than just the admin user.

Before you add nodes that you may want to share with other users, you should first create at least one Compliance organization and a corresponding Compliance team to which those nodes will be associated.

In such a scenario, you'll need to switch to the new organization and then add nodes while you are working under that organization.

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.

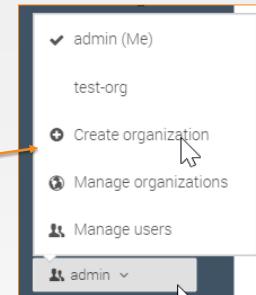
Objectives:

- 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.

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 is already part of this and any new organization teams.

Organizations		
Actions		
Id	Name	Actions
chef	chef	

Teams		
Owners		
Member	Actions	
admin		
<input type="text" value="Add user"/>		
admin		
jane-doe		

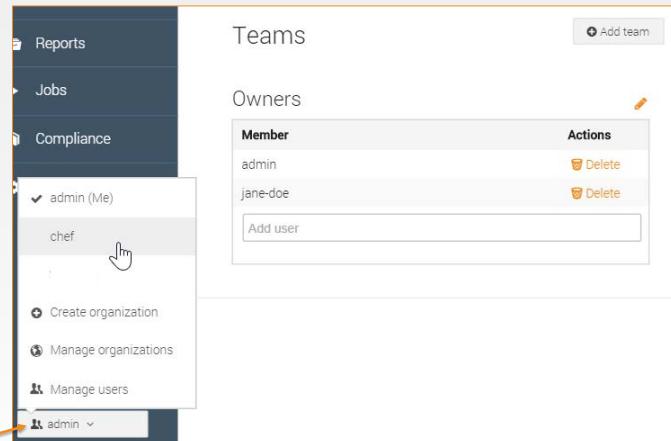
Slide 15

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.



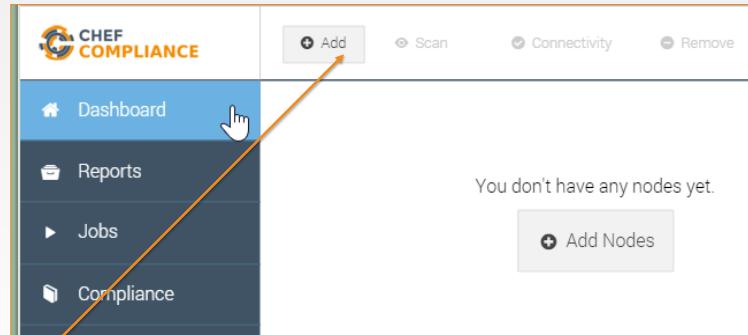
Slide 16

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.



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.

The screenshot shows the 'Add 1 node' dialog box. It has the following fields:

- Enter nodes (IPs or hostnames): 52.90.148.31, 198.51.100.1, 192.0.2.1:8080, www.example.com, server.example.com:22
- Add to environment: Choose an Environment
- Access configuration:
 - SSH (selected)
 - WinRM
- Username: chef
- >Password-based login is generally not recommended and should be limited to development and legacy systems. Make sure you have a sufficiently complex password configured.
....
- Use login with [private key](#) instead.
- Sudo Configuration:
 - Disable sudo
- Optional sudo password

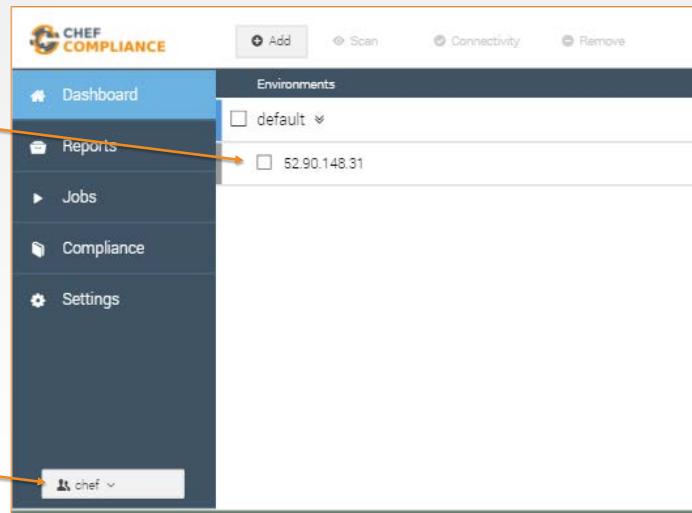
At the bottom right, there are 'Cancel' and 'Add 1 node' buttons. The 'Add 1 node' button is highlighted with a mouse cursor icon.

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.

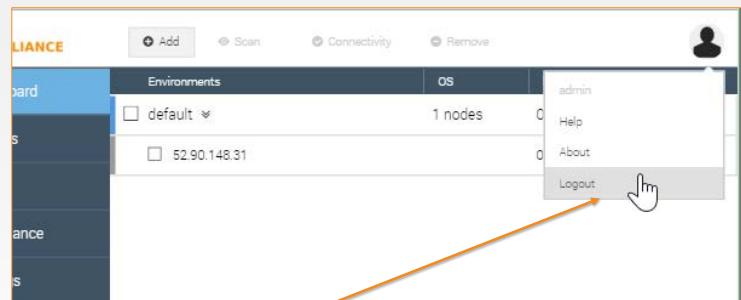


Slide 19

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**.

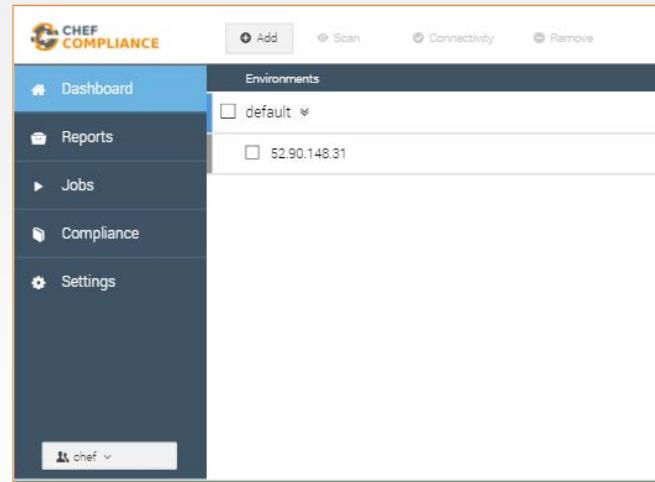


Slide 20

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?

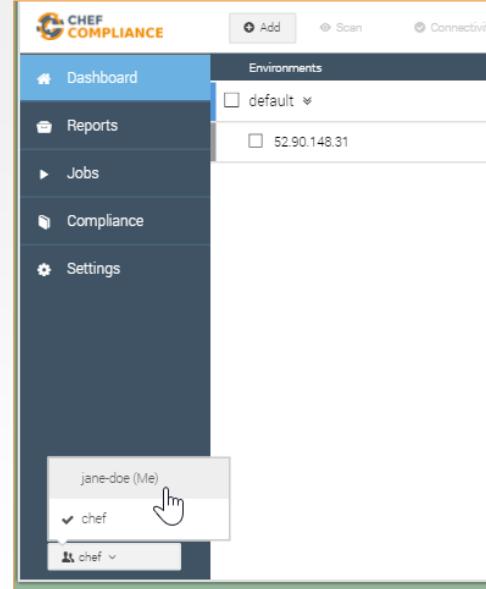


Slide 21

GL: Using Your New Organization

9. From the bottom-left of the Compliance Dashboard, switch back and forth between your 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



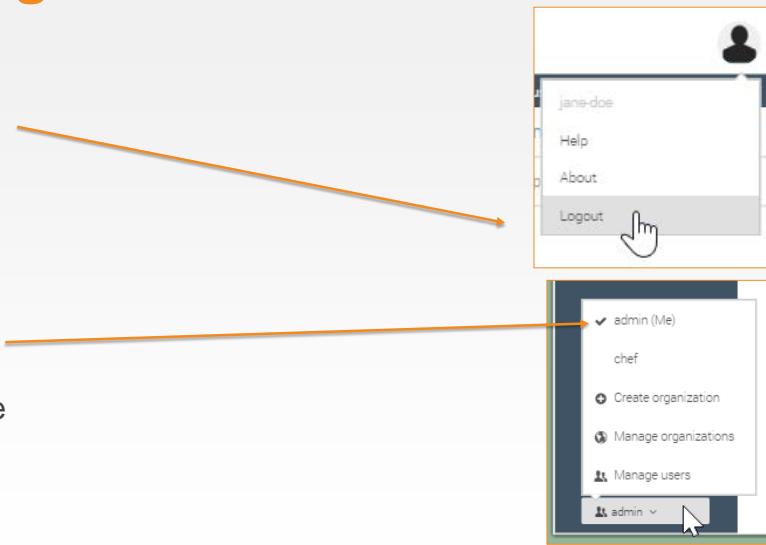
Teams have their own set of permissions.

Team permissions are completely separate and independent of user permissions.

Slide 23

GL: Managing Team Permissions

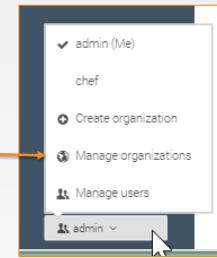
1. Log out of the Compliance Web UI and log back in as **admin**.
2. If you like you can switch back to the admin organization although it really doesn't matter for the next task.



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.



ID	Name	Actions
chef	chef	Edit Teams Delete

Slide 25

GL: Managing Team Permissions

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

The team permissions screen should display.



A screenshot of a modal dialog box titled 'Team Permissions'. It contains fields for 'Team name' (set to 'Owners') and 'Team ID' (set to 'owners'). Below these are four checkboxes under the heading 'Team Permissions': 'Node management' (checked), 'Compliance scan' (checked), 'Patch management' (checked), and 'Security automation' (checked). At the bottom of the dialog are 'Cancel' and 'Save change' buttons. An orange arrow points from the text 'The team permissions screen should display.' to the 'Save change' button.

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.

The screenshot shows a modal dialog for managing team permissions. It includes fields for 'Team name' (set to 'Owners') and 'Team ID' (set to 'owners'). Below these are four checkboxes under 'Team Permissions': 'Node management' (checked), 'Compliance scan' (checked), 'Patch management' (checked), and 'Security automation' (checked). At the bottom right are 'Cancel' and 'Save change' buttons. Two orange arrows point from the text descriptions in the slide to the corresponding checked boxes in the screenshot.

Team name	Owners
Team ID	owners
Team Permissions	<input checked="" type="checkbox"/> Node management <input checked="" type="checkbox"/> Compliance scan <input checked="" type="checkbox"/> Patch management <input checked="" type="checkbox"/> Security automation
<input type="button" value="Cancel"/> <input type="button" value="Save change"/>	

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.

Team name
Owners 

Team ID
owners

Team Permissions

Node management

Compliance scan

Patch management

Security automation

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.

Slide 29

Review Questions

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

Team name
Owners 

Team ID
owners

Team Permissions

Node management

Compliance scan

Patch management

Security automation

Slide 30



©2016 Chef Software Inc.

9: Further Resources

Further Resources

Other Places to Talk About, Practice, and Learn Chef Compliance

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

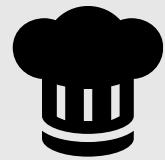


docs.chef.io

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



©2016 Chef Software Inc.

9-8



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



Thank You!